Local Prime Factor Decomposition of Approximate Strong Product Graphs

Von der Fakultät für Mathematik und Informatik
der Universität Leipzig
angenommene

D I S S E R T A T I O N

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM
(Dr. rer. nat.)

im Fachgebiet
Informatik

vorgelegt
von Diplom Wirtschaftsmathematiker *Marc Hellmuth*
geboren am 25. Juni 1980 in Nordhausen

Die Annahme der Dissertation wurde empfohlen von:

1. Professor Dr. Peter F. Stadler (Leipzig, Deutschland)
2. Professor Dr. Sandi Klavžar (Ljubljana, Slowenien)

Die Verleihung des akademischen Grades erfolgt mit Bestehen der Verteidigung am 22.04.2010 mit dem Gesamtprädikat *summa cum laude*.

Acknowledgements

Let me thank you very much !!

peter F. stadler and wilfried imrich

paula

werner kloeckl, daniel merkle, lydia gringmann, maribel hernandez-rosales, steve hoffmann, phil ostermeier, konstantin klemm, sven findeisz, and the entire *beer*informatics community

my family

christine rahn, marlen pelny, dietrich becker, gilbert spiegel, min choe, and my old friends in nordhausen

jens steuck and petra pregel

josef leydold, manja marz, sonja prohaska, and martin middendorf

... and all other persons that are in the *closed neighborhood* of mine !!

Abstract

In practice, graphs often occur as perturbed product structures, so-called *approximate* graph products. The practical application of the well-known prime factorization algorithms is therefore limited, since most graphs are prime, although they can have a product-like structure.

This work is concerned with the *strong graph product*. Since strong product graphs G contain subgraphs that are itself products of subgraphs of the underlying factors of G, we follow the idea to develop local approaches that cover a graph by factorizable patches and then use this information to derive the global factors.

First, we investigate the local structure of strong product graphs and introduce the *backbone* $\mathbb{B}(G)$ of a graph G and the so-called *S1-condition*. Both concepts play a central role for determining the prime factors of a strong product graph in a unique way. Then, we discuss several graph classes, in detail, *NICE*, *CHIC* and *locally unrefined* graphs. For each class we construct local, quasi-linear time prime factorization algorithms. Combining these results, we then derive a new local prime factorization algorithm for all graphs.

Finally, we discuss approximate graph products. We use the new local factorization algorithm to derive a method for the recognition of approximate graph products. Furthermore, we evaluate the performance of this algorithm on a sample of approximate graph products.

Contents

1 Introduction

Graphs and in particular graph products arise in a variety of different contexts, from computer science to theoretical biology, computational engineering or in studies on social networks.

In practical applications, we observe perturbed product structures, so-called *approximate* graph products, since structures derived from real-life data are notoriously incomplete and/or plagued by measurement errors. As a consequence, the structures need to be analyzed in a way that is robust against inaccuracies, noise, and perturbations in the data.

The problem of computing approximate graph products was posed several years ago in a theoretical biology context [56]. The authors provided a concept concerning the topological theory of the relationships between genotypes and phenotypes. In this framework a so-called "character" (trait or *Merkmal*) is identified with a factor of a generalized topological space that describes the variational properties of a phenotype. The notion of a character can be understood as a property of an organism that can vary independently of other traits from generation to generation. Characters thus are not necessarily the same as observable properties such as arms, legs, fingers, a spinal chord, etc, although such observables of course often are instantiations of characters. The important biological distinction is *whether* such measurable attributes (or combinations thereof) form a "coordinate axis" along

which the character states (e.g. the lengths of arms or fingers) can vary independently of other traits, or whether the underlying genetics dictates dependencies among the observables [41].

This question can be represented as a graph problem in the following way: Consider a set $\mathbb{X}$ of "phenotypes", that is, representations of distinct organisms, each of which is characterized by a list of properties such as body shape, eye color, presence or absence of certain bones, etc. If one knows about the phylogenetic relationships between the members of $\mathbb{X}$, we can estimate which combinations of properties are interconvertible over short evolutionary time-scales. This evolutionary "accessibility relation" introduces a graph-structure on $\mathbb{X}$ [7, 17, 18, 52].

In particular, a phenotype space inherits its structure from an underlying sequence space. Sequence spaces are Hamming graphs, that is, Cartesian products of complete graphs, see [10, 11]. The structure of localized subsets turns out to be of particular interest. Gavrilets [19], Grüner [21], and Reidys [48], for example, describe subgraphs in sequence spaces that correspond to the subset of viable genomes or to those sequences that give rise to the same phenotype. The structure of these subgraphs is intimately related to the dynamics of evolutionary processes [30, 54]. However, since characters are only meaningfully defined on subsets of phenotypes it is necessary to use a local definition [56]: A character corresponds to a factor in a factorizable induced subgraph with non-empty interior (where x is an interior vertex of $H \subset G$ if x and all its neighbors within G are in H).

Other applications of graph products can be found in rather different areas as computer graphics and theoretical computer science. In [1, 2], the authors provide a framework, called *TopoLayout*, to draw undirected graphs based on the topological features they contain. Topological features are detected recursively, and their subgraphs are collapsed into single nodes, forming a graph hierarchy. The final layout is drawn using an appropriate algorithm for each topological feature [1]. Graph products have a well understood structure, that can be drawn in an effective way. Hence, for an extension of this framework in particular approximate graph products are of interest.

Reasons and motivations to study graph products or graphs that have a product-like structure can be found in many other areas, e.g. for the formation of finite element models or construction of localized self-equilibrating systems in computational engineering [35–37]. Other motivations can be found in discrete mathematics. A natural question is what can be said about a graph invariant of an (approximate) product if one knows the corresponding invariants of the factors. There are many contributions, treating this problem, e.g. [4, 6, 22, 23, 26, 42].

In all applications of practical interest, the graphs in question have to be either obtained from computer simulations (e.g. within the RNA secondary structure model as in [7, 17, 18]) or they need to be estimated from measured data. In both cases, they are known only approximately. In order to

deal with such inaccuracies, a mathematical framework is needed that allows us to deal with graphs that are only approximate products.

Given a graph G that has a product-like structure, the task is to find a graph H that is a nontrivial product and a good approximation of G, in the sense that H can be reached from G by a small number of additions or deletions of edges and vertices. In fact, a very small perturbation, such as the deletion or insertion of a single edge, can destroy the product structure completely, modifying a product graph to a prime graph [13, 58].

In this thesis, we are in particular interested in the so-called *strong* graph product, that is one of the four standard products. The observation that strong product graphs contain subgraphs that are themselves products of subgraphs of the underlying factors, so-called subproducts, leads to the idea to factorize those subgraphs and to use the local factorizations for the construction of a global one.

First, we introduce the necessary basic definitions in **Chapter 2**. Moreover, we deal with two graph products, the *Cartesian* and the *strong* product and show how one computes the prime factors of a graph with respect to both products. In the last part of this chapter, we introduce several other graph classes that will become powerful tools in later considerations.

In order to cover a graph G by its subproducts and to use the information provided by the factorization of those subgraphs to construct the factors of G, we are concerned with several important tools and techniques that will help us to realize this purpose in **Chapter 3**. As it turns out, the so-called *S1-condition* and the *backbone* $\mathbb{B}(G)$ of a graph G, that is a subset of the vertex set of G, will play a central role.

After this, we are concerned with a local approach that recognizes the prime factors of a graph by covering it with induced neighborhoods that satisfy certain properties in **Chapter 4**. In particular, the term *thinness* of graphs is essential. A graph is *thin* if any two of its vertices can be distinguished by their respective neighborhoods. We introduce the class of *NICE* and *CHIC* graphs and show that the information provided by the local factorization of thin induced neighborhoods of backbone vertices is sufficient to determine the prime factors of those graphs. Moreover, we derive quasi-linear time algorithms that determine the prime factors of NICE and CHIC graphs using neighborhood information only.

As it turns out, not all graphs have this property. In **Chapter 5**, we therefore consider graphs that cannot be covered by those thin neighborhoods only and extend the previous work to graphs that have a local factorization that is not finer than the global one. We call this property *locally unrefined*. We then show how one can cover such a graph by its neighborhoods, in order to determine its prime

factors. As results we derive polynomial-time algorithms to check whether a graph is locally unrefined and to compute its prime factor decomposition.

In **Chapter 6**, we use the previous findings and provide a general local approach for the prime factor decomposition for all kinds of graphs. The algorithm makes use of several different subproducts. As it turns out, in this approach we have to enlarge the subproducts, e.g. from neighborhoods to unions of neighborhoods, for the general case. We explain how the general local approach works and show that time complexity of this approach is quasi-linear in the number of vertices of G.

Finally, we discuss approximate graph products in **Chapter 7**. We use the new local factorization algorithm to derive a method for the recognition of approximate graph products. At the end, we perform experimental tests and we evaluate the performance of this algorithm on a sample of approximate graph products.

2

The Basics

We begin this chapter with basic definitions that are quite similar to those ones in [8]. We proceed to introduce two graph products, the *Cartesian* and the *strong* product. In particular we are interested in the strong product, but as it turns out, the Cartesian product is closely related to the strong product and plays a central role in the prime factor decomposition of strong product graphs. We then explain how one decomposes a given graph into its prime factors with respect to both products and give an overview of the well-known prime factorization algorithms. In the last part of this chapter we introduce several graph classes, like *Hamming graphs*, *Subproducts*, and *S-prime graphs*, that will become powerful tools in later considerations.

2.1 Graphs

The cardinality of a set X, i.e. the number of its elements, is denoted by $|X|$. The abbreviation *gcd* stands for the *greatest common divisor*. A set $\mathscr{X} = \{X_1, \ldots, X_n\}$ of nonempty, disjoint subsets of a set X is called a *partition* of X, if $\cup_{i=1}^{n} X_i = X$. Logarithms are taken to the base 2, denoted by log.

A graph $G = (V, E)$ is an ordered pair of sets consisting of a set V of vertices and a set E of edges,

that are 2-element (unordered) subsets of V. Such graphs are also called *undirected* graphs.Note that by definition graphs cannot have edges e with $|e| = 1$. A *simple* graph is an undirected graph such that there is at most *one* edge between any two different vertices.

To avoid ambiguity, we always assume that $V \cap E = \emptyset$. If there is a risk of confusion we refer to the vertex set of G as $V(G)$ and to its edge set E as $E(G)$. A vertex v is *incident* with an edge e if $v \in e$. The two vertices incident with e are its *endpoints*, and e *joins* its endpoints. An edge $\{x,y\} \in E(G)$ is usually written as (x,y) and the vertices x and y are said to be *adjacent* or *neighbors*. Furthermore we say two edges are *incident* if they share a common endpoint.

A *path* is a graph $P = (V,E)$ of the form $V = \{v_1,\dots,v_n\}$ and $E = \{(v_1,v_2),(v_2,v_3),\dots(v_{n-1},v_n)\}$, where the vertices v_i are all distinct. A *cycle* $C = (V,E)$ is a *closed* path, i.e, a graph of the form $V = \{v_1,\dots,v_n\}$ and $E = \{(v_1,v_2),(v_2,v_3),\dots(v_{n-1},v_n),(v_n,v_1)\}$. A path P, respectively a cycle C, with n vertices will be denoted by P_n, respectively by C_n. A cycle C_4 is called *square*. The length of a path is defined as the number of its edges. The *distance* $d_G(x,y)$ in G between two vertices $x,y \in V(G)$ is defined as the shortest path, connecting them. If no such path exists we set $d_G(x,y) := \infty$. If there is no risk of confusion we write $d(x,y)$ instead of $d_G(x,y)$. The largest distance between any two vertices in G is the *diameter* of G.

A graph G is *connected* if for any two of its vertices there is a path connecting them.

Remark 2.1. From here on we always deal with connected, undirected and simple graphs $G = (V,E)$ with finite vertex set V.

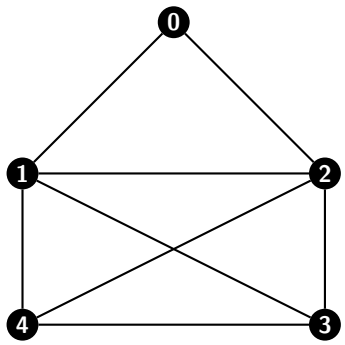


Figure 2.1: Shown is a finite, connected, undirected and simple graph.

Given $G = (V,E)$, we will write $G \ddagger (u,v)$ for the graph with vertex set V and edge set $E \ddagger (u,v)$ for each of the set operations $\ddagger \in \{\setminus, \cup, \cap\}$.

If all vertices of a graph $G = (V,E)$ are pairwise adjacent, G is *complete* and denoted by $K_{|V|}$. The

graph K_2 is called *edge* and K_3 is called *triangle*. A graph is *nontrivial* if it has at least two vertices. Hence, the complete graph K_1 is trivial.

The *open* neighborhood $N(v)$ of a vertex $v \in V$ is the set of all vertices that are adjacent to v. We define the *k-neighborhood* of vertex v as the set $N_k[v] = \{x \in V(G) \mid d_G(v,x) \leq k\}$. We call a 1-neighborhood $N_1[v] = N(v) \cup \{v\}$ also *closed* neighborhood or just neighborhood, denoted by $N[v]$, unless there is a risk of confusion. To avoid ambiguity, we sometimes write $N^G[v]$ to indicate that $N[v]$ is taken with respect to G.

The degree $\deg(v)$ of a vertex v is the number of adjacent vertices, or, equivalently, the number of incident edges. For a given graph $G = (V,E)$ the average degree $\overline{\deg}(G)$ is defined as $\frac{\sum_{v \in V} \deg(v)}{|V|}$. The maximum degree is denoted by Δ.

If for two graphs H and G holds $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ then H is a called a *subgraph* of G, denoted by $H \subseteq G$. H is a *spanning* subgraph of G if $V(H) = V(G)$. If $H \subseteq G$ and all pairs of adjacent vertices in G are also adjacent in H then H is called a *(vertex) induced* subgraph. An *edge induced* graph H of G is a subgraph with edge set $E(H) \subseteq E(G)$ and vertex set $V(H) = \cup_{e \in E(H)} e$. The subgraph of a graph G that is induced by a vertex set $W \subseteq V(G)$, respectively an edge set $F \subseteq E(G)$ is denoted by $\langle W \rangle$, respectively $\langle F \rangle$.

A subset D of $V(G)$ is a *dominating set* for G, if for all vertices in $V(G) \setminus D$ there is at least one adjacent vertex from D. We call D *connected dominating set*, if D is a dominating set and the subgraph $\langle D \rangle$ is connected.

A *homomorphism* $\phi : V(G) \to V(H)$ is an adjacency preserving mapping, i.e., if $(x,y) \in E(G)$ then $(\phi(x), \phi(y)) \in E(H)$. We call two graphs G and H *isomorphic*, and write $G \simeq H$, if there exists a bijection $\phi : V(G) \to V(H)$ with $(x,y) \in E(G) \iff (\phi(x), \phi(y)) \in E(H)$ for all $x,y \in V(G)$. Such a map ϕ is called an *isomorphism*; if $G = H$, it is called an *automorphism*.

Throughout this contribution we often use an algorithm, called *breadth-first search (BFS)*, that traverses all vertices of a graph $G = (V,E)$ in a particular order. We introduce the ordering of the vertices of V by means of breadth-first search as follows: Select an arbitrary vertex $v \in V$ and create a sorted list $BFS(v)$ of vertices beginning with v; append all neighbors $v_1, \ldots, v_{\deg(v)}$ of v; then append all neighbors of v_1 that are not already in this list; continue recursively with $v_2, v_3, \ldots$ until all vertices of V are processed. In this way, we build levels where each v in level i is adjacent to some vertex w in level $i-1$ and vertices u in level $i+1$. We then call the vertex w the *parent* of v, denoted by $parent(v)$, and vertex v a *child* of w.

2.2 Product Graphs

Defining graph products can be done in various ways. Usually one wants to define a product that satisfies the three basic properties:

1. The vertex set of a product is the Cartesian product of the vertex sets of the factors.
2. The product of a simple graph is a simple graph.
3. Adjacency in the product depends on the adjacency properties of the projections of pairs of vertices into the factors.

As shown in [31], there are 256 possibilities to define such a graph product, but only six of them are commutative, associative and have a unit, see [32]. If one wishes the product to depend on the structure of both factors and if the homomorphism property of the projections into the factors, that will be defined later on, plays a role, the number of products decreases to 4. In this contribution we are concerned with two of these 4 products, the Cartesian and the strong product. In particular, we are interested in the strong product, but as it turns out the Cartesian product is closely related to the strong product and plays a central role in the prime factorization of strong product graphs. Consequently, we will also deal with the Cartesian product.

Definition 2.2. The vertex set of the *Cartesian product* $G_1 \square G_2$ and the *strong product* $G_1 \boxtimes G_2$ of two graphs G_1 and G_2 is the set

$$V(G) \times V(H) = \{(v_1, v_2) \mid v_1 \in V(G), v_2 \in V(H)\},$$

that is, the Cartesian product of the vertex sets of the factors.

Two vertices (x_1, x_2), (y_1, y_2) are adjacent in the Cartesian product $G_1 \square G_2$ if one of the following conditions is satisfied:

$$\begin{aligned} &\text{(i)} && (x_1, y_1) \in E(G_1) \text{ and } x_2 = y_2 \\ &\text{(ii)} && (x_2, y_2) \in E(G_2) \text{ and } x_1 = y_1 \end{aligned}$$

Two vertices (x_1, x_2), (y_1, y_2) are adjacent in the strong product $G_1 \boxtimes G_2$ if one of the following conditions is satisfied:

$$\begin{aligned} &\text{(i)} && (x_1, y_1) \in E(G_1) \text{ and } x_2 = y_2 \\ &\text{(ii)} && (x_2, y_2) \in E(G_2) \text{ and } x_1 = y_1 \\ &\text{(iii)} && (x_1, y_1) \in E(G_1) \text{ and } (x_2, y_2) \in E(G_2) \end{aligned}$$

The definition of the edge sets shows that the Cartesian product is closely related to the strong product and indeed it plays a central role in the factorization of the strong products. Consequently, the edges of a strong product that satisfy (i) or (ii) are called *Cartesian*, the others *non-Cartesian*.

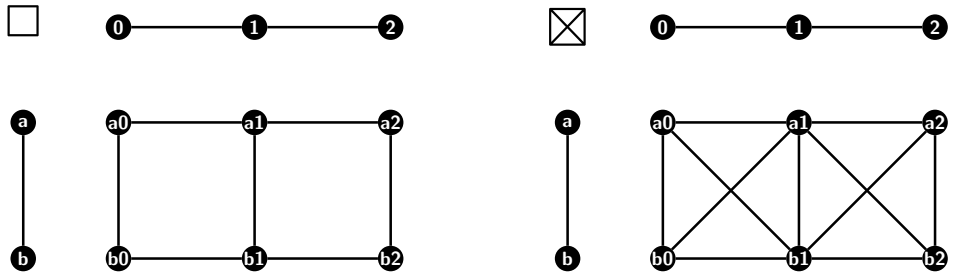


Figure 2.2: Left: A Cartesian Product graph. Right: A strong product graph

The one-vertex complete graph K_1 serves as a unit for both products, as $K_1 \square H = H$ and $K_1 \boxtimes H = H$ for all graphs H. It is well-known that both products are associative and commutative, see [32]. Hence a vertex x of the Cartesian product $\square_{i=1}^{n} G_i$, respectively the strong product $\boxtimes_{i=1}^{n} G_i$ is properly "coordinatized" by the vector $c(x) := (c_1(x), \ldots, c_n(x))$ whose entries are the vertices $c_i(x)$ of its factor graphs G_i. Two adjacent vertices in a Cartesian product graph, respectively endpoints of a Cartesian edge in a strong product, therefore differ in exactly one coordinate. Often we will write $(x_1, \ldots, x_n)$ for the coordinates of x unless there is a risk of confusion.

The mapping $p_j(x) = x_j$ of a vertex x with coordinates $(x_1, \ldots, x_n)$ is called *projection* of x onto the $j-th$ factor. For a set W of vertices of $\square_{i=1}^{n} G_i$, resp. $\boxtimes_{i=1}^{n} G_i$, we define $p_j(W) = \{p_j(w) \mid w \in W\}$. Sometimes we also write p_A if we mean the projection onto factor A.

In both products $\square_{i=1}^{n} G_i$ and $\boxtimes_{i=1}^{n} G_i$, a *G_j-fiber* or *G_j-layer* through vertex x with coordinates $(x_1, \ldots, x_n)$ is the vertex induced subgraph G_j^x in G with vertex set $\{(x_1, \ldots x_{j-1}, v, x_{j+1}, \ldots, x_n) \in V(G) \mid v \in V(G_j)\}$. Thus, G_j^x is isomorphic to the factor G_j for every $x \in V(G)$. For $y \in V(G_j^x)$ we have $G_j^x = G_j^y$, while $V(G_j^x) \cap V(G_j^z) = \emptyset$ if $z \notin V(G_j^x)$. With a *horizontal* fiber we mean the subgraph of G induced by vertices of one and the same fiber, i.e., we mean a particular G_i^x-fiber without mentioning this particularly, if there is no risk of confusion. With *parallel G_i*-fibers we mean all fibers with respect to a given factor G_i. Edges of (not necessarily different) G_i-fibers are said to be edges *of one and the same* factor G_i.

Note, the coordinatization of a product is equivalent to a (partial) edge coloring of G in which edges (x,y) share the same color c_k if x and y differ only in the value of a single coordinate k, i.e., if $x_i = y_i$, $i \neq k$ and $x_k \neq y_k$. This colors the *Cartesian edges* of G (with respect to the *given* product

representation). It follows that for each color c the set $E_c = \{e \in E(G) \mid c(e) = c\}$ of edges with color c spans G. The connected components of $\langle E_c \rangle$ are isomorphic subgraphs of G.

We state now some well-known lemmas concerning several properties of product graphs that will be used throughout this contribution. The first lemma deals with the connectedness of graphs and their product.

Lemma 2.3 ([32])**.** *Let G be a Cartesian product $\square_{i=1}^{n} G_i$, respectively, a strong product $\boxtimes_{i=1}^{n} G_i$. Then G is connected if and only if every factor G_i is connected.*

For later reference we note that the distance of two vertices in a product graph is determined by distances within the factors:

Lemma 2.4 ([32])**.** *Let $G = \square_{i=1}^{n} G_i$ and $u, v \in V(G)$. Then it holds:*

$$d_G(u,v) = \sum_{i=1}^{n} d_{G_i}(u_i, v_i).$$

Lemma 2.5 ([32])**.** *Let $G = \boxtimes_{i=1}^{n} G_i$ and $u, v \in V(G)$. Then it holds:*

$$d_G(u,v) = \max_{1 \le i \le n} d_{G_i}(u_i, v_i).$$

2.3 Prime Factor Decomposition (PFD)

In this section, we are concerned with the *Prime Factor Decomposition*, for short *PFD*, of graphs with respect to the Cartesian and the strong product. For this purpose, we first state when a graph is said to be prime.

Definition 2.6. A graph G is *prime* with respect to the Cartesian, respectively the strong product, if it cannot be written as a Cartesian, respectively a strong product, of two nontrivial graphs, i.e., the identity $G = G_1 \star G_2$ ($\star = \square, \boxtimes$) implies that $G_1 \simeq K_1$ or $G_2 \simeq K_1$.

2.3.1 The Cartesian Product

As shown by Sabidussi [49] and independently by Vizing [55], all finite connected graphs have a unique prime factor decomposition with respect to the Cartesian product.

Theorem 2.7 ([49, 55])**.** *Every connected graph has a unique representation as a Cartesian product of prime graphs, up to isomorphisms and the order of the factors.*

A well-known counterexample for the non-uniqueness of the PFD of disconnected graphs is based on results of Nakayama and Hashimoto [47]. It is not hard to see that the identity

$$(K_1 + K_2 + K_2^2) \square (K_1 + K_2^3) = (K_1 + K_2^2 + K_2^4) \square (K_1 + K_2)$$

holds, where $+$ denotes the disjoint union and where powers are taken with respect to the Cartesian product. Moreover, an easy proof that the factors on the left- and right-hand side are indeed prime can be found in [32].

In 1985, Feigenbaum et al. [15] developed the first polynomial time algorithm that finds the prime factorization of connected graphs with respect to the Cartesian product running in $O(|V|^{4.5})$ time. Later, Winkler [57] presented an $O(|V|^4)$ time algorithm which is based on a method of isometrically embedding graphs into Cartesian products by Graham and Winkler [20]. Feder [12] continued with an algorithm that requires $O(|V| \cdot |E|)$ time. The latest and fastest approach is due to Imrich and Peterin that runs in $O(|E|)$ time, see [33].

However, the main idea for the PFD of a Cartesian product G is to compute an equivalence relation Π, defined on the edge set $E(G)$, also called *product relation*. Let $G = \square_{i=1}^{n} G_i$ be a Cartesian product, where the factors are not necessarily prime. With respect to this representation we define a product relation Π on $E(G)$, as follows:

$$e \Pi f \text{ if there is an } i \text{ such that } |p_i(e)| = |p_i(f)| = 2.$$

Expressed in words, $e \Pi f$ if the projection of the endpoints of both edges e and f maps onto the same factor G_i.

The finest product relation Π leads to the prime factorization of a connected graph, i.e., a prime factor is isomorphic to one connected component of G that is induced by the edges that are in the same relation.

A well-known property of the Cartesian product is the following one.

Lemma 2.8 (Square Property [34])**.** *Let G be a Cartesian product. If e and f are incident edges of different fibers, then there exists exactly one square without diagonals that contains e and f.*

Furthermore any two opposite edges of a diagonal-free square are edges from copies of one and the same factor.

Every product relation Π satisfies the square property [32]. A very important feature of equivalence relations defined on the edge set of a given graph G is stated in the next lemma.

Lemma 2.9 ([34])**.** *Let γ be an equivalence relation on the edge set $E(G)$ of a connected graph. Suppose γ has the equivalence classes $\gamma_1, \ldots, \gamma_k, \ldots$ and satisfies the square property. Then every vertex of G meets every γ_i, i.e., every vertex is incident to an edge of each equivalence class.*

2.3.2 The Strong Product

As shown by Dörfler and Imrich [9] and independently by McKenzie [43], all finite connected graphs have a unique prime factor decomposition with respect to the strong product.

Theorem 2.10 ([9, 43])**.** *Every connected graph has a unique representation as a strong product of prime graphs, up to isomorphisms and the order of the factors.*

As in the case of the Cartesian product there is a counterexample for the non-uniqueness of the PFD of disconnected graphs based on results of Nakayama and Hashimoto [47]. The following identity holds:

$$(K_1 + K_2 + K_2^2) \boxtimes (K_1 + K_2^3) = (K_1 + K_2^2 + K_2^4) \boxtimes (K_1 + K_2),$$

where $+$ denotes the disjoint union and where powers are taken with respect to the strong product. A proof that the factors on the left- and right-hand side are prime can be found in [32].

The prime factor decomposition with respect to the strong product works basically as follows. Given a strong product G with specific property, one computes a subgraph $\mathbb{S}(G)$ of G, the so-called *Cartesian skeleton*. The skeleton $\mathbb{S}(G)$ is decomposed with respect to the Cartesian product and this information is used to construct the prime factors of the original graph G with respect to the strong product. However, before we proceed to explain this approach in more detail we have to deal with the specific property a graph G has to have: *thinness*.

Thinness

It is important to notice that although the PFD of a strong product is unique, the coordinatizations might not be. Figure 2.3 shows that the reason for the non-unique coordinatizations is the existence of automorphisms that interchange the vertices b and d, but fix all the others. This is possible because b and d have the same closed neighborhoods. Thus, an important issue in the context of strong graph products is whether or not two vertices can be distinguished by their neighborhoods. This is captured

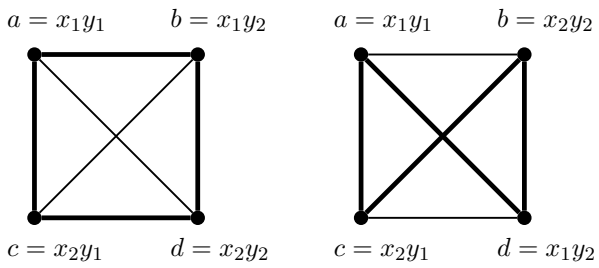


Figure 2.3: The edge (a,b) is Cartesian in the left, and non-Cartesian in the right coordinatization

by the relation S defined on the vertex set of G, which was first introduced by Dörfler and Imrich [9]. This relation is essential in the studies of the strong product.

Definition 2.11. Let G be a given a graph G and $x,y \in V(G)$ be arbitrary vertices. The vertices x and y are in relation S if $N[x] = N[y]$. A graph is *S-thin*, or *thin* for short, if no two vertices are in relation S.

In [16], vertices x and y with xSy are called *interchangeable*. Note that *xSy* implies that x and y are adjacent since, by definition, $x \in N[x]$ and $y \in N[y]$. Clearly, S is an equivalence relation. The graph G/S is the usual quotient graph, more precisely:

Definition 2.12. The *quotient graph* G/S of a given graph G has vertex set

$$V(G/S) = \{S_i \mid S_i \text{ is an equivalence class of } S\}$$

and $(S_i,S_j) \in E(G/S)$ whenever $(x,y) \in E(G)$ for some $x \in S_i$ and $y \in S_j$.

Note that the relation S on G/S is trivial, that is, its equivalence classes are single vertices [32]. Thus G/S is thin. The importance of thinness lies in the uniqueness of the coordinatizations, i.e., the property of an edge being Cartesian or not does not depend on the choice of the coordinates. As a consequence, the Cartesian edges are uniquely determined in an S-thin graph, see [9, 16].

Lemma 2.13. *If a graph G is thin, then the set of Cartesian edges is uniquely determined and hence the coordinatization is unique.*

For later usage we also define S-classes w.r.t. subgraphs of a given graph G.

Definition 2.14. Let $H \subseteq G$ be an arbitrary subgraph of a given graph G. Then $S_H(x)$ is defined as the set

$$S_H(x) = \left\{v \in V(H) \mid N^G[v] \cap V(H) = N^G[x] \cap V(H)\right\}.$$

If $H = \langle N^G[v]\rangle$ for some $v \in V(G)$ we set $S_v(x) := S_{\langle N^G[v]\rangle}(x)$

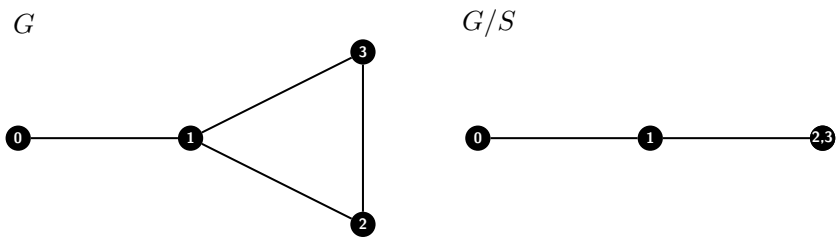


Figure 2.4: A graph G and its quotient graph G/S. The S-classes are $S_G(0) = \{0\}$, $S_G(1) = \{1\}$, and $S_G(2) = S_G(3) = \{2,3\}$.

Important basic properties, first proved by Dörfler and Imrich [9], concerning the thinness of graphs are given now. Alternative proofs can be found in [32].

Lemma 2.15. *For any two graphs G_1 and G_2 holds $(G_1 \boxtimes G_2)/S \simeq G_1/S \boxtimes G_2/S$. Furthermore, for every $x = (x_1, x_2) \in V(G)$ holds $S_G(x) = S_{G_1}(x_1) \times S_{G_2}(x_2)$.*

This result directly implies the next corollaries, see [32].

Corollary 2.16. *A graph is thin if and only if all of its factors with respect to the strong product are thin.*

Corollary 2.17. *Let G be a strong product $G = G_1 \boxtimes G_2$. Consider a vertex $x \in V(G)$ with coordinates (x_1, x_2). Then for every $z \in S_G(x)$ holds $z_i \in S_{G_i}(x_i)$, i.e. the i-th coordinate of z is contained in the S-class of the i-th coordinate of x.*

The Cartesian Skeleton

As mentioned before, the key idea of finding the PFD of a graph G with respect to the strong product is to find the PFD of a subgraph $\mathbb{S}(G)$ of G, the so-called *Cartesian skeleton*, with respect to the Cartesian product and construct the prime factors of G using the information of the PFD of $\mathbb{S}(G)$.

Definition 2.18. A subgraph H of a graph $G = G_1 \boxtimes G_2$ with $V(H) = V(G)$ is called *Cartesian skeleton* of G, if it has a representation $H = H_1 \square H_2$ such that $V(H_i^v) = V(G_i^v)$ for all $v \in V(G)$ and $i \in \{1,2\}$. The Cartesian skeleton H is denoted by $\mathbb{S}(G)$.

In other words, the H_i-fibers of the Cartesian skeleton $\mathbb{S}(G) = H_1 \square H_2$ of a graph $G = G_1 \boxtimes G_2$ induce the same partition as the G_i-fibers on the vertex sets $V(\mathbb{S}(G)) = V(G)$.

This concept was first introduced by Feigenbaum and Schäffer in [16]. In this approach, edges are marked as Cartesian if the neighborhoods of their endpoints fulfill some (strictly) maximal conditions

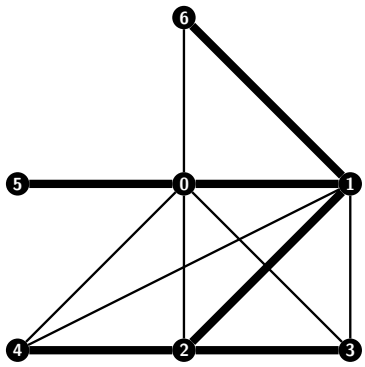


Figure 2.5: A prime graph G and its Cartesian Skeleton $\mathbb{S}(G)$ induced by thick-lined edges. Thin-lined edges are marked as dispensable in the approach of Hammack and Imrich. On the other hand, the thick-lined edges are marked as Cartesian in the approach of Feigenbaum and Schäffer. However, in both cases the resulting Cartesian skeleton $\mathbb{S}(G)$ spans G. Hence, the vertex sets of the $\mathbb{S}(G)$-fiber (w.r.t. Cartesian product) and the G-fiber (w.r.t. strong product) induce the same partition $V(\mathbb{S}(G)) = V(G)$ of the respective vertex sets.

in collections of neighborhoods or subsets of neighborhoods in G. This approach is technically tricky and complex.

A more transparent and also the fastest and latest approach is due to Hammack and Imrich, see [24]. In distinction to the approach of Feigenbaum and Schäffer edges are marked as dispensable. All edges that are dispensable will be removed from G. The resulting graph $\mathbb{S}(G)$ is the desired Cartesian skeleton and will be decomposed with respect to the Cartesian product. For an example see Figure 2.5.

Definition 2.19. An edge (x,y) of G is *dispensable* if there exists a vertex $z \in V(G)$ for which both of the following statements hold.

1. (a) $N[x] \cap N[y] \subset N[x] \cap N[z]$ or (b) $N[x] \subset N[z] \subset N[y]$
2. (a) $N[x] \cap N[y] \subset N[y] \cap N[z]$ or (b) $N[y] \subset N[z] \subset N[x]$

Some important results, concerning the Cartesian skeleton are summarized in the following theorem.

Theorem 2.20 ([24])**.** *Let $G = G_1 \boxtimes G_2$ be a strong product graph. If G is connected, then $\mathbb{S}(G)$ is connected. Moreover, if G_1 and G_2 are thin graphs then*

$$\mathbb{S}(G_1 \boxtimes G_2) = \mathbb{S}(G_1) \square \mathbb{S}(G_2).$$

Any isomorphism $\varphi : G \to H$, as a map $V(G) \to V(H)$, is also an isomorphism $\varphi : \mathbb{S}(G) \to \mathbb{S}(H)$.

Remark 2.21. Notice that the set of all Cartesian edges in a strong product $G = \boxtimes_{i=1}^{n} G_i$ of connected, thin prime graphs are uniquely determined and hence its Cartesian skeleton. Moreover, since by Theorem 2.20 and Definition 2.18 of the Cartesian skeleton $\mathbb{S}(G) = \square_{i=1}^{n} \mathbb{S}(G_i)$ of G we know that $V(\mathbb{S}(G)_i^v) = V(G_i^v)$ for all $v \in V(G)$. Thus, we can assume without loss of generality that the set of *all* Cartesian edges in a strong product $G = \boxtimes_{i=1}^{n} G_i$ of connected, thin graphs is the edge set of the Cartesian skeleton $\mathbb{S}(G)$ of G w.r.t. this factorization, see [32].

Algorithm

Now, we are able to give a brief overview of the global approach that decomposes given graphs into their prime factors with respect to the strong product, see also Figure 2.6 and 2.7.

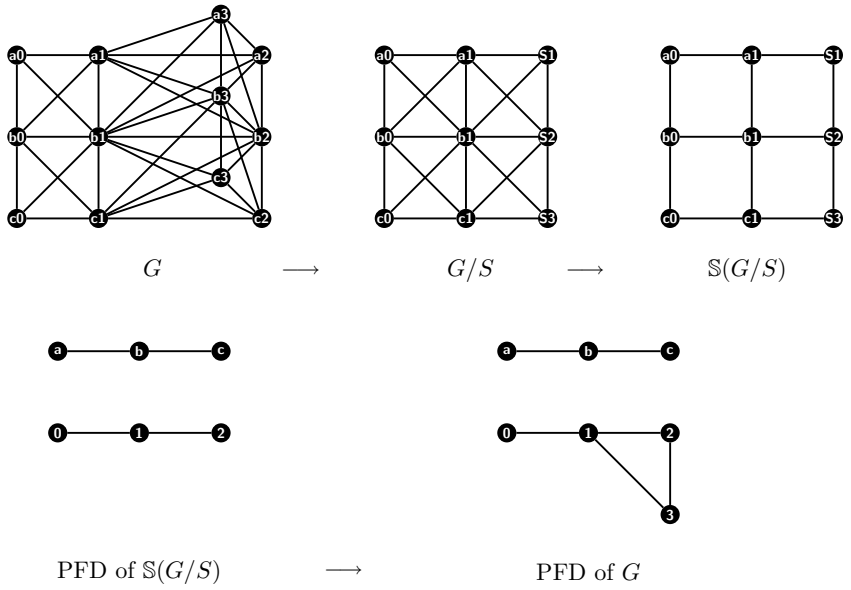


Figure 2.6: Illustrated are the basic steps of the PFD of strong product graphs, see Algorithm 1.

Given an arbitrary graph G, one first extracts a possible complete factor K_l of maximal size, resulting in a graph G', i.e., $G \simeq G' \boxtimes K_l$, and computes the quotient graph $H = G'/S$. This graph H is thin and therefore the Cartesian edges of $\mathbb{S}(H)$ can be uniquely determined. Now, one computes the prime factors of $\mathbb{S}(H)$ with respect to the Cartesian product and utilizes this information to determine the prime factors of G' by usage of an additional operation stated in the next lemma.

Lemma 2.22. *[32] Suppose that it is known that a given graph G that does not admit any complete graphs as a factor is a strong product graph $G_1 \boxtimes G_2$, and suppose that the decomposition $G/S = G_1/S \boxtimes G_2/S$ is known. Then G1 and G2 can be determined from G, G_1/S and G_2/S.*

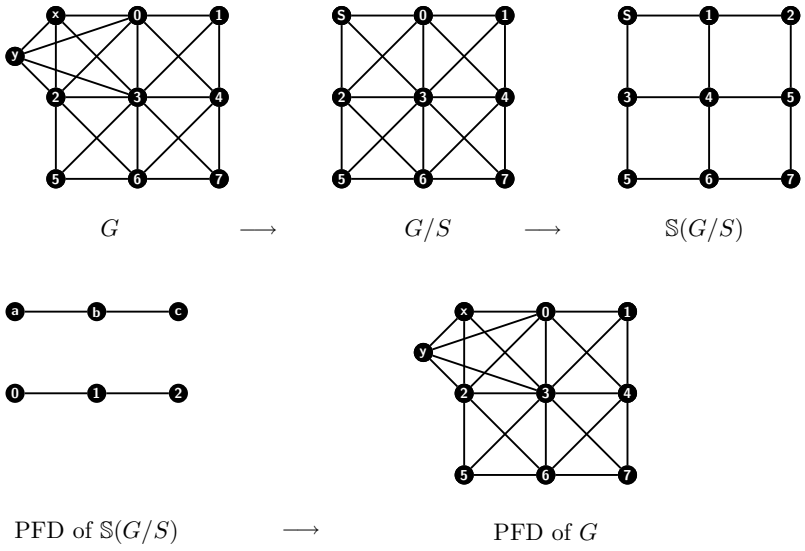


Figure 2.7: Illustrated are the basic steps of the PFD of strong product graphs, see Algorithm 1.

In fact, if $D(x_1,x_2)$ denotes the size of the S-equivalence class of G that is mapped into $(x_1,x_2) \in G_1/S \boxtimes G_2/S$, then the size $D(x_1)$ of the equivalence class of G_1 mapped into $x_1 \in G_1/S$ is $gcd\{D(x_1,y) \mid y \in V(G_2)\}$. Analogously for $D(x_2)$.

By repeated application of Lemma 2.22 one can determine the prime factors of G', see [32]. Notice that $G \simeq G' \boxtimes K_l$. The prime factors of G are then the prime factors of G' together with the complete factors $K_{p_1},\dots,K_{p_j}$, where $p_1 \dots p_j$ are the prime factors of the integer l. This approach is summarized in Algorithm 1 and 2.

Algorithm 1 PFD of graphs w.r.t. $\boxtimes$

1: **INPUT:** a graph G
2: Compute $G = G' \boxtimes K_l$, where G' has no nontrivial factor isomorphic to a complete graph K_r;
3: Determine the prime factorization of K_l, that is, of l;
4: compute $H = G'/S$;
5: compute PFD and prime factors $H_1,\dots,H_n$ of H with Algorithm 2
6: By repeated application of Lemma 2.22 find all minimal subsets J of $I = \{1,2,\dots,n\}$ such that there are graphs A and B with $G = A \boxtimes B$, $A/S = \boxtimes_{i\in J} H_i$ and $B = \boxtimes_{j\in J\setminus I} H_j$. Save A as prime factor.
7: **OUTPUT:** The prime factors of G;

Algorithm 2 PFD of *thin* graphs w.r.t. $\boxtimes$

INPUT: a *thin* graph G
compute the Cartesian skeleton $\mathbb{S}(G)$;
factor $\mathbb{S}(G) = \square_{i \in I} H_i$ and assign coordinates to each vertex;
$J \leftarrow I$;
for $k = 1, \ldots, |I|$ **do**
  for each $S \subset J$ with $|S| = k$ **do**
    compute $A = \square_{i \in S} V(H_i)$ and $A' = \square_{i \in I \setminus S} V(H_i)$;
    compute $B_1 = \langle p_A(G) \rangle$ and $B_2 = \langle p_{A'}(G) \rangle$;
    if $B_1 \boxtimes B_2 \simeq G$ **then**
      save B_1 as prime factor;
      $J \leftarrow J \setminus S$;
    end if
  end for
end for
OUTPUT: The prime factors of G;

However, Algorithm 1 and 2 just give an overview of the top level control structure to determine the PFD of a given graph. Applying some smart ideas together with slight modifications on those Algorithms one can bound the time complexity as stated in the next lemma.

Lemma 2.23 ([24])**.** *The PFD of a given graph $G = (V, E)$ with bounded maximum degree Δ can be computed in $O(|E|\Delta^2)$ time.*

2.4 Graph Classes

In this section, we will introduce some special kinds of graphs that will be important and useful in the sequel. We start to define *Hamming graphs* and will proceed to describe particular *subproducts* of given graphs. At the end of this section so-called *S-prime* graphs are introduced, that are a special class of prime graphs and will become a powerful tool for later considerations.

2.4.1 Hamming Graphs

We state here the definition of so-called *Hamming graphs*, that have comprehensively been studied, see e.g. [3, 44–46].

Definition 2.24. A graph G is a *Hamming graph* iff G can be written in the form

$$G = \square_{i=1}^{n} K_{k_i},$$

where $k_i \geq 2$ for all i. If $k_i = 2$ for all i then G is called a *hypercube* of dimension n.

Note that the distance between two vertices in a Hamming graph coincides with the number of positions, in which they differ, which is also known as Hamming distance [25].

2.4.2 Subproducts

As already mentioned, the aim of this contribution is to provide algorithms that cover and decompose given graphs by usage of so-called *subproducts*, also known as *boxes* [53].

Definition 2.25. A *subproduct* of a product $G \boxtimes H$, resp. $G \square H$, is defined as the strong product, resp. the Cartesian product, of subgraphs of G and H, respectively.

As shown in [28], it holds that 1-neighborhoods are subproducts:

Lemma 2.26 ([28])**.** *For any two graphs G and H holds* $\langle N^{G \boxtimes H}[(x,y)] \rangle = \langle N^G[x] \rangle \boxtimes \langle N^H[y] \rangle$.

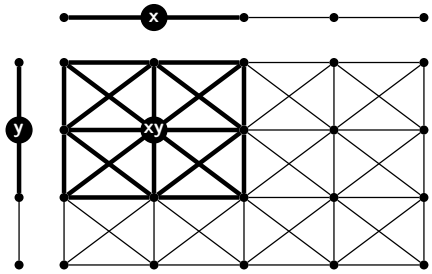


Figure 2.8: The 1-neighborhood $\langle N[(x,y)] \rangle = \langle N[x] \rangle \boxtimes \langle N[y] \rangle$ is highlighted by thick lined edges

For applications to approximate products it would be desirable to use small subproducts. Unfortunately, it will turn out that 1-neighborhoods, which would be small enough for our purpose, are not sufficient to cover a given graph in general while providing enough information to recognize the global factors. However, we want to avoid to use 2-neighborhoods, although they are subproducts as well, they have diameter 4 and are thus quite large. Therefore, we will define further subgraphs, that are smaller than 2-neighborhoods, and prove that these subgraphs are subproducts.

Definition 2.27. Given a graph G and an arbitrary edge $(v,w) \in E(G)$. The *edge-neighborhood* of (v,w) is defined as

$$\langle N[v] \cup N[w] \rangle$$

and the $N^*_{v,w}$*-neighborhood* is defined as

$$N^*_{v,w} = \langle \bigcup_{x \in N[v] \cap N[w]} N[x] \rangle.$$

If there is no risk of confusion we will denote $N^*_{v,w}$-neighborhoods just by N^*-neighborhoods. We will show in the following that in addition to 1-neighborhoods also edge-neighborhoods of Cartesian edges and N^*-neighborhoods are subproducts and hence, natural candidates to cover a given graph as well. We show first, given a subproduct H of G, that the subgraph that is induced by vertices contained in the union of 1-neighborhoods $N[v]$ with $v \in V(H)$, is itself a subproduct of G.

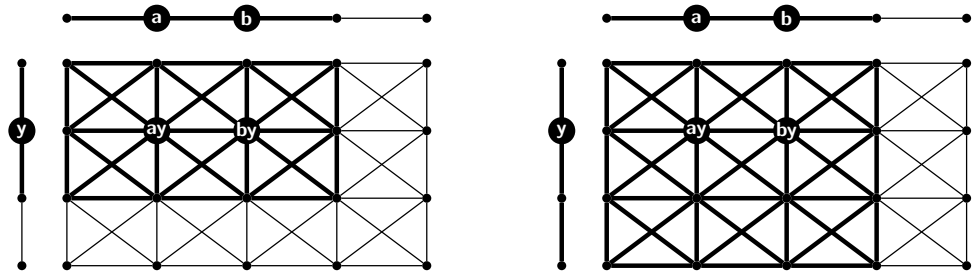


Figure 2.9: Shown is a strong product graph of two paths. Notice that the 2-neighborhood $\langle N_2[(by)] \rangle$ of vertex (by) is isomorphic to G.
lhs.: The edge-neighborhood $\langle N[(a,y)] \cup N[(b,y)] \rangle = \langle (N[a] \cup N[b]) \rangle \boxtimes \langle N[y] \rangle$.
rhs.: The N^*-neighborhood $N^*_{(ay),(by)} = \langle \cup_{z \in N[a] \cap N[b]} N[z] \rangle \boxtimes \langle \cup_{z \in N[y]} N[z] \rangle$.

Lemma 2.28. *Let $G = G_1 \boxtimes G_2$ be a strong product graph and $H = H_1 \boxtimes H_2$ be a subproduct of G. Then*

$$H^* = \left\langle \cup_{v \in V(H)} N^G[v] \right\rangle$$

is a subproduct of G with $H^* = H_1^* \boxtimes H_2^*$, where H_i^* is the induced subgraph of factor G_i on the vertex set $V(H_i^*) = \bigcup_{v_i \in V(H_i)} N^{G_i}[v_i]$, $i = 1,2$.

Proof. It suffices to show that $V(H^*) = V(H_1^*) \times V(H_2^*)$. For the sake of convenience, we denote $V(H_i)$ by V_i, for $i = 1,2$. We have:

$$V(H^*) = \bigcup_{v \in V(H)} N^G[v] = \bigcup_{v \in V_1 \times V_2} N^G[v].$$

Since the induced neighborhood of each vertex $v = (v_1, v_2)$ in G is the product of the corresponding neighborhoods $N^{G_1}[v_1] \boxtimes N^{G_2}[v_2]$ we can conclude:

$$V(H^*) = \bigcup_{\{v_1 \in V_1\} \times (v_2 \in V_2\}} (N^{G_1}[v_1] \times N^{G_2}[v_2]) \qquad = \bigcup_{v_1 \in V_1} N^{G_1}[v_1] \times \bigcup_{v_2 \in V_2} N^{G_2}[v_2]$$
$$= V(H_1^*) \times V(H_2^*)$$

□

Lemma 2.29. *Let G be a nontrivial strong product graph and (v,w) be an arbitrary edge of G. Then $\langle N^G[v] \cap N^G[w] \rangle$ is a subproduct.*

Proof. Let v and w have coordinates (v_1, v_2) and (w_1, w_2), respectively. Since $N^G[v] = N^{G_1}[v_1] \times N^{G_2}[v_2]$ we can conclude that

$$\begin{aligned} N^G[v] \cap N^G[w] &= (N^{G_1}[v_1] \times N^{G_2}[v_2]) \cap (N^{G_1}[v_1] \times N^{G_2}[v_2]) \\ &= (N^{G_1}[v_1] \cap N^{G_1}[w_1]) \times (N^{G_2}[v_2] \cap N^{G_2}[w_2]). \end{aligned}$$

□

Lemmas 2.26, 2.28 and 2.29 directly imply the next corollary.

Corollary 2.30. *Let G be a given graph. Then for all $v \in V(G)$ and all edges $(v,w) \in E(G)$ holds:*

$$\langle N_2[v] \rangle \text{ and } N^*_{v,w}$$

is a subproduct of G. Moreover, if the edge (v,w) is Cartesian than the edge-neighborhood

$$\langle N[v] \cup N[w] \rangle$$

is a subroduct of G.

Notice that $\langle N[v] \cup N[w] \rangle$ could be a product, i.e., not prime, even if (v,w) is non-Cartesian in G. However, the edge-neighborhood of a single non-Cartesian edge is not a subproduct, in general.

2.4.3 S-prime Graphs

In this section so-called *S-prime* graphs are considered. This graph class is a subset of prime graphs with special properties and will be used later on for the designed covering algorithms. The results of this subsection have been submitted to *Discrete Mathematics*, [27].

Definition 2.31. A graph S is *S-prime* (S stands for "subgraph") if for all graphs G and H with $S \subseteq G \star H$ holds: $S \subseteq H$ or $S \subseteq G$, where $\star$ denotes an arbitrary graph product. A graph is *S-composite* if it is not S-prime.

The class of S-prime graphs was introduced and characterized for the direct product by Sabidussi in 1975 [50]. He showed that the only S-prime graphs with respect to the direct product are complete graphs or complete graphs minus an edge. Analogous notions of S-prime graphs with respect to other products are due to Lamprey and Barnes [39, 40]. They showed that the only S-prime graphs w.r.t. the strong product and the lexicographic product are the single vertex graph K_1, the disjoint union $K_1 \cup K_1$ and the complete graph on two vertices K_2.

Remark 2.32. In this section we will consider the Cartesian product only. Therefore, the terms S-prime and S-composite refer to this product from here on.

Not much is known about the structure of S-prime graphs, although Klavžar *et al.* [38] and Brešar [5] proved several characterizations of S-prime graphs. For our purposes, the characterization of S-composite graphs in terms of particular colorings [38] is of most direct interest. Before we proceed, we introduce some notation, that is only needed in this section.

A *k-coloring* of G is a surjective mapping $F : V(G) \to \{1, \ldots, k\}$. This coloring need not be proper, i.e., adjacent vertices may receive the same color. A path P in G is *well-colored* by F if for any two consecutive vertices u and v of P we have $F(u) \neq F(v)$. Following [38], we say that F is a *path-k-coloring* of G if $F(u) \neq F(v)$ holds for the endpoints of every well-colored u,v-path P in G. For $k = 1$ and $k = |V|$ there are trivial path-k-colorings: For $k = 1$ the coloring is constant and hence there are no well-colored paths. On the other hand, if a different color is used for every vertex, then every path, of course, has distinctly colored endpoints. A path-k-coloring is nontrivial if $2 \leq k \leq |V(G)| - 1$.

Theorem 2.33 ([38])**.** *A connected graph G is S-composite if and only if there exists a nontrivial path-k-coloring.*

The next corollary, which follows directly from Theorem 2.33, will be useful in the subsequent discussion.

Corollary 2.34. *Consider an S-prime graph S and let F be a path-k-coloring of S. If there are two distinct vertices $u, v \in V(S)$ with $F(u) = F(v)$ then F is constant, i.e., $k = 1$.*

Now consider a product graph $\square_i G_i$. We say that all vertices *within* the G_i-layer G_i^x have the same color if $F(a) = F(b)$ holds for all vertices $a, b \in V(G_i^x)$. Note that this does not imply that vertices of different G_i-layer receive the same color.

The main topic of this section are *diagonalized* Cartesian product graphs.

Definition 2.35. A graph G is called a *diagonalized* Cartesian product, whenever there is an edge $(u,v) \in E(G)$ such that $H = G \setminus (u,v)$ is a nontrivial Cartesian product and u and v have maximal distance in H.

For an example of a diagonalized Cartesian product see Figure 2.10.

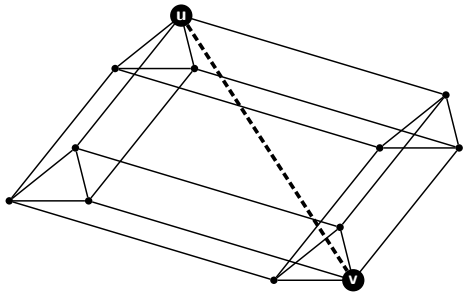


Figure 2.10: A diagonalized Cartesian Product of the graph $K_2 \square K_2 \square K_3$.

We will show that diagonalized Cartesian products of S-prime Graphs are S-prime. Moreover, we will give a necessary and sufficient condition for path-k-colorings of Cartesian products of S-prime graphs.

Path-k-colorings of Cartesian Products of S-prime graphs

Let us start with a brief preview of this paragraph. We first establish that every nontrivial Cartesian product $G_1 \square G_2$ has a nontrivial path-k-coloring. For instance, choose $k = |V(G_1)|$ and assign to every vertex x with coordinates (x_1, x_2) the color x_1.

Given a Cartesian product $G = \square_{i=1}^{n} S_i$ of S-prime graphs with a nontrivial path-k-coloring F, first we will show that there is an S_i-layer on which F is constant. Next, we prove that is true for all S_i-layers. We then proceed to show that F is constant even on any H-layer with $H = \square_{j \in J} S_j$, provided that certain conditions are satsfied. This eventually leads us to necessary and sufficient conditions for path-k-colorings. This result, in turn, will be demonstrated to imply that diagonalized Cartesian products of S-prime graphs are S-prime.

We start our exposition with a simple necessary condition:

Lemma 2.36. *Let $H \subseteq G$ and suppose F is a path-k-coloring of G. Then the restriction $F_{|V(H)}$ of F on $V(H)$ is a path-k-coloring of H. Moreover, if $V(H) = V(G)$ and F is a nontrivial path-k-coloring*

of G, then it is also a nontrivial path-k-coloring of H.

Proof. Suppose H is not path-k-colored. Then there is a u,v-path $P_{u,v}$ in H that is well-colored, but u and v have the same color. This path $P_{u,v}$ is also contained in G, contradicting the assumption that F is a path-k-coloring of G. The second statement now follows directly from $|V(G)| = |V(H)|$. □

Lemma 2.37. *Let F be a nontrivial path-k-coloring of G. Then there are adjacent vertices $u,v \in V(G)$ with $F(u) = F(v)$.*

Proof. Since $k \leq |V(G)| - 1$ it follows that there are at least two vertices of the same color, say x and y. Assume now there is a path $P_{x,y}$ from x to y, such that all consecutive vertices have different colors. Then $P_{x,y}$ would be well-colored. But the endpoints of P_{xy} satisfy $F(x) = F(y)$ so that F cannot be a path-k-coloring, a contradiction. Thus there are consecutive, and hence adjacent, vertices with the same color. □

For later reference, we state the following observation that can be verified by explicitly enumerating all colorings, see Figure 2.11 for a subset of cases.

Lemma 2.38. *The hypercube $Q_2 = K_2 \square K_2$ has no path-3-coloring. Every path-2-coloring has adjacent vertices with the same color.*

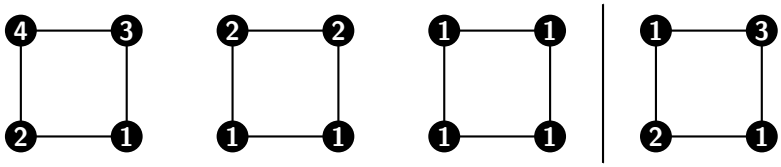


Figure 2.11: Possible path-k-coloring of a square Q_2 for $k = 1,2,4$. A possible well coloring that is not a path-3-coloring is shown on the right-hand side graph

We next show that F is constant on each S_j-layer whenever there is one S_j-layer that contains two distinct vertices with the same color. More precisely:

Lemma 2.39. *Let $G = \square_{i=1}^{n} S_i$ be a given Cartesian product of S-prime graphs and let F be a nontrivial path-k-coloring of G. Furthermore let $u,w \in V(S_j^u)$ be two distinct vertices satisfying $F(u) = F(w)$. Then $F(x) = F(y)$ holds for all vertices $x,y \in V(S_j^b)$ in each S_j-layer S_j^b.*

Proof. Corollary 2.34 and Lemma 2.36 imply that all vertices of the layer S_j^u have the same color. For $b \in V(S_j^u)$ there is nothing to show. Thus, assume $b \notin V(S_j^u)$, i.e., $S_j^u \neq S_j^b$, and an arbitrary

edge $e = (u,v) \in E(S_j^u)$. Let $\tilde{u} \in V(S_j^b)$ be the vertex with coordinates $c_j(\tilde{u}) = c_j(u)$. Moreover, let $P_{u,\tilde{u}} := (u = u_1, u_2, .., u_l = \tilde{u})$ be a path from u to $\tilde{u}$ such that $c_j(u_k) = c_j(u)$ for all $k = 1, \ldots, l$. None of the edges (u_k, u_{k+1}) is contained in an S_j-layer. By definition of the Cartesian product there is a unique square (u, u_2, v_2, v) where v_2 has coordinates $c_i(v_2) = c_i(u_2)$ for $i \neq j$ and $c_j(v_2) = c_j(v)$. Lemma 2.38 now implies that the only F on the square is either constant or a path-2-coloring, i.e., the assumption $F(u) = F(v)$ implies $F(u_2) = F(v_2)$.

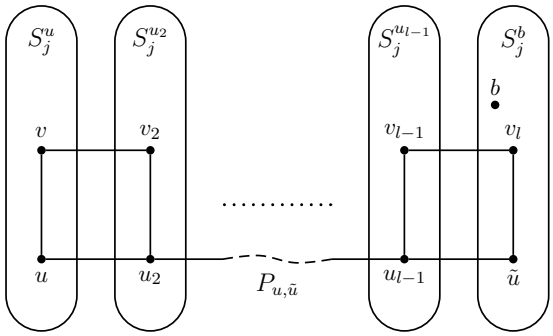


Figure 2.12: Idea of the proof of Lemma 2.39. The path $P_{u,\tilde{u}}$ connects vertices u and u_k ($k = 2, \ldots, l$) of distinct S_j-layers. If $F(u_{k-1}) = F(v_{k-1})$ then the squares $(u_{k-1}, u_k, v_k, v_{k-1})$ located in adjacent S_j-layers must admit a path-1-coloring or a path-2-coloring, enforcing that u_k and v_k must have the same color. This, in turn, is used to show that F is constant on the entire layer $S_j^{u_k}$.

By induction on the length of the path $P_{u,\tilde{u}}$ we see that $F(u_k) = F(v_k)$, whenever $c_i(v_k) = c_i(u_k)$ for all $i \neq j$ and $c_j(v_k) = c_j(v)$. The assumption $\tilde{u} \in V(S_j^b)$ and our choice of the coordinates implies $(u_l, v_l) = (\tilde{u}, v_l) \in E(S_j^b)$. We apply Lemma 2.38 to the square $(u_{l-1}, \tilde{u}, v_l, v_{l-1})$ with $F(u_{l-1}) = F(v_{l-1})$ to infer $F(\tilde{u}) = F(v_l)$. Corollary 2.34 and Lemma 2.36 imply that for all vertices $x, y \in V(S_j^b)$ holds $F(x) = F(y)$. □

It is imporant to notice that Lemma 2.39 only implies that F is constant on S_j-layers, but it does not imply that all S_j-layers receive the same color.

Corollary 2.40. *Let $G = \square_{i=1}^n S_i$ be a given product of S-prime graphs and let F be a nontrivial path-k-coloring of G. Then there is a $j \in I_n$ such that, for every $v \in V(G)$, F is constant on S_j^v.*

Proof. The assertion follows directly from Lemma 2.37, Lemma 2.39, and the definition of the Cartesian product. □

Lemma 2.41. *Let F be a nontrivial path-k-coloring of the Cartesian product $G = \square_{i=1}^n S_i$ of S-prime graphs S_i. Let $H = \square_{j \in J} S_j$ be the product of a subset of factors of G, where $J \subseteq I_n$ denotes an*

arbitrary subset of indices. Moreover, let H^a be an H-layer such that F is constant on $V(H^a)$. Then F is constant within each H-layer.

Proof. Let H^a be an H-layer defined as above and assume $H^a \neq H^b$. By assumption, F is constant on $V(H^a)$. Thus F is also constant on each S_j-layer $S_j \subseteq H^a$, $j \in J$, and Lemma 2.39 then implies that F is also constant within every S_j-layer with $j \in J$. Now choose two arbitrary vertices $x,y \in V(H^b)$. By connectedness of H^b there is a path $P_{x,y}$ from x to y consisting only of vertices of this H-layer H^b. Notice that any two consecutive vertices $x_k, x_{k+1} \in P_{x,y}$ are contained in some S_j-layer such that $j \in J$ and therefore $F(x_k) = F(x_{k+1})$. Therefore, the coloring F must be constant along P, hence $F(x) = F(y)$. Thus F is constant on $V(H^b)$. □

Next we consider two (not necessarily prime) factors H_1, H_2 of a Cartesian product of S-prime graphs and ask under which conditions a path-k-coloring on $(H_1 \square H_2)$-layers must be constant.

Lemma 2.42. *Let F be a nontrivial path-k-coloring on the Cartesian product $G = \square_{i=1}^{n} S_i$ of S-prime graphs S_i. Let $H_1 = \square_{j \in J} S_j$ and $H_2 = \square_{k \in K} S_k$ be two distinct Cartesian products of factors S_i of G, where $J, K \subseteq I_n$ and $J \cap K = \emptyset$. Then F is constant on each $(H_1 \square H_2)$-layer whenever F is constant on some H_1-layer H_1^a and on some H_2-layer H_2^b.*

Proof. Let H_1^a and H_2^b as constructed above. Lemma 2.41 implies that all vertices within each H_1 layer and within each H_2-layer, resp., have the same color. For all vertices $z \in V(H_1^a)$ there is an H_2-layer H_2^z, Thus for all vertices $x,y \in V(H_2^z)$ holds $F(x) = F(y) = F(z) = F(a)$. By definition of the Cartesian product, this implies in particular that all vertices within the layer $(H_1 \square H_2)^a$ have the same color $F(a)$. Hence we can apply Lemma 2.41 and conclude that all vertices within each $(H_1 \square H_2)$-layer have the same color. □

Now we are in the position to characterize nontrivial path-k-colorings.

Lemma 2.43. *Let F be a nontrivial path-k-coloring of the Cartesian product $G = \square_{i=1}^{n} S_i$ of S-prime graphs S_i, and consider two distinct vertices $u, v \in V(G)$ satisfying $F(u) = F(v)$. Let $J = \{ j \mid c_j(u) \neq c_j(v) \} \subseteq I_n$ denote the index set of the coordinates in which u and v differ, and let $H = \square_{j \in J} S_j$ be the Cartesian product of the corresponding factors S_j of G. Then F is constant within each H-layer H^b.*

Proof. First assume that $v \in V(S_l^u)$ for some l, which implies that $J = \{l\}$ by definition of the Cartesian product. In this case, the statement follows directly from Lemma 2.39.
Now assume that there is no l such that $v \in V(S_l^u)$. Lemma 2.39 and Corollary 2.40 together imply

that there is an index i such that all vertices within each S_i-layer have the same color. In particular, this is true for S_i^u and S_i^v. Together with Lemma 2.39, this observation implies that, since $F(u) = F(v)$, F is constant on $V(S_i^u) \cup V(S_i^v)$. Now let $\tilde{u} \in V(S_i^v)$ be the vertex with coordinates $c_i(u) = c_i(\tilde{u})$ and denote by $J_1 = \{j \mid c_j(u) \neq c_j(\tilde{u})\} = J \setminus \{i\}$ the set of indices in which the coordinates of u and $\tilde{u}$ differ. Notice that $J \setminus \{i\} = J$, if $v = \tilde{u}$.

Let $P_{u,\tilde{u}} := (u = u_1, u_2, .., u_k = \tilde{u})$ be a path from u to $\tilde{u}$ such that for all vertices $x \in P_{u,\tilde{u}}$ holds $c_r(x) = c_r(u)$ for all $r \in I_n \setminus J_1$. In other words, no edge of an S_r-layer, $r \notin J_1$, is contained in the path $P_{u,\tilde{u}}$, and hence in particular no edge of an S_i-layer. From $F(u) = F(\tilde{u})$ and the fact that G is path-k-colored, we can conclude that there is an edge $(u_l, u_{l+1}) \in P_{u,\tilde{u}}$ of some layer different from S_i such that $F(u_l) = F(u_{l+1})$.

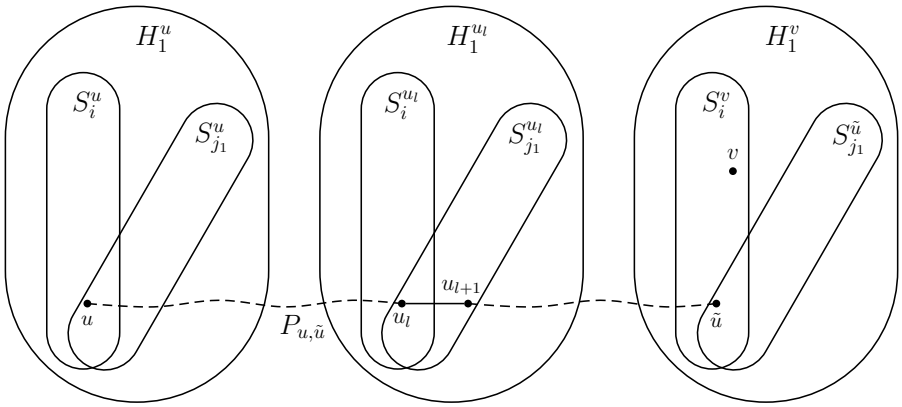


Figure 2.13: Idea of the proof of Lemma 2.43. The path $P_{u,\tilde{u}}$ connects a pair of vertices with the same color in S_i^u to S_i^v. It therefore must contain two consecutive vertices u_l and u_{l+1} with the same color. It follows that all vertices within the layer $S_i^{u_l}$ and $S_{j_1}^{u_l}$ have the same color $F(u_l)$ and finally one shows that all vertices within each H_1-layer with $H_1 = S_i \square S_{j_1}$ have the same color.

These consecutive vertices u_l and u_{l+1} differ in exactly one coordinate c_{j_1} for some $j_1 \in J_1$, hence u_l and u_{l+1} are contained in some S_{j_1}-layer. Lemma 2.39 implies that all vertices of this layer $S_{j_1}^{u_l}$ and therefore all vertices within each S_{j_1}-layer have the same color. Lemma 2.42 now implies that F is constant on each H_1-layer with $H_1 = S_i \square S_{j_1}$, and in particular, all vertices $x, y \in V(H_1^u) \cup V(H_1^v)$ have the same color, we have again two different layers that have the same color. Just as before we will construct a path between these layers, which implies that the endpoints of this path have the same color. Since G is path-k-colored, this path must contain an edge (u_t, u_{t+1}) with $F(u_t) = F(u_{t+1})$.

More precisely, let $\tilde{u}$ be a vertex of this new H_1-layer H_1^v such that $c_i(\tilde{u}) = c_i(u)$ and $c_{j_1}(\tilde{u}) = c_{j_1}(u)$. Again we choose a Path $P_{u,\tilde{u}}$ constructed as above, where J_1 is replaced by $J_2 = J_1 \setminus \{j_1\}$. In other

words for all vertices $x \in P_{u,\tilde{u}}$ holds $c_r(x) = c_r(u)$ for all $r \in I_n \setminus J_2$, i.e. in particular no edge of $P_{u,\tilde{u}}$ is contained in any H_1-layer. Notice that $|J_2| = |J_1| - 1$. Again we can conclude that there are consecutive vertices $u_t, u_{t+1} \in P_{u,\tilde{u}}$ such that $F(u_t) = F(u_{t+1})$, since $F(\tilde{u}) = F(u)$ and G is path-k-colored. Let these consecutive vertices u_t and u_{t+1} differ in coordinate c_{j_2} for some $j_2 \in J_2$. Using the same arguments as before we can infer that all vertices in between each $H_2 = (S_i \square S_{j_1} \square S_{j_2})$-layer must have the same color.

Repeating this procedure generates, in each step, a new index set J_s with $|J_s| = |J_{s-1}| - 1$ for $s = 2, \ldots, |J_1|$, and all vertices within each H_s-layer with $H_s = S_i \square \left(\square_{j \in J_1 \setminus J_s} S_j\right) \square S_{j_s}$ for some $j_s \in J_s$ are shown to have the same color. For $s^* = |J_1|$ we have $|J_{s^*}| = 1$. Moreover the path $P_{u,\tilde{u}}$ with $c_r(\tilde{u}) = c_r(u)$ for all $r \in I_n \setminus \{j^*\}$ with $j^* \in J_{s^*}$ consists only of vertices that are included in this S_{j^*}-layer $S^u_{j^*}$. Since $F(u) = F(\tilde{u})$ and $u, \tilde{u} \in S^u_{j^*}$ we can conclude that all vertices $x \in S^u_{j^*}$ have the same color $F(u)$. From Lemma 2.41 and Lemma 2.42 it follows that F is constant on each H_{s^*}-layer, where $H_{s^*} = \left(S_i \square (\square_{j \in J_1 \setminus J_{s^*}} S_j) \square S_{j^*}\right)$. Since $\{i\} \cup (J_1 \setminus J_{s^*}) \cup \{j^*\} = \{i\} \cup ((J \setminus \{i\}) \setminus \{j^*\}) \cup \{j^*\} = J$, we conclude that all vertices within each $(\square_{j \in J} S_j)$-layer have the same color, completing the proof of the lemma. □

Since two vertices with maximal distance contained in a Cartesian product of nontrivial factors differ in all coordinates we can conclude the following corollary.

Corollary 2.44. *Let F be a path-k-coloring of the Cartesian product $G = \square_{i=1}^n S_i$ of S-prime graphs S_i and suppose $u, v \in V(G)$ are two vertices with maximal G-distance that have the same color. Then F is constant on G, i.e., $k = 1$.*

Characterization

We are now in the position to give a complete characterization of path-k-colorings of Cartesian products of S-prime graphs.

Theorem 2.45 (Path-k-coloring of Cartesian products of S-prime Graphs)**.** *Let $G = \square_{j=1}^n S_j$ be a Cartesian product of S-prime graphs. Then F is a path-k-coloring of G if and only if there exists an index set $I \subseteq I_n$ such that the following two conditions hold for the graph H defined as $H = \square_{i \in I} S_i$ for $I \neq \emptyset$ and $H = K_1$ for $I = \emptyset$.*

1. *$F(a) = F(b)$ for all $a, b \in V(H^x)$ for all $x \in V(G)$ and*
2. *$F(a) \neq F(b)$ for all $a \in V(H^x)$ and $b \in V(H^y)$ with $H^x \neq H^y$.*

The coloring F consists of $k = |V(G)|/|V(H)|$ distinct colors. F is nontrivial if and only if $I \neq I_n$ and $I \neq \emptyset$.

Proof. Let F be an arbitrary path-k-coloring of G. If F is trivial, then it follows that $k = 1$ or $k = |V(G)|$ and thus we can conclude that $I = I_n$ or $I = \emptyset$, respectively. In both cases, conditions (1) and (2) are satisfied. If F is nontrivial, then $k \leq |V(G)| - 1$ and there are two vertices with the same color. Conditions (1) and (2) now follow directly from Lemma 2.42 and Lemma 2.43.

We will prove the converse by contraposition. Thus assume that F satisfied properties (1) and (2) for some $I \subseteq I_n$ and F is not a path-k-coloring of G. Thus, there must be a well colored path $P_{u,v}$ between two vertices u and v with $F(u) = F(v)$. If there is an edge $(a,b) \in P_{u,v}$ such that (a,b) is contained in an H-layer H^x for some $x \in V(G)$ we would contradict Condition (1). Thus assume there is no edge $(a,b) \in P_{u,v}$ that lies in any H-layer. Notice that this implies that u and v are not contained in the same H-layer, otherwise some edge $(a,b) \in P_{u,v}$ must be an edge of an H-layer, by definition of the Cartesian product. Since $P_{u,v}$ is a well colored path between u and v with $F(u) = F(v)$ and $H^u \neq H^v$, we contradict Condition (2).

It remains to show that F consists of $k = |V(G)|/|V(H)|$ different colors. For $I = I_n$ and $I = \emptyset$ this assertion is trivially true. Therefore assume $I \neq I_n$ and $I \neq \emptyset$. Condition (2) implies that all pairwise different H-layers are colored differently and from Condition (1) we can conclude that all vertices in between each H-layer have the same color. Thus we have just as many colors as H-layers exists. In a Cartesian product $G = H \square H'$ the number of different H-layers is $|V(H')| = |V(G)|/|V(H)|$ and thus $k = |V(G)|/|V(H)|$.

Finally, we have to show that F is nontrivial if and only if $I \neq I_n$ and $I \neq \emptyset$. If F is nontrivial the assumption is already shown at the beginning of this proof. Thus assume now that $I = I_n$, i.e., $H = \square_{i \in I} S_i = G$. Condition (1) implies that all vertices $v \in V(G)$ have the same color and hence $k = 1$, contradicting that F is nontrivial. Now let $I = \emptyset$, i.e. $H = K_1$. As for all vertices $v, x \in V(G)$ holds $v \in V(K_1^x)$ if and only if $v = x$, we can conclude that $F(a) \neq F(b)$ for all $a, b \in V(G)$. Hence $k = |V(G)|$, again contradicting that F is nontrivial. □

In the following, let F_I denote a path-k-coloring F of a Cartesian product G of S-prime graphs S_i that satisfies the conditions of Theorem 2.45 with index set I. We can now proceed proving the main result of this subsection.

Theorem 2.46. *The diagonalized Cartesian Product of S-prime graphs is S-prime.*

Proof. Let $G = H \cup (u,v)$ be a diagonalized Cartesian product of graphs S_i, i.e., $H = \square_{i=1}^{n} S_i$ is a Carte-

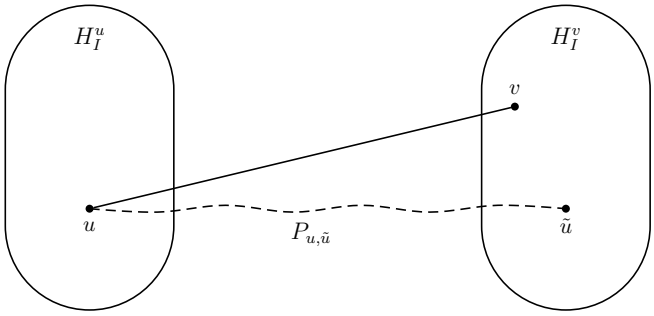


Figure 2.14: Sketch of the proof of Theorem 2.46. The H_I-layers H_I^u and H_I^v are connected by a well-colored path $P_{u,\tilde{u}}$ with distinct colors at the endpoints, $F_I(u) \neq F_I(\tilde{u})$. The path $P^* = P_{u,\tilde{u}} \cup (u,v)$ is well colored, but $F_I(u) = F_I(v)$, i.e., F_I is not a path-k-coloring.

sian product of S-prime graphs and the vertices u and v have maximal distance in H. Lemma 2.36 shows that any nontrivial path-k-coloring of G gives rise to a nontrivial path-k-coloring of H, which in turn implies that there is a nontrivial subset $I \subset I_n$ and an according nontrivial path-k-coloring F_I such that the conditions of Theorem 2.45 are satisfied for H. We can conclude that $F_I(u) \neq F_I(v)$, since otherwise the coloring of H is trivial with $k = 1$ according to Corollary 2.44 and F_I would be constant. Let H_I denote the Cartesian product $\square_{i \in I} S_i$ of prime factors of G and let H_I^u and H_I^v be the $H_I - layer$ containing u and v, respectively. Clearly, $H_I^u \neq H_I^v$, since $I \neq \{1,\ldots,n\}$, by definition of the Cartesian product and since u and v have maximal distance in H. Let $\tilde{u} \in V(S_i^v)$ be the vertex with coordinates $c_i(\tilde{u}) = c_i(u)$ for all $i \in I$. Note that $v \neq \tilde{u}$, because $c_i(\tilde{u}) = c_i(u) \neq c_i(v)$ for all $i \in I$, otherwise u and v would not have maximal distance.

Let $P_{u,\tilde{u}}$ be a path between u and $\tilde{u}$ such that for all vertices $x \in P_{u,\tilde{u}}$ holds $c_i(x) = c_i(u)$ for all $i \in I$. Thus no edge of any H_I-layer is contained in this path $P_{u,\tilde{u}}$. From Theorem 2.45 and the fact that F_I is nontrivial, it follows that $F_I(a) \neq F_I(b)$ for all $a \in V(H_I^x)$ and $b \in V(H_I^y)$ with $H_I^x \neq H_I^y$. This is true in particular also for any two distinct vertices a and b in the path $P_{u,\tilde{u}}$, since $H_I^a \neq H_I^b$ by choice of the coordinates. Thus $P_{u,\tilde{u}}$ is well colored. Moreover it holds $F_I(u) \neq F_I(\tilde{u})$.

Now consider the path $P^* = P_{u,\tilde{u}} \cup (u,v)$ in G, which is by construction a well colored path from v to $\tilde{u}$. However, $F_I(v) = F_I(\tilde{u})$. Thus F_I is not a path-k-coloring of G for any nontrivial $I \subset I_n$. Theorem 2.33 and Lemma 2.36 imply that $G = H \cup (u,v)$ is S-prime, from which the statement follows. □

Corollary 2.47. *Diagonalized Hamming graphs, and thus diagonalized Hypercubes, are S-prime.*

We conclude this section with an example that shows that not every diagonalized Cartesian product

is S-prime, see Figure 2.15.

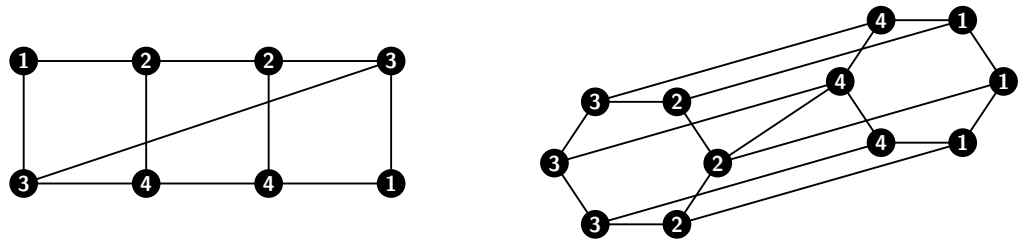


Figure 2.15: Shown are two diagonalized Cartesian products that have a nontrivial path-4-coloring. Therefore these graphs are S-composite.

3

The Local Way to Go

One easily realizes that almost all graphs are prime, see [13]. Even a small perturbation of a product graph, such as the deletion or insertion of a single edge, often leads to a prime graph, although the graph still has a product-like structure. Hence, naturally arising questions are: How can one recover the structure of a disturbed product? Is it possible to recover the original factors of a disturbed product? How can at least some parts of a disturbed product be recognized as a product?

As shown in Section 2.4.2, there are several subgraphs of a given product graph G that are itself products of subgraphs of the factors of G. This leads directly to the following idea: We try to cover a given disturbed product G by subproducts that are itself undisturbed, see Figure 3.1. If the graph G is not too much disturbed, we would expect to be able to cover most of it by 1-neighborhoods or other small subproducts and to use these information for the construction of a strong product H that approximates G. The graph G will be called *approximate* graph product.

In this chapter we introduce several important tools for the realization of this idea. We first start with the so-called *S1-condition*, that is a property of an edge, that allows us to determine Cartesian edges, even if the given graph is not thin. We then introduce the so-called *backbone* $\mathbb{B}(G)$ of a given graph G that is defined on the cardinality of equivalence classes of a particular relation S. In the last

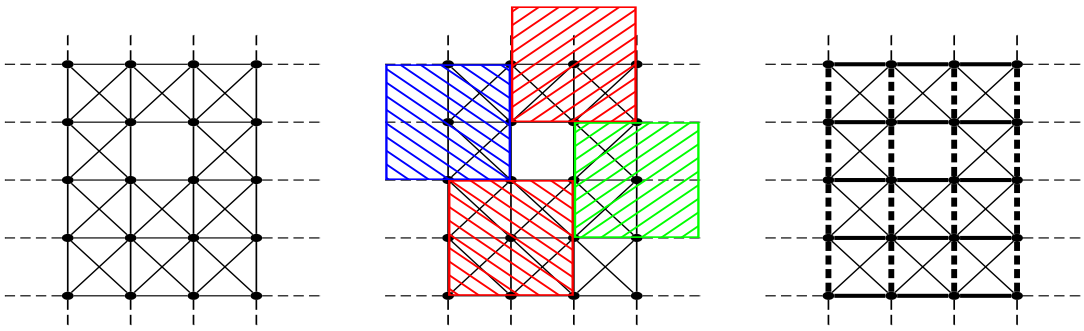


Figure 3.1: One covers a given disturbed product G by undisturbed subproducts and try use the information provided by the PFD of those subproducts for the construction of a global PFD.

part of this chapter we are concerned with the so-called *color-continuation*, that is a condition that has to be met in order to identify different local fibers as related to copies of coinciding or different global factors.

3.1 Tools

3.1.1 The S1-condition

The main idea of our approach is to construct the Cartesian skeleton of G by considering only PFDs of suitable subproducts. The main obstacle is that even though G is thin, this is not necessarily true for subgraphs, Fig. 3.2. Hence, although the Cartesian edges are uniquely determined in G, they need not to be unique in those subgraphs. In order to investigate this issue in some more detail, we also defined S-classes w.r.t. subgraphs H of a given graph G, Definition 2.14. Remind:

$$S_H(x) = \left\{v \in V(H) \mid N^G[v] \cap V(H) = N^G[x] \cap V(H)\right\}$$

As mentioned, if $H = \langle N^G[y]\rangle$ we set

$$S_y(x) := S_{\langle N^G[y]\rangle}(x) = \left\{v \in N^G[y] \mid N^G[v] \cap N^G[y] = N^G[x] \cap N^G[y]\right\}.$$

In other words, $S_y(x)$ is the S-class that contains x in the subgraph $\langle N[y]\rangle$. Notice that $N[x] \subseteq N[v]$ holds for all $v \in S_x(x)$. If G is additionally thin, then $N[x] \subsetneq N[v]$.

Since the Cartesian edges are globally uniquely defined in a thin graph, the challenge is to find a way to determine enough Cartesian edges from local information, even if $\langle N[v]\rangle$ is not thin. The following property will play a crucial role for this purpose:

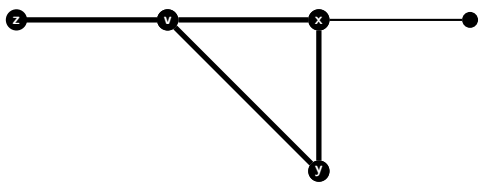


Figure 3.2: A thin graph where $\langle N[v]\rangle$ is not thin. The S-classes in $\langle N[v]\rangle$ are $S_v(v) = \{v\}$, $S_v(z) = \{z\}$ and $S_v(x) = S_v(y) = \{x,y\}$.

Definition 3.1. Given a graph G. An edge $(x,y) \in E(G)$ satisfies the *S1-condition* in an induced subgraph $H \subseteq G$ if

1. $x,y \in V(H)$ and
2. $|S_H(x)| = 1$ or $|S_H(y)| = 1$.

Note that $|S_H(x)| = 1$ for all $x \in V(H)$, if H is thin. From Lemma 2.15 we can directly infer that the cardinality of an S-class in a product graph G is the product of the cardinalities of the corresponding S-classes in the factors. Applying this fact together with Lemma 2.26 to the subgraph of G induced by a closed neighborhoods $N[v]$ immediately implies Corollary 3.2.

Corollary 3.2. *Consider a strong product $G = G_1 \boxtimes G_2$ and two vertices $v,x \in V(G)$ with coordinates (v_1,v_2) and (x_1,x_2), s.t. $v_i,x_i \in V(G_i)$ and $v_i \in N[x_i]$ for $i = 1,2$. Then $S_v(x) = S_{v_1}(x_1) \times S_{v_2}(x_2)$ and therefore $|S_v(x)| = |S_{v_1}(x_1)| \cdot |S_{v_2}(x_2)|$.*

Lemma 3.3. *Let $G = \boxtimes_{i=1}^{n} G_i$ be a strong product graph containing two S-classes $S_G(x)$, $S_G(y)$ that satisfy*

(i) *$(S_G(x), S_G(y))$ is a Cartesian edge in G/S and*

(ii) *$|S_G(x)| = 1$ or $|S_G(y)| = 1$.*

Then all edges in G induced by vertices of $S_G(x)$ and $S_G(y)$ are Cartesian and copies of one and the same factor.

Proof. For simplicity, we write $S(\,.\,)$ for $S_G(\,.\,)$. We may assume w.l.o.g. that $|S(x)| = 1$. Corollary 3.2 implies that for every factor G_i of G, $1 \leq i \leq n$, holds

$$|S_{G_i}(x_i)| = 1$$

In the following, $S(v)_m$ denotes the m-th coordinate of vertex $S(v)$ in G/S. Being a Cartesian edge means that $S(x)$ and $S(y)$ coincide in every, but one, say the j-th coordinate w.r.t. the factorization of G/S, i.e. $\forall i \neq j$ holds $S(x)_i = S(y)_i$. By Lemma 2.15 this is $S_{G_i}(x_i) = S_{G_i}(y_i)$.

Corollary 2.17 implies that the i-th coordinate ($i \neq j$) of every vertex in $S(x) \cup S(y)$ is in $S_{G_i}(x_i) \cup S_{G_i}(y_i) = S_{G_i}(x_i)$, which is a set of cardinality 1. Hence, all vertices in $S(x) \cup S(y)$ have the same i-th coordinate. This is equivalent to the claim of the lemma. □

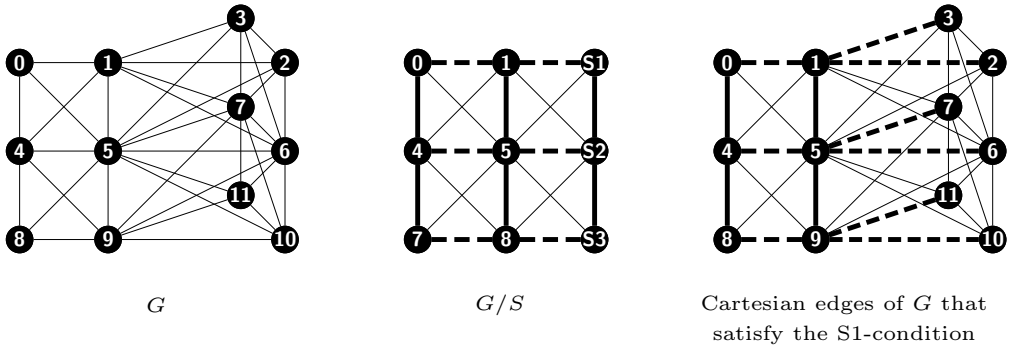


Figure 3.3: Determining Cartesian edges that satisfy the *S1-condition*. Given a graph G, one computes its quotient graph G/S. Since G/S is thin the Cartesian edges of G/S are uniquely determined. Now one factorizes G/S and computes the prime factors of G with Algorithm 1. Apply Lemma 3.3 to identify all Cartesian edges with respective colors (thick and dashed lined) in G that satisfy the *S1-condition*.

Remark 3.4. Whenever we find a Cartesian edge (x,y) in a neighborhood $\langle N[z]\rangle$ such that one endpoint of (x,y) is contained in a S-class of cardinality 1 in $\langle N[z]\rangle/S$, i.e., such that $S_z(x) = \{x\}$ or $S_z(y) = \{y\}$, we can therefore conclude that all edges in $\langle N[z]\rangle$ induced by vertices of $S_z(x)$ and $S_z(y)$ are also Cartesian and are copies of one and the same factor, see Figure 3.3.

Note, even if $\langle N[z]\rangle/S$ has more factors than $\langle N[z]\rangle$ Algorithm 1 indicates which factors have to be merged to one factor. Again we can conclude that all edges in $\langle N[z]\rangle$ that satisfy the *S1-condition* are Cartesian and are copies of one and the same factor, see Figure 3.4.

Moreover, since $\langle N[z]\rangle \subseteq G$ is a subproduct of a strong product graph G, it follows that any Cartesian edge of $\langle N[z]\rangle$ that satisfy the *S1-condition* is a Cartesian edge in G.

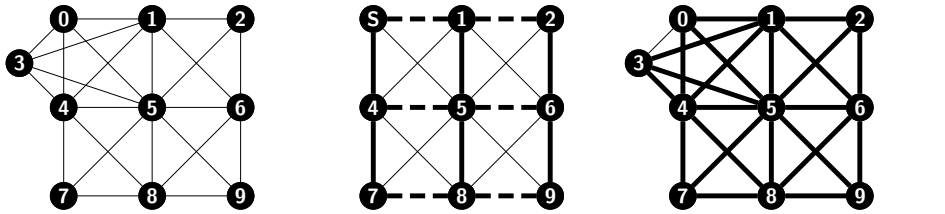
3.1.2 The Backbone $\mathbb{B}(G)$

We consider here a subset of $V(G)$ that is essential for our algorithms.

Definition 3.5. The *backbone* of a thin graph G is the vertex set

$$\mathbb{B}(G) = \{v \in V(G) \mid |S_v(v)| = 1\}.$$

Elements of $\mathbb{B}(G)$ are called *backbone vertices*.



G G/S Cartesian edges of G that satisfy the S1-condition

Figure 3.4: Determining Cartesian edges that satisfy the *S1-condition*. We factorize G/S and compute the prime factors of G with Algorithm 1. Notice that it turns out that the factors induced by thick and dashed lined edges have to be merged to one factor. Apply now Lemma 3.3 to identify all Cartesian edges in G that satisfy the *S1-condition*. In this case it is clear that the edge $(0,3)$ has to be Cartesian as well and belongs to the single prime factor G.

Clearly, the backbone $\mathbb{B}(G)$ and the *S1-condition* are closely related, since all edges (x,y) that contain a backbone vertex, say x, satisfy the *S1-condition* in $\langle N[x]\rangle$. If the backbone $\mathbb{B}(G)$ of a given graph G is nonempty then Corollary 3.2 implies that no factor of G is isomorphic to a complete graph, otherwise we would have $|S_v(v)| > 1$ for all $v \in V(G)$. The last observations lead directly to the next corollary.

Corollary 3.6. *Given a graph G with nonempty backbone $\mathbb{B}(G)$ then for all $v \in \mathbb{B}(G)$ holds: all edges $(v,x) \in E(\langle N[v]\rangle)$ satisfy the* S1-condition *in $N[v]$.*

We start exploring properties of the backbone $\mathbb{B}(G)$ of thin graphs. Our immediate goal is to establish that the backbone $\mathbb{B}(G)$ of thin graphs G is a connected dominating set. This allows us to cover the entire graph by closed neighborhoods of the backbone vertices only. Moreover, we prove that it suffices to exclusively use information about the neighborhood of backbone vertices, to find *all* Cartesian edges that satisfy the *S1-condition* in arbitrary closed neighborhoods, even those edges (x,y) with $x,y \notin \mathbb{B}(G)$

Lemma 3.7. *Let G be a thin, connected simple graph and $v \in V(G)$ with $|S_v(v)| > 1$. Then there exists a vertex $y \in S_v(v)$ s.t. $|S_y(y)| = 1$.*

Proof. Let $|S_v(v)| > 1$. Since G is finite we can choose a vertex $y \in S_v(v)$ that has a maximal closed neighborhood in G among all vertices in $S_v(v)$. Moreover $N[y]$ is maximal in G among all vertices of $V(G)$. Assume not. Then there is a vertex z s.t. $N[y] \subset N[z]$, but then $z \in S_v(v)$, a contradiction to the maximality of $N[y]$ among all vertices in $S_v(v)$. Since G is thin $N[y]$ is strictly maximal.

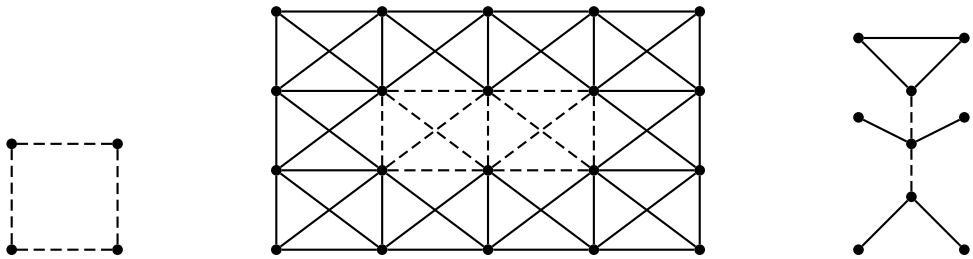


Figure 3.5: Examples of backbones. The induced subgraph of the backbone vertices of each graph is highlighted by the dashed lines.

Furthermore $|S_y(y)| = 1$, otherwise there is a $z \in S_y(y),\ z \neq y$ s.t. $N[z] \cap N[y] = N[y]$. Since G is thin, there is a $x \in N[z]$ with $x \notin N[y]$ and thus $N[y] \subsetneq N[z]$, but this is a contradiction to the fact that $N[y]$ is strictly maximal. □

Lemma 3.8. *Let G be a thin graph and v an arbitrary vertex of G. Then $v \in \mathbb{B}(G)$ if and only if $N[v]$ is a strictly maximal neighborhood in G.*

Proof. If $N[v]$ is a strictly maximal neighborhood in G then $|S_v(v)| = 1$ which is shown analogously to the last part of the last proof.

Let now $v \in \mathbb{B}(G)$. Assume $N[v]$ is not strictly maximal. Then there is a vertex $z \in V(G)$ different from v such that $N[v] \subseteq N[z]$. Thus, $N[v] \cap N[z] = N[v]$, $z \in S_v(v)$ and $|S_v(v)| > 1$, contradicting that $v \in \mathbb{B}(G)$. □

Lemma 3.9. *Let G be a thin connected simple graph. Then the backbone $\mathbb{B}(G)$ is a dominating set for G.*

Proof. We have to show that for all $v \in V(G)$ there exists a vertex $w \in N[v]$ s.t. $|S_w(w)| = 1$. If $\langle N[v] \rangle$ is thin or $|S_v(v)| = 1$, there is nothing to show. If $|S_v(v)| > 1$, then the statement follows from Lemma 3.7. □

Lemma 3.10. *Let G be a thin connected simple graph. Then the set of adjacent vertices v and w with $|S_w(w)| = 1$ or $|S_v(v)| = 1$ induces* one *connected subgraph H of G.*

Proof. Assume H consists of at least two components and let $\mathscr{C}$ denote the set of these components. Since G is connected we can choose components $C, C' \in \mathscr{C}$ s.t. there are vertices $x \in C,\ y \in C'$ that are adjacent in G. Since G is finite and $x, y \in N[x]$ there is a maximal closed neighborhood $N[z]$ in G containing x and y. The thinness of G implies that $N[z]$ is strictly maximal. This implies, analogously

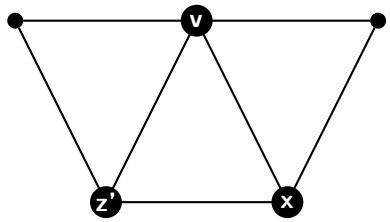


Figure 3.6: A thin graph G with backbone $\mathbb{B}(G) = \{v\}$. Thus there is no vertex $w \in N(v)$ s.t. $|S_w(w)| = 1$. Moreover notice that $|S_{z'}(x)| = 1$ but $x, z' \notin \mathbb{B}(G)$. Lemma 3.14 implies that there is a vertex $z \in \mathbb{B}(G)$ such that $|S_z(x)| = 1$. In this example holds $z = v$.

as in the proof of Lemma 3.7, that $|S_z(z)| = 1$ contradicting that x and y are in different components of H. □

Lemma 3.11. *Let G be a thin connected graph. Then the set of adjacent vertices v and w with $|S_w(w)| = 1$ and $|S_v(v)| = 1$ induces* one *connected subgraph H of G, i.e. the backbone $\mathbb{B}(G)$ induces a connected subgraph H of G.*

Proof. Assume H consists of at least two connected component. Let C be any such connected component. From Lemma 3.10 we can conclude that the subgraph M of G induced by all vertices of edges (v,w) with $|S_w(w)| = 1$ or $|S_v(v)| = 1$ is connected. Hence, in M there is path $P = \{x = x_0, x_1, x_2, ..., x_{n-1}, x_n = y\}$ from $x \in C$ to $y \in C'$, where C' is any other connected component.

W.l.o.g., we may assume that $P \cap V(C) = \{x\}$. (Otherwise we replace P by $\{x_m, x_{m+1}, ..., x_n = y\}$, where $m = \max\{i \mid x_i \in P \cap V(C)\}$.) This implies that x_1 is not in $\mathbb{B}(G)$. But then x_2 must be in a component $C'' \neq C$ from $\mathbb{B}(G)$, since every edge in M contains at least one vertex which is in $\mathbb{B}(G)$.

Notice that neither x nor x_2 are in $S_{x_1}(x_1)$, otherwise $(x, x_2) \in E(G)$ and C and C'' would be connected. By Lemma 3.7 we can choose a $z \in S_{x_1}(x_1), z \neq x, x_2$ with $|S_z(z)| = 1$. Thus C and C'' are connected. Contradiction. □

From Lemma 3.9 and Lemma 3.11 we can directly infer the next Theorem.

Theorem 3.12. *Let G be a thin graph. Then the backbone $\mathbb{B}(G)$ is a connected dominating set for G.*

Notice that if G is thin and $|\mathbb{B}(G)| > 1$, then *every* vertex has an adjacent vertex that is in the backbone. Clearly this is not true whenever $|\mathbb{B}(G)| = 1$, as the example in Figure 3.6 shows.

Lemma 3.13. *Let G be a thin graph with a backbone consisting of a single vertex $\mathbb{B}(G) = \{v\}$. Then $|S_v(w)| = 1$ for all $w \in V(G)$.*

Proof. Theorem 3.12 implies that $\langle N[v] \rangle \simeq G$ and thus $S_v(w) = S_G(w)$ for all $w \in V(G)$. Since G is thin every S class in G is trivial and therefore also in $\langle N[v] \rangle$. □

Lemma 3.14. *Let G be a thin graph and (x,y) an arbitrary edge in $E(G)$. If there exists a vertex $z' \in N[x] \cap N[y]$ with $|S_{z'}(x)| = 1$ then there exists even a vertex $z \in N[x] \cap N[y]$ with the following properties:*

$$z \in \mathbb{B}(G) \text{ and } |S_z(x)| = 1.$$

Proof. If $z' \in \mathbb{B}(G)$ there is nothing to show.
Now suppose $|S_{z'}(z')| > 1$. By Lemma 3.7 we can choose a vertex $z \in S_{z'}(z')$ with $|S_z(z)| = 1$. Since $z \in S_{z'}(z')$, we can conclude that $N[z'] \subset N[z]$ and thus $x,y \in N[z]$ and therefore $z \in N[x] \cap N[y]$.
It remains to show that $|S_z(x)| = 1$. Assume $|S_z(x)| > 1$ then there is a vertex $w \in S_z(x)$ different from x. The definition of $S_z(x)$ implies $N[w] \cap N[z] = N[x] \cap N[z]$, which implies that $w \in N[z']$, since $z' \in N[x] \cap N[z]$. Moreover we can conclude

$$N[w] \cap N[z] \cap N[z'] = N[x] \cap N[z] \cap N[z']. \tag{3.1.1}$$

Since $N[z'] \subset N[z]$, we can cancel the intersection with $N[z]$ in equation 3.1.1 to obtain

$$N[w] \cap N[z'] = N[x] \cap N[z'].$$

But then $w \in S_{z'}(x)$ and thus $|S_{z'}(x)| > 1$, contradicting $|S_{z'}(x)| = 1$. Hence $|S_z(x)| = 1$. □

Lemma 3.15. *Let $(x,y) \in E(G)$ be an arbitrary edge in a thin graph G such that $|S_x(x)| > 1$. Then there exists a vertex $z \in \mathbb{B}(G)$ s.t. $z \in N[x] \cap N[y]$.*

Proof. Since $|S_x(x)| > 1$ and by applying Lemma 3.7 we can choose a vertex $z \in S_x(x)$ with $z \in \mathbb{B}(G)$. Since $z \in S_x(x)$ it holds $N[x] \subset N[z]$ and hence $y \in N[z]$, and the claim follows. □

Corollary 3.16. *Let G be a thin graph and (x,y) an arbitrary edge in $E(G)$ that does not satisfy the* S1-condition *in any* 1*-neighborhood. Then there exists a vertex $z \in \mathbb{B}(G)$ s.t. $z \in N[x] \cap N[y]$, i.e. the edges (z,x) and (z,y) satisfy the* S1-condition *in $\langle N[z] \rangle$.*

We prove now that if at least one edge of a fiber G_i^x satisfies the *S1-condition* in a 1-neighborhood, then all vertices contained in G_i^x have an endpoint in an edge $e \in E(G_i^x)$ that satisfies the *S1-condition* in a 1-neighborhood.

Lemma 3.17. *Let $G = \boxtimes_{j=1}^{n} G_j$ be the strong product of thin graphs and $(x,y) \in E(G)$ be a Cartesian edge, where x and y differ in coordinate i. Moreover let (x,y) satisfy the* S1-condition *in a* 1*-neighborhood. Then for all edges $(a,b) \in E(G_i^x)$ at least one of the following statements is true:*

1. *(a,b) satisfies the* S1-condition *in a* 1*-neighborhood.*
2. *There are edges $(\tilde{z},a),(\tilde{z},b) \in E(G_i^x)$ that satisfy the* S1-condition *a* 1*-neighborhood.*
   In this case, knowing that $(\tilde{z},a),(\tilde{z},b)$ belong to G_i^x implies that (a,b) is necessarily also an edge of G_i^x.

Furthermore, the vertices incident with edges of G_i^x that satisfy the S1-condition *in* 1*-neighborhoods induce a single connected subgraph $H \subseteq G_i^x$.*

Proof. By associativity and commutativity of the strong product it suffices to show this for the product $G = G_1 \boxtimes G_2$ of two thin (not necessarily prime) graphs. Notice that $G_i^x = G_i^y$, since x and y differ only in coordinate i. Furthermore let (x_1,x_2) denote the coordinates of x. The notation of the coordinates of a, b, and y is analogous. W.l.o.g. assume $i = 2$ and $|S_z(x)| = 1$ with $z = (z_1,z_2) \in N[x] \cap N[y]$. Corollary 3.2 implies $|S_{z_1}(x_1)| = 1$ and $|S_{z_2}(x_2)| = 1$. The idea of the rest of the proof is to shift properties of (a_2,b_2), the projection of (a,b) into the factor G_2, to (a,b).

Case (a) (a_2,b_2) satisfies the *S1-condition* in a 1-neighborhood w.r.t. G_2. Then we may assume w.l.o.g. that there is a $v_2 \in G_2$ with $|S_{v_2}(a_2)| = 1$ and $a_2,b_2 \in N[v_2]$. Since $x_1 = a_1$, Corollary 3.2 implies $|S_{(z_1,v_2)}(a)| = 1$. Lemma 2.5 shows that $a,b \in N[(z_1,v_2)]$. Hence (a,b) satisfies the *S1-condition* in $N[(z_1,v_2)]$.

Case (b) (a_2,b_2) does not satisfy the *S1-condition* a 1-neighborhood w.r.t. G_2. Then Corollary 3.16 implies the existence of a vertex $v_2 \in G_2$ such that both (v_2,a_2) and (v_2,b_2) satisfy the *S1-condition* in $N^{G_2}[v_2]$. Case (a) shows that $((a_1,v_2),a)$ and $((a_1,v_2),b)$ satisfy the *S1-condition* in the respective 1-neighborhood.

Since $\mathbb{B}(G_2)$ is a connected dominating set for G_2, the subgraph of G_2 induced by all vertices of edges that satisfy the *S1-condition* in 1-neighborhoods w.r.t. G_2 is connected. Since we can shift every edge that satisfies the *S1-condition* in a 1-neighborhood w.r.t. G_2 to an edge that satisfies the *S1-condition* in a 1-neighborhood w.r.t. G in G_i^x, H is connected. □

From Lemma 3.14 and 3.17 we can directly conclude the next Theorem. that highlights the impor-

tance of $\mathbb{B}(G)$ for the identification of Cartesian edges.

Theorem 3.18. *All Cartesian edges that satisfy the* S1-condition *in an arbitrary induced neighborhood also satisfy the* S1-condition *in the induced neighborhood of a vertex of the backbone* $\mathbb{B}(G)$. *If at least one edge of* G_i^x *in* $G = \boxtimes_{j=1}^n G_j$ *satisfies the* S1-condition *in a 1-neighborhood, then all vertices of* G_i^x *are contained in edges of* G_i^x *that satisfy the* S1-condition *in 1-neighborhoods.*

This result implies that it makes sense to give the following definition:

Definition 3.19. An entire G_i^x-fiber satisfies the *S1-condition* in 1-neighborhoods, whenever one of its edges does.

Taken together, the latter results allow us to identify all Cartesian edges of G_i^x-fiber that satisfy the *S1-condition* in 1-neighborhoods, using exclusively information about the 1-neighborhoods of the backbone vertices.

Last, we will show that for a given subproduct H of a thin graph G that entirely contains at least one 1-neighborhood of a backbone vertex $x \in \mathbb{B}(G)$, the set of Cartesian edges of H that satisfy the *S1-condition* in H, induce a connected subgraph of H. This holds even if H is not thin. For this we need the next two lemmas.

Lemma 3.20. *Let G be a given thin graph, $x \in \mathbb{B}(G)$ and $H \subseteq G$ an arbitrary induced subgraph such that $N[x] \subseteq V(H)$. Then $|S_H(x)| = 1$ and $x \in \mathbb{B}(H)$.*

Proof. First notice that Lemma 3.8 and $x \in \mathbb{B}(G)$ implies that $\langle N[x] \rangle$ is strictly maximal in G. Since $\langle N[x] \rangle \subseteq H \subseteq G$ we can conclude that $\langle N[x] \rangle$ is strictly maximal in H. Hence, it holds $|S_H(x)| = 1$ and in particular $x \in \mathbb{B}(H)$, applying Lemma 3.8 again. □

Lemma 3.21. *Let G be a given thin graph and $H \subseteq G$ be a subproduct of G such that there is a vertex $x \in \mathbb{B}(G)$ with $N[x] \subseteq V(H)$. Then the set of all Cartesian edges of H that satisfy the* S1-condition *in H induce a connected subgraph of H.*

Proof. Let $\boxtimes_{i=1}^n H_i$ be any factorization of H and (a,b) be an arbitrary Cartesian edge of H (w.r.t. to this factorization) that satisfies the *S1-condition* in H. W.l.o.g we assume that $|S_H(a)| = 1$. We denote the coordinates of a with $(a_1, \ldots, a_n)$ and the ones of x with $(x_1, \ldots, x_n)$. Clearly, the coordinatization need not to be unique, since H is not supposed to be thin. However, we will construct a path P from a to x that consists of Cartesian edges (v,w) such that $|S_H(v)| = 1$ and $|S_H(w)| = 1$. Those Cartesian edges are uniquely determined in H, independently from the coordinatization.

Notice that Lemma 3.20 implies that $|S_H(x)| = 1$, since $N[x] \subseteq V(H)$. Moreover, from Corollary 3.2 we can conclude that $|S_{H_i}(x_i)| = 1$ for all i. Analogously, $|S_{H_i}(a_i)| = 1$ for all i. The index set I denotes the set of position where a and x differ. W.l.o.g we assume that $I = \{1,2,\ldots,k\}$. The path P has edge set $\{(x,v^1),(v^2,v^3),\ldots,(v^{k-1},a)\}$ with vertices v^j that have respective coordinates $(a_1,a_2,\ldots,a_j,x_{j+1},\ldots,x_n)$, $j = 1,\ldots,k-1$. Corollary 3.2 implies that for all those vertices holds $|S_H(v^k)| = 1$ and hence in particular for all edges $(u,w) \in \{(x,v^1),(v^2,v^3),\ldots,(v^{n-1},a)\}$ holds $|S_H(u)| = 1$ and $|S_H(w)| = 1$, i.e., those Cartesian edges are uniquely determined in H. Finally, since all edges have endpoints differing in exactly one coordinate all edges are Cartesian and hence all those Cartesian edges (a,b) are connected to vertex x by a path of Cartesian edges that satisfy the *S1-condition*, from what the statement follows. □

Corollary 3.22. *Let G be a given thin graph, $x \in \mathbb{B}(G)$ and let $H \subseteq G$ denote one of the subproducts $\langle N[x]\rangle$, $N^*_{x,y}$ or $\langle N[x] \cup N[y]\rangle$. In the latter case we assume that the edge (x,y) is Cartesian in H. Then the set of all Cartesian edges of H that satisfy the* S1-condition *in H induce a connected subgraph of H.*

3.1.3 The Color-Continuation

The concept of covering a graph by suitable subproducts and to determine the global factors needs some additional improvements. Since we want to determine the global factors, we need to find their fibers. This implies that we have to identify different locally determined fibers as belonging to different or belonging to one and the same global fiber. For this purpose, we formalize the term *product coloring*, *color-continuation* and *combined coloring*.

Definition 3.23. A *product coloring* of a strong procuct graph $G = \boxtimes_{i=1}^n G_i$ of $n \geq 1$ (not necessarily prime) factors is a mapping P_G from a subset $E' \subseteq E(G)$, that is a set of Cartesian edges of G, into a set $C = \{1,\ldots,n\}$ of colors, such that all such edges in G_i-fibers receive the same color i.

Definition 3.24. A *partial product coloring* of a graph $G = \boxtimes_{i=1}^n G_i$ is a product coloring that is only defined on edges that additionally satisfy the *S1-condition* in G.

Note, in a thin graph G a product coloring and a partial product coloring conincide, since all edges statisfy the *S1-condition* in G.

Definition 3.25. Let $H_1,H_2 \subseteq G$ and P_{H_1}, resp. P_{H_2}, be partial product colorings of H_1, resp. H_2. Then P_{H_2} is a *color-continuation* of P_{H_1} if for every color c in the image of P_{H_2} there is an edge in H_2 with color c that is also in the domain of P_{H_1}.

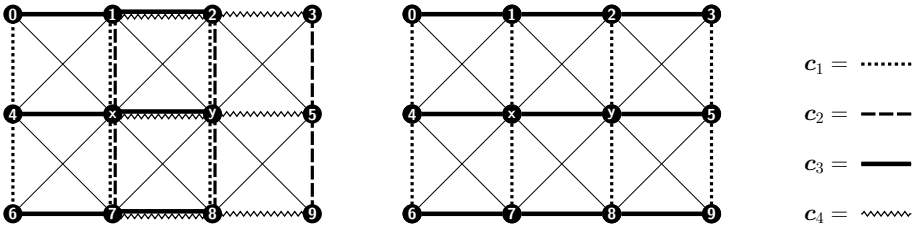


Figure 3.7: Shown is a thin graph G that is a strong poduct of two pathes. If one computes the PFD of the neighborhood $\langle N[x]\rangle$ one receives a (partial) product coloring with colors c_1 and c_3. The (partial) product coloring of $\langle N[y]\rangle$ has colors c_2 and c_4. Since on edge (x,y), resp. $(x,1)$, both colors c_1 and c_2, resp. c_3 and c_4 are represented we can identify those colors and merge them to one color. Hence, the product coloring $P_{\langle N[x]\rangle}$ is a color-continuation of $P_{\langle N[y]\rangle}$ and vice versa.

The *combined coloring* on $H_1 \cup H_2$ uses the colors of P_{H_1} on H_1 and those of P_{H_2} on $H_2 \setminus H_1$.

In other words, for all newly colored edges with color c in H_2, which are Cartesian edges in H_2 that satisfy the *S1-condition* in H_2, we have to find a representative edge that satisfy the *S1-condition* in H_1 and was already colored in H_1. If H_1 and H_2 are thin we can ignore the *S1-condition*, since all edges satisfy this condition in H_1 and H_2, see Figure 3.7.

In Chapter 5 we are concerned with so-called *locally unrefined* graphs. For this we introduce a particular product coloring that is a restricted version of the previous definitions. Here we claim only that all edges of a particular G_i-fiber G_i^x receive the same color. For an example see Figure 3.8.

Definition 3.26. Let G_j^x be a fiber of an arbitrary factor G_j of G. An (x,j)- *product coloring* of a graph $G = \boxtimes_{i=1}^n G_i$ is a mapping F_G from a subset E' of the set of Cartesian edges of G into a set C of colors, such that all edges in this particular G_j^x-fiber receive the same color.

Definition 3.27. Let G_j^x be a fiber of an arbitrary factor G_j of G. An (x,j)- *partial product coloring ((x,j)-PPC)* of a graph $G = \boxtimes_{i=1}^n G_i$ is a (x,j)- product coloring that is only defined on edges that additionally satisfy the *S1-condition* in G.

Definition 3.28. Let $H_1, H_2 \subset G$ and F_{H_1} be a (x,j)-PPC of H_1. Then F_{H_2} is a (x,j)-*color-continuation* of F_{H_1} if there is a color c in the image of F_{H_2} that is also in the domain of F_{H_1}. More formally:

$$\exists \text{ edge } e \in Dom(F_{H_1}) \cap Dom(F_{H_2}) \cap E(G_j^x)$$

that satisfies the *S1-condition* in both H_1 and H_2.
The *combined* (x,j)-*PPC* on $H_1 \cup H_2$ uses the color of F_{H_1} on H_1 and colors all edges f of H_2 with $F_{H_2}(f) = c$ with the color $F_{H_1}(e)$.

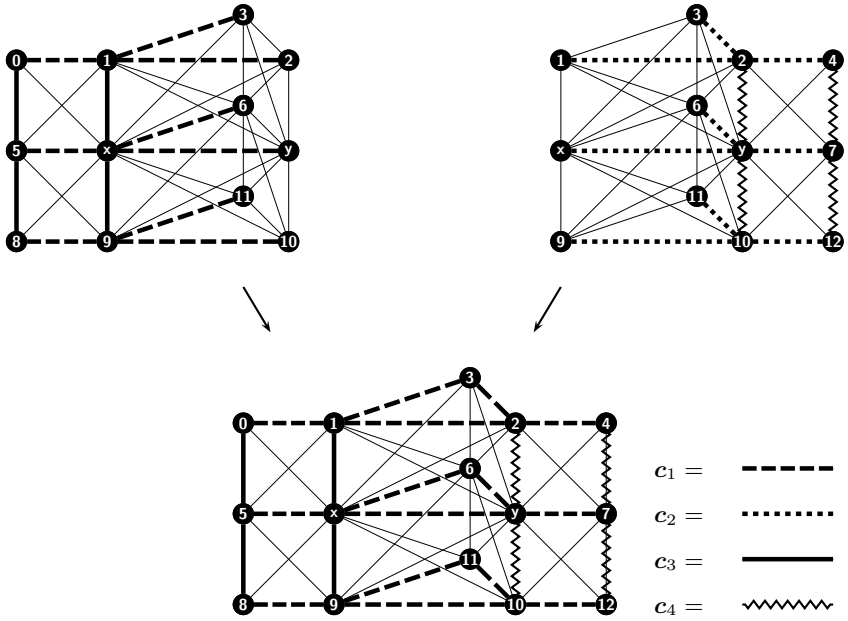


Figure 3.8: Shown is a thin graph G that is a strong poduct of a path and a path containing a triangle. The backbone $\mathbb{B}(G)$ consists of the vertices x and y. Both neighborhoods $\langle N[x]\rangle$ and $\langle N[y]\rangle$ are not thin. After computing the PFD of $\langle N[x]\rangle$, resp. of $\langle N[y]\rangle$ one receives a partial product coloring with colors c_1 and c_3, resp. with colors c_2 and c_4. In this example the partial product coloring of $P_{\langle N[y]\rangle}$ is not a color-continuation of $P_{\langle N[x]\rangle}$ since no edge with color c_4 is colored in $\langle N[x]\rangle$. If we denote the factor induced by one component of dashed-line fibers by G_1 we can observe that the $(x,1)$- partial product coloring $F_{\langle N[y]\rangle}$ is a $(x,1)$-color-continuation of $F_{\langle N[x]\rangle}$ and vice versa.

We will now provide several properties of (partial) product colorings. The next Lemma, which was stated for equivalence classes w.r.t. to a product relation in [34], is a restatement of Lemma 2.9.

Lemma 3.29 ([34])**.** *Let G be a thin strong product graph and let P_G be a product coloring of G. Then every vertex of $V(G)$ is incident to at least one edge with color c for all colors c in the image of P_G.*

Lemma 3.30. *Let G be a thin strong product graph, $H \subseteq G$ be a non-thin subproduct of G and $x \in V(H)$ be a vertex with $|S_H(x)| = 1$. Moreover let P_H be a partial product coloring of H. Then vertex x is contained in at least one edge with color c for all colors c in the image of P_G.*

Proof. Notice that H does not contain complete factors, otherwise Corollary 3.2 implies that $|S_H(x)| > 1$. Now, the statement follows directly from Lemma 3.3, Corollary 3.6 and Lemma 3.29 □

We show in the following that in a given thin strong product graph G a partial product coloring

P_H of a subproduct $H \subseteq G$ is always a color-continuation of a partial product coloring $P_{\langle N[x]\rangle}$ of any neighborhood $N[x]$ with $N[x] \subseteq V(H)$ and $x \in \mathbb{B}(G)$ and vice versa.

Lemma 3.31. *Let G be a thin graph and $x \in \mathbb{B}(G)$. Moreover let P^1 and P^2 be arbitrary partial product colorings of the induced neighborhood $\langle N[x]\rangle$.*

Then P^2 is a color-continuation of P^1 and vice versa.

Proof. Let C^1 and C^2 denote the images of P^1 and P^2, respectively. Note, that the PFD of $\langle N[x]\rangle$ is the finest possible decomposition, i.e. the number of used colors becomes maximal. Moreover every fiber with respect to the PFD of $\langle N[x]\rangle$ that satisfies the *S1-condition*, is contained in any decomposition of $\langle N[x]\rangle$. In other words any prime fiber that satisfies the *S1-condition* is a subset of a fiber that satisfies the *S1-condition* with respect to any decomposition of $\langle N[x]\rangle$.

Moreover since $x \in \mathbb{B}(G)$ it holds that $|S_x(x)| = 1$ and thus every edge containing vertex x satisfies the *S1-condition* in $\langle N[x]\rangle$. Lemma 3.3 implies that all Cartesian edges (x,v) can be determined as Cartesian in $\langle N[x]\rangle$ Together with Lemma 3.30 we can infer that each color of C^1, resp. C^2 is represented at least on edges (x,v) contained in the prime fibers, which completes the proof. □

Lemma 3.32. *Let $G = \boxtimes_{i=1}^{n} G_i$ be a thin strong product graph. Furthermore let H be a subproduct of G with partial product coloring P_H and $\langle N[x]\rangle \subseteq H$ with $x \in \mathbb{B}(G)$.*

Then P_H is a color-continuation of the partial product coloring P_N of $\langle N[x]\rangle$ and vice versa.

Proof. First notice that Lemma 3.20 implies that $x \in \mathbb{B}(H)$ and in particular $|S_H(x)| = 1$. Thus every edge containing vertex x satisfies the *S1-condition* in H as well as in $\langle N[x]\rangle$. Moreover Lemma 3.30 implies that every color of the partial product coloring P_H, resp. P_N, is represented at least on edges (x,v).

Since $\langle N[x]\rangle$ is a subproduct of the subproduct H of G we can conclude that the PFD of H induces a local (not neccessarily prime) decomposition of $\langle N[x]\rangle$ and hence a partial product coloring of $\langle N[x]\rangle$. Lemma 3.31 implies that any partial product coloring of $\langle N[x]\rangle$ and hence in particular the one induced by P_H is a color-continuation of P_N.

Conversely, any product coloring P_N of $\langle N[x]\rangle$ is a color-continuation of the product coloring induced by the PFD of $\langle N[x]\rangle$. Since $\langle N[x]\rangle$ is a subproduct of H it follows that every prime fiber of $\langle N[x]\rangle$ that satisfies the *S1-condition* is a subset of a prime fiber of H that satisfies the *S1-condition*. This holds in particular for the fibers through vertex x, since $|S_x(x)| = 1$ and $|S_H(x)| = 1$. By the

same arguments as in the proof of Lemma 3.31 one can infer that every product coloring of H is a color-continuation of the product coloring induced by the PFD of H, which completes the proof. □

We can infer now the following Corollaries.

Corollary 3.33. *Let $G = \boxtimes_{i=1}^{n} G_i$ be a thin strong product graph, $(v,w) \in E(G)$ be a Cartesian edge of G and H denote the edge-neighborhood $\langle N[v] \cup N[w] \rangle$. Then the partial product coloring P_H of H is a color-continuation of the partial product coloring $P_{N[v]}$ of $\langle N[v] \rangle$, resp. of the partial product coloring $P_{N[w]}$ of $\langle N[w] \rangle$ and vice versa.*

Corollary 3.34. *Let $G = \boxtimes_{i=1}^{n} G_i$ be a thin strong product graph and $(v,w) \in E(G)$ be an arbitrary edge of G. Then the partial product coloring P^* of the $N^*_{v,w}$-neighborhood is a color-continuation of the partial product coloring $P_{N[v]}$ of $\langle N[v] \rangle$, resp. of the partial product coloring $P_{N[w]}$ of $\langle N[w] \rangle$ and vice versa.*

4

NICE and CHIC Graphs

Given a graph G we want to recognize its prime factors by covering G by suitable subproducts $H \subseteq G$. If those subproducts H are thin and hence, $|S_H(v)| = 1$ for all $v \in V(H)$, then *all* Cartesian edges in H are uniquely determined. Thus, a first natural way to cover G would be covering it by thin subproducts H only. Graphs that can be covered by thin 1-neighborhoods only will be called *thin-N coverable*. As it turns out not all graphs have this property, but we will introduce large classes of thin-N coverable graphs, so-called *NICE* and *CHIC* graphs and show that the information provided by the local PFDs is sufficient to determine the prime factors of those graphs. Moreover, we will derive quasi-linear time algorithms that determine the prime factors of NICE and CHIC graphs using 1-neighborhood information only.

4.1 Thin-N coverable Graphs

Definition 4.1. A graph G is *thin-N coverable* if there is a dominating set σ of G such that for all $v \in \sigma$ holds $\langle N[v] \rangle$ is thin. We call σ a *thin* dominating set. If σ is ordered, we denote it with $\sigma^{\gg}$ and call it *covering sequence*.

We give now a characterization of thin-N coverable graph.

Lemma 4.2 ([28]). *Let $G = \boxtimes_{i=1}^{n} G_i$ be a strong product. G is thin-N coverable if and only if all of its factors are thin-N coverable.*

Proof. By associativity and commutativity of the strong product it suffices to show this for the product $G = G_1 \boxtimes G_2$ of two (not necessarily prime) graphs.

Suppose every factor is thin-N coverable. Hence there are thin dominating sets $\sigma_i \subseteq V(G_i)$, $i = 1,2$. Then, the neighborhoods of vertices in $\sigma_1 \times \sigma_2$ cover G. To see this we choose $v = (v_1, v_2) \in V(G)$ arbitrarily. By the choice of σ_i there are thin neighborhoods $N[v_i']$ that contain v_i and from Corollary 2.16 and Lemma 2.26 we can conclude that $N[(v_1', v_2')]$ is a thin neighborhood containing v.

For the converse let $v_i \in V(G_i)$ be arbitrarily chosen. Let $v \in V(G)$ with i-th coordinate v_i. By assumption it is in the thin closed neighborhood of some vertex v', thus by Lemma 2.26 vertex v_i is contained in $N[v_i']$, the neighborhood of the i-th coordinate of v' in G_i. Corollary 2.16 implies that $N[v_i']$ is thin. □

Clearly, if $v \in \sigma$ then $|S_v(v)| = 1$ and hence $v \in \mathbb{B}(G)$. Therefore, it holds $\sigma \subseteq \mathbb{B}(G)$. Notice that thin-N coverable does not imply that all edges of G are covered by thin induced neighborhoods, see Figure 4.1.

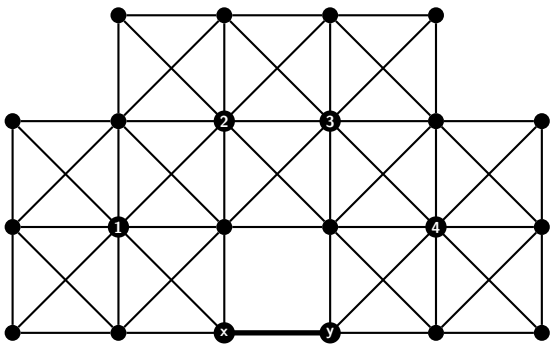


Figure 4.1: Shown is a thin-N coverable graph G with thin dominating set $\sigma = \{1,2,3,4\}$. Notice, that in this example holds $\sigma = \mathbb{B}(G)$. The thick edge (x,y) cannot be covered by thin neighborhoods, since neither $\langle N[x] \rangle$ nor $\langle N[y] \rangle$ is thin.

The class of NICE, respectively CHIC graphs are defined as subclasses of thin-N coverable graphs that satisfy some conditions. NICE graphs were first introduced in [28]. For the recognition of the prime factors of a given NICE graph, the introduced algorithm requires a covering sequence $\sigma^{\gg} = \{v_1, \ldots, v_k\}$ that guarantees that the color-continuation from $\langle N[v_i] \rangle$ to $\langle N[v_{i+1}] \rangle$ never fails.

However, recognizing whether such a covering sequence exists in general and if so determining it, is not provided by this algorithm and indeed a disadvantage and the main obstacle for a fast and constructive approach.

For CHIC graphs we do not need such an ordering of σ, but we claim, in distinction from NICE graphs, that the induced subgraph $\langle\sigma\rangle$ is connected. However, we will show how to solve the problem if the color-continuation fails.

As it turns out, the class of NICE and CHIC graphs has a non-empty intersection but nevertheless they are not identical. For an example of a graph that is NICE and CHIC see Figure 4.1.

4.2 NICE

In this section we briefly summarize the results of [28]. We start with the definition of NICE graphs.

Definition 4.3. A graph G is *thin-N intersection coverable*, in short *NICE*, if it has a covering sequence $\sigma^{\gg} = \{v_1, \ldots, v_k\}$ such that for all $i = 1, \ldots, k-1$ the product coloring of $\langle N[v_{i+1}]\rangle$ is a color-continuation of the combined coloring of $\bigcup_{j=1}^{i} E(\langle N[v_j]\rangle)$ defined by the product colorings of each $\langle N[v_j]\rangle$.

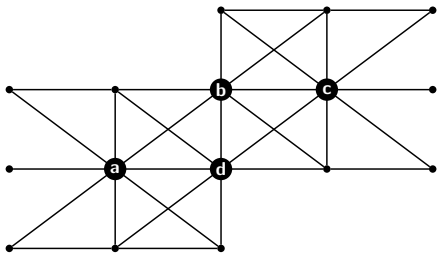


Figure 4.2: A prime graph G with $\sigma = \{a, c\}$ and $\mathbb{B}(G) = \{a, b, c, d\}$ that can be covered by thin neighborhoods only. Both thin neighborhoods $\langle N[a]\rangle$ and $\langle N[c]\rangle$ are prime and thus all edges receive the same color. Therefore the single color used in each neighborhood can be continued on the edge (b,d). Hence G is NICE. Notice that the induced subgraph $\langle\sigma\rangle$ is not connected.

As shown in [28] the product of NICE graphs is a NICE graph.

Lemma 4.4. *Let $G = \boxtimes_{i=1}^{n} G_i$ be a strong product graph for which all factors are NICE. Then G is NICE.*

We give now a short overview of Algorithm 3 that decomposes NICE graphs with given covering sequence $\sigma^{\gg}$ into its prime factors.

Algorithm 3 NICE graph decomposition

1: **INPUT:** a NICE graph G with a covering sequence $\sigma^{\gg} = \{v_1, \ldots, v_k\}$
2: compute PFD of $\langle N[v_1] \rangle$ and properly color its Cartesian edges;
3: **for** i=2,...,k **do**
4: Compute PFD of $\langle N[v_i] \rangle$ and properly color its Cartesian edges;
5: compute the combined coloring of $\langle \cup_{j=1}^{i-1} N[v_j] \rangle$ and $\langle N[v_i] \rangle$;
6: **end for**
7: $I \leftarrow \{1, \ldots, num\_comp\}$;
8: $J \leftarrow I$;
9: **for** $k = 1$ to num_comp **do**
10: **for** each $S \subset J$ with $|S| = k$ **do**
11: compute two connected components A, A' of G induced by the colored edges of G with color $i \in S$, and $i \in I \backslash S$, resp;
12: compute $H_1 = \langle p_A(G) \rangle$ and $H_2 = \langle p_{A'}(G) \rangle$;
13: **if** $H_1 \boxtimes H_2 \backsimeq G$ **then**
14: save H_1 as prime factor;
15: $J \leftarrow J \backslash S$;
16: **end if**
17: **end for**
18: **end for**
19: **OUTPUT:** The prime factors of G;

In the first part (line 2 – 6) every induced neighborhood of vertices in the order of their appearance in the covering sequence $\sigma^{\gg}$ is decomposed with respect to the strong product, all the product colorings of the induced neighborhoods are combined in order to obtain a partial product coloring of G. It might happen that the coloring returned by the first part of the algorithm is finer than the coloring of the global PFD of G, for an example see Figure 4.3. Every induced neighborhood $\langle N[x] \rangle$ is a strong product of two factors, but the graph itself is prime. Another example can be seen in Figure 4.5. Thus, colors may need to be combined to determine the factors of the global PFD which is performed in the second part of the algorithm (line 7 – 18). Finally, the algorithm returns the prime factors of G.

As shown in [28], Algorithm 3 computes the PFD of NICE graphs in quasi-linear time.

Theorem 4.5. *For a NICE graph $G = (V, E)$ with bounded maximum degree Δ and given covering sequence Algorithm 3 determines the prime factors of G w.r.t. the strong product in $O(|V|\Delta^4)$ time.*

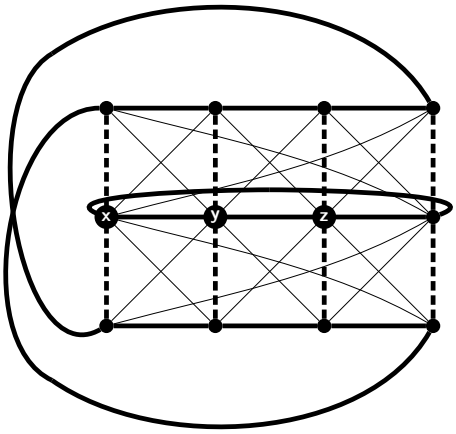


Figure 4.3: A so-called *twisted* product with covering sequence $\sigma = \{x, y, z\}$, with product coloring (induced by tick and dashed edges) after running the first part of Algorithm 3. The components are combined in the second part.

4.3 CHIC

As mentioned before, the main disadvantage of the approach for NICE graphs is that a covering sequence must be provided, which guarantees that the color-continuation always works. Unfortunately, there is no algorithm known that determines if a graph is NICE and computes such a covering sequence. Clearly, one could exhaustively enumerate all possibilities for such a sequence $\sigma^{\gg}$ and test if the color-continuation works, but this is not efficient at all. To solve this problem we introduce the class of CHIC graphs that is a subclass of thin-N coverable graphs. In distinction from NICE graphs we abandon that the vertices of the covering sequence can be ordered with respect to Definition 4.3. Thus, the color-continuation does not need to work as for NICE graphs. Instead, we suppose that the thin dominating set σ is a *connected* dominating set.

Definition 4.6. A graph G is *connected thin-N coverable*, in short *CHIC*, if it has a *connected* thin dominating set σ, i.e., the subgraph induced by σ is connected.

Notice that we can order the vertices of σ via a BFS-ordering applied in the induced subgraph $\langle\sigma\rangle$, since $\langle\sigma\rangle$ is connected. In the sequel we assume that $\sigma^{\gg}$ is ordered in this way. We show now that the product of CHIC graphs is again a CHIC graph.

Lemma 4.7. *Let $G = \boxtimes_{i=1}^{n} G_i$ be a strong product graph for which all factors are CHIC. Then G is CHIC.*

Proof. Since the strong product is commutative and associative it suffices to show this for the product

$G = G_1 \boxtimes G_2$ of two CHIC (not necessarily prime) graphs. From Lemma 4.2 we can conclude that G is thin-N coverable. Let $\sigma_1 = \{x_1, \ldots, x_k\}$ and $\sigma_2 = \{y_1, \ldots, y_m\}$ be connected thin dominating sets of G_1, resp. G_2. Corollary 2.16 and Lemma 2.26 imply that the induced neighborhood $\langle N[v] \rangle$ with coordinates $v = (x_i, y_j)$ and $x_i \in \sigma_1$, $y_j \in \sigma_2$ is thin. Moreover, by definition of the strong product and since each factor can be covered by respective neighborhoods of each $x_i \in \sigma_1$ and each $y_j \in \sigma_2$ we can infer that the whole graph G is covered by the neighborhoods of those vertices $v = (x_i, y_j)$. Thus, the set σ consisting of all vertices $v = (x_i, y_j)$ with $x_i \in \sigma_1$ and each $y_j \in \sigma_2$ is a thin dominating set. Since $\langle \sigma_1 \rangle$ and $\langle \sigma_2 \rangle$ induce connected subgraphs in each factor G_1, respectively G_2 we can apply Lemma 2.3 and conclude that the product of $\langle \sigma_1 \rangle$ and $\langle \sigma_2 \rangle$ is connected. Thus $\sigma = \{v_{i,j} \mid v_{i,j} = (x_i, y_j), i = 1, \ldots, k, j = 1, \ldots, m\}$ is a connected thin dominating set for G. □

4.3.1 Solving the Color-Continuation Problem

As argued, we do not demand that the covering sequence $\sigma^{\gg}$ guarantees that the color-continuation always works. Indeed, there are examples where the color-continuation fails, see Figure 4.4. In the following we discuss this problem and show how to solve it. First we prove a lemma for later usage.

Lemma 4.8. *Let $G = \boxtimes_{l=1}^{n} G_l$ be a thin strong product graph and $(v,w) \in E(G)$ a non-Cartesian edge. Let J denote the set of indices where v and w differ and $U \subseteq V(G)$ be the set of vertices u with coordinates $u_i = v_i$, if $i \notin J$ and $u_i \in \{v_i, w_i\}$, if $i \in J$. Then the induced subgraph $\langle U \rangle \subseteq \mathbb{S}(G)$ on U consisting of Cartesian edges of G only is a hypercube of dimension $|J|$.*

Proof. Notice that the coordinization of G is unique, since G is thin. Moreover, since the strong product is commutative and associative we can assume w.l.o.g. that $J = \{1, \ldots, k\}$. Note, that $k > 1$, otherwise the edge (v,w) would be Cartesian.

Assume that $k = 2$. We denote the coordinates of v, resp. of w, by (v_1, v_2, X), resp. by (w_1, w_2, X). By defintion of the strong product we can conclude that $(v_i, w_i) \in E(G_i)$ for $i = 1, 2$. Thus the set of vertices with coordinates (v_1, v_2, X) (v_1, w_2, X),(w_1, v_2, X), and (w_1, w_2, X) induce a complete graph K_4 in G. Clearly, the subgraph consisting of Cartesian edges only is a Q_2.

Assume now the assumption is true for $k = m$. We have to show that the statement holds also for $k = m + 1$. Let J={1,...,m+1} and let U_1 and U_2 be a partition of U with $U_1 = \{u \in U \mid u_{m+1} = v_{m+1}\}$ and $U_2 = \{u \in U \mid u_{m+1} = w_{m+1}\}$. Thus each U_i consists of vertices that differ only in the first m coordinates. Notice, by definition of the strong product and by construction of both sets U_1 and U_2 there are vertices a, b in each U_i that differ in all m coordinates that are adjacent in G and hence non-Cartesian in G. Thus, by induction hypothesis the subgraphs $\langle U_i \rangle$ induced by each U_i

consisting of Cartesian edges only is a Q_m. Let $\langle U \rangle$ be the subgraph with vertex set U and edge set $E(\langle U_1 \rangle) \cup E(\langle U_2 \rangle) \cup \{(a,b) \in E(G) \mid a = (X, v_{m+1}, Y)$ and $b = (X, w_{m+1}, Y)\}$. By definition of the strong product the edges (a,b) with $a = (X, v_{m+1}, Y)$ and $b = (X, w_{m+1}, Y)$ induce an isomorphism between $\langle U_1 \rangle$ and $\langle U_2 \rangle$ which implies that $\langle U \rangle \simeq Q_m \square K_2 \simeq Q_{m+1}$. □

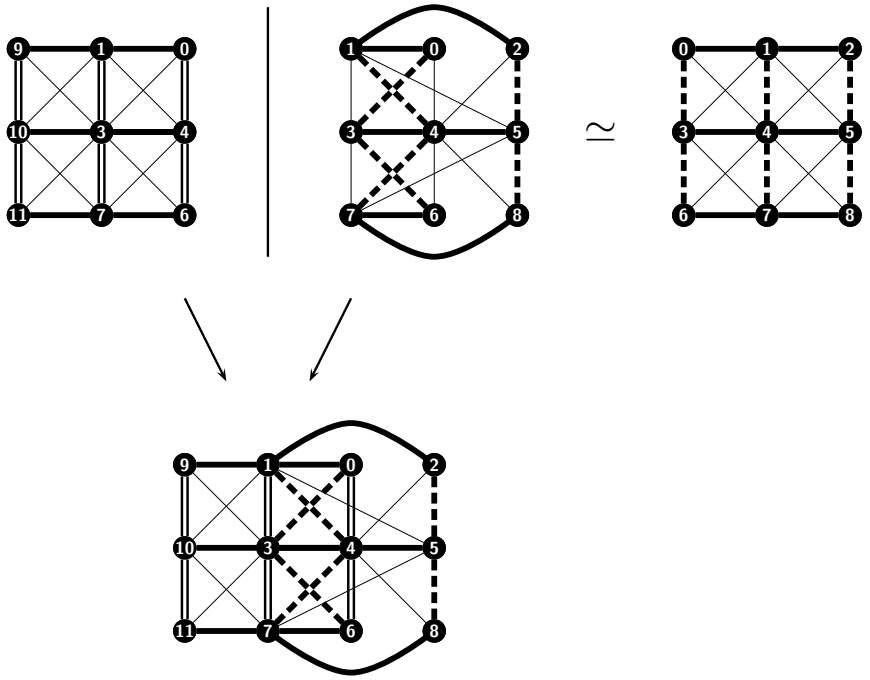


Figure 4.4: In the lower part a graph G is shown with $\sigma = \{3,4\}$ and backbone $\mathbb{B}(G) = \{1,3,4,7\}$. Since $\langle \sigma \rangle$ is a thin connected dominating set we can conclude that G is CHIC. Notice that neither $\langle N[1] \rangle$ nor $\langle N[7] \rangle$ are thin.
Consider the induced neighborhoods $\langle N[3] \rangle$ and $\langle N[4] \rangle$, depicted in the upper part. The colorings of the edges w.r.t. the PFD of each neighborhood are shown as thick dashed edges, thick-lined edges and double-lined edges, respectively. If we cover the graph from $N[3]$ to $N[4]$ the color-continuation fails, e.g. on edge $(1,4)$, since $(1,4)$ is determined as non-Cartesian in $\langle N[3] \rangle$. This holds for all edges in $\langle N[3] \rangle$ that received the color "thick dash" in $\langle N[3] \rangle$. The same holds for the color "double-lined" if we cover the graph from $N[4]$ to $N[3]$. Hence the color-continuation always fails and therefore G is not NICE. If we force the edge $(1,4)$ to be Cartesian in $\langle N[3] \rangle$ Lemma 4.9 implies that the colors "thick-lined" and "double-lined" have to be merged to one color, since the subgraph with edge set $\{(0,1),(0,4),(1,3),(3,4)\} \cup \{(1,4)\}$ is a diagonalized hypercube Q_2.

Consider now a strong product graph G and two given thin subproducts $H_1, H_2 \subseteq G$. Let the Cartesian edges of each subgraph be colored with respect to a product coloring of H_1, respectively H_2 that is at least as fine as the product coloring of G w.r.t. to its PFD. As stated in Definition 3.25 we

have a proper color-continuation from H_1 to H_2 if for all colored edges with color c in H_2 there is a representative edge that is colored in H_1. Assume the color-continuation fails, i.e., there is a color c in H_2 such that for all edges $e_c \in E(H_2)$ with color c holds that e_c is not colored in H_1, for an example see Figure 4.4.

The open question is: "What can we do if the color-continuation fails?" In the sequel we assume that such an edge e_c with color c is contained in $E(H_1)$. The strategy will then be as follows. As claimed, the product colorings of H_1 and H_2 are at least as fine as the one of G and H_1, H_2 are subproducts of G, which implies that colored Cartesian edges in each H_i are Cartesian edges in G. Notice that e_c is determined as non-Cartesian in H_1, otherwise it would have been colored. But since e_c is determined in H_2 as Cartesian, we can infer that e_c must be Cartesian in G. Thus we can force the edge e_c, that is non-Cartesian in H_1, to be Cartesian in H_1. The now arising questions is: "What happens with the factorization of H_1?" The answer is given in the next lemma.

Lemma 4.9. *Let $G = \boxtimes_{l=1}^{n} G_l$ be a thin strong product graph, where each G_l, $l = 1, \ldots, n$ is prime. Let $H = \boxtimes_{l=1}^{m} H_l \subseteq G$ be a thin subproduct of G such that there is a non-Cartesian edge $(v,w) \in E(H)$ that is Cartesian in G. Let J denote the set of indices where v and w differ w.r.t. to the coordinatization of H. Then the factor $\boxtimes_{i \in J} H_i$ of H is a subgraph of a prime factor G_l of G.*

Proof. In this proof factors w.r.t. the Cartesian product and the strong product, respectively, are called Cartesian factors and strong factors, respectively. First notice that Cartesian edges in G as well as in H are uniquely determined, since both graphs are thin. Therefore, a Cartesian edge of $G = \boxtimes_{l=1}^{n} G_l$ that is a non-Cartesian edge in a subproduct $H = \boxtimes_{l=1}^{m} H_l$ of G implies that $m > n$, i.e., the factorization of H is a refinement of the factorization induced by the global PFD. Since H is a thin subproduct of G with a refined factorization, it follows that Cartesian edges of H are Cartesian edges of G. Therefore we can conclude that strong factors of H are entirely contained in strong factors of G.

We denote the subgraph of H that consists of all Cartesian edges of H only, i.e., its Cartesian skeleton, by $\mathbb{S}(H)$, hence $\mathbb{S}(H) = \square_{l=1}^{m} H_l$. Let $U \subseteq V(H)$ be the set of vertices u with coordinates $u_i = v_i$, if $i \notin J$ and $u_i \in \{v_i, w_i\}$, if $i \in J$. Notice that Lemma 4.8 implies that for the induced subgraph w.r.t. the Cartesian skeleton $\langle U \rangle \subseteq \mathbb{S}(H)$ holds $\langle U \rangle \simeq Q_{|J|}$. Moreover, the distance $d_{\langle U \rangle}(v,w)$ between v and w in $\langle U \rangle$ is $|J|$, that is the maximal distance that two vertice can have in $\langle U \rangle$. If we claim that (v,w) has to be an edge in $\langle U \rangle$ we receive a diagonalized hypercube $\langle U \rangle^{diag}$. Corollary 2.47 implies that $\langle U \rangle^{diag}$ is S-prime and hence $\langle U \rangle^{diag}$ must be contained entirely in a Cartesian factor $\widetilde{H}$ of a graph $H^* = \widetilde{H} \square H'$ with $\mathbb{S}(H) \cup (v,w) \subset H^*$. This implies that $\langle U \rangle^{diag} \subseteq \widetilde{H}^u$ for all $u \in V(H^*)$, i.e., $\langle U \rangle^{diag}$ is entirely contained in all $\widetilde{H}^u$-layer in H^*. Note that all $\widetilde{H}$-layer H^u contain at least one edge of every

H_i-layer H_i^u of the previously determined factors H_i, $i \in J$ of H.

Now, all Cartesian factors of $\mathbb{S}(H) = \square_{l=1}^{m} H_l$ coincide with the strong factors of $H = \boxtimes_{l=1}^{m} H_l$ and hence, in particular the factors $H_i,\ i \in J$. Moreover, since H is a subproduct of G and the factorization of H is a refinement of G it holds that Cartesian factors $H_i,\ i \in J$ of $\mathbb{S}(H)$ must be entirely contained in strong prime factors of G. This implies that for all $i \in J$ the H_i-layer H_i^u must be entirely contained in the layer of strong factors of G. We denote the set of all already determined strong factors $H_i,\ i \in J$ of H with $\mathscr{H}$.

Assume the graph $H^* = \square_{j=1}^{s} K_j$ with $\mathbb{S}(H) \cup (v,w) \subseteq H^*$ and $V(H^*) = V(\mathbb{S}(H))$ has a factorization such that $\square_{i \in J} H_i \cup (v,w) \not\subseteq K_j$ for all Cartesian factors K_j. Since $\mathbb{S}(H) \cup (v,w) \subseteq H^*$, we can conclude that $\langle U \rangle^{diag} \subseteq H^*$. Since $\langle U \rangle^{diag}$ is S-prime it must be contained in a Cartesian factor K_r of H^*. This implies that $\langle U \rangle^{diag} \subseteq K_r^u$ for all $u \in V(H^*)$, i.e., for all K_r-layer of this particular factor K_r. Since $\square_{i \in J} H_i \cup (v,w) \not\subseteq K_r$, we can conclude that there is an already determined factor H_i such that $H_i^u \not\subseteq K_r^u$ for all $u \in V(H^*)$. Furthermore, all K_r-layer K_r^u contain at least one edge of each H_i-layer H_i^u of the previously determined factors H_i, $i \in J$ of H. We denote with e the edge of the H_i-layer H_i^u that is contained in the K_r-layer K_r^u. This edge e cannot be contained in any K_j-layer, $j \neq r$. This implies that $H_i^u \not\subseteq K_j^u$ for any K_j-layer, $j = 1, \ldots, s$.

Thus, there is an already determined factor $H_i \in \mathscr{H}$ with $H_i^u \not\subseteq K_j^u,\ u \in V(H^*)$ for all K_j-layer, $j = 1, \ldots, s$. Therefore, none of the layer of this particular H_i are subgraphs of layer of any Cartesian factor K_j of H^*. This means that H^* is not a subproduct of G or a refinement of H, both cases contradict that $H_i \in \mathscr{H}$.

Therefore, we can conclude that $\langle U \rangle^{diag} \subseteq \square_{i \in J} H_i \cup (v,w) \subseteq \widetilde{H}$ for a Cartesian factor $\widetilde{H}$ of H^*. As argued, Cartesian factors are subgraphs of its strong factors and hence, we can infer that $\square_{i \in J} H_i$ and hence $\boxtimes_{i \in J} H_i$ must be entirely contained in a strong factor of H and hence in a strong factor of G, since H is a subproduct. □

4.3.2 Recognition and PFD of CHIC Graphs

We give now a short overview of the approach that recognizes if a graph G is CHIC and that decomposes G into its prime factors if G is CHIC, see Algorithm 4.

One first computes the backbone $\mathbb{B}(G)$ of the given graph G. The set σ consists then of all vertices $x \in \mathbb{B}(G)$ that have a thin 1-neighborhood. To determine if G is *CHIC* one has to check if $\langle \sigma \rangle$ is a connected dominating set. If this is the case the vertices of σ are ordered via BFS applied in the induced subgraph $\langle \sigma \rangle \subseteq G$. This ordered set is denoted by $\sigma^{\gg}$.

Algorithm 4 Recognition and Decomposition of CHIC graph

INPUT: a graph G
compute the backbone $\mathbb{B}(G)$;
$\sigma \leftarrow \{x \in \mathbb{B}(G) \mid \langle N[x] \rangle$ is thin$\}$;
if σ is not a connected dominating set **then**
  STOP and return "G is not CHIC";
end if
compute $\sigma^{\gg} = \{v_1, \ldots, v_k\}$ via BFS along $\langle \sigma \rangle$;
compute PFD of $\langle N[v_1] \rangle$ and properly color its Cartesian edges;
for i=2,...,k **do**
  $H \leftarrow \langle \cup_{j=1}^{i-1} N[v_j] \rangle$
  compute PFD of $\langle N[v_i] \rangle$ and properly color its Cartesian edges;
  compute the combined coloring of H and $\langle N[v_i] \rangle$;
  if color-continuation from H to $\langle N[v_i] \rangle$ fails **then**
    $C \leftarrow \{$color $c \mid$ color-continuation for c fails$\}$
    $W \leftarrow \{v_1, \ldots, v_{i-1}\}$
    Solve-Color-Continuation-Problem(H, $\langle N[v_i] \rangle$, W, C);
  end if
end for
$I \leftarrow \{1, \ldots, num\_comp\}$;
$J \leftarrow I$;
for $k = 1$ to num_comp **do**
  for each $S \subset J$ with $|S| = k$ **do**
    compute two connected components A, A' of G induced by the colored edges of G with color $i \in S$, and $i \in I \backslash S$, resp;
    compute $H_1 = \langle p_A(G) \rangle$ and $H_2 = \langle p_{A'}(G) \rangle$;
    if $H_1 \boxtimes H_2 \backsimeq G$ **then**
      save H_1 as prime factor;
      $J \leftarrow J \backslash S$;
    end if
  end for
end for
OUTPUT: Prime factors of G and product colored Cartesian Skeleton of G w.r.t. to this PFD;

After that, one covers G by the neighborhoods of the vertices $v_i \in \sigma^{\gg}$ according to their BFS-ordering. Let v_i be an arbitrary vertex of $\sigma^{\gg}$. We compute the prime factorization of $\langle N[v_i]\rangle$, properly color its Cartesian edges and compute the combined coloring of $H = \cup_{j=1}^{i-1}\langle N[v_j]\rangle$ and $\langle N[v_i]\rangle$. If the color-continuation fails we use Algorithm 5 to solve this problem by application of Lemma 4.9. Hence, all product colorings of the used induced neighborhoods are combined in order to obtain a product coloring of G.

As in the case of NICE graphs, it might happen that the coloring returned by the first part of the algorithm is finer than the coloring of the global PFD of G and thus colors may need to be combined to determine the factors of the global PFD. This is performed in the second part of the algorithm

Finally the algorithm returns the prime factors of G.

Algorithm 5 Solve-Color-Continuation-Problem

1: **INPUT:** a product colored graph H, a product colored graph $\langle N[v_i]\rangle$, a set of vertices $\{v_1,\ldots,v_{i-1}\}$, a set C of colors;
2: take $v \in \{v_1,\ldots,v_{i-1}\}$ with $(v,v_i) \in E(H)$;
3: compute coordinates of $\langle N[v]\rangle$ with respect to the combined product coloring of H;
4: {differ in "i" if color "i"}
5: **for** all colors $c \in C$ {color-continuation fails} **do**
6: take on representant $e_c = (v,w) \in E(\langle N[v_i]\rangle)$;
7: merge all colors in H where v and w differ to one color;
8: **end for**
9: compute the combined coloring of H and $\langle N[v_i]\rangle$;
10: **OUTPUT:** colored graph H, colored graph $\langle N[v_i]\rangle$;

Lemma 4.10. *Let G be a given graph. Then Algorithm 4 recognizes whether G is CHIC and if G is CHIC it determines the prime factors of G w.r.t. the strong product.*

Proof. Given an arbitrary graph G the algorithm recognizes whether the set of vertices with thin induced neighborhoods is a connected dominating set and thus determines whether G is CHIC or not.

If G is CHIC the ordered set $\sigma^{\gg}$ is computed via a breadth-first search in $\langle\sigma\rangle$ which can be done since σ is a *connected* dominating set.

Let $\langle N[v_i]\rangle$ be a neighborhood where the color-continuation fails from $H = \cup_{j=1}^{i-1}\langle N[v_j]\rangle$ to $\langle N[v_i]\rangle$. Notice that there is a vertex $v \in \{v_1,\ldots,v_{i-1}\}$ with $v \in N[v_i]$, since $\sigma^{\gg}$ implies a BFS-ordering of the vertices of σ. Thus it holds $\langle N[v]\rangle \subseteq H$. Let c denote the color in $\langle N[v_i]\rangle$ such that for all edges $e \in E(\langle N[v_i]\rangle)$ with color c holds that e was not colored in H. Since the combined coloring in H

implies a product coloring of $\langle N[v]\rangle$ we can compute the coordinates of the vertices in $\langle N[v]\rangle$ with respect to this coloring. Notice that the coordinization in $\langle N[v]\rangle$ is unique since $\langle N[v]\rangle$ is thin. Now Lemma 3.29 implies that there is at least one edge $e \in \langle N[v_i]\rangle$ with color c that contains vertex v. Let us denote this edge by $e_c = (v,w)$. Clearly, it holds $(v,w) \in E(\langle N[v]\rangle)$. Hence, this edge is not determined as Cartesian in H, and thus in particular not in $\langle N[v]\rangle$ otherwise e_c would have been colored in $\langle N[v]\rangle$. But since e_c is determined as Cartesian in $\langle N[v_i]\rangle$ and moreover, since $\langle N[v_i]\rangle$ is a subproduct of G, we can infer that e_c must be Cartesian in G. Therefore, we claim that the non-Cartesian edge (v,w) in $\langle N[v]\rangle$ has to be Cartesian in $\langle N[v]\rangle$. Notice that the product coloring in $\langle N[v]\rangle$ induced by the combined colorings of all $\langle N[v_j]\rangle$, $j = 1,\ldots,i-1$ is as least as fine as the product coloring of G. Thus, we can apply Lemma 4.9 and together with the unique coordinization of $\langle N[v]\rangle$ directly conclude that all colors $i \in C$, where C denotes the set of coordinates where v and w differ, have to be merged to one color. This is done in Algorithm 5. This implies that we always get a color-continuation for each color c that is based on those additional edges (v,w) as defined above. Hence, we always get a proper combined coloring, even if the color-continuation previously failed. We end with a combined coloring F_G on $G = \cup_{v\in\sigma}\langle N[v]\rangle$ where the domain of F_G consists of all edges that were determined as Cartesian edges in the previously used $\langle N[v]\rangle$ with $v \in \sigma$. By construction of F_G and the combined colorings used at each step from $\langle N[v_i]\rangle$ to $\langle N[v_{i+1}\rangle$, $v_i, v_{i+1} \in \sigma^{\gg}$ we know that the number of colors in the image of F_G is at most as many colors that were used in the first neighborhood $\langle N[v_1]\rangle$. This number is at most $\log_2(\Delta)$, because every product of k nontrivial factors must have at least 2^k vertices.

Notice that the Cartesian edges of every $\langle N[v]\rangle$, $v \in \sigma$, together with their endpoints, form a connected spanning subgraph of $\langle N[v]\rangle$, $v \in \sigma$. Since any two vertices of G are connected via σ it follows that the edges in the domain of F_G, together with their endpoints, form a connected spanning subgraph of G.

Let now G_i be a prime factor of the input graph G. We have to show that it is returned by our algorithm. It is trivial that for some subset $S \subset J$, S will contain all colors that occur in a particular G_i-fiber G_i^a which contains vertex a. Every vertex $y \in N[x]$ is incident to an edge with every color used in the PFD of $\langle N[x]\rangle$, and hence also with every color of F_G on the same edge set. Thus the set of S-colored edges in G_i^a spans G_i^a.

Since the global PFD induces a local decomposition, every layer in an induced closed neighborhood with respect to a local prime factor is a subset of a layer with respect to a global prime factor. Thus we never identify colors that occur in copies of different global prime factors. In other words, the number of colors in the image of F_G might be larger than the number of prime factors of G and hence

the coloring F_G is a refinement of the product coloring of the global PFD. This guarantees that a connected component of the graph, induced by all edges with a color in S, induces a graph that is isomorphic to G_i. The same arguments show that the colors not in S lead to the appropriate cofactor. Thus G_i will be recognized. □

Lemma 4.11. *Given a graph $G = (V,E)$ with bounded maximum degree then Δ Algorithm 4 recognizes whether G is CHIC in $O(|V| \cdot \Delta^3)$ time.*

Proof. For determining the backbone $\mathbb{B}(G)$ we have to check for a particular vertex $v \in V(G)$ whether there is a vertex $w \in N[v]$ with $N[w] \cap N[v] = N[v]$. This can be done in $O(\Delta^2)$ time for a particular vertex w in $N[v]$. Since this must be done for all vertices in $N[v]$ we end in time-complexity $O(\Delta^3)$. This step must be repeated for all $|V|$ vertices of G. Hence the time complexity for determining $\mathbb{B}(G)$ is $O(|V| \cdot \Delta^3)$.

Checking if $\langle \mathbb{B}(G) \rangle$ is connected can be done via a breadth-first search in $O(|V| + |E|)$ time. Since the number of edges is bounded by $O(|V| \cdot \Delta)$ we can conclude that this task needs $O(|V| \cdot \Delta)$ time. Checking if $\mathbb{B}(G)$ is a dominating set can be in $O(|V|)$ time.

Hence we end in an overall time complexity of $O(|V| \cdot \Delta^3)$. □

Lemma 4.12. *Let G be CHIC graph and $\sigma^{\gg} = \{v_1, \ldots, v_n\}$ be its ordered covering sequence. Furthermore, let $H = \langle \cup_{j=1}^{i-1} N[v_j] \rangle$ with $v_j \in \sigma^{\gg}$ be a product colored subgraph of G and $\langle N[v_i] \rangle$ be a product colored neighborhood with v_i as the next vertex in $\sigma^{\gg}$. Assume the color-continuation from H to $\langle N[v_i] \rangle$ fails and let C denote the set of colors where it fails. Given the latter items as input in Algorithm 5, then Algorithm 5 computes the combined coloring of H and $\langle N[v_i] \rangle$ in $O(\Delta^2)$ time.*

Proof. Taking a vertex $v \in \{v_1, \ldots, v_{i-1}\}$ with $(v, v_i) \in E(H)$ can be done in linear time in the number of edges of $\langle N[v_i] \rangle$ that is in $O(\Delta^2)$ time.

Computing the coordinates of the product colored neighborhood $\langle N[v] \rangle$ can be done via a breadth-first search in $\langle N[v] \rangle$ in $O(|N[v]| + |E(\langle N[v] \rangle)|) = O(\Delta + \Delta^2) = O(\Delta^2)$ time.

Notice that by the color-continuation property H can have at most as many colors as there are colors for the first neighborhood $\langle N[v_1] \rangle$. This number is at most $\log(\Delta)$, because every product of k non-trivial factors must have at least 2^k vertices. Thus the for-loop is repeated at most $\log(\Delta)$ times. All tasks in between the for-loop can be done in $O(\Delta)$ time and hence the for-loop takes $O(\log(\Delta) \cdot \Delta)$ time.

Computing the combined color can also be done linear in the number of edges of $\langle N[v_i]\rangle$ and thus in $O(\Delta^2)$ time.

Therefore, the total time complexity is $O(\Delta^2)$. □

Lemma 4.13. *Given a graph $G = (V,E)$ with bounded maximum degree Δ then Algorithm 4 recognizes whether G is CHIC and if G is CHIC then it determines the prime factors of G w.r.t. the strong product in $O(|V| \cdot \Delta^4)$ time.*

Proof. Determining the backbone $\mathbb{B}(G)$ and checking whether $\langle \mathbb{B}(G)\rangle$ is a connected dominating set can be done $O(|V| \cdot \Delta^3)$ time, see Lemma 4.11.

Computing $\sigma^{\gg} = \{v_1, \ldots, v_k\}$ via the breadth-first search takes $O(|V| + |E|)$ time. Since the number of edges is bounded by $|V| \cdot \Delta$ we can conclude that this task needs $O(|V| \cdot \Delta)$ time.

Each neighborhood has at most $\Delta + 1$ vertices and hence at most $(\Delta + 1) \cdot \Delta$ edges. Together with Lemma 2.23 we can conclude that the PFD each neighborhood and therefore the computation in Line 8 needs $O((\Delta + 1) \cdot \Delta \cdot \Delta^2) = O(\Delta^4)$ time.

The first for-loop will be repeated at most |V| times. Computing H in Line 10, i.e., adding a neighborhood to H, can be done in linear time in the number of edges of this neighborhood, that is in $O(\Delta^2)$ time. The PFD of $\langle N[v_i]\rangle$ in Line 11 takes $O(\Delta^4)$ time and the combined coloring of H and $\langle N[v_i]\rangle$ in Line 12 can be done in constant time. For checking if the color-continuation is valid, one has to check at most for all edges of $\langle N[v_i]\rangle$ if a respective colored edge was also colored in H, which can be done in $O(\Delta^2)$ time. As shown in Lemma 4.12, the complexity of Algorithm 5 is $O(\Delta^2)$. Thus, the time complexity of the first for-loop is $O(|V| \cdot \Delta^4)$.

For the second part (Line 19 – 30) we observe that the size of I is the number of used colors. By the same arguments as in the proof of Lemma 4.12 we can conclude that this number is bounded by $\log(\Delta)$. Hence we also have at most Δ sets S, i.e., color combinations, to consider. In Line 24 we have to find connected components of graphs and in Line 25 we have to perform an isomorphism test for a fixed bijection. Both tasks take linear time in the number of edges of the graph and hence $O(|V| \cdot \Delta)$ time. Thus the total complexity of this part is $O(|V| \cdot \Delta^3)$ time.

The overall time complexity of Algorithm 4 is therefore $O(|V| \cdot \Delta^4)$ time. □

4.4 Relation between NICE and CHIC graphs

In this section, we treat the relation between NICE and CHIC graphs. One can observe that the classes of NICE and CHIC graphs have a non-empty intersection, although they are not identical. For an example of a graph that is NICE and CHIC see Figure 4.1 and 4.5. A graph that is CHIC but not NICE is shown in Figure 4.4. Conversely, a graph that is NICE but not CHIC is depicted in Figure 4.2.

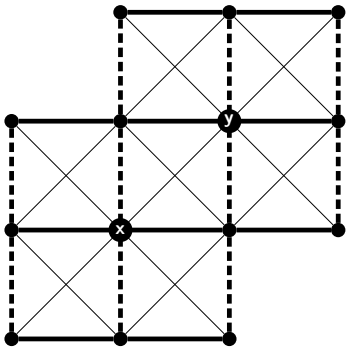


Figure 4.5: A graph with covering sequence $\sigma = \{x, y\}$ that is NICE and CHIC. After running the first part of both Algorithms the assigned coloring consists of two colors, although the graph is prime. The components are combined in the last part of both Algorithms.

As shown in [28] we have:

Lemma 4.14. *Let $G = \boxtimes_{i=1}^{n} G_i$ be the strong product of n triangle-free nontrivial connected graphs different from K_2. Then G is thin.*

Lemma 4.15. *Every triangle-free nontrivial connected graph G different from K_2 is NICE.*

We show that the latter lemma holds for CHIC graphs, too.

Lemma 4.16. *Every triangle-free nontrivial connected graph G different from K_2 is CHIC.*

Proof. First notice that since $G \not\simeq K_2$ and since G is connected that for every vertex $v \in V(G)$ with $\deg(v) = 1$ there is vertex $w \in V(G)$ with $\deg(w) > 1$.

Let $w \in V(G)$ be a vertex with $\deg(w) > 1$. Assume $\langle N[w] \rangle$ is not thin. Then there are vertices $x, y \in N[w]$ with $N[x] \cap N[w] = N[y] \cap N[w]$ and hence there are edges $(x,y), (x,w), (y,w) \in E(G)$, contradicting that G is triangle-free. Hence for all vertices w with $\deg(w) > 1$ holds $\langle N[w] \rangle$ is thin.

Notice that this implies that $\sigma = \mathbb{B}(G)$. Lemma 4.14 implies that G is thin. From Theorem 3.12 we can conclude that σ is a connected dominating set. □

Theorem 4.17. *Let $\boxtimes_{i=1}^{n} G_i$ be the PFD of the connected thin graph G. If G does not contain a clique K_m with $m \geq 3 \cdot 2^{n-1}$, then G is NICE and CHIC.*

Proof. By the thinness of G we know that no factor of G is isomorphic to K_2. If G is the strong product of n prime factors, where at least one of them contains a triangle, then G contains a complete graph K_m with $m \geq 3 \cdot 2^{n-1}$. Hence every prime factor is triangle-free. The statement follows now directly from Lemma 4.2, 4.4, 4.7, 4.15 and Lemma 4.16 □

We conclude this section with the observation that thin graphs need neither be NICE nor CHIC nor thin-N coverable. For examples compare Figures 4.6 and 4.7.

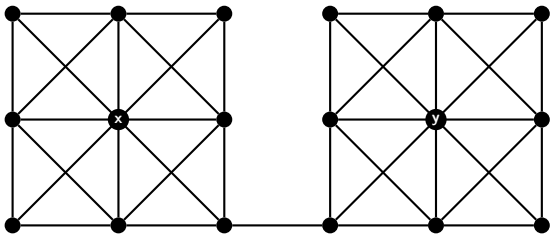


Figure 4.6: A graph G that can be covered by thin neighborhoods $\langle N[x] \rangle$ and $\langle N[y] \rangle$. The graph is thin-N coverable, but neither NICE nor CHIC, because there is no covering sequence.

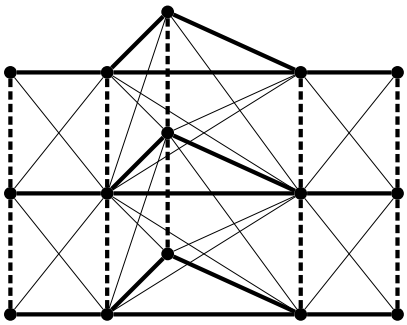


Figure 4.7: A thin graph with the property that all induced neighborhoods are not thin, consequently no covering sequence σ exists. The fibers of the prime factors are marked with thick and dashed edges

5

Locally unrefined Graphs

In this chapter, we are concerned with graphs that cannot be covered by thin 1-neighborhoods only and extend the work of the last chapter to a new class of graphs, which are graphs whose local factorization is not finer than the global one. Will call this property *locally unrefined*.

Definition 5.1. Let G be a given graph. We denote the number of prime factors of G by $|PF(G)|$
The graph class Υ of *locally unrefined graphs* consists of all S-thin graphs with the property that $|PF(G)| = |PF(\langle N[v]\rangle)|$ for all $v \in \mathbb{B}(G)$.
The graph class Υ_n is the set of all graphs $G \in \Upsilon$ with $|PF(G)| = n$.

Note, there are also NICE and CHIC graphs that are locally unrefined, e.g. the graph in Figure 3.7. See Figure 4.3 and 4.4 for examples of graphs, that are NICE and CHIC, but not locally unrefined. However, in this chapter we are interested in an approach that can also deal with non-thin neighborhoods, which is another step towards a local covering algorithm that works for *all* graphs.

We show in the following, how the prime factors of a locally unrefined graph G can be determined, by covering G by 1-neighborhoods of the backbone vertices only. Moreover, we will derive polynomial-time local algorithms for computing the product coloring and the Cartesian skeleton of G, and for recognizing whether G is locally unrefined.

Remark 5.2. We want to emphasize that in this chapter the term *S1-condition* refers to 1-neighborhoods, if not stated explicitly differently. This means, in this chapter, an edge (x,y) satisfies the *S1-condition*, if there is a vertex $z \in V(G)$ with $x,y \in N[z]$ such that $|S_z(x)| = 1$ or $|S_z(y)| = 1$.

5.1 Determining the Prime Factors of $G \in \Upsilon$

Note, we can identify at least one edge (x,y) of each prime factor that belongs to the backbone of G, i.e. $x \in \mathbb{B}(G)$ or $y \in \mathbb{B}(G)$, even if the decomposition of subproducts is finer than the global one. Applying Theorem 3.18 we can do much more in the case of locally unrefined graphs $G \in \Upsilon$: Once we have found an edge (x,y) of a G_i^x-fiber that satisfies the *S1-condition* we can identify *all* edges of that G_i^x-fiber as Cartesian.

Therefore it remains to show, how to color a G_i^x-fiber of a given product graph $G \in \Upsilon$ with $x \in \mathbb{B}(G)$ in a way that all edges of the G_i^x-fiber receive the same color. For this we will need the restricted version of a partial product coloring to individual G_i^x-fibers, the (x,j)- *partial product coloring (*(x,j)-*PPC)*, see Definition 3.27.

We start with the definition of a (x,j)-*covering sequence*.

Definition 5.3. A finite sequence $\sigma_{(x,j)} = (v_i)_{i=0}^k$ of vertices of G is a (x,j)-*covering sequence* if

1. for all $v \in V(G_j^x)$ there exists a vertex $w \in \sigma_{(x,j)}$ with $v \in N[w]$ and
2. if for all $i > 0$ every PPC of $\langle N[v_{i+1}]\rangle$ is a (x,j)-color-continuation of the combined (x,j)-coloring of $\bigcup_{l=1}^{i} E(\langle N[v_l]\rangle)$ defined by the (x,j)-PPC of each $\langle N[v_l]\rangle$.

In this chapter we call a (x,j)-*covering sequence* simply *covering sequence* if there is no risk of confusion.

In our approach we will use the breadth-first search algorithm, explained in Section 2, in a slightly modified way. Let $v \in \mathbb{B}(G)$ be the start vertex. We then decompose the neighborhood of v w.r.t. to its strong prime factor decomposition. Then we fix one color c of one fiber, say G_i^v, and append only those neighbors v_j of v to the current list $BFS(v)$ if

1. they are not already in this list and
2. $v_j \in \mathbb{B}(G)$ and
3. the edge (v,v_j) has the color c of the corresponding G_i^v-fiber.

This will be done recursively for the remaining vertices w fixing the color in each neighborhood

$\langle N[w] \rangle$ of the underlying G_i^v-fiber. Therefore, $BFS(v)$ is a sorted BFS-list on the vertex set $\mathbb{B}(G) \cap V(G_i^v)$.

First we show that in a prime graph $G \in \Upsilon$ such a $BFS(x)$ ordering on the vertices of $\mathbb{B}(G)$ leads to a $(x,1)$-covering sequence of G.

Lemma 5.4. *Let $G \in \Upsilon$ be prime and let x be an arbitrary vertex of the backbone $\mathbb{B}(G) = \{w_1, \ldots, w_m\}$. Then $BFS(x)$ on the vertices of $\mathbb{B}(G)$ is a $(x,1)$-covering sequence.*

Proof. By Theorem 3.12 holds that for all $v \in V(G)$ there is a vertex $w \in BFS(x)$ such that $v \in N[w]$. Thus item (1) of Definition 5.3 is fulfilled.

Notice that $|PF(\langle N[v] \rangle)| = 1$ for all $v \in BFS(x)$ since $G \in \Upsilon$. Thus all edges in such $\langle N[v] \rangle$ are Cartesian and get exactly one color.

Now, take two arbitrary consecutive vertices v_i, v_{i+1} from $BFS(x)$. If v_i and v_{i+1} are adjacent then v_{i+1} is a child of v_i and the edge (v_i, v_{i+1}) satisfies the *S1-condition* in $\langle N[v_i] \rangle$ as well as in $\langle N[v_{i+1}] \rangle$, since $v_i, v_{i+1} \in \mathbb{B}(G)$. Therefore the edge (v_i, v_{i+1}) is colored in the neighborhoods of both adjacent vertices and we get a proper $(x,1)$-color-continuation from $\langle N[x] \rangle \cup \bigcup_{l=1}^{i} \langle N[v_l] \rangle$ to $\langle N[v_{i+1}] \rangle$.

If v_i and v_{i+1} are not adjacent (thus $v_i \neq x$) then there must be parents $u, w \in BFS(x)$ of v_i and v_{i+1}, respectively and we can apply the latter argument. Therefore $BFS(x)$ is a proper $(x,1)$-covering sequence. □

We will now directly transfer that knowledge to (non prime) product graphs. For this we will introduce in Algorithm 6 how to get a proper coloring on all G_i^x-fiber with $x \in \mathbb{B}(G)$. The correctness is proved in the following lemma. Remind that $\Upsilon_n \subset \Upsilon$ denotes the set of graphs $G \in \Upsilon$ with $|PF(G)| = n$.

Lemma 5.5. *Let $G \in \Upsilon_n$ and x be an arbitrary vertex of $\mathbb{B}(G)$. Then Algorithm 6 properly colors all edges of each G_i^x-fiber for $i = 1, \ldots, n$.*

Proof. We show in the sequel that the BFS covering of vertices of $\mathbb{B}(G) \cap V(G_i^x)$, i.e of vertices along Cartesian edges (a,b) of G_i^x with $a, b \in \mathbb{B}(G)$, leads to a proper (x,i)-covering sequence.

First notice that for each $x \in \mathbb{B}(G)$ holds $|PF(\langle N[x] \rangle)| = |PF(G)| = n$, since $G \in \Upsilon_n$. Moreover, all Cartesian edges (v,w) with $v, w \in \mathbb{B}(G) \cap V(G_i^x)$ satisfy the *S1-condition* and therefore can be determined as Cartesian, by applying Lemma 2.26 and Lemma 3.3. Hence, any such edge (v,w) was properly colored both in $\langle N[w] \rangle$ and in $\langle N[v] \rangle$. Applying Theorem 2.10 leads to the requested PPC.

We show next that for all vertices $y \in G_i^x$ there is a vertex $w \in N[y]$ with $w \in BFS(x)$, implying that item (1) of Definition 5.3 is fulfilled. Since $\mathbb{B}(G_i)$ is a connected dominating set for factor G_i

Algorithm 6 Color G_i^x-fiber

INPUT: a graph $G \in \Upsilon_n$ and a vertex $x \in \mathbb{B}(G)$
compute PFD of $\langle N[x] \rangle$ and properly color the Cartesian edges in $\langle N[x] \rangle$ that satisfy the *S1-condition* with colors $c_1, \ldots, c_n$;
$L_i \leftarrow \emptyset$, $i = 1, \ldots, n$;
for $i = 1, \ldots, n$ **do**
    mark x;
    add all neighbors $v \in \mathbb{B}(G)$ of x with color c_i in list L_i in the order of their covering;
    while $L_i \neq \emptyset$ **do**
        take first vertex v from the front of L_i;
        delete v from L_i;
        if v is not marked **then**
            mark v;
            compute PFD of $\langle N[v] \rangle$ and properly color the Cartesian edges in $\langle N[v] \rangle$ that satisfy the *S1-condition*;
            combine the colors on edge $(parent(v), v)$;
            add all neighbors $w \in \mathbb{B}(G)$ of v with color c_i to the end of list L_i in the order of their covering;
        end if
    end while
    for all edges (v, w) that do not satisfy the *S1-condition* **do**
        if there are edges (z, v) and (z, w) that have color c_i **then**
            mark (v, w) as Cartesian and assign color c_i to (v, w);
            {Notice that these edges (z, v) and (z, w) satisfy the *S1-condition*}
        end if
    end for
end for
OUTPUT: G with colored G_j^x-fiber, $j = 1, \ldots, n$;
{Notice that every G_j^x-fiber is isomorphic to one prime factor of G}

we can conclude that for all vertices $y_i \in V(G_i)$ there is a vertex $w_i \in N[y_i]$ such that $w_i \in \mathbb{B}(G_i)$. Suppose that the coordinates for the chosen vertices x are $(x_1,\dots,x_n)$. Then, $w \in N[y]$ has coordinates $(x_1,\dots,x_{i-1},w_i,x_{i+1},\dots,x_n)$. Corollary 3.2 implies $|S_{x_j}(x_j)| = 1$ for $j = 1,\dots,n$. Furthermore, we have $S_w(w) = \prod_{j=1}^{i-1} |S_{x_j}(x_j)| \cdot |S_{w_i}(w_i)| \cdot \prod_{j=i+1}^{n} |S_{x_j}(x_j)| = 1$. Thus $w \in V(G_i^x) \cap \mathbb{B}(G)$ and consequently $w \in BFS(x)$.

Moreover, since all those edges (w,y) with $y \notin \mathbb{B}(G)$ satisfy the *S1-condition* and the fact that $G \in \Upsilon_n$ we can conclude that these edges are properly colored in the neighborhood $\langle N[w]\rangle$. Therefore $BFS(x)$ along vertices of $\mathbb{B}(G) \cap V(G_i^x)$ constitutes a proper (x,i)-covering sequence $\sigma_{x,i}$.

Finally, consider Line 17 – 22 of the algorithm. Theorem 3.12 and Lemma 3.17 imply that the remaining edges (y,y') of G_i^x that do not satisfy the *S1-condition* are induced by vertices of Cartesian edges (z,y) and (z,y') that do satisfy the *S1-condition*. As shown above, all those edges (z,y) and (z,y') are already colored with the same color in some $\langle N[w]\rangle$ with $w \in V(G_i^x) \cap \mathbb{B}(G)$. It follows that we obtain a complete coloring in G_i^x.

This procedure is repeated independently for all colors c_i in $\langle N[x]\rangle$, $i = 1,\dots,n$. This completes the proof. □

Lemma 5.6. *Algorithm 6 determines the prime factors w.r.t. the strong product of a given graph $G = (V,E) \in \Upsilon$ with bounded maximum degree Δ in time complexity $O(|V| \cdot \log_2(\Delta) \cdot (\Delta)^5)$.*

Proof. The time complexity of Algorithm 6 is determined by the complexity of the breadth-first search and the decomposition of each neighborhood in each step.
Notice that the number of vertices of every neighborhood $N[v]$ is at most $\Delta+1$. Thus the number of edges of every neighborhood $\langle N[v]\rangle$ is bounded by $(\Delta+1)\Delta$ and hence the PFD of each neighborhood can be computed in $O(\Delta^4)$, see Lemma 2.23. The number of colors is bounded by the number of factors in each neighborhood, which is at most $\log_2(\Delta+1)$. The breadth-first search takes at most $O(|V|+|E|)$ time for each color. Since the number of edges in G is bounded by $|V| \cdot \Delta$ we can conclude that the time complexity of the breadth-first search is $O(|V|+|V| \cdot \Delta) = O(|V| \cdot \Delta)$. Thus we end in an overall time complexity of $O((|V| \cdot \Delta) \cdot \log_2(\Delta) \cdot (\Delta)^4)$ which is $O(|V| \cdot \log_2(\Delta) \cdot (\Delta)^5)$. □

Remark 5.7. If $G \in \Upsilon$, it is sufficient to use Algorithm 6 to identify a single G_i-fiber through exactly one vertex $x \in \mathbb{B}(G)$ in order to determine the corresponding prime factor of G. For $G \in \Upsilon$ we would therefore be ready at this point.

There is, however, no known sufficient condition to establish that $G \in \Upsilon$, except of course by computing the PFD of G. Moreover, as discussed in [28], it will be very helpful to determine as many

identificable fibers as possible for applications to approximate graph products. However, this task will be treated in the next section.

5.2 Detection and product coloring of the Cartesian skeleton

As shown before, we can identify and even color G_i^x-fiber that satisfy the *S1-condition* in a way that all edges of this fiber receive the same color, whenever $x \in \mathbb{B}(G)$. We will generalize this result for all fibers that satisfy the *S1-condition* in Lemma 5.8. This provides that we get a big part of the Cartesian skeleton colored such that all edges of identified G_i^y-fibers received the same color. Moreover we will show how to identify colors of different colored G_i-fibers. Furthermore we introduce a method to determine Cartesian edges of fibers that do not satisfy the *S1-condition*.

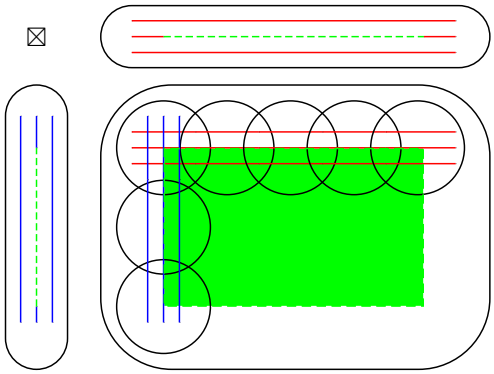


Figure 5.1: The Backbone of the factors is depicted as green dashed line. The backbone of the product graph G is sketched as a green rectangle. Starting with some vertex $x \in \mathbb{B}(G)$ we go along backbone vertices of G with fixed color, i.e. we apply the BFS algorithm only on vertices of $\mathbb{B}(G) \cap G_i^x$ for all i. Applying Lemma 5.5, 5.8 and 5.9 we can color all G_i-fibers that satisfy the *S1-condition* in this way.

5.2.1 Identify Colors of all G_i^x-fibers that satisfy the *S1-condition*

Lemma 5.8. *Let $G \in \Upsilon_n$ and G_i^y with $y \notin \mathbb{B}(G)$ be an arbitrary fiber that satisfies the* S1-condition. *Let $z \in \mathbb{B}(G)$ such that $|S_z(a)| = 1$ or $|S_z(b)| = 1$ for some edge $(a,b) \in G_i^y$. Then the (z,i)-covering sequence $\sigma_{z,i}$ is also a (y,i)-covering sequence.*

Proof. The existence of such a vertex z follows directly from Lemma 3.14. W.l.o.g. let $|S_z(a)| = 1$, otherwise switch the labels of vertices a and b. If $G_i^y = G_i^z$ then the assertion follows directly from Lemma 5.5. Thus we can assume that $G_i^y \neq G_i^z$.

W.l.o.g. let vertex z have coordinates $(z_1,\dots,z_i,\dots,z_n)$ and vertex a have coordinates $(a_1,\dots,a_i,\dots,a_n)$. In the following $\hat{z}$ will denote the vertex in G_i^y with coordinates $\hat{z}_j = a_j$ for $j \neq i$ and $\hat{z}_i = z_i$, in short with coordinates $(a_1,\dots,z_i,\dots,a_n)$. Thus we can infer that $G_i^{\hat{z}} = G_i^y$. For the sake of convenience we will denote all vertices with coordinates $(z_1,\dots,w_i,\dots,z_n)$ and $(a_1,\dots,w_i,\dots,a_n)$ with w and $\hat{w}$, respectively. Note, that w and $\hat{w}$ are adjacent, by choice of their coordinates and by definition of the strong product.

Moreover, since $a \in N[z]$ and because of the coordinates of the vertices $\hat{u},\hat{w} \in G_i^y$ we can infer that $\hat{u} \in N[w]$ holds for all vertices $\hat{u} \in N[\hat{w}]$ by definition of the strong product. More formally, $N[\hat{w}] \cap V(G_i^y) \subseteq N[w]$, see Figure 5.2.

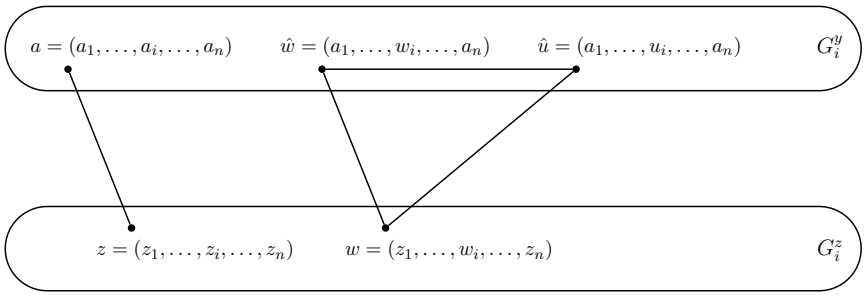


Figure 5.2: $N[\hat{w}] \cap V(G_i^y) \subseteq N[w]$

Let $\sigma_{z,i} = (z, v^1,\dots,v^m)$ be a proper (z,i)-covering sequence, based on the BFS approach explained above, consisting of all backbone vertices of G contained in G_i^z. Furthermore let w be any vertex of $\sigma_{z,i}$. Notice that for all such vertices w holds $|S_w(w)| = 1$ and therefore in particular $|S_{w_i}(w_i)| = 1$, by applying Corollary 3.2. Thus for all such vertices $\hat{w}$ holds

$$|S_w(\hat{w})| = \prod_{j=1}^{i-1} |S_{z_i}(a_i)| \cdot |S_{w_i}(w_i)| \cdot \prod_{j=i+1}^{n} |S_{z_i}(a_i)| = 1,$$

by applying Corollary 3.2 again. Hence all edges $(\hat{u},\hat{w}) \in E(\langle N[\hat{w}]\rangle) \cap E(G_i^y)$ satisfy the *S1-condition* in the closed induced neighborhood of the vertex w, since $N[\hat{w}] \cap V(G_i^y) \subseteq N[w]$. Moreover since $\mathbb{B}(G_i)$ is a connected dominating set we can infer that item (1) of Definition 5.3 is fulfilled.

It remains to show that we also get a proper color-continuation. The main challenge now is to show that for all vertices $(parent(v), v)$ contained in $BFS(z)$ there is an edge $(a,b) \in G_i^y$ that satisfies the *S1-condition* in both $\langle N[parent(v)]\rangle$ and$\langle N[v]\rangle$. This implies that we can continue the color of the G_i^y-fiber on that edge (a,b).

Therefore, let $\hat{v}$ and $\hat{w}$ be any two adjacent vertices of G_i^y with coordinates as mentioned above such that $v_i, w_i \in \mathbb{B}(G_i)$. Thus by choice of the coordinates v and w are adjacent vertices such that $|S_v(v)| =$

$|S_w(w)| = 1$ and hence $v,w \in BFS(z)$. As shown above $|S_v(\hat{v})| = 1$ and $|S_w(\hat{w})| = 1$. Therefore the edge $(\hat{v},\hat{w})$ satisfies the *S1-condition* in both $\langle N[v]\rangle$ and $\langle N[w]\rangle$, since $N[\hat{v}] \cap V(G_i^y) \subseteq N[v]$ and $N[\hat{w}] \cap V(G_i^y) \subseteq N[w]$. The connectedness of $\mathbb{B}(G_i)$ and $G \in \Upsilon_n$ implies that any such edge is properly colored with c by means of the color-continuation. Since $\mathbb{B}(G_i)$ is also a dominating set it holds that all vertices $\hat{u}$ with $|S_u(\hat{u})| > 1$ have an adjacent vertex $\hat{w}$ with $|S_w(\hat{w})| = 1$. Since $N[\hat{w}] \cap V(G_i^y) \subseteq N[w]$, we can infer that $\hat{u} \in N[w]$ and therefore all these edges satisfy the *S1-condition* and are colored with c. Hence, property (2) of Definition 5.3 is satisfied. □

Lemma 5.9. *Let $G \in \Upsilon_n$ and G_i^y with $y \notin \mathbb{B}(G)$ be an arbitrary fiber that satisfies the* S1-condition. *Furthermore let $z \in \mathbb{B}(G)$ with $|S_z(a)| = 1$ or $|S_z(b)| = 1$ for some edge $(a,b) \in G_i^y$. Then Algorithm 6 properly colors all edges of each such G_i^y-fiber with vertex z as an input vertex.*

Proof. Lemma 3.14 implies that there is a $z \in \mathbb{B}(G)$ such that $|S_z(a)| = 1$ or $|S_z(b)| = 1$ for some edge $(a,b) \in G_i^y$. As shown in Lemma 5.8 each such G_i^y-fiber that satisfies the *S1-condition* with $y \notin \mathbb{B}(G)$ can be covered and colored via the corresponding (z,i)-covering sequence $\sigma_{z,i}$. By the way, since $G \in \Upsilon_n$ and Theorem 2.10 we can directly color all edges of G_i^y with the same color c as the G_i^z-fiber. Furthermore, by applying Lemma 3.17 all remaining edges of $(a,b) \in E(G_i^y)$ are induced by vertices of Cartesian edges $(a,\tilde{z}),(b,\tilde{z}) \in E(G_i^y)$ which are satisfying the *S1-condition* and thus already colored with color c. Thus all these edges (a,b) must be Cartesian edges of G_i^y (by definition of the strong product) and thus also obtain color c. □

5.2.2 Identification of Parallel Fibers

As shown in the last subsection we are able to identify all edges of a G_i^x-fiber that satisfies the *S1-condition* as Cartesian in such a way that all these edges in G_i^x get the same color. An example of the colored Cartesian edges of a product graph after coloring all horizontal fibers that satisfy the *S1-condition* is shown in Figure 5.3.

It remains to show how we can identify colors of different colored G_i-fibers. For this the Square Property (Lemma 2.8) is crucial. In the following, we investigate how we can find the necessary squares and under which conditions we can identify colors of differently colored fibers that belong to one and the same factor.

Before stating the next lemma, we explain its practical relevance. Let $G = \boxtimes_{l=1}^{n} G_l \in \Upsilon$ be a strong product graph. In this case, different fibers of the same factor may be colored differently, see Figure 5.3 for an example. We will show that in this case there is a square of Cartesian edges containing one

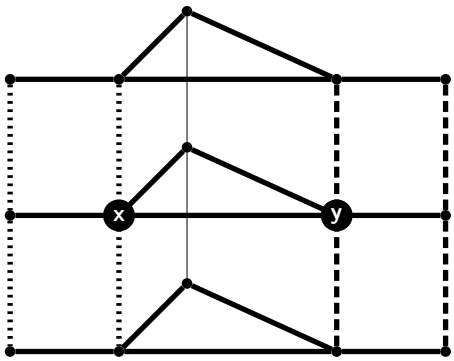


Figure 5.3: Cartesian skeleton of a strong product graph. Its factors are induced by one thick and one dashed colored component. Application of Algorithm 6 identifies Cartesian edges in three distinct color classes indicated by thick lines and the two types of dashed lines. The edges drawn as thin lines are not identified as Cartesian because they do not satisfy the *S1-condition*. The backbone of G consists of the vertices x and y.

Cartesian edge of each of the fibers G_i^a and G_i^x, if these fibers are connected by an arbitrary Cartesian edge of some G_j-fiber. The other two Cartesian edges then belong to two distinct G_j-fibers G_j^A and G_j^B, both of which satisfy the *S1-condition*. The existence of such a square implies that G_i^a and G_i^x are copies of the same factor. Thus we can identify the fibers that belong to the same factor after computing a proper horizontal fiber coloring as explained in previous subsection. Moreover we will show in Lemma 5.11 that all parallel fibers that satisfy the *S1-condition* are connected by a path of Cartesian edges. This provides that we can color *all* G_i-fibers with the same color applying Lemma 2.8 and 5.10.

Lemma 5.10. *Let $G = \boxtimes_{l=1}^{n} G_l$ be the strong product of thin graphs. Let there be two different fiber G_i^a and G_i^x that satisfy the* S1-condition.
Furthermore let there exist an index $j \in \{1,\dots n\}$ s.t. $(p_j(a), p_j(x)) \in E(G_j)$ and $p_k(a) = p_k(x)$ for all $k \neq i, j$.
Then there is a square $A\widehat{A}\widehat{B}B$ in G with

1. *$(A,\widehat{A}) \in E(G_i^x)$ and $(B,\widehat{B}) \in E(G_i^a)$ and*
2. *$(A,B) \in E(G_j^A)$ and $(\widehat{A},\widehat{B}) \in E(G_j^{\widehat{A}})$, whereby $G_j^A \neq G_j^{\widehat{A}}$ and at least one edge of G_j^A and at least one edge of $G_j^{\widehat{A}}$ satisfies the* S1-condition.

Proof. Since the strong product is commutative and associative it suffices to show this for the product $G = G_1 \boxtimes G_2 \boxtimes G_3$ of thin (not necessarily prime) graphs. W.l.o.g. choose $i = 1$, $j = 2$ and $k = 3$. W.l.o.g., let x have coordinates (x_1, x_2, x_3) and a have coordinates (a_1, a_2, x_3). Now we have to

distinguish the following cases for the three different graphs.

Before we proceed we fix a particular notation for the coordinates of certain vertices and edges, which we will maintain throughout the rest of the proof.

- For G_1:

  1. $|\mathbb{B}(G_1)| > 1$, i.e. there is an edge $(v_1, \hat{v}_1) \in E(G_1)$ with $v_1, \hat{v}_1 \in \mathbb{B}(G)$ and
  2. not (1): $|\mathbb{B}(G_1)| = |\{v_1\}| = 1$.

- For G_2:

  A. the edge (a_2, x_2) satisfies the *S1-condition* in G_2
  B. not (A).

  Notice that in case (A) there is by definition a vertex $z_2 \in N[a_2] \cap N[x_2]$ with $|S_{z_2}(a_2)| = 1$ or $|S_{z_2}(x_2)| = 1$. In the following we will assume w.l.o.g. that in this case holds $|S_{z_2}(x_2)| = 1$.
  Case (B) implies that $|S_{x_2}(x_2)| > 1$. By Theorem 3.12 we can conclude that there is a vertex $\tilde{x}_2 \in N[x_2]$ with $|S_{\tilde{x}_2}(\tilde{x}_2)| = 1$, which implies that the edge $(x_2, \tilde{x}_2)$ satisfies the *S1-condition* in G_2.

- For G_3:

  i. $x_3 \in \mathbb{B}(G_3)$
  ii. not (i): $x_3 \notin \mathbb{B}(G_3)$.

  For the sake of convenience define $\tilde{p}_3 = x_3$ if we have case (i). In case (ii) let $\tilde{p}_3 = z_3$ with $z_3 \in N[x_3]$ s.t. $|S_{z_3}(x_3)| = 1$. Notice that such a vertex z_3 has to exist in G_3, otherwise $|S_{z_3}(x_3)| > 1$ for all $z_3 \in N[x_3]$. But then for all $z, x \in V(G)$ with $z \in N[x]$ with coordinates $z = (\ ,\ , z_3)$ and $x = (\ ,\ , x_3)$, resp., holds $|S_z(x)| = \prod_{i=1}^{3} |S_{z_i}(x_i)| > 1$. Hence none of the edges of G_1^q and G_1^x satisfies the *S1-condition*, contradicting the assumption. However, notice that $\tilde{p}_3$ is chosen such that $|S_{\tilde{p}_3}(x_3)| = 1$.

In all cases we will choose the coordinates of the vertices of the square $A\widehat{A}\widehat{B}B$ as follows: $A = (v_1, x_2, x_3)$, $B = (v_1, a_2, x_3)$ with $v_1 \in \mathbb{B}(G_1)$ and $\widehat{A} = (\hat{v}_1, x_2, x_3)$, $\widehat{B} = (\hat{v}_1, a_2, x_3)$, $v_1 \neq \hat{v}_1$. By choice holds $(A, \widehat{A}) \in G_1^x$, $(B, \widehat{B}) \in E(G_1^q)$, $(A, B) \in G_2^A$ and $(\widehat{A}, \widehat{B}) \in E(G_2^{\widehat{A}})$ whereby $G_2^A \neq G_2^{\widehat{A}}$, see Figure 5.4.

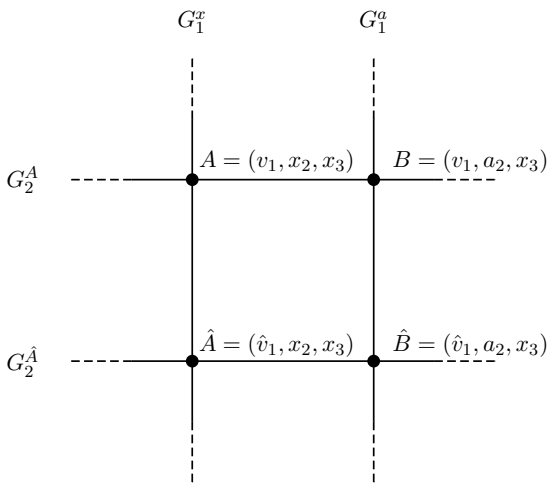


Figure 5.4: General notation of the chosen square $A\hat{A}\hat{B}B$.

It remains to show that at least one edge of both fibers G_2^A and $G_2^{\hat{A}}$ satisfies the *S1-condition*. This part of the proof will become very technical.
In Figure 5.5 and 5.6 the ideas of the proofs are depicted.

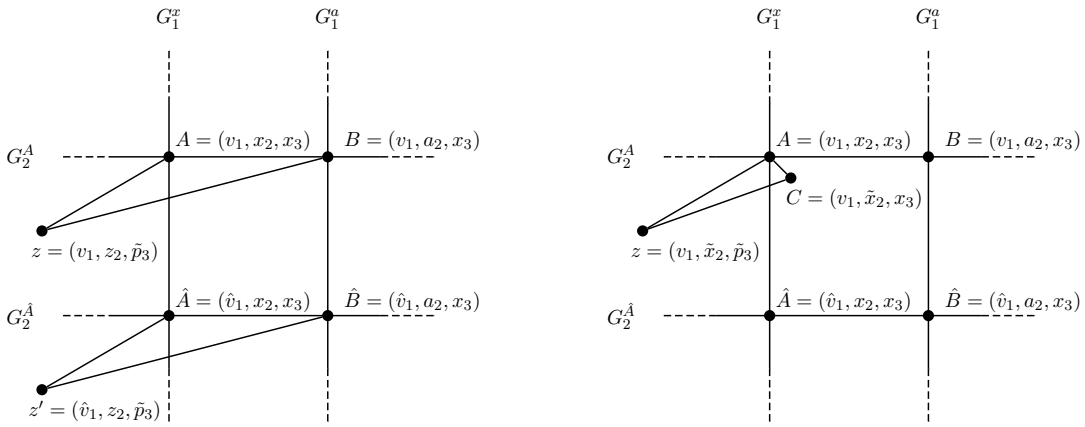


Figure 5.5: Left: Case 1.*A.i.* and *ii.*. Right: Case 1.*B.i.* and *ii*.

Cases 1.A.i and 1.A.ii :
Let $v_1, \hat{v}_1 \in \mathbb{B}(G_1)$ with $(v_1, \hat{v}_1) \in E(G_1)$. Let $z_2 \in N[x_2]$ with $|S_{z_2}(x_2)| = 1$ in G_2. Choose $z \in V(G)$ with coordinates $(v_1, z_2, \tilde{p}_3)$.

By definition of the strong product the edges (z, A) and (z, B) do exist in G and therefore $z \in N[A] \cap N[B]$. Moreover Corollary 3.2 implies that $|S_z(A)| = 1$. Therefore the edge (A, B) is satisfying the *S1-condition* in G in both cases (i) and (ii).

The same argument holds for the edge $(\widehat{A},\widehat{B})$ by choosing $z \in V(G)$ with coordinates $(\hat{v}_1, z_2, \tilde{p}_3)$.

Case 1.B.i and 1.B.ii :

Let $v_1, \hat{v}_1 \in \mathbb{B}(G_1)$ with $(v_1,\hat{v}_1) \in E(G_1)$ and let $\tilde{x}_2 \in N[x_2]$ with $|S_{\tilde{x}_2}(\tilde{x}_2)| = 1$. Choose $z \in V(G)$ with coordinates $(v_1, \tilde{x}_2, \tilde{p}_3)$. By definition of the strong product holds that $(z,A) \in E(G)$.

In case (i) we can conclude from Corollary 3.2 that $|S_z(z)| = 1$. Moreover, in case (i) holds by definition of the strong product that $(z,A) \in G_2^A$ and we are ready.

Otherwise in case (ii) choose the vertex C with coordinates $(v_1, \tilde{x}_2, x_3)$. Then $z \in N[A] \cap N[C]$ and $|S_z(C)| = 1$. Since $(A,C) \in G_2^A$ the assertion for G_2^A follows.

The same arguments hold for $G_2^{\widehat{A}}$ by choosing $z \in V(G)$ with coordinates $(\hat{v}_1, \tilde{x}_2, \tilde{p}_3)$ and C with coordinates $(\hat{v}_1, \tilde{x}_2, x_3)$.

Cases 2.A.i and 2.A.ii :

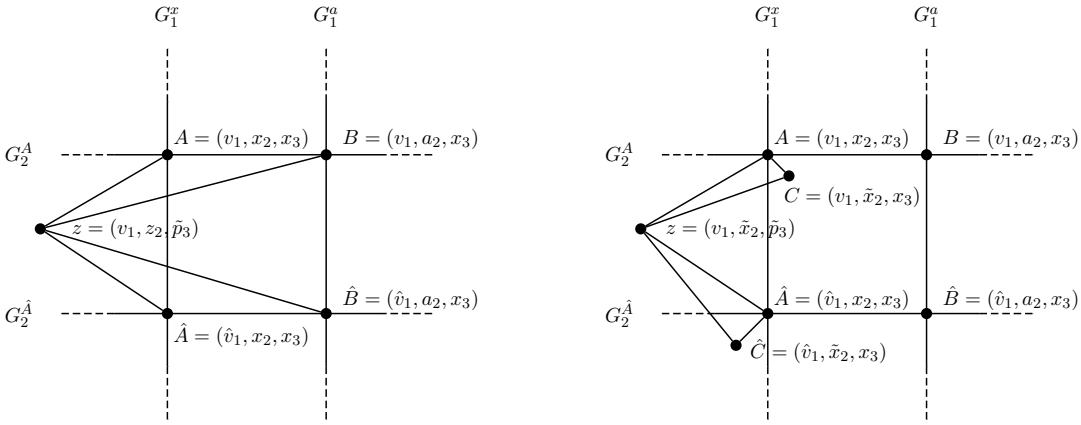


Figure 5.6: Left: Case 2.*A.i.* and *ii.*. Right: Case 2.*B.i.* and *ii.*

Let $v_1 \in \mathbb{B}(G_1)$ and $\hat{v}_1 \in N[v_1]$. Let $z_2 \in N[x_2] \cap N[a_2]$ with $|S_{z_2}(x_2)| = 1$ in G_2. Choose $z \in V(G)$ with coordinates $(v_1, z_2, \tilde{p}_3)$.

In order to show that the conditions are fulfilled for G_2^A we proceed as in cases in (1.A.i) and (1.A.ii):

By definition of the strong product there are non-Cartesian edges $(z,\widehat{A})$ and $(z,\widehat{B})$ and thus $z \in N[\widehat{A}] \cap N[\widehat{B}]$. Now, Lemma 3.13 implies that $|S_{v_1}(\hat{v}_1)| = 1$ and therefore by applying Corollary 3.2 we can conclude that $|S_z(\widehat{A})| = 1$, and the assertion follows for $G_2^{\widehat{A}}$.

Case 2.B.i and 2.B.ii :

Let $v_1 \in \mathbb{B}(G_1)$ and $\hat{v}_1 \in N[v_1]$. Let $\tilde{x}_2 \in N[x_2]$ with $|S_{\tilde{x}_2}(\tilde{x}_2)| = 1$ Choose $z \in V(G)$ with coordinates $(v_1, \tilde{x}_2, \tilde{p}_3)$.

That the conditions are fulfilled for G_2^A can be shown analogously, as in cases in (1.B.i) and (1.B.ii).

To show that the conditions are also fulfilled in case (2.B.i) and (2.B.ii) for $G_2^{\widehat{A}}$ choose $z \in V(G)$ with coordinates $(v_1, \tilde{x}_2, \tilde{p}_3)$ and a vertex C with coordinates $(\hat{v}_1, \tilde{x}_2, x_3)$. Clearly $(\widehat{A}, \widehat{C}) \in E(G_2^{\widehat{A}})$. Furthermore, by definition of the strong product: $z \in N[\widehat{A}]$ and $z \in N[\widehat{C}]$, and thus $z \in N[\widehat{A}] \cap N[\widehat{C}]$. By applying Corollary 3.2 we conclude that $|S_z(\widehat{C})| = 1$, using that Lemma 3.13 implies $|S_{v_1}(\hat{v}_1)| = 1$. Thus the edge $(\widehat{A}, \widehat{C})$ satisfies the *S1-condition*, and the assertion follows for $G_2^{\widehat{A}}$. □

It is important to notice that the square $A\widehat{A}\widehat{B}B$ in the construction of Lemma 2.8 is exclusively composed of Cartesian edges. The lemma can therefore be applied to determine whether two fibers G_i^a and G_i^x, which have been colored differently in the initial steps, are copies of the same factor, and hence, whether their colors need to be identified. As we shall see below, this approach is in fact sufficient to identify all fibers belonging to a common factor.

Lemma 5.11. *Let $G = \boxtimes_{j=1}^n G_j$ be the strong product of thin graphs. Furthermore let $G_i^{y_1}, \ldots, G_i^{y_m}$ be all G_i-fibers in G satisfying the* S1-condition*. Then there is a connected path $\mathscr{P}$ in G consisting only of vertices of $\mathscr{X} = \{x_1, \ldots, x_m\}$ with $x_j \in V(G_i^{y_j})$ s.t. each edge $(x_k, x_l) \in \mathscr{P}$ is Cartesian.*

Proof. Since the strong product is commutative and associative it suffices to show this for the product $G = G_1 \boxtimes G_2$ of two thin (not necessarily prime) graphs. W.l.o.g. let $i = 1$. Moreover, we can choose w.l.o.g. the vertices $x_1, \ldots, x_m$ such that $p_1(x_k) = x$ for $k = 1, \ldots, m$. Moreover by applying Theorem 3.12 we can choose x such that $x \in \mathbb{B}(G_1)$.

Consider first all vertices v with coordinates (x, v_2) such that $v_2 \in \mathbb{B}(G_2)$. From Theorem 3.12 follows that $\mathbb{B}(G_2)$ is connected. Thus there is a connected path $\mathscr{P}_2$ consisting only of such vertices v. Moreover, each edge (a, b) with $a, b \in V(\mathscr{P}_2)$ and thus with coordinates (x, a_2) and (x, b_2), resp., is Cartesian. Furthermore, all corresponding G_1^v-fibers are satisfying the *S1-condition*, since for each edge (v, w) holds $|S_v(v)| = 1$, by applying Corollary 3.2. Therefore all vertices v with coordinates (x, v_2) with $v_2 \in \mathbb{B}(G_2)$ are also contained in $\mathscr{X}$. Hence all those G_i^v-fibers are connected by such a path $\mathscr{P}_2$ with $V(\mathscr{P}_2) \subset \mathscr{X}$.

Let now $\tilde{v}$ be any vertex in $\mathscr{X} \setminus V(\mathscr{P}_2)$. Hence $p_2(\tilde{v}) \notin B(G_2)$. Theorem 3.12 implies that for all those vertices $p_2(\tilde{v}) \notin B(G_2)$ there is an adjacent vertex $p_2(v)$ in G_2 s.t. $p_2(v) \in \mathbb{B}(G_2)$. Thus we can conclude that for all vertices $\tilde{v} \in \mathscr{X} \setminus V(\mathscr{P}_2)$ with coordinates $(x, p_2(\tilde{v}))$ there is an adjacent vertex $v \in V(\mathscr{P}_2)$ with coordinates $(x, p_2(v))$, what from the assertion follows. □

5.2.3 Detection of unidentified Cartesian Edges

One open question still remains: How can we identify a Cartesian (x,y) edge that does not satisfy the *S1-condition* in any 1-neighborhood, i.e., if for all $z \in N[x] \cap N[y]$, we have both $|S_z(x)| > 1$ and $|S_z(y)| > 1$? Figures 5.7 and 5.8 show examples of product graphs, in which not all fibers were determined by the approach outline in the previous two sections.

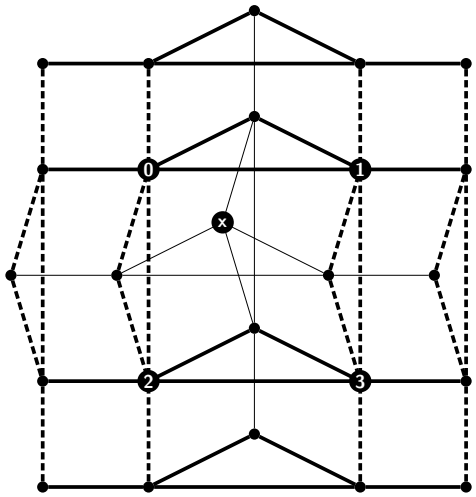


Figure 5.7: Cartesian Skeleton of the strong product G of two prime factors induced by the dashed and bold lined fibers. Application of Algorithm 6 to all fibers determines a part of the Cartesian Skeleton H that consists only of the edges drawn as dashed or bold lines. While the bold and dashed fibers identify the true factors, we miss the copies shown by thin lines. None of these edges satisfies the *S1-condition* in an induced 1-neighborhood. The backbone $\mathbb{B}(G)$ consists of the vertices $0,1,2$ and 3.

Unfortunately, we do not see an efficient possibility to resolve the missing cases by utilizing only the information contained in the fibers that already have been identified so far and the structure of 1-neighborhoods. We therefore introduce a method which relies on the identification of Cartesian edges within N^*-neighborhoods.

Of course, it would be desirable if smaller structure were sufficient. Natural candidates would be to exploit the *S1-condition* in edge-neighborhoods of the form $\langle N[x] \cup N[x'] \rangle$, where (x,x') is a Cartesian edge. However, the example in Figure 5.8 shows that the information contained in these subproducts is still insufficient.

Note, that we refine the already known results of [29], where analogous results were stated for 2-neighborhoods. We will show that every Cartesian (x,y) edge that does not satisfy the *S1-condition* can be determined as Cartesian in the $N^*_{x,y}$-neighborhood.

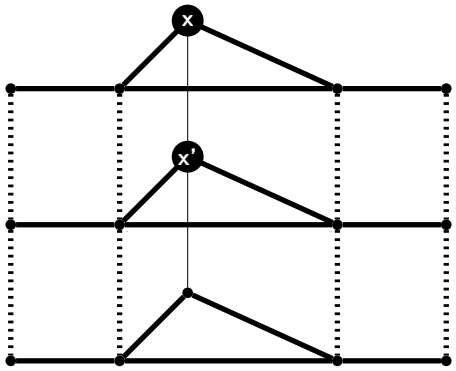


Figure 5.8: Cartesian skeleton of a thin strong product graph whose factors are induced by one thick and dashed component. The fiber whose edges are drawn as thin lines does not satisfy the *S1-condition*. Moreover, even in the subgraph induced by the neighborhoods of x and x', which is the product of a path and a K_3, the *S1-condition* is violated for the Cartesian edge.

Lemma 5.12. *Let G be a thin graph and (v,w) be any edge of G. Let N^* denote the $N^*_{v,w}$-neighborhood. Then it holds that $|S_{N^*}(v)| = 1$ and $|S_{N^*}(w)| = 1$, i.e., the edge (v,w) satisfies the* S1-condition *in N^*.*

Proof. Assume that $|S_{N^*}(v)| > 1$. Thus there is a vertex $x \in S_{N^*}(v)$ different from v with $N[x] \cap N^* = N[v] \cap N^*$, which implies that $w \in N[x]$ and hence $x \in N[v] \cap N[w]$. Since $N[v] \subseteq N^*$ and $N[x] \subseteq N^*$ we can conclude that $N[v] = N[v] \cap N^* = N[x] \cap N^* = N[x]$, contradicting that G is thin. Analogously, one shows that the statement holds for vertex w. □

Next, we prove that the PFD of an arbitrary N^*-neighborhood is not finer than the PFD of a given graph $G \in \Upsilon_n$. This implies that each Cartesian edge in G that is contained in N^* and satisfies the *S1-condition* in N^* can be determined as Cartesian in N^*.

Lemma 5.13. *Let $G \in \Upsilon_n$ and let (x,y) be an arbitrary edge in $E(G)$. Then $|PF(\langle N^*_{x,y}\rangle)| = n$.*

Proof. Notice that $|PF(G)| = n$ and $|PF(N[x])| = n$, since $G \in \Upsilon_n$. Since $N^*_{x,y}$ is a subproduct of G (Corollary 2.30) we can conclude that the PFD of $N^*_{x,y}$ has at least $|PF(G)|$ factors. Futhermore, since $\langle N[x]\rangle$ is subproduct of $N^*_{x,y}$ we can infer that $\langle N[x]\rangle$ has at least as many prime factors as $N^*_{x,y}$. Therefore we have

$$n = |PF(G)| \leq |PF(N^*_{x,y})| \leq |PF(\langle N[x]\rangle)| = n,$$

and thus $|PF(N^*_{x,y})| = n$. □

From Lemma 5.12 and 5.13 we can conclude that any Cartesian edge (x,y) of some fiber that does not satisfy the *S1-condition* can be determined as Cartesian in its $N^*_{x,y}$-neighborhood. Thus, we can identify *all* Cartesian edges of G.

The last step, we have to consider is to identify such fibers as copy of the corresponding factor. This can be done in a simple way. Consider that we have now identified *all* Cartesian edges of G. Notice that for all new identified G_i^a-fibers holds $a \notin \mathbb{B}(G)$, otherwise each edge containing vertex a of this fiber would satisfy the *S1-condition* in $\langle N[a] \rangle$ and we would have identified this fiber. However, for each such vertex a there is a vertex $x \in N[a]$ with $x \in \mathbb{B}(G)$, since $\mathbb{B}(G)$ is a connected dominating set. Thus, the corresponding G_i^x-fiber satisfies the *S1-condition* and is therefore already identified and colored as G_i-fiber. Hence, again we can apply the square property to determine such a new identified G_i^a-fiber belonging to a copy of the factor G_i by identifying the colors of the G_i^a-fiber with the color of the G_i^x-fiber.

5.2.4 Algorithm and Time Complexity

We will now summarize the algorithm for determining the colored Cartesian skeleton of a given graph $G \in \Upsilon$ w.r.t. to its PFD and give the top level control structure, which are proved to be correct in the previous subsections. Furthermore , we will determine the time complexity, which is stated in the following lemma.

Algorithm 7 Cartesian skeleton and Product Coloring of G

1: **INPUT:** Graph $G \in \Upsilon$.
2: Compute the backbone $\mathbb{B}(G)$;
3: **for** all x in $\mathbb{B}(G)$ **do**
4: Color all G_i^x-fibers (and G_i^y-fibers that satisfy the *S1-condition*) with Algorithm 6;
5: **end for**
6: Determine unidentified Cartesian edges in N^*-neighborhoods;
7: Compute all squares in the induced Cartesian skeleton of G and identify the colors of parallel fibers applying Lemma 2.8;
8: **OUTPUT:** Product coloring of G with respect to its PFD;

Lemma 5.14. *Algorithm 7 determines the colored Cartesian skeleton with respect to its PFD of a given graph $G = (V,E) \in \Upsilon$ with bounded maximum degree Δ in $O(|V|^2 \cdot \log_2(\Delta) \cdot \Delta^5)$ time.*

Proof. **1. Determining the backbone** $\mathbb{B}(G)$: we have to check for a particular vertex $v \in V(G)$ whether there is a vertex $w \in N[v]$ with $N[w] \cap N[v] = N[v]$. This can be done in $O(\Delta^2)$ for a particular

vertex w in $N[v]$. Since this must be done for all vertices in $N[v]$ we end in time-complexity $O(\Delta^3)$. This step must be repeated for all $|V|$ vertices of G. Hence the time complexity for determining $\mathbb{B}(G)$ is $O(|V| \cdot \Delta^3)$.

2. For-Loop. The time complexity of Algorithm 6 is $O(|V| \cdot \log_2(\Delta) \cdot (\Delta)^5)$. The for-loop is repeated for all backbone vertices. Hence we can conclude that the time complexity of the for-loop is $O(|V| \cdot |V| \cdot \log_2(\Delta) \cdot \Delta^5)$.

3. Determine unidentified Cartesian edges in N^*-neighborhoods. Notice that each N^*-neighborhood has at most $1 + \Delta \cdot (\Delta - 1)$ vertices. Therefore the number of edges in each N^*-neighborhood is bounded by $(1 + \Delta \cdot (\Delta - 1)) \cdot \Delta$. By Lemma 2.23 the computation of the PFD of each N^* and hence the assignment to an edge of being Cartesian is bounded by $O(((1 + \Delta \cdot (\Delta - 1)) \cdot \Delta) \cdot \Delta^2) = O(\Delta^5)$. Again, this will be repeated for all vertices and thus the time complexity is $O(|V| \cdot \Delta^5)$.

4. Compute all squares. Take an edge (x,y) and check whether there is an edge (x_i,y_j) for all neighbors $x_1,\ldots,x_l \neq y$ of x and $y_1,\ldots,y_k \neq x$ of y. Notice that $l,k \leq \Delta - 1$. This leads to all squares containing the edge (x,y) and requires at most $(\Delta - 1)^2$ comparisons. Since we need diagonal-free squares we also have to check that there is no (Cartesian) edge (x,y_j) and no edge (x_i,y). This will be done for all $|E|$ edges. Thus we end in time complexity $O(|E| \cdot (\Delta - 1)^3)$, which is $O(|V| \cdot \Delta^4)$, since the number of edges in G is bounded by $|V| \cdot \Delta$.

Considering all steps we end in an overall time complexity $O(|V|^2 \cdot \log_2(\Delta) \cdot \Delta^5)$. □

5.3 Recognition of Graphs $G \in \Upsilon$

In this section we will provide an algorithm that tests whether a given graph is element of Υ in polynomial time.

Lemma 5.15. *Algorithm 8 recognizes if a given graph G is in class* Υ.

Proof. Lemma 2.26 implies that the PFD of any neighborhood in a graph G has at least $|PF(G)|$ factors and hence $MAX \geq |PF(G)|$. Thus if $MAX = |PF(G)|$ then none of the decomposed neighborhoods was locally finer. If in addition the isomorphism test is true we can conclude that we have found the correct factors and that $G \in \Upsilon$. □

Lemma 5.16. *Algorithm 8 recognizes if a given a given graph $G = (V,E)$ with bounded maximum degree Δ is in class Υ in $O(|V|^2 \cdot \log_2(\Delta) \cdot \Delta^5)$ time.*

Algorithm 8 Recognition if $G \in \Upsilon$

1: **INPUT:** thin Graph G.
2: compute the colored Cartesian skeleton of G with Algorithm 7 and remind the number of prime factors in each decomposed neighborhood;
3: $MAX \leftarrow$ maximal number of prime factors of decomposed neighborhoods;
4: compute the possible prime factors $G_1, \dots, G_m$ of G by taking one connected component of the Cartesian skeleton of each color $1, \dots, m$;
5: **if** $\boxtimes_{i=1}^{m} G_i \backsimeq G$ and $MAX = m$ **then**
6: IS_IN_$\Upsilon \leftarrow$ true;
7: **else**
8: IS_IN_$\Upsilon \leftarrow$ false;
9: **end if**
10: **OUTPUT:** IS_IN_Υ;

Proof. Algorithmus 7 takes $O(|V|^2 \cdot \log_2(\Delta) \cdot \Delta^5)$ time. Computing the maximum MAX of the number of prime factors of each decomposed neighborhood can be done in linear time in the number of vertices. By the same arguments as in the proof of Lemma 4.13 we can conclude that extracting the possible factors and the isomorphism test for a fixed bijection can be done in $O(|V| \cdot \Delta)$ time. Thus we end in $O(|V|^2 \cdot \log_2(\Delta) \cdot \Delta^5)$ time. □

6

A General Local Approach

In this chapter, we use and summarize the previous results and provide a general local approach for the PFD of thin graphs G. Notice that even if the given graph G is not thin, the provided Algorithm works on G/S. The prime factors of G can then be constructed by using the information of the prime factors of G/S as shown in Section 2.3.2.

The new algorithm makes use of several different subproducts. As it turns out it will not be enough to use 1-neighborhoods only. We also need edge-neighborhoods and N^*-neighborhoods. Notice that edge-neighborhoods are not always proper subproducts of a given graph. Therefore, we treat this problem first and show how the local information that is provided by an edge-neighborhood can be used to determine if this edge-neighborhood is a proper subproduct or not. Then, we proceed to explain how the general local approach works as well as to give a proof of the correctness of this algorithm. In the last part of this chapter, we show that the time complexity of the new algorithm is quasi-linear in the number of vertices of G.

6.1 Dispensability

As mentioned, the general local approach needs in addition to 1-neighborhoods also edge-neighborhoods and N^*-neighborhoods. Notice that Corollary 2.30 implies that for each edge (x,y) the respective N^*-neighborhood $N^*_{x,y}$ is a subproduct, while this is not true for the edge-neighborhood $\langle N[x] \cup N[y] \rangle$ if (x,y) is non-Cartesian in G. Notice that a non-Cartesian edge of G might be Cartesian in its edge-neighborhood. Therefore, we cannot use the information provided by the PFD of $\langle N[x] \cup N[y] \rangle$ to figure out if (x,y) is Cartesian in G. On the other hand, an edge that is Cartesian in a subproduct H of G must be Cartesian in G. To check if an edge (x,y) is Cartesian in $\langle N[x] \cup N[y] \rangle$ that is Cartesian in G as well we use the *dispensable*-property provided by Hammack and Imrich, see [24] and Section 2.3.2.

We show that an edge (x,y) that is dispensable in G is also dispensable in $\langle N[x] \cup N[y] \rangle$. Conversely, we can conclude that every edge that is indispensable in $\langle N[x] \cup N[y] \rangle$ must be indispensable and therefore Cartesian in G. This implies that every edge-neighborhood $\langle N[x] \cup N[y] \rangle$ is a proper subproduct of G if (x,y) is indispensable in $\langle N[x] \cup N[y] \rangle$.

Recall, an edge (x,y) of G is *dispensable* if there exists $z \in V(G)$ for which both of the following statements hold, Definition 2.19.

1. (a) $N[x] \cap N[y] \subset N[x] \cap N[z]$ or (b) $N[x] \subset N[z] \subset N[y]$
2. (a) $N[x] \cap N[y] \subset N[y] \cap N[z]$ or (b) $N[y] \subset N[z] \subset N[x]$

Remark 6.1. As mentioned in [24] we have:

- $N[x] \subset N[z] \subset N[y]$ implies $N[x] \cap N[y] \subset N[y] \cap N[z]$.
- $N[y] \subset N[z] \subset N[x]$ implies $N[x] \cap N[y] \subset N[x] \cap N[z]$.
- If (x,y) is indispensable then $N[x] \cap N[y] \subset N[x] \cap N[z]$ and $N[x] \cap N[y] \subset N[y] \cap N[z]$ cannot both be true.

Lemma 6.2. *Let (x,y) be an arbitrary edge of a given graph G and $H = \langle N[x] \cup N[y] \rangle$ Then it holds:*

$$N[x] \cap N[y] \subset N[x] \cap N[z]$$

if and only if

$$N[x] \cap N[y] \cap H \subset N[x] \cap N[z] \cap H.$$

Proof. First notice that $N[x] \cap N[y] \cap H = N[x] \cap N[y]$. Furthermore, since $N[x] \cap N[z] \subseteq N[x] \subseteq V(H)$ we can conclude that $(N[x] \cap N[z]) \cap H = N[x] \cap N[z]$, from what the assertion follows. □

Lemma 6.3. *Let (x,y) be an arbitrary edge of a given graph G and $H = \langle N[x] \cup N[y] \rangle$. If*

$$N[x] \subset N[z] \subset N[y]$$

then

$$N[x] \cap H \subset N[z] \cap H \subset N[y] \cap H$$

.

Proof. First notice that $N[x] \cap H = N[x]$, $N[y] \cap H = N[y]$, and $N[z] \cap H = (N[z] \cap N[x]) \cup (N[z] \cap N[y])$. Since $N[x] \subset N[z] \subset N[y]$ we can conclude that $(N[z] \cap N[x]) \cup (N[z] \cap N[y]) = (N[x]) \cup (N[z]) = N[z]$. Therefore $N[x] \cap H = N[x] \subset N[z] = N[z] \cap H$ and $N[z] \cap H = N[z] \subset N[y] = N[y] \cap H$. □

Notice that the converse does not hold in general, since $N[z] \cap H \subset N[y] \cap H = N[y]$ does not imply that $N[z] \subset N[y]$. However, by symmetry, Remark 6.1, Corollary 2.30, Lemma 6.2 and 6.3 we can conclude the next corollary.

Corollary 6.4. *If an edge (x,y) of a thin strong product graph G is* indispensable *in $\langle N[x] \cup N[y] \rangle$ and therefore Cartesian in G then the edge-neighborhood $\langle N[x] \cup N[y] \rangle$ is a subproduct of G.*

One aim of our new approach will be to detect all Cartesian edges of the Cartesian skeleton $\mathbb{S}[G]$ of a given graph G. As already shown, only Cartesian edges that satisfy the *S1-condition* can be identified locally as Cartesian. In some cases it might happen that even edge-neighborhoods $H = \langle N[x] \cup N[y] \rangle$ of globally Cartesian edges (x,y) do not provide enough information to identify those edges as Cartesian edges in H, e.g., if $|S_H(x)| > 1$ and $|S_H(y)| > 1$, see Figure 6.1 and 6.2. However, Lemma 5.12 implies that *every* edge $(x,y) \in E(G)$ satisfies the *S1-condition* in its $N^*_{x,y}$-neighborhood if G is thin.

6.2 Algorithm and Time Complexity

First, we give an overview of the algorithm. Then, we proceed to prove the correctness of the new local approach and in the last part of this section, we treat its time complexity.

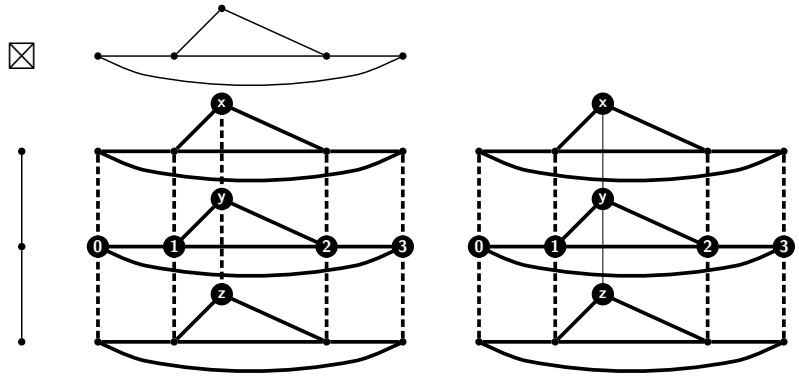


Figure 6.1: Depicted is the colored Cartesian skeleton of the thin strong product graph G after running the first while-loop of Algorithm 9 with different BFS-orderings $\mathbb{B}_{BFS}$ of the backbone vertices. The backbone $\mathbb{B}(G)$ consists of the vertices $0,1,2$ and 3.
lhs.: $\mathbb{B}_{BFS} = 2,1,3,0$. In this case the color-continuation from $N[2]$ to $N[1]$ fails. hence we compute the PFD of the edge-neighborhood $\langle N[2] \cup N[1] \rangle$. Notice that the Cartesian edges (x,y) and (y,z) satisfy the *S1-condition* in $\langle N[2] \cup N[1] \rangle$ and will be determined as Cartesian. In all other steps the color-continuation works.
rhs.: $\mathbb{B}_{BFS} = 3,0,2,1$. In all cases ($N[3]$ to $N[0]$, $N[3]$ to $N[2]$, $N[0]$ to $N[1]$) the color-continuation works. However, after running the first while-loop there are missing Cartesian edges (x,y) and (y,z) that do not satisfy the *S1-condition* in any of the previously used subproducts $N[3]$, $N[0]$, $N[2]$ and $N[1]$. Moreover, the edge-neighborhoods $\langle N[x] \cup N[y] \rangle$ as well as $\langle N[z] \cup N[y] \rangle$ are the product of a path and a K_3 and the *S1-condition* is violated for the Cartesian edges in its edge-neighborhood. These edges will be determined in the second while-loop of Algorithm 9 using the respective N^*-neighborhoods.

Given an arbitrary thin graph G, first the backbone vertices are ordered via the *breadth-first search (BFS)*. After this, the neighborhood of the first vertex x from the ordered BFS-list $\mathbb{B}_{BFS}$ is decomposed. Then the next vertex $y \in N[x] \cap \mathbb{B}_{BFS}$ is taken and the edges of $\langle N[y] \rangle$ are colored with respect to the neighborhoods PFD. If the color-continuation does not fail, then the Algorithm proceeds with the next vertex $y' \in N[x] \cap \mathbb{B}_{BFS}$. If the color-continuation fails, the Algorithm proceeds with the edge-neighborhood $\langle N[x] \cup N[y] \rangle$. If it turns out that (x,y) is indispensable in $\langle N[x] \cup N[y] \rangle$ and hence, that $\langle N[x] \cup N[y] \rangle$ is a proper subproduct (Corollary 6.4) the algorithm proceeds to decompose and to color $\langle N[x] \cup N[y] \rangle$. If it turns out that (x,y) is dispensable in $\langle N[x] \cup N[y] \rangle$ the N^*-neighborhoods $N^*_{x,y}$ is factorized and colored. In all previous steps edges are marked as "checked" if they satisfy the *S1-condition*, independent from being Cartesian or not.

After this, the N^*-neighborhoods of all edges that do not satisfy the *S1-condition* in any of the previously used subproducts, i.e, 1-neighborhoods, edge-neighborhoods or N^*-neighborhoods, are decomposed and again the edges are colored.

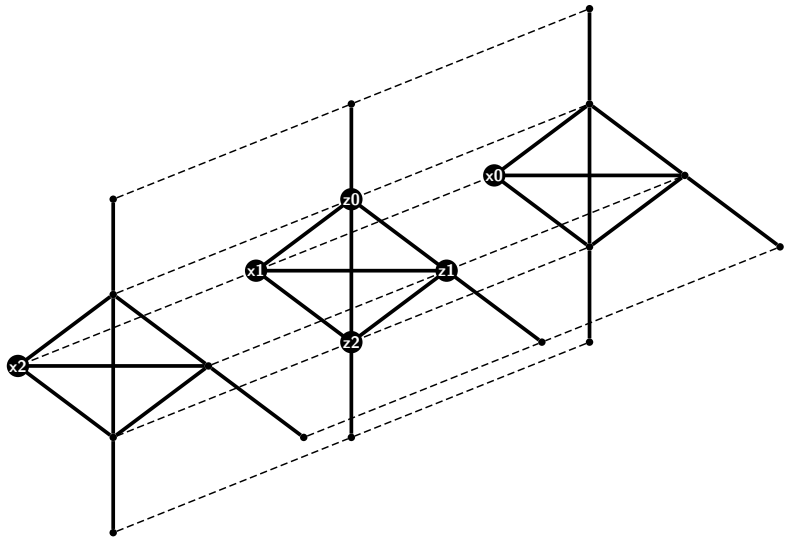


Figure 6.2: The Cartesian skeleton of the thin product graph G of two prime factors induced by one connected component of thick and dashed lined edges. The backbone $\mathbb{B}(G)$ consists of the vertices z_1, z_2 and z_3. In *none of any* edge-neighborhood H holds $|S_H(x_i)| = 1$, $i = 1,2,3$. Hence the fiber induced by vertices x_1, x_2 and x_3 does not satisfy the *S1-condition* in any edge-neighborhood. To identify this particular fiber it is necessary to use N^*-neighborhoods. By Lemma 5.12 N^*-neighborhoods are also sufficient.

Finally, the Algorithm checks which of the recognized factors have to be merged into the prime factors $G_1, \ldots, G_n$ of G.

Theorem 6.5. *Given a thin graph G then Algorithm 9 determines the prime factors of G w.r.t. the strong product.*

Proof. We have to show that every prime factor G_i of G is returned by our algorithm.

First, the algorithm scans all backbone vertices in their BFS-order stored in $\mathbb{B}_{BFS}$, which can be done, since G is thin and hence $\langle \mathbb{B}(G) \rangle$ is connected (Theorem 3.12).

1. Starting with the first neighborhoods $N[x]$ with x as first vertex in $\mathbb{B}_{BFS}$, we proceed to cover the graph with neigborhoods $N[y]$ with $y \in \mathbb{B}_{BFS}$ and $y \in N[x]$. If the color-continuation does not fail, we can apply Lemma 3.21 and Lemma 3.30 and conclude that the determined Cartesian edges in $\langle N[x] \rangle$, resp. in $\langle N[y] \rangle$, i.e., the Cartesian edges that satisfy the *S1-condition* in $\langle N[x] \rangle$, resp. in $\langle N[y] \rangle$, induce a connected subgraph of $\langle N[x] \cup N[y] \rangle$.
2. If the color-continuation fails, we check if $\langle N[x] \rangle$ and $\langle N[y] \rangle$ are thin. If both neighborhoods are thin we can use Algorithm 5 to get a proper color-continuation from $\langle N[x] \rangle$ to $\langle N[y] \rangle$, see

Algorithm 9 General Approach

INPUT: a thin graph G
compute backbone-vertices of G, order them in BFS and store them in $\mathbb{B}_{BFS}$;
$x \leftarrow$ first vertex of $\mathbb{B}_{BFS}$;
$W \leftarrow x$;
FactorSubgraph($\langle N[x] \rangle$);
while $\mathbb{B}_{BFS} \neq \emptyset$ **do**
  $H \leftarrow \langle \cup_{w \in W} N[w] \rangle$;
  for all $y \in N[x] \cap \mathbb{B}_{BFS}$ **do**
    FactorSubgraph($\langle N[y] \rangle$);
    compute the combined coloring of H and $\langle N[y] \rangle$;
    if color-continuation fails from H to $N[y]$ **then**
      if $\langle N[x] \rangle$ and $\langle N[y] \rangle$ are thin **then**
        $C \leftarrow \{$color $c \mid$ color-continuation for c fails$\}$;
        CombineFactors(H, $\langle N[y] \rangle$, W, C); (Algorithm 5)
        mark all vertices and all edges of $\langle N[y] \rangle$ as "checked";
      else if (x,y) is indispensable in $\langle N[x] \cup N[y] \rangle$ **then**
        FactorSubgraph($\langle N[x] \cup N[y] \rangle$);
      else
        FactorSubgraph($N^*_{x,y}$);
      end if
      compute the combined coloring of H and $\langle N[y] \rangle$;
    end if
  end for
  delete x from $\mathbb{B}_{BFS}$;
  $x \leftarrow$ first vertex of $\mathbb{B}_{BFS}$;
end while
while there exists a vertex $x \in V(H)$ that is not marked as "checked" **do**
  if there exists edges (x,y) that are not marked as "checked" **then**
    FactorSubgraph($N^*_{x,y}$);
  else
    take an arbitrary edge $(x,v) \in E(H)$;
    FactorSubgraph($N^*_{x,v}$);
  end if
end while
for each edge $e \in E(H)$ **do**
  assign color of e to edge $e \in E(G)$;
end for
check and merge factors with Algorithm 11;
OUTPUT: G with colored G_j-fiber, and Factors of G;

Algorithm 10 FactorSubgraph

INPUT: a graph H
compute the PFD of H and color the Cartesian edges in H that satisfy the *S1-condition*;
mark all vertices x with $|S_H(x)| = 1$ as "checked";
mark all edges that satisfy the *S1-condition* as "checked";
Return partially colored H;

Algorithm 11 Check Factors

INPUT: a thin colored graph G
take one connected component $G_1^*, \ldots, G_l^*$ of each color $1, \ldots, l$ in G;
$I \leftarrow \{1, \ldots, l\}$;
$J \leftarrow I$;
for $k = 1$ to l **do**
  for each $S \subset J$ with $|S| = k$ **do**
    compute two connected components A, A' of G induced by the colored edges of G with color $i \in S$, and $i \in I \backslash S$, resp;
    compute $H_1 = \langle p_A(G) \rangle$ and $H_2 = \langle p_{A'}(G) \rangle$;
    if $H_1 \boxtimes H_2 \backsimeq G$ **then**
      save H_1 as prime factor;
      $J \leftarrow J \backslash S$;
    end if
  end for
end for

Section 4.3.2. Since both neighborhoods are thin, for all vertices v in $N[x]$, resp. $N[y]$, holds $|S_x(v)| = 1$, resp. $|S_y(v)| = 1$. Hence all edges in $\langle N[x]\rangle$, resp. $\langle N[y]\rangle$, satisfy the *S1-condition*. Therefore, the Cartesian edges span $\langle N[x]\rangle$ and $\langle N[y]\rangle$ and thus, by the color-contiuation property, $\langle N[x] \cup N[y]\rangle$ as well.

3. If one of the neighborhoods is not thin then we check whether the edge (x,y) is dispensable or not w.r.t. $\langle N[x] \cup N[y]\rangle$. If this edge is indispensable then Corollary 6.4 implies that $\langle N[x] \cup N[y]\rangle$ is a proper subproduct. Moreover, Lemma 3.20 implies that $|S_{\langle N[x]\cup N[y]\rangle}(x)| = 1$. and $|S_{\langle N[x]\cup N[y]\rangle}(y)| = 1$. From Lemma 3.21 we can conclude that the determined Cartesian edges of $\langle N[x] \cup N[y]\rangle$ induce a connected subgraph of $\langle N[x] \cup N[y]\rangle$.

4. Finally, if (x,y) is dispensable in $\langle N[x] \cup N[y]\rangle$ we can not be assured that $\langle N[x] \cup N[y]\rangle$ is a proper subproduct. In this case we factorize $N^*_{x,y}$. Again, Lemma 3.20 implies that $|S_{N^*_{x,y}}(x)| = 1$ and $|S_{N^*_{x,y}}(y)| = 1$. Moreover, from Lemma 3.21 follows that all Cartesian edges that satisfy the *S1-condition* on $N^*_{x,y}$ induce a connected subgraph of $N^*_{x,y}$.

Assume now that $H = \langle \cup_{w\in W} N[w]\rangle$. Clearly, the previous four steps are valid for all consecutive backbone vertices $x,y \in \mathbb{B}_{BFS}$. We have to show that we always get a proper color-continuation from H to $N[y]$ after these four steps (Line 21). This follows immediately from Lemma 3.32 and Corollaries 3.33 and 3.34 since $N[x] \subseteq H$. Moreover, since we always get a proper color-continuation from H to $N[y]$ using these four steps and the latter arguments concerning induced connected subgraphs we can conclude that all determined Cartesian edges induce a connected subgraph of $H = \langle \cup_{w\in\mathbb{B}(G)} N[w]\rangle$. Notice that $H = \langle \cup_{w\in\mathbb{B}(G)} N[w]\rangle = G$, since $\mathbb{B}(G)$ is a dominating set. The first while-loop will terminate since $\mathbb{B}_{BFS}$ is finite.

Therefore, those edges have been identified as Cartesian or if they have not been identified as Cartesian they are at least connected to Cartesian edges that satisfy the *S1-condition*. To see this assume that the edge (v,w) satisfies the *S1-condition* but is non-Cartesian. W.l.o.g. we assume $|S_H(v)| = 1$. Hence all Cartesian edges containing vertex v satisfy the *S1-condition*. Lemma 3.30 implies that each color of each Factor is represented on edges containing vertex v. Thus, edges (v,w) that satisfy the *S1-condition* but are not determined as Cartesian are connected to Cartesian edges that satisfy the *S1-condition*.

In all previous steps vertices x are marked as "checked" if there is a used subproduct K such that $|S_K(x)| = 1$. Edges are marked as "checked" if they satisfy the *S1-condition*. In the second while-loop all vertices that are not marked as "checked", i.e., $|S_K(x)| > 1$ for all used subproducts K, are treated. For all those vertices the N^*-neighborhoods $N^*_{x,y}$ are decomposed and colored. Lemma 5.12 implies that $|S_{N^*_{x,y}}(x)| = 1$ and $|S_{N^*_{x,y}}(y)| = 1$. Hence all Cartesian edges containing vertex x or y satisfy the

S1-condition. Lemma 3.30 implies that each color of every Factor is represented on edges containing vertex x, resp., y. Lemma 3.21 implies that all Cartesian edges that satisfy the *S1-condition*in $N^*_{x,y}$ induce a connected subgraph of Lemma $N^*_{x,y}$.

It remains to show that we get always a proper color-continuation. Since $|S_K(x)| > 1$ for all used subproducts K, we can conclude in particular that $|S_x(x)| > 1$. Therefore, we can apply Lemma 3.15 and conclude that there exists a vertex $z \in \mathbb{B}(G)$ s.t. $z \in N[x] \cap N[y]$ and hence $\langle N[z] \rangle \subseteq N^*_{x,y}$. This neighborhood $\langle N[z] \rangle$ was alreay colored in one of the previous steps since $z \in \mathbb{B}(G)$. Lemma 3.20 implies that $|S_{N^*_{x,y}}(z)| = 1$ and thus each color of each factor of $N^*_{x,y}$ is represented on edges containing vertex z and all those edges can be determined as Cartesian via the *S1-condition*. We get a proper color-continuation from the already colored subgraph H to $N^*_{x,y}$ since $N[z] \subseteq H$ and $N[z] \subseteq N^*_{x,y}$, which follows from Lemma 3.32 and Corollary 3.34.

Finally, as argued before all edges that satisfy the *S1-condition* are connected to Cartesian edges that satisfy the *S1-condition*. Notice that this are *all* edges of G after the while-loop has terminated. Thus, the set of determined Cartesian edges induce a connected *spanning* subgraph G. By the color-continuation property we can conclude that the final number of colors on G is at most the number of colors that were used in the first neighborhood. This number is at most $\log\Delta$, since every product of k non-trivial factors must have at least 2^k vertices. Let's say we have l colors. As shown before, all vertices are "checked" and thus we can conclude from Lemma 3.30 and the color-continuation property that each vertex $x \in V(G)$ is incident to an edge with color c for all $c \in \{1,\ldots,l\}$. Thus, we end with a combined coloring F_G on G where the domain of F_G consists of all edges that were determined as Cartesian in the previously used subproducts.

It remains to verify which of the possible factors are prime factors of G. This task is done by using Algorithm 11. Clearly, for some subset $S \subset J$, S will contain all colors that occur in a particular G_i-fiber G_i^a which contains vertex a. Together with the latter arguments we can conclude that the set of S-colored edges in G_i^a spans G_i^a. Since the global PFD induces a local decomposition, even if the used subproducts are not thin, every layer that satisfies the *S1-condition* in a used subproduct with respect to a local prime factor is a subset of a layer with respect to a global prime factor. Thus, we never identify colors that occur in copies of different global prime factors. In other words, the coloring F_G is a refinement of the product coloring of the global PFD, i.e., it might happen that there are more colors than prime factors of G. This guarantees that a connected component of the graph induced by all edges with a color in S induces a graph that is isomorphic to G_i. The same arguments show that the colors that are not in S lead to the appropriate cofactor. Thus G_i will be recognized.

□

Remark 6.6. Algorithm 9 is a generalization of the previous results and computes the PFD of NICE, CHIC and locally unrefined thin graphs. Moreover, even if we do not claim that the given graph G is thin one can compute the PFD of G as follows: We apply Algorithm 9 on G/S. The prime factors of G can be constructed by using the information of the prime factors of G/S as shown in Section 2.3.2.

In the last part of this section, we show that the time complexity to decompose any connected thin graph G into its prime factors with respect to the strong product is $O(|V| \cdot \Delta^6)$ time

Lemma 6.7. *Given a thin graph $G = (V,E)$ with bounded maximum degree Δ, then Algorithm 9 determines the prime factors of G w.r.t. the strong product in $O(|V| \cdot \Delta^6)$ time.*

Proof. For determining the backbone $\mathbb{B}(G)$ we have to check for a particular vertex $v \in V(G)$ whether there is a vertex $w \in N[v]$ with $N[w] \cap N[v] = N[v]$. This can be done in $O(\Delta^2)$ time for a particular vertex w in $N[v]$. Since this must be done for all vertices in $N[v]$ we end in time-complexity $O(\Delta^3)$. This step must be repeated for all $|V|$ vertices of G. Hence, the time complexity for determining $\mathbb{B}(G)$ is $O(|V| \cdot \Delta^3)$. Computing $\mathbb{B}_{BFS}$ via the breadth-first search takes $O(|V| + |E|)$ time. Since the number of edges is bounded by $|V| \cdot \Delta$ we can conclude that this task needs $O(|V| \cdot \Delta)$ time.

We consider now the Line 6 – 26 of the algorithm. The while-loop runs at most $|V|$ times. Computing H in Line 7, i.e., adding a neighborhood to H, can be done in linear time in the number of edges of this neighborhood, that is in $O(\Delta^2)$ time. The for-loop runs at most Δ times. The PFD of $\langle N[y] \rangle$ can be computed in $O(\Delta^4)$ time, see Lemma 2.23. The computation of the combined coloring of H and $\langle N[y] \rangle$ can be done in constant time. For checking if the color-continuation is valid one has to check at most for all edges of $\langle N[v_i] \rangle$ if a respective colored edge was also colored in H, which can be done in $O(\Delta^2)$ time. Notice that all "if" and "else" conditions are bounded by the complexity of the PFD of the largest subgraph that is used and therefore by the complexity of the PFD of $N^*_{x,y}$. As shown in the proof of Lemma 5.14, the number of edges in each N^*-neighborhood is bounded by $(1 + \Delta \cdot (\Delta - 1)) \cdot \Delta$. Lemma 2.23 implies that the PFD of each N^*-neighborhood takes $O(\Delta^5)$ time. Considering all steps of Line 6 – 26 we end in an overall time complexity $O(|V| \cdot \Delta \cdot \Delta^5) = O(|V| \cdot \Delta^6)$.

Using the same arguments, one shows that the time complexity of the second while-loop is $O(|V| \cdot \Delta^5)$. The last for-loop (Line 35–37) needs $O(|E|) = O(V \cdot \Delta)$ time.

Finally, we have to consider Line 38. Using the same arguments as in the proof of Lemma 4.13 we can conclude that extracting the possible factors and the isomorphism test for a fixed bijection (Algorithm 11) can be done in $O(|V| \cdot \Delta)$ time. Considering all steps of Algorithm 9 we end in an overall time complexity $O(|V| \cdot \Delta^6)$. □

7

Approximate Graph Products

In this chapter we discuss approximate strong graph products. First, we analyze the complexity to determine such products. We then explain how Algorithm 9 can be modified in order to recognize approximate products. In the last part of this chapter, we evaluate the performance of this algorithm on a sample of approximate graph products and try to answer the following questions:

1. How often do we find both original factors in the disturbed product depending on the percentage of perturbation, respectively the ratio of backbone prime 1-neighborhoods?
2. Depending on the percentage of perturbation how fast does the number of backbone prime 1-neighborhoods grow?
3. How large is the maximal factorized subgraph of the disturbed product depending on the percentage of perturbation, respectively the ratio of backbone prime 1-neighborhoods?

7.1 Complexity

For a formal definition of approximate graph products we begin with the definition of the distance between two graphs. We say the *distance* $d(G,H)$ between two graphs G and H is the smallest integer

k such that G and H have representations G', H' for which the sum of the symmetric differences between the vertex sets of the two graphs and between their edge sets is at most k. That is, if

$$|V(G') \triangle V(H')| + |E(G') \triangle E(H')| \leq k.$$

A graph G is a *k-approximate graph product* if there is a product H such that

$$d(G,H) \leq k.$$

Now, we investigate the complexity of recognizing k-approximate graph products. We first show that k-approximate graph products can be recognized in polynomial time for constant values of k. To this end, we begin with a bound on the number of graphs of distance k from a given connected graph G.

Lemma 7.1. *Let G be a connected graph on n vertices. Then the number of connected graphs of distance $\leq k$ from G is $O(n^{2k})$.*

Proof. We bound the number of graphs (including also disconnected graphs) H of distance $\leq k$ from G. First let $V(H) = V(G)$ and $E(H) = E(G)$. We modify the edge set $E(H)$. We have $\binom{n}{2} = (n)(n-1)/2 = O(n^2)$ ways to select a pair of vertices in $V(G)$. If a selected pair is an edge of $E(G)$ we delete it from $E(H)$, otherwise we add the corresponding edge. We do this i-times and obtain $O(n^{2i})$ graphs. Summing over all i from 0 to k, this yields $O(n^{2k})$ graphs, and in particular all graphs will distance of at most k from G that have the same vertex set as G.

Now we allow the vertex set to change. Suppose we only add $j \leq k$ isolated vertices. We proceed with $V(H) = V(G) \cup \{v_1, \ldots, v_j\}$ and $E(H) = E(G)$. Now we have $(n+j)(n+j-1)/2 = O(n^2)$ ways to select pairs in $V(H)$. Hence we can re-use the argument above to see that this generates no more than $O(n^{2k})$ distinct graphs.

Finally, suppose we add l_1 and delete l_2 vertices. Of course, we have $l_1 + l_2 \leq k$. For a fixed l_1, we know from the previous paragraph that there a no more than $O(n^{2l_1})$ distinct graphs. In each of them, we have at most $\binom{n}{l_2} \in O(n^{l_2})$ ways to delete vertices that were already there in $V(G)$. (Note that deleting a newly inserted vertex is equivalent to reducing l_1 and hence need not be considered). For fixed l_1 and l_2, we can proceed by adding or deleting edges. Now we have $\binom{n+l_1-l_2}{2} \in O(n^2)$ ways to select, and we can repeat this no more than $i \leq k - l_1 - l_2$ times, giving us access to no more than $O(n^{2k})$ graphs. There are $O(k^2)$ ways of choosing l_1 and l_2, hence we have no more than $O(k^2 \cdot n^{2k})$.

The lemma follows by treating k as a prescribed constant. □

Lemma 7.2. *For fixed k all strong and Cartesian k-approximate graph products can be recognized in polynomial time.*

Proof. For a given graph G the number of graphs of distance at most k is $O(n^{2k})$. The observation that every one of these graphs can be factored in polynomial time completes the proof. □

Without the restriction on k the problem of finding a product of closest distance to a given graph G is NP-complete for the Cartesian product. This has been shown by Feigenbaum and Haddad [14]. They proved that the following problem is NP-complete:

Problem 1 (Smallest Factorable Extension)**.** Input: A graph G and positive integers n and m.
Question: Is there a factorizable graph H such that $G \subseteq H$ and $|V(H)| \leq n$ and $|E(H)| \leq m$.

Problem 2 (Largest Factorable Subgraph)**.** Input: A graph G and positive integers n and m.
Question: Is there a factorizable graph H such that $H \subseteq G$ and $|V(H)| \geq n$ and $|E(H)| \geq m$.

We conjecture that this also holds for the strong product.

7.2 Recognition of Approximate Graph Products

In this section, we will show how Algorithm 9 can be modified and be used to recognize approximate products and how one can get a suggestion of the structure of the global factors. We do not claim that the given Algorithm finds an optimal solution in general.

First, consider the graph G of Figure 7.1. It approximates $P_5 \boxtimes P_7^T$, where P_7^T denotes a path that contains a triangle. Suppose we are unaware of this fact. Clearly, if G is non-prime, then every subproduct is also non-prime. We factor every suitable subproduct of backbone vertices (1-neighborhood, edge-neighborhood, N^*-neighborhood) that is not prime and try to use the information to find a product that is either identical to G or approximates it.

The graph G is thin and thus the backbone is a connected dominating set. The backbone $\mathbb{B}(G)$ consists of the vertices $0,1,\ldots,5$ and all vertices marked with "x". The induced neighborhood of all "x" marked vertices is prime. We do not use those neighborhoods but the ones of the vertices $0,1,\ldots,5$, factor their neighborhoods and consider the Cartesian edges that satisfy the *S1-condition* in the factorizations. There are two factors for every such neighborhood and thus, two colors for the Cartesian edges in every neighborhood. If two neighborhoods have a Cartesian edge that satisfy the *S1-condition* in common, we identify their colors. Notice that the color-continuation fails if we go

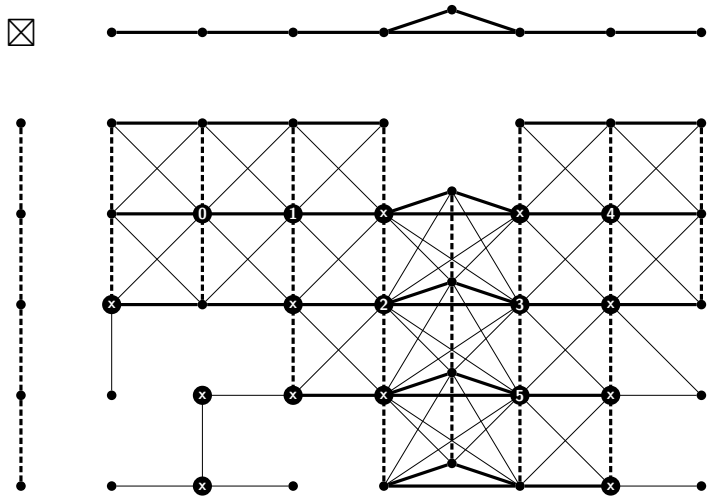


Figure 7.1: An approximate product G of the product of a path and a path containing a triangle. The resulting colored graph after application of the modified Algorithm 9 is highlighted with thick and dashed edges. We set $P = 1$, i.e., we do not use prime subproducts and hence only the vertices $0, 1, \ldots, 5$ are used. Taking out one maximal component of each color would lead to appropriate approximate factors of G.

from $\langle N[2]\rangle$ to $\langle N[3]\rangle$. Since $\langle N[2] \cup N[3]\rangle$ is not prime we factor the edge-neighborhood and get a proper color-continuation. In this way we end up with two colors altogether, one for the horizontal Cartesian edges and one for the vertical ones. If G is a product, then the edges of the same color span a subgraph with isomorphic components, that are either isomorphic to one and the same factor or that span isomorphic layers of one and the same factor.

Clearly, the components are not isomorphic in our example. But, under the assumption that G is an approximate graph product, we take one component for each color. In this example it would be useful to take a component of maximal size, say the one consisting of the horizontal edges through vertex 2, and the vertical ones through vertex 3. This components are isomorphic to the original factors P_5 and P_7^T. It is now easily seen that G can be obtained from $P_5 \boxtimes P_7^T$ by the deletion of edges.

As mentioned, Algorithm 9 has to be modified for the recognition of approximate products G. First note that we might possibly find fibers of the original prime factors, even if we do not cover the whole input graph by our algorithm.

Deleting or adding edges in a product graph H, resulting in a disturbed product graph G, usually makes the graph prime and also the neighborhoods $\langle N^G[v]\rangle$ that are different from $\langle N^H[v]\rangle$ and hence the subproducts (edge-neighborhood, N^*-neighborhood) that contain $\langle N^G[v]\rangle$. We call such subproducts *disturbed*.

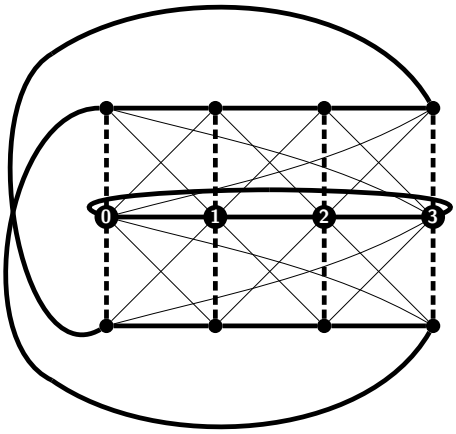


Figure 7.2: Shown is a prime graph G, also known as twisted product, with $\mathbb{B}(G) = \{0,1,2,3\}$. Each PFD of 1-neighborhoods leads to two factors. Notice that G can be considered as an approximate product of a path P_3 and a cycle C_4. After application of the modified Algorithm 9 with $P = 1$ we end with the given coloring (thick and dashed lines). Taking one minimal component of each color would lead to appropriate approximate factors of G.

In Algorithm 9 we therefore only use those subproducts of backbone vertices that are at least not prime. Moreover, we can restrict the set of allowed backbone vertices much more and use only those subproducts that have more than $P \geq 1$ prime factors and limit therefore the number of allowed subproducts. Hence, no prime regions or subproducts that have less or equal than P prime factors are used and therefore we don't identify colors of different locally determined fibers to only P colors. After coloring the graph one would take out one component of each color to determine the (approximate) factors. For many kinds of approximate products the connected components of graphs induced by the edges in one component of each color will not be isomorphic. In our case, where the approximate product was obtained by deleting edges, it is easy to see that we should take the maximal connected component of each color. Some examples for approximate products can be seen in Figure 7.1, 7.2, and 7.3.

The isomorphism test (line 38) in Algorithm 9 will not be applied. Thus, in prime graphs G we would not merge colors if the product of the corresponding approximate prime factors is not isomorphic to G.

We summarize the modifications we apply to Algorithm 9:

1. We do not claim that the given (disturbed) product is thin.
2. Theorem 3.12 and item 1. implies that we can not assume that the backbone is connected. Hence we only compute a BFS-ordering on connected components induced by backbone ver-

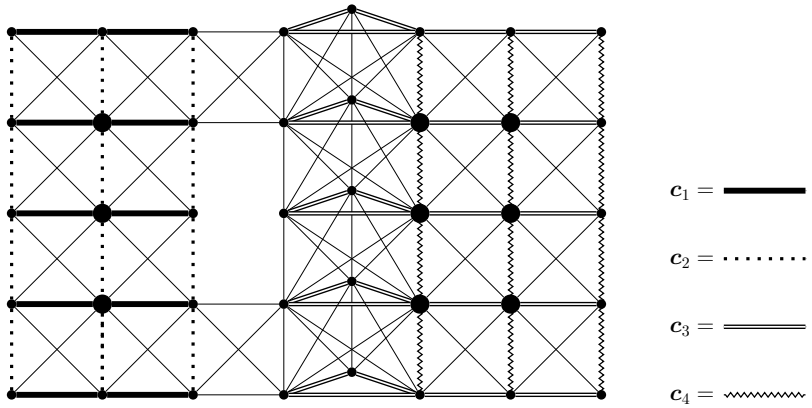


Figure 7.3: An approximate product G of the prime factors shown in Figure 7.1. In this example G is not thin. Obviously, this graph seems to be less disturbed than the one in Figure 7.1. The thick vertices indicate the backbone vertices with more then $P = 1$ prime factors. Application of the modified Algorithm 9 on G (without computing G/S), choosing $P = 1$ and using only thick vertices backbone vertices leads to a coloring with the four colors c_1, c_2, c_3 and c_4. This is due to the fact that the color-continuation fails, which would not be the case if we would allow to use also prime regions.

tices.

3. We only use those subproducts (1-neighborhoods, edge-neighborhood, N^*-neighborhood) that have more than $P \geq 1$ prime factors, where P is a fixed integer.
4. We do not apply the isomorphism test (line 38).
5. After coloring the graph, we take one minimal, maximal, or arbitrary connected component of each color. The choice of this component depends on the problem one wants to be solved.

Remark 7.3. In the remaining part of this chapter Algorithm 9 together with the applied modifications 1. – 5. will be called *modified Algorithm 9*.

7.3 Experimental Results

To complete this chapter, we perform in this section experimental tests concerning the recognition of approximate products. To disturb a product graph G one can apply several modifications on G like deleting edges, deleting vertices, adding vertices and edges, shifting edges etc. Here, we focus on the first kind of perturbation, i.e., deleting edges, and investigate how the modified Algorithm 9 behaves. Moreover, we try to answer the following questions:

1. How often do we find both original factors in the disturbed product depending on the percentage

of perturbation, respectively the ratio of backbone prime 1-neighborhoods?

2. Depending on the percentage of perturbation how fast does the number of backbone prime 1-neighborhoods grow?

3. How large is the maximal factorized subgraph of the disturbed product depending on the percentage of perturbation, respectively the ratio of backbone prime 1-neighborhoods?

7.3.1 A Measure of Perturbation by Deleting Edges

When deleting an edge (a,b) in a given product graph one disturbs more than only the 1-neigborhoods $\langle N[a]\rangle$ and $\langle N[b]\rangle$ in general, see Figure 7.4. In fact, all neighborhoods $\langle N[z]\rangle$ with $z \in \langle N[a] \cap N[b]\rangle$ are disturbed.

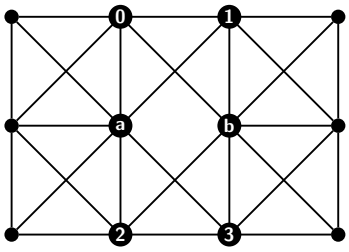


Figure 7.4: Deleting the edge (a,b) in the given strong product graph G disturbs all neighborhoods $\langle N^G[z]\rangle$ with $z \in N^G[a] \cap N^G[b] = \{a,b,0,1,2,3\}$

Definition 7.4. Let $G = (V,E)$ be a given graph and $a,b \in V$. We call $U_{a,b} = N[a] \cap N[b]$ the *common environment* of *a* and *b* or *environment* of *a* and *b* for short. $\overline{U}_G$ denotes the average number of vertices contained in environments of any two connected closed neighborhoods of G, i.e.,

$$\overline{U}_G = \frac{\sum_{(a,b)\in E} |U_{a,b}|}{|E|}$$

Clearly, if for some graphs G_1 and G_2 holds $\overline{U}_{G_1} > \overline{U}_{G_2}$ then the probability to disturb more neighborhoods in G_1 as in G_2 by deleting an edge is higher. Notice that $\overline{U}_G$ becomes maximal among all graphs with n vertices if $G \simeq K_n$.

We show in the following how for a given product graph $G = G_1 \boxtimes G_2$ the value $\overline{U}_G$ depends on the values $\overline{U}_{G_1}$ and $\overline{U}_{G_2}$. For this, we first state a well-known lemma concerning the number of vertices and edges in a strong product graph and treat afterwards the average degree of strong product graphs.

Lemma 7.5 ([32])**.** *Let $G = G_1 \boxtimes G_2$ be a strong product graph. Then it holds:*

$$|V(G)| = |V(G_1)| \cdot |V(G_2)|$$

and

$$|E(G)| = |V(G_1)| \cdot |E(G_2)| + |V(G_2)| \cdot |E(G_1)| + 2|E(G_1)| \cdot |E(G_2)|.$$

By definition of the strong product we can directly infer the next lemma.

Lemma 7.6. *Let $G = G_1 \boxtimes G_2$ be a strong product graph and $v = (v_1, v_2) \in V(G)$ be an arbitrary vertex. Then it holds:*

$$\deg(v) = \deg(v_1) + \deg(v_2) + \deg(v_1)\deg(v_2)$$

We show now that the average degree of a given strong product graph depends on the average degrees of its factors.

Lemma 7.7. *Let $G = (V,E) = G_1 \boxtimes G_2$ be a strong product graph of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, then it holds:*

$$\overline{\deg}(G) = \overline{\deg}(G_1) + \overline{\deg}(G_2) + \overline{\deg}(G_1)\overline{\deg}(G_2)$$

Proof.

$$\begin{aligned}
\overline{\deg}(G) &= \frac{\sum_{v \in V} \deg(v)}{|V|} = \frac{\sum_{(v_1,v_2) \in V} \deg((v_1,v_2))}{|V|} \\
&= \frac{\sum_{v_1 \in V_1} \sum_{v_2 \in V_2} (\deg(v_1) + \deg(v_2) + \deg(v_1)\deg(v_2))}{|V_1| \cdot |V_2|} \\
&= \frac{\sum_{v_1 \in V_1} \sum_{v_2 \in V_2} \deg(v_1)}{|V_1| \cdot |V_2|} + \frac{\sum_{v_1 \in V_1} \sum_{v_2 \in V_2} \deg(v_2)}{|V_1| \cdot |V_2|} + \frac{\sum_{v_1 \in V_1} \sum_{v_2 \in V_2} \deg(v_1)\deg(v_2)}{|V_1| \cdot |V_2|} \\
&= \frac{|V_2| \sum_{v_1 \in V_1} \deg(v_1)}{|V_1| \cdot |V_2|} + \frac{|V_1| \sum_{v_2 \in V_2} \deg(v_2)}{|V_1| \cdot |V_2|} + \frac{\sum_{v_1 \in V_1} \deg(v_1) \sum_{v_2 \in V_2} \deg(v_2))}{|V_1| \cdot |V_2|} \\
&= \overline{\deg}(G_1) + \overline{\deg}(G_2) + \overline{\deg}(G_1)\overline{\deg}(G_2)
\end{aligned}$$

□

Definition 7.8. Let $G = (V,E)$ be a given graph. We denote with $\overline{N}_G$ the average number of vertices contained in closed neighborhoods of G, i.e.,

$$\overline{N}_G = \frac{\sum_{v \in V} |N[v]|}{|V|}$$

Lemma 7.9. *Let $G = (V,E)$ be a given graph and $\overline{N}_G$ be the average number of vertices in each closed neighborhood of G. Then it holds:*

$$\overline{N}_G = \overline{\deg}(G) + 1$$

Proof. Note that $|N[v]| = \deg(v) + 1$.

$$\overline{N}_G = \frac{\sum_{v \in V} |N[v]|}{|V|} = \frac{\sum_{v \in V} (\deg(v) + 1)}{|V|} = \frac{\sum_{v \in V} \deg(v) + |V|}{|V|} = \frac{\sum_{v \in V} \deg(v)}{|V|} + 1 = \overline{\deg}(G) + 1$$

□

Lemma 7.10. *Let $G = (V,E) = G_1 \boxtimes G_2$ be a strong product graph of two graphs $G_1 = (V_1,E_1)$ and $G_2 = (V_2,E_2)$. Let $\overline{N}_G$ be the average number of vertices in each closed neighborhood of G. Then it holds:*

$$\overline{N}_G = \overline{N}_{G_1} \cdot \overline{N}_{G_2}$$

Proof. Lemmas 7.7 and 7.9 imply that

$$\overline{N}_G = \overline{\deg}(G_1) + \overline{\deg}(G_2) + \overline{\deg}(G_1)\overline{\deg}(G_2) + 1.$$

Since $\overline{\deg}(G_i) = \overline{N}_{G_i} - 1$, $i = 1,2$ we can infer that

$$\overline{N}_G = \overline{N}_{G_1} - 1 + \overline{N}_{G_2} - 1 + (\overline{N}_{G_1} - 1)(\overline{N}_{G_2} - 1) + 1.$$

Hence, $\overline{N}_G = \overline{N}_{G_1} \cdot \overline{N}_{G_2}$. □

Lemma 7.11. *Let $G = (V,E) = G_1 \boxtimes G_2$ be a strong product graph of two graphs $G_1 = (V_1,E_1)$ and $G_2 = (V_2,E_2)$. Then it holds:*

$$\overline{U}_G = \frac{|V_1| \cdot |E_2| \cdot \overline{N}_{G_1} \cdot \overline{U}_{G_2} + |V_2| \cdot |E_1| \cdot \overline{N}_{G_2} \cdot \overline{U}_{G_1} + 2|E_1| \cdot |E_2| \cdot \overline{U}_{G_1} \cdot \overline{U}_{G_2}}{|E|}$$

Proof. The coordinates of vertices v are denoted by (v_1, v_2) and 1-neighborhoods in a factor G_i are denoted by N^i for $i = 1, 2$. Applying Lemma 2.26 and basic set theory the following equations can be inferred.

$$\begin{aligned}
|E| \cdot \overline{U}_G &= \sum_{(a,b)\in E} |U_{a,b}| = \sum_{(a,b)\in E} |N[a] \cap N[b]| \\
&= \sum_{((a_1,a_2),(b_1,b_2))\in E} |(N^1[a_1] \times N^2[a_2]) \cap (N^1[b_1] \times N^2[b_2])| \\
&= \sum_{((a_1,a_2),(b_1,b_2))\in E} |(N^1[a_1] \cap N^1[b_1]) \times (N^2[a_2] \cap N^2[b_2])| \\
&= \sum_{((a_1,a_2),(b_1,b_2))\in E} |N^1[a_1] \cap N^1[b_1]| \cdot |N^2[a_2] \cap N^2[b_2]| \\
&= \underbrace{\sum_{y\in V_2} \sum_{((a_1,y),(b_1,y))\in E} |N^1[a_1] \cap N^1[b_1]| \cdot |N^2[y]|}_{A:=} + \\
&\quad \underbrace{\sum_{x\in V_1} \sum_{((x,a_2),(x,b_2))\in E} |N^1[x]| \cdot |N^2[a_2] \cap N^2[b_2]|}_{B:=} + \\
&\quad \underbrace{\sum_{((a_1,a_2),(b_1,b_2))\in E, a_1\neq b_1, b_2\neq b_2} |N^1[a_1] \cap N^1[b_1]| \cdot |N^2[a_2] \cap N^2[b_2]|}_{C:=}
\end{aligned}$$

First, we consider term A.

$$\begin{aligned}
A &= \sum_{y\in V_2} \sum_{((a_1,y),(b_1,y))\in E} |N^1[a_1] \cap N^1[b_1]| \cdot |N[y]| \\
&= \sum_{y\in V_2} |N^2[y]| \sum_{((a_1,y),(b_1,y))\in E} |N^1[a_1] \cap N^1[b_1]| \\
&= \sum_{y\in V_2} |N^2[y]| \sum_{(a_1,b_1)\in E_1} |N^1[a_1] \cap N^1[b_1]| \\
&= \frac{|V_2| \cdot |E_1| \sum_{y\in V_2} |N^2[y]| \sum_{(a_1,b_1)\in E_1} |N^1[a_1] \cap N^1[b_1]|}{|V_2||E_1|} \\
&= |V_2| \cdot |E_1| \cdot \overline{N}_{G_2} \cdot \overline{U}_{G_1}
\end{aligned}$$

Analogously it can be shown that $B = |V_1| \cdot |E_2| \cdot \overline{N}_{G_1} \cdot \overline{U}_{G_2}$.

$$\begin{aligned}
C &= \sum_{((a_1,a_2),(b_1,b_2))\in E, a_1\neq b_1, a_2\neq b_2} |N^1[a_1] \cap N^1[b_1]| \cdot |N^2[a_2] \cap N^2[b_2]| \\
&= 2 \sum_{(a_1,b_1)\in E_1} \sum_{(a_2,b_2)\in E_2} |N^1[a_1] \cap N^1[b_1]| \cdot |N^2[a_2] \cap N^2[b_2]| \\
&= \frac{2|E_1||E_2| \sum_{(a_1,b_1)\in E_1} |N^1[a_1] \cap N^1[b_1]| \sum_{(a_2,b_2)\in E_2} |N^2[a_2] \cap N^2[b_2]|}{|E_1||E_2|} \\
&= 2|E_1| \cdot |E_2| \cdot \overline{U}_{G_1} \cdot \overline{U}_{G_2}
\end{aligned}$$

Hence,

$$\overline{U}_G = \frac{A+B+C}{|E|} = \frac{|V_1| \cdot |E_2| \cdot \overline{N}_{G_1} \cdot \overline{U}_{G_2} + |V_2| \cdot |E_1| \cdot \overline{N}_{G_2} \cdot \overline{U}_{G_1} + 2|E_1| \cdot |E_2| \cdot \overline{U}_{G_1} \cdot \overline{U}_{G_2}}{|E|}$$

□

As already mentioned, if for some graphs G_1 and G_2 holds $\overline{U}_{G_1} > \overline{U}_{G_2}$ the chance to disturb more neighborhoods in G_1 as in G_2 by deleting an edge is higher. As shown the value of $\overline{U}_G$ of a product graph G depends on the respective value $|V_i|$, $|E_i|$, $\overline{N}_{G_i}$, and $\overline{U}_{G_i}$ of its factors G_i. We will use this fact when examining the experimental results.

7.3.2 Data Set

We now give an overview of the data set and the resulting product graphs that are used in our experiments.

Prime Graph Data Set For the experiment a small basic data set containing four different prime graphs P, C, T, and I is chosen, see Figure 7.5.

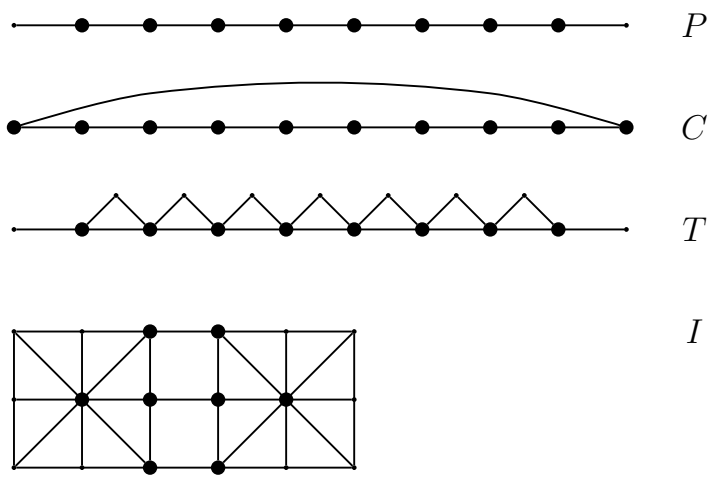


Figure 7.5: Four prime graphs P, C, T and I that are used to compute different product graphs as test set. Backbone vertices are highlighted as thick dots.

The graph denoted by P is a path and the graph C is a closed path. These graphs have the simplest structure. Both graphs are NICE and CHIC and can therefore be covered by thin 1-neighborhoods only. The backbone vertex set of C contains all vertices $V(C)$ while the backbone vertex set of P contains only the "interior" vertices as shown in Figure 7.5. The graph T is a path that contains 7 triangles. This graph cannot be covered by thin 1-neighborhoods. Hence, for a (non-trivial) product graph $G = H \boxtimes T$ the edge-neighborhoods and N^*–neighborhoods will become crucial when computing the PFD of G with the modified Algorithm 9. The graph I has the densest structure, i.e., I is the graph where the most edges have to be removed such that I becomes disconnected. Moreover, the

respective values $|V(I)|$, $|E(I)|$, $\overline{U}_I$, and $\overline{N}_I$ are the largest ones among the values of the other graphs.

	P	C	T	I
$\|V(G)\|$	10	10	17	18
$\|E(G)\|$	9	10	23	35
$\|\mathbb{B}(G)\|$	8	10	8	8
$\overline{U}_G$	2	2	2.91	3.37
$\overline{N}_G$	2.8	3	3.71	4.89

Table 7.1: Depicted are the values $|V(G)|$, $|E(G)|$, $|\mathbb{B}(G)|$, $\overline{U}_G$, and $\overline{N}_G$ of the respective prime graphs that are used to compute the product graphs of out test data set.

In Table 7.1 the number of vertices and edges as well as the average number of vertices in neighborhoods $\overline{N}_G$ and the average number of vertices in environments of adjacent vertices $\overline{U}_G$ for $G \in \{P,C,T,I\}$ are represented. One can observe that $\overline{N}_P < \overline{N}_C < \overline{N}_T < \overline{N}_I$ and furthermore that $\overline{U}_P = \overline{U}_C < \overline{U}_T < \overline{U}_I$. As an example consider graph P and I. Deleting one edge (a,b) in P would disturb exactly the two neighborhoods $\langle N[a]\rangle$ and $\langle N[b]\rangle$ only. On the other hand, in graph I one averagely disturbs 3.37 neighborhoods when deleting an edge. Moreover, deleting an arbitrary edge in graph I would averagely disturb more neighborhoods than in all other graphs $G \in \{P,C,T\}$.

Product Graph Data Set Our test set of product graphs consists of all possible combinations of products of two of the prime graphs P, C, T, and I. As shown in Lemma 7.10 and 7.11 the values of $\overline{N}_G$ and $\overline{U}_G$ of a product graph G depend on the number of vertices, the number of edges, and the respective values $\overline{N}_{G_i}$ and $\overline{U}_{G_i}$ of the factors. Hence, it is easy to see why $\overline{U}_{I\boxtimes I}$ and $\overline{N}_{I\boxtimes I}$ becomes maximal and why $\overline{U}_{P\boxtimes P}$ and $\overline{N}_{P\boxtimes P}$ becomes minimal among all other products and why the values of the other products range between them.

Procedure The (modified) Algorithm 9 was implemented in *C++*. In addition, the *Boost Graph Library* was used [51]. Given one of the computed strong product graphs G we randomly disturb the product by removing edges from it. The number of edges that will be removed from G in each step depends on the number of edges |E(G)| of G. To be more precise, in each step we delete $\frac{i}{100}|E(G)|$ of edges with $i = 0.5, 1, 1.5, \ldots, 20$. After randomly deleting i% of edges we use the modified Algorithm 9 with $P = 1$, i.e., we do not allow to use subproducts that are prime, to compute a partial colored subgraph of G. Each step is repeated 200 times for each graph in the product graph data set.

	$P \boxtimes P$	$P \boxtimes C$	$C \boxtimes C$	$P \boxtimes T$	$C \boxtimes T$	$P \boxtimes I$	$C \boxtimes I$	$T \boxtimes T$	$T \boxtimes I$	$I \boxtimes I$
$\lvert V(G)\rvert$	100	100	100	170	170	180	180	289	306	324
$\lvert E(G)\rvert$	342	370	400	797	860	1142	1230	1840	2619	3710
$\lvert \mathbb{B}(G)\rvert$	64	80	100	64	80	64	80	64	64	64
$\overline{U}_G$	4.84	4.92	5	6.80	6.92	8	8.15	9.47	11.13	13.1
$\overline{N}_G$	7.84	8.4	9	10.38	11.12	13.69	14.67	13.73	18.12	23.90

Table 7.2: The used products and their respective values $|V(G)|$, $|E(G)|$, $|\mathbb{B}(G)|$, $\overline{U}_G$ and $\overline{N}_G$.

7.3.3 Experiment and Results

Recovering both original factors

To investigate how often both factors of the original graph product can be recovered in dependence of the ratio of perturbation we proceed as follows. After randomly deleting a fixed percentage of edges of each graph, we apply the modified Algorithm 9 to color the disturbed product. Then, one maximal connected component of each color is taken, to determine the approximate prime factors of G. After this, we check whether two of the determined factors are isomorphic to the original ones.

So as not to bias the results we must apply an additional step when checking the recognized approximate factors. Note, that it might happen that we get different but isomorphic factors although the corresponding layers of the original graph are parallel fibers. In this case we do not allow to treat those factors as different. As an example consider the colored graph in Figure 7.3. Here we would check if the factor that corresponds to a fiber with zigzag lines and the one that corresponds to a fiber with dotted lines are in parallel fibers of the original graphs. As it turns out they do and hence, they would not be treated as different and only one approximate factor that corresponds to the path P_5 would have been determined as an original factor. In particular, this approach is important for the four strong product graphs $P \boxtimes P$, $C \boxtimes C$, $T \boxtimes T$, and $I \boxtimes I$. Using this additional step, we can be assured that we found two isomorphic factors that do not appear in parallel fibers and hence, we do not bias the results.

Figure 7.6 shows the relative frequency of instances where both factors of the original graph product were recovered in dependence of the ratio of perturbation.

One immediately observes that the ratio of disturbed product graphs where both underlying factors were recovered decreases very fast and that there is a remarkable difference between the different

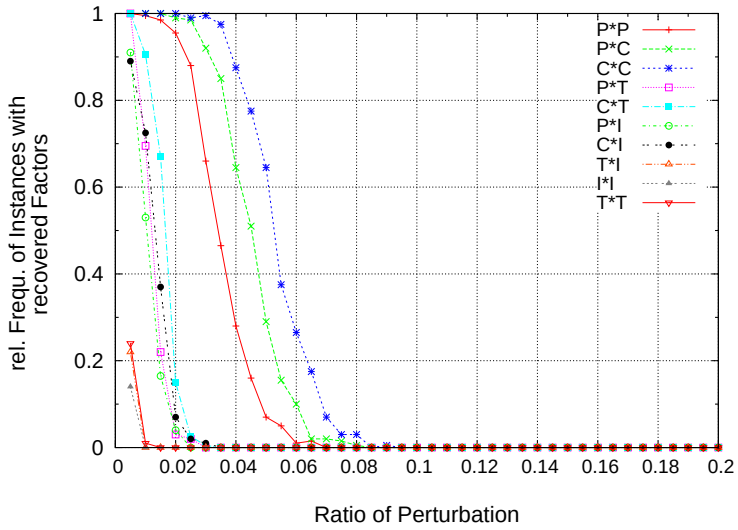


Figure 7.6: The plot shows the relative frequency of instances where both factors of the original graph product are recovered in dependence of the ratio of perturbation.

product graphs. The most "successful" candidate is the graph $C \boxtimes C$, followed by $P \boxtimes C$, and $P \boxtimes P$. The least "successful" ones are the instances of the graphs $T \boxtimes I$, $T \boxtimes T$, and $I \boxtimes I$. In the latter three graphs even a perturbation of only 1% disturbs the product graphs so much that it was not possible to recover the original factors. Likewise in the other graphs the chance to determine the underlying factors decreases also very fast, e.g., for the graphs $C \boxtimes T$, $C \boxtimes I$, $P \boxtimes T$, and $P \boxtimes I$ only 2% of perturbation is needed such that the percentage of instances with recovered underlying factors is less than 20%. Even in the most "successful" candidates $C \boxtimes C$, $P \boxtimes C$, and $P \boxtimes P$, a disturbance of $5-7\%$ leads to approximate graph products where only in $\sim 10\%$ of the instances the original factors were recognized.

Thus, there arise two questions:

1. Why do these graphs behave differently?
2. Why does the number of disturbed graphs with recovered original factors decreases so fast?

Why do these graphs behave differently? To explain this, we take the values $\overline{U}_G$ into account. Indeed, one can observe that the most "successful" candidates are the graphs with smallest values $\overline{U}_G$ that are $C \boxtimes C$, $P \boxtimes C$, and $P \boxtimes P$. The least "successful" candidates are the graphs $T \boxtimes T$, $T \boxtimes I$, and $I \boxtimes I$ that have largest values $\overline{U}_G$. Clearly, if for some graphs G_1 and G_2 holds $\overline{U}_{G_1} > \overline{U}_{G_2}$ then the probability to disturb more neighborhoods in G_1 as in G_2 by deleting an edge is higher. Moreover, the

graphs $T \boxtimes T$, $T \boxtimes I$, and $I \boxtimes I$ have also the largest number of edges. Thus, for a fixed percentage of removed edges, the number of removed edges in the latter three graphs is much higher as in the other ones. Taking together the latter arguments, we can conclude that for a fixed percentage of removed edges in $T \boxtimes T$, $T \boxtimes I$, and $I \boxtimes I$ more neighborhoods are disturbed as in the other graphs. Hence, even for a small ratio of perturbation the product structure of these graphs is heavily disturbed and thus, the underlying original factors cannot be recovered. The last arguments also explain why the plots of the instances of the other graphs are arranged in such a way.

Furthermore, one can observe that there is a remarkable difference between the graphs $C \boxtimes C$, $P \boxtimes C$, and $P \boxtimes P$, although the values $\overline{U}_{C \boxtimes C} = 5$, $\overline{U}_{P \boxtimes C} = 4.92$, and $\overline{U}_{P \boxtimes P} = 4.84$ and the number of removed edges for a fixed ratio of perturbation are quite similar. For example, for a perturbation of 5% there are 64.5% of instances in $C \boxtimes C$, 29% of instances in $P \boxtimes C$, and 7% of instances in $P \boxtimes P$ where both underlying factors where recognized.

To understand this phenomena we also take the ratio of backbone vertices into account. Notice that $|V(C \boxtimes C)| = |V(P \boxtimes C)| = |V(P \boxtimes P)| = 100$. Therefore, the ratio of backbone vertices are as follows: $|\mathbb{B}(C \boxtimes C)|/100 = 1$, $|\mathbb{B}(P \boxtimes C)|/100 = 0.8$, and $|\mathbb{B}(P \boxtimes P)|/100 = 0.64$. Note, the main obstacle for determining the prime factors is to obtain a proper color-continuation by usage of the respective subproducts. Hence, if one want to recover the underlying factors, there must be connected subgraphs in the perturbed product, that can be covered by non-prime subproducts and that contain at least one entire fiber of each factor. Moreover, one must ensure that at least one fiber of each factor gets exactly one color. Now, the ratio of 1-neighborhoods that can be used in $C \boxtimes C$ is higher than in $P \boxtimes C$ as well as it is higher in $P \boxtimes C$ than in $P \boxtimes P$. Hence, one can assume that the chance to find some connected undisturbed regions that contain an entire fiber of the original factors and thus, to determine at least one fiber of each factor, becomes higher for the approximate products of $C \boxtimes C$ as for $P \boxtimes C$ and $P \boxtimes P$ and higher for $P \boxtimes C$ as for $P \boxtimes P$.

In general, all instances of graphs with nearly the same values $\overline{U}_G$ are more "successful" if they have a cycle as factor. One can see that the values $\overline{U}_G$ are similar for all products $H \boxtimes C$ and $H \boxtimes P$, for a fixed factor $H \in \{P, C, T, I\}$. As argued, the ratio of backbone vertices plays an important role. But in addition, the structure of the factors and therefore the structure of the (disturbed) products has to be taken into account. Consider the prime factors P and C. Note, the deletion of a single edge in the path P would decompose it into two disconnected subgraphs. If one deletes an edge in the cycle C, this graph remains connected. Now, after application of the modified Algorithm 9 the disconnected path would have been colored with two different colors, one color for each connected component, while the disturbed cycle can entirely be covered by 1-neighborhoods and moreover, all of its edges

receive the same color. This gives rise to the assumption, that for a fixed percentage of removed edges in approximate products of $H \boxtimes C$, there are usually more "ways" to get a proper color-continuation than in approximate products of $H \boxtimes P$ with $H \in \{P, C, T, I\}$, see Figure 7.7.

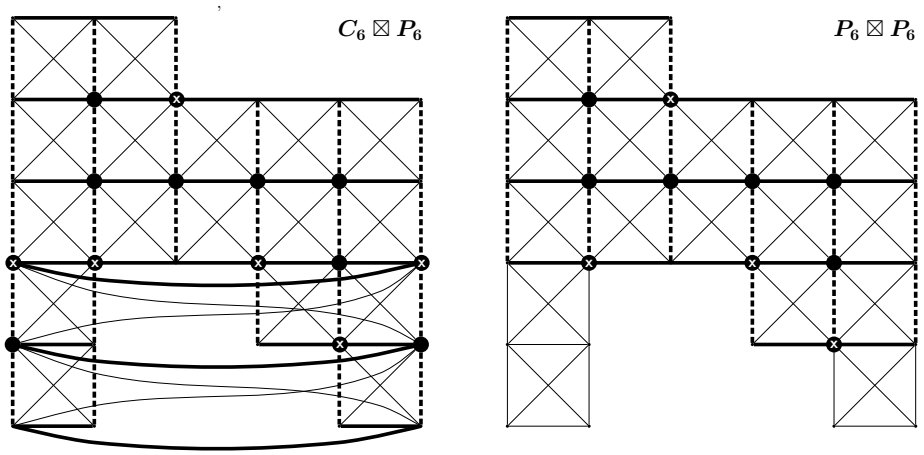


Figure 7.7: Shown are the colored graphs of disturbed products of $C_6 \boxtimes P_6$ and $P_6 \boxtimes P_6$ after applying modified Algorithm 9. In both graphs nearly the same percentage of edges are removed (left 31% and right 27%). The set of backbone vertices $\mathbb{B}$ consists of all thick-dotted vertices. For all backbone vertices that are marked with an "x" the induced neighborhood is prime and hence, they are not used to cover the graph. Although the number of backbone vertices in the disturbed product $C_6 \boxtimes P_6$ is larger as in in the disturbed product $P_6 \boxtimes P_6$ the ratio of prime 1-neighborhood is nearly the same, i.e., the number of backbone prime 1-neighborhoods divided by $|\mathbb{B}|$ is 0.42 in the left and 0.4 in the right graph. However, in the left graph there are more possibilities to cover it with undisturbed connected neighborhoods than in the right graph. Hence, by taking a maximal connected component of each color in the disturbed product $C_6 \boxtimes P_6$ the underlying factors would be recovered but not in the disturbed product $P_6 \boxtimes P_6$. Note, one can regard the right graph as an approximate product of $C_6 \boxtimes P_6$. In this case, the right graph is much more disturbed than the left graph.

Why does the number of disturbed graphs with recovered original factors decreases so fast? Figure 7.8 shows the ratio of prime 1-neighborhoods (number of prime backbone 1-neighborhoods divided by the number of all backbone 1-neighborhoods of the disturbed product) in dependence of the ratio of perturbation.

One can see that in all disturbed products the average number of prime neighborhoods increases fast, e.g. in the graphs $P \boxtimes P, P \boxtimes C$, and $C \boxtimes C$ only 5% of perturbation results in about 65% of prime 1-neighborhoods. More prime 1-neighborhoods can be found in the other graphs where only 5% of perturbation results in more than 90% prime 1-neighborhoods. Clearly, the more 1-neighborhoods are prime the fewer 1-neighborhoods can be used to recover the underlying factors. Taking into account

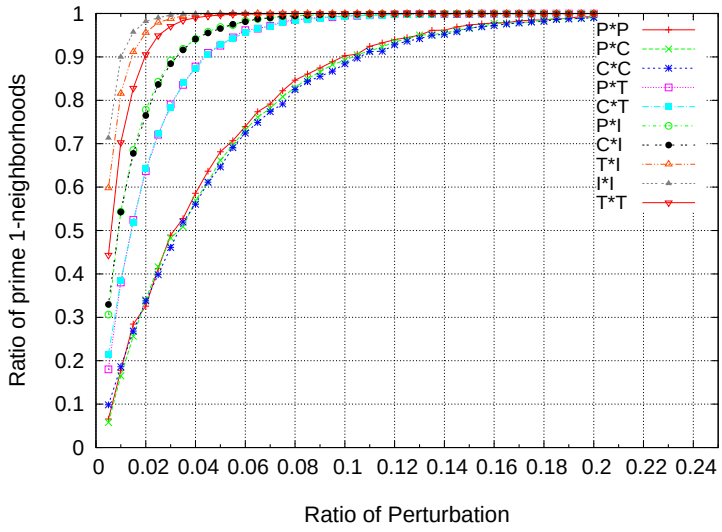


Figure 7.8: The plot shows the (relative) ratio of prime 1-neighborhoods, i.e., the number of backbone prime 1-neighborhoods divided by the number of all backbone 1-neighborhoods of the disturbed product, in dependence of the ratio of perturbation.

the respective values $\overline{U}_G$, one can easily see why the average number of prime 1-neighborhoods increases very fast. For example, consider the graph $G = C \boxtimes C$ with $\overline{U}_G = 5$. If we delete 1% of edges, i.e., four edges, we would have about $4 \cdot 5 = 20$ disturbed 1-neighborhoods. It holds $|\mathbb{B}(G)| = 100$ for this particular graph. Hence, on average 20% of used 1-neighborhoods are disturbed, even if we delete only four edges. As observable in Figure 7.8, the average ratio of prime 1-neighborhoods with a measured value of 0.186 is slightly less than 0.2, which might be explained with the circumstance that in some cases edges are removed from already disturbed neighborhoods. Moreover, the plot in Figure 7.8 shows that the ratio of prime 1-neighborhoods does not increase linearly. Again, we argue that with an increasing percentage of deleted edges the probability to remove an edge from an already disturbed neighborhood increases. Note, in these test cases usually removing one edge from a 1-neighborhood leads to a prime neighborhood and removing additional edges from this neighborhood preserves the property of being prime.

In addition, as one can see the graphs are grouped corresponding to the similarity of their values $\overline{U}_G$ and the number of their edges. The graphs $P \boxtimes P$, $P \boxtimes C$, and $C \boxtimes C$ are in one cluster. Their respective values $\overline{U}_G \in \{4.84, 4.92, 5\}$ and the number of removed edges for a fixed percentage perturbation are similar. The same holds for $P \boxtimes T, C \boxtimes T$ with $\overline{U}_G \in \{6.80, 6.92\}$ and $P \boxtimes I, C \boxtimes I$ with $\overline{U}_G \in \{8, 8.15\}$. Clearly, if the values $\overline{U}_{G_1}$ and $\overline{U}_{G_2}$ for two graphs are similar and moreover, the number of removed edges for a fixed percentage perturbation is alike, then almost the same number of neighborhoods in

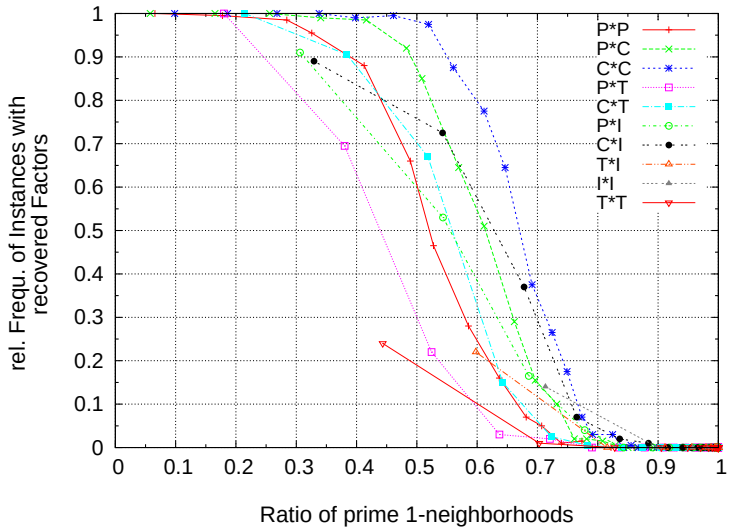


Figure 7.9: Shown is the relative frequency of instances where both factors of the original graph product are recovered in dependence of the relative ratio of prime 1-neighborhoods.

both graphs will be disturbed, after deleting a fixed percentage of edges. The remaining graphs $T \boxtimes T$, $T \boxtimes I$, and $I \boxtimes I$ are all clustered within a single group.

Figure 7.9 shows the relative frequency of instances where both factors of the original graph product were recovered in dependence of the ratio of prime 1-neighborhoods.

With an increasing number of prime 1-neighborhoods the number of graphs, where the underlying factors can be recovered, decreases. As observable, if $30-40\%$ of used 1-neighborhoods are prime then in more than 68% of the respective instances the underlying factors were recovered. In $P \boxtimes C$ and $C \boxtimes C$ there are even more; in $\sim 99\%$ of the instances the original factors were determined in the disturbed product graphs. If more than 70% of used 1-neighborhoods are prime then the chance to recover the original factors is less than 30% in all samples and if more than 90% of used 1-neighborhoods are prime then in no case the underlying factors was recognized. Again, it can be seen that in graphs, that have almost coinciding values $\overline{U}_G$, the chance to find the original factors in those graphs that have a cycle as factor is slightly better.

Maximal Factorized Subgraphs

In the remaining part of this section we will discuss maximal factorized subgraphs. For this purpose, we analyze the ratio of the maximal factorized subgraph in the disturbed product graph. Note that

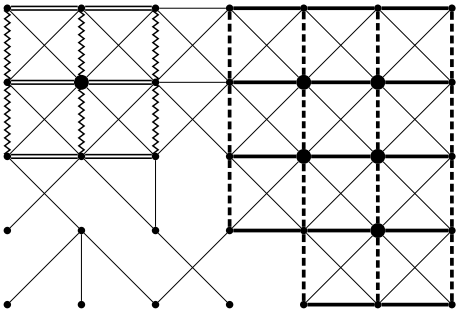


Figure 7.10: An approximate product graph of the product of two pathes P_5 and P_7 with a perturbation of 19%, i.e., 20 edges are removed. All backbone vertices with non-prime 1-neighborhoods are highlighted by thick dots. The computed subgraph of the Cartesian skeleton after application of modified Algorithm 9 is not connected. Four different colors are used. The maximal factorized subgraph H is the one induced by the vertices that are incident to the thick-lined and dashed-lined edges. In this particular example holds: Ratio $H = \frac{1}{2}\left(\frac{|V(H)|}{|V(G)|} + \frac{|E(H)|}{|E(G)|}\right) = \frac{1}{2}\left(\frac{19}{35} + \frac{51}{106}\right) = \frac{1}{2}(0.54 + 0.48) = 0.51$. Notice that $|\mathbb{B}(G)| = 15$ if the graph would not be disturbed. In the disturbed product only 6 of 15 backbone 1-neighborhoods are not prime. However, even if only 40% of originally non-prime neighborhoods can be used one can factorize more than 51% of the original graph in this example.

the recognized Cartesian skeleton of the disturbed product after application of modified Algorithm 9 need not be connected, see Figure 7.10.

Therefore, we take one maximal connected component of the computed Cartesian skeleton, i.e., a connected component with a maximal number of vertices and among all those subgraphs the ones having a maximal number of edges. The edges of the maximal factorized subgraph in the disturbed product are then the edges of the subgraph of the Cartesian skeleton and all non-Cartesian edges between those edges. Let H be such a subgraph of a disturbed product G' and let G be the original undisturbed product. We calculate the ratio of a maximal factorized subgraph as follows:

$$\text{Ratio } H = \frac{1}{2}\left(\frac{|V(H)|}{|V(G)|} + \frac{|E(H)|}{|E(G)|}\right)$$

Figure 7.11 and 7.12 show the relative ratio of maximal factorized subgraphs in dependence of the ratio of perturbation, respectively in dependence of the relative ratio of prime 1-neighborhoods.

In both plots, one can see that with an increasing ratio of perturbation and hence, with an increasing number of prime 1-neighborhoods, the size of maximal factorized subgraphs decreases. For a fixed percentage of removed edges in all graphs the size of maximal factorized subgraphs decreases in accordance with the decrease of the respective values $\overline{U}_G$. As observable, for a disturbance of 2% in the graphs $T \boxtimes T$, $T \boxtimes I$, and $I \boxtimes I$, the maximal factorized subgraphs averagely represent $\sim 10\%$ of

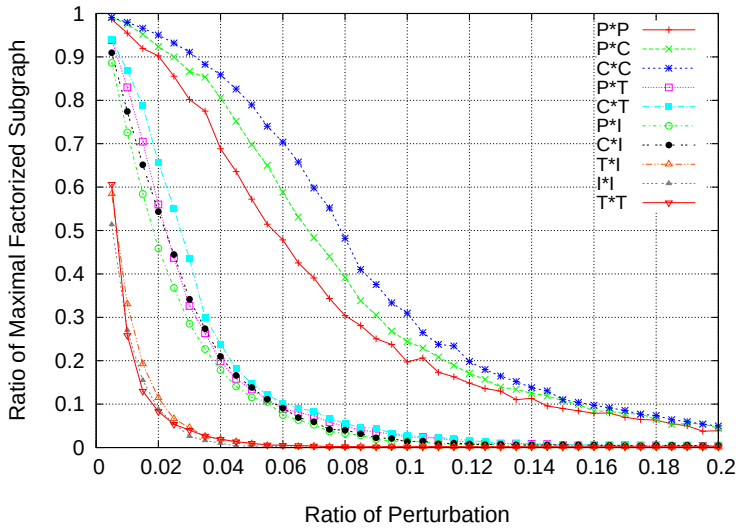


Figure 7.11: The plot shows the relative ratio of maximal factorized subgraphs in dependence of the ratio of perturbation.

the original product graph. If one removes 2% of the edges in the graphs $C \boxtimes T$, $C \boxtimes I$, $P \boxtimes T$, and $P \boxtimes I$, the maximal factorized subgraphs averagely represent $46 - 66\%$ of the original product. The graphs $C \boxtimes C$, $C \boxtimes P$, and $P \boxtimes P$ can be disturbed much more. If 10% of their edges were deleted it was possible to factorize at least $20 - 30\%$ of their subgraphs, but only $0 - 4\%$ in the other samples. As already argued, the values $\overline{U}_G$ and cardinalities of the edge sets, explain why the percentage of maximal factorized subgraphs in the graphs $T \boxtimes T$, $T \boxtimes I$, and $I \boxtimes I$ decreases faster than in the other graphs and why $C \boxtimes C$, $C \boxtimes P$, and $P \boxtimes P$ are more robust against perturbation. Again, it can be seen that the algorithm performs on graphs with almost coinciding values $\overline{U}_G$ that have a higher ratio of backbone vertices or that have a cycle as factor slightly better.

In Figure 7.12 it is observable that for a fixed ratio of used prime 1-neighborhoods the algorithm performs worst on the graph $T \boxtimes T$. Note, in order to receive a proper color-continuation in approximate products of this graph $T \boxtimes T$, it is crucial to use edge-neighborhoods and N^*-neighborhoods, since none of its 1-neighborhoods are thin. Thus, due to the structure of the graph $T \boxtimes T$, the computation of a proper color-continuation is much harder compared to other graphs. Even if parts of the approximate product of $T \boxtimes T$ can be factorized with 1-neighborhoods, in each step edge-neighborhoods and N^*-neighborhoods have to be factorized, to receive a proper color-continuation. Hence, in addition to the information provided by the PFD of 1-neighborhoods, we must use more "non-local" information. Therefore, it might happen that in the disturbed product many 1-neighborhoods can be factorized, but not the respective edge-neighborhoods and N^*–neighborhoods. Hence, the color-

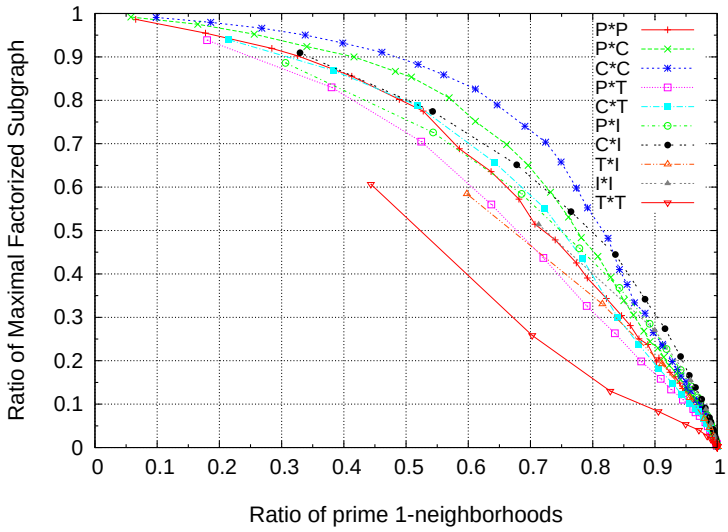


Figure 7.12: The plot shows the relative ratio of maximal factorized subgraphs in dependence of the relative ratio of prime 1-neighborhoods.

continuation fails and thus, the maximal factorized subgraphs become smaller.

A remarkable result that can be observed in Figure 7.12 is that for all graphs, except the graph $T \boxtimes T$, that have a measured ratio of prime 1-neighborhoods of $\sim 80\%$, at least $30\% - 50\%$ of the original graphs was recovered. Even if $\sim 95\%$ of used 1-neighborhoods were prime it was possible to recover $7\% - 16\%$ of the underlying graph. The latter result is promising and shows that the algorithm for the recognition of approximate products computes suitable results and factorizes large parts of the disturbed products even if a large amount of 1-neighborhoods is prime. Moreover, we have only counted *maximal* factorized subgraphs and there might be more factorized subgraphs. Therefore, one would expect that the ratio is larger if we take all factorized subgraphs into account.

Summary

Starting with the question: "How often do we find both original factors in the disturbed product depending on the percentage of perturbation?" We found that the ratio of disturbed product graphs, where both underlying were recovered, decreases very fast and that there is remarkable difference between the different product graphs.

To understand the latter observation we took the values $\overline{U}_G$, the ratio of the backbone vertices and also the structure of the graphs into account. For graphs G and H with $\overline{U}_G > \overline{U}_H$ the chance to

recover the underlying factors for H is higher than for G. Modified Algorithm 9 performs on graphs with almost coinciding values $\overline{U}_G$ that have a higher ratio of backbone vertices or that have a cycle as factor slightly better.

However, even a small percentage of perturbation leads to disturbed product graphs, where only few instances of the data set have a structure, for which it was possible to determine the original underlying factors. To understand why the ratio of disturbed product graphs, where both underlying factors were recovered, decreases so fast the relation between perturbation and the ratio of prime 1-neighborhoods was investigated. In general it was observed that only slight perturbations lead to a high ratio of prime 1-neighborhoods. A perturbation of 6% leads to more than 70% prime 1-neighborhoods in all graphs of the data set. Clearly, the more 1-neighborhoods are prime the fewer 1-neighborhoods can be used to recover the underlying factors. Taking the latter observations into account we can empirically conclude why only few instances of slightly disturbed products have a structure where it was possible to determine the original underlying factors.

In the last part of this section we investigated maximal factorized subgraphs and tried to find how large maximal factorized subgraph of disturbed product are in dependence on the ratio of disturbance, and therefore, in dependence on the ratio of prime 1-neighborhoods.

We observed that with an increasing percentage of perturbation and hence, with an increasing number of prime 1-neighborhoods, the size of maximal factorized subgraphs decreases. We found that for almost all graphs with measured ratio of prime 1-neighborhoods of $\sim 80\%$, at least $30-50\%$ of the original graphs was recovered. Even if $\sim 95\%$ of 1-neighborhoods were prime, it was possible to recover $8-16\%$ of the original underlying product graph. Moreover, only maximal factorized subgraphs were counted, it is expected that this ratio is larger if we take all factorized subgraphs into account. The latter result is promising and shows that the algorithm for the recognition of approximate products computes good results.

8

Summary and Outlook

Motivated by the fact that, in practical applications, graphs often occur as perturbed product structures, we investigated the local structure of strong product graphs and developed various new algorithms that work on a local level, i.e, by usage of suitable subgraphs, to decompose strong product graphs into their prime factors.

We realized that the term *thinness* plays a central role. The major task for the prime factor decomposition of a strong product graph is to determine its Cartesian skeleton, which is only uniquely determined in thin graphs. We observed that, although a graph can be thin, this holds not necessarily for its subproducts. To treat this problem we introduced the concepts *S1-condition* and the *backbone* $\mathbb{B}(G)$ of a graph G. These tools turned out to be essential for uniquely determining parts of the Cartesian skeleton, even if the used subproducts are not thin.

We then introduced the graph classes of *NICE*, *CHIC*, and *locally unrefined* graphs. Moreover, we investigated various local structural properties and derived polynomial-time algorithms that work on a local level for the PFD of those graphs. After all, we used these results to construct a new local, quasi-linear time algorithm that computes the PFD of all graphs.

Finally, approximate graph products were discussed. To derive an algorithm for the recognition

of approximate graph products the new local algorithm was modified. At the end, the performance of this algorithm was evaluated on a sample of approximate products. We perturbed given strong product graphs by deleting edges. We found that even a small percentage of perturbation leads to disturbed product graphs, where only few instances of the data set have a structure, for which it was possible to determine the original underlying factors. We explained this phenomena with the observation that only slight perturbations lead to a high ratio of prime 1-neighborhoods. After this, maximal factorized subgraphs were investigated. We observed that with an increasing percentage of perturbation and hence, with an increasing number of prime 1-neighborhoods the size of maximal factorized subgraphs decreases. We found that for almost all graphs with a measured ratio of prime 1-neighborhoods of $\sim 80\%$, at least $30-50\%$ of the original graphs was recovered. Even if $\sim 95\%$ of 1-neighborhoods were prime, it was possible to recover $8-16\%$ of the original underlying product graph. We concluded that the algorithm for the recognition of approximate products computes good results.

The future research should be focused on providing and developing heuristics for approximately factorizable graphs based on the new local decomposition algorithm. Moreover, how can the problems be solved even if the used subproducts are approximate products?

Furthermore, one should generalize the current problem to the factorization of directed graphs, weighted graphs or hypergraphs and ask under which conditions those product graphs have unique prime factors and how they could be computed fast. Moreover, how can approximate graph products of those graphs be recognized?

In addition, one should also treat other graph products, e.g. the Cartesian, the direct, and lexicographic product, and ask under which conditions it is possible to recognize approximate products using local working approaches of those products.

It is the current state of the art to decide whether a graph is prime or not by computation of its prime factorization. Also the new developed algorithms need non-prime subgraphs. Therefore, one should develop a graph preprocessing from which (at least) necessary conditions can be derived to decide whether a prime graph is very similar to a product graph or not, using statistical approaches, e.g. degree distributions or shortest paths distributions. Those approaches would be very important to consent or invalidate several theories, that make explicit statements about the product-like structure of graphs, in different contexts, as e.g. in theoretical biology [56].

Bibliography

[1] D. Archambault, T. Munzner, and D. Auber. TopoLayout: Multilevel graph layout by topological features. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):305–317, 2007.

[2] D. Archambault, T. Munzner, and D. Auber. Grouseflocks: Steerable exploration of graph hierarchy space. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):900–913, 2008.

[3] H. J. Bandelt, H.M. Mulder, and E. Wilkeit. Quasi-median graphs and algebras. *J. Graph Theory*, 18(7):681–703, 1994.

[4] I. Banič, R. Erveš, and J. Žerovnik. The edge fault-diameter of Cartesian graph bundles. *Eur. J. Comb.*, 30(5):1054–1061, 2009.

[5] B. Brešar. On subgraphs of Cartesian product graphs and S-primeness. *Discr. Math.*, 282:43–52, 2004.

[6] B. Brešar and S. Špacapan. Edge-connectivity of strong products of graphs. *Discuss. Math., Graph Theory*, 27(2):333–343, 2007.

[7] J. Cupal, S. Kopp, and P. F. Stadler. RNA shape space topology. *Artificial Life*, 6:3–23, 2000.

[8] R. Diestel. *Graph Theory*, volume 3 of *Graduate Texts in Mathematics*. Springer-Verlag, Heidelberg, 2005.

[9] W. Dörfler and W. Imrich. Über das starke Produkt von endlichen Graphen. *Österreih. Akad. Wiss., Mathem.-Natur. Kl., S.-B .II*, 178:247–262, 1969.

[10] A. W. M. Dress and D. S. Rumschitzki. Evolution on sequence space and tensor products of representation spaces. *Acta Appl. Math.*, 11:103–111, 1988.

[11] M. Eigen, J. McCaskill, and P. Schuster. The molecular quasispecies. *Adv. Chem. Phys.*, 75:149–263, 1989.

[12] T. Feder. Product graph representations. *J. Graph Theory*, 16:467–488, 1992.

[13] J. Feigenbaum. Product graphs: some algorithmic and combinatorial results. Technical Report STAN-CS-86-1121, Stanford University, Computer Science, 1986. PhD Thesis.

[14] J. Feigenbaum and R. A. Haddad. On factorable extensions and subgraphs of prime graphs. *SIAM J. Disc. Math*, 2:197–218, 1989.

[15] J. Feigenbaum, Hershberger J., and A. A. Schäffer. A polynomial time algorithm for finding the prime factors of Cartesian-product graphs. *Discrete Appl. Math.*, 12:123–128, 1985.

[16] J. Feigenbaum and A. A. Schäffer. Finding the prime factors of strong direct product graphs in polynomial time. *Discrete Math.*, 109:77–102, 1992.

[17] W. Fontana and P. Schuster. Continuity in Evolution: On the Nature of Transitions. *Science*, 280:1451–1455, 1998.

[18] W. Fontana and P. Schuster. Shaping Space: The Possible and the Attainable in RNA Genotype-Phenotype Mapping. *J. Theor. Biol.*, 194:491–515, 1998.

[19] S. Gavrilets and J. Gravner. Percolation on the fitness hypercube and the evolution of reproductive isolation. *J. Theor. Biol.*, 184:51–64, 1997.

[20] R.L. Graham and P.M. Winkler. On isometric embeddings of graphs. *Trans. Amer. Math. Soc.*, 288:527–536, 1985.

[21] W. Gruener, R. Giegerich, D. Strothmann, C. Reidys, J. Weber, I. L. Hofacker, P. F. Stadler, and P. Schuster. Analysis of RNA sequence structure maps by exhaustive enumeration. I. neutral networks. *Monath. Chem.*, 127:355–374, 1996.

[22] R. Hammack. On direct product cancellation of graphs. *Discrete Math.*, 309(8):2538–2543, 2009.

[23] R. Hammack and Z. Bradshaw. Minimum cycle bases of direct products of graphs with cycles. *Ars Math. Contemp.*, 2(1):101–119, 2009.

[24] R. Hammack and W. Imrich. On Cartesian skeletons of graphs. *Ars Mathematica Contemporanea*, 2(2):191–205, 2009.

[25] R.W. Hamming. Error detecting and error correcting codes. *Bell System Technical Journal*, 29(2):147–160, 1950.

[26] P. Hell, Z. Pan, T. Wong, and X. Zhu. Adaptable chromatic number of graph products. *Discrete Math.*, 309(21):6153–6159, 2009.

[27] M. Hellmuth, L. Gringmann, and P. F. Stadler. Diagonalized Cartesian products of S-prime graphs are S-prime. 2009. Submitted to *Discrete Mathematics*.

[28] M. Hellmuth, W. Imrich, W. Klöckl, and P. F. Stadler. Approximate graph products. *European Journal of Combinatorics*, 30:1119 – 1133, 2009.

[29] M. Hellmuth, W. Imrich, W. Klöckl, and P. F. Stadler. Local algorithms for the prime factorization of strong product graphs. *Math. Comput. Sci*, 2(4):653–682, 2009.

[30] M. A. Huynen, P. F. Stadler, and Walter Fontana. Smoothness within ruggedness: the role of neutrality in adaptation. *Proc. Natl. Acad. Sci., USA*, 93:397–401, 1996.

[31] W. Imrich and H. Izbicki. Associative products of graphs. *Monatshefte für Mathematik*, 80(4):277–281, 1975.

[32] W. Imrich and S Klavžar. *Product graphs*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience, New York, 2000.

[33] W. Imrich and I. Peterin. Recognizing Cartesian products in linear time. *Discrete Math.*, 307:472–482, 2007.

[34] W. Imrich and J. Žerovnik. Factoring Cartesian-product graphs. *J. Graph Theory*, 18(6), 1994.

[35] A. Kaveh and K. Koohestani. Graph products for configuration processing of space structures. *Comput. Struct.*, 86(11-12):1219–1231, 2008.

[36] A. Kaveh and R. Mirzaie. Minimal cycle basis of graph products for the force method of frame analysis. *Communications in Numerical Methods in Engineering*, 24(8):653–669, 2008.

[37] A. Kaveh and H. Rahami. An efficient method for decomposition of regular structures using graph products. *Intern. J. for Numer. Methods in Engineering*, 61(11):1797–1808, 2004.

[38] S. Klavžar, A. Lipovec, and M. Petkovšek. On subgraphs of Cartesian product graphs. *Discr. Math.*, 244:223–230, 2002.

[39] R. H. Lamprey and B. H. Barnes. A new concept of primeness in graphs. *Networks*, 11:279–284, 1981.

[40] R. H. Lamprey and B. H. Barnes. A characterization of Cartesian-quasiprime graphs. *Congressus Numerantium*, 109:117–121, 1995.

[41] R. C. Lewontin. Adaptation. *Sci. Am.*, 239:156–169, 1978.

[42] M. Lu, G. Chen, and X. Xu. On super edge-connectivity of product graphs. *Appl. Math. Comput.*, 207(2):300–306, 2009.

[43] R. McKenzie. Cardinal multiplication of structures with a reflexive relation. *Fund. Math. LXX*, pages 59–101, 1971.

[44] M. Mollard. Two characterizations of generalized hypercube. *Discrete Mathematics*, 93(1):63 – 74, 1991.

[45] H. M. Mulder. *The Interval Function of a Graph*, volume 132 of *Mathematical Centre tracts*. Amsterdam: Mathematisch Centrum, 1980.

[46] H. M. Mulder. Interval-regular graphs. *Discrete Mathematics*, 41(3):253 – 269, 1982.

[47] T. Nakayama and J. Hashimoto. On a problem of g. birkhoff. *Proceedings of the American Mathematical Society*, 1(2):141–142, 1950.

[48] C. M. Reidys and P. F. Stadler. Neutrality in fitness landscapes. *Appl. Math. & Comput.*, 117:321–350, 2001.

[49] G. Sabidussi. Graph multiplication. *Mathematische Zeitschrift*, 72:446–457, 1959.

[50] G. Sabidussi. Subdirect representations of graphs in infinite and finite sets. *Colloq. Math. Soc. Janos Bolyai*, 10:1199–1226, 1975.

[51] J. Siek, L. Lie-Quan, and A. Lumsdaine. *The Boost Graph Library: User Guide and Reference Manual*, volume 1 of *C++ In-Depth Series*. Addison-Wesley Professional, 2002.

[52] B. M. R. Stadler, P. F. Stadler, G. P. Wagner, and W. Fontana. The topology of the possible: Formal spaces underlying patterns of evolutionary change. *J. Theor. Biol.*, 213:241–274, 2001.

[53] C. Tardif. A fixed box theorem for the Cartesian product of graphs and metric spaces. *Discrete Math.*, 171(1-3):237–248, 1997.

[54] E. van Nimwegen, J. P. Crutchfield, and M. A. Huynen. Neutral evolution of mutational robustness. *Proc. Natl. Acad. Sci. USA*, 96:9716–9720, 1999.

[55] V. G. Vizing. The Cartesian product of graphs. *Vycisl. Sistemy*, 9:30–43, 1963.

[56] G. Wagner and P. F. Stadler. Quasi-independence, homology and the unity of type: A topological theory of characters. *J. Theor. Biol.*, 220:505–527, 2003.

[57] P.M. Winkler. Factoring a graph in polynomial time. *European J. Combin.*, 8:209–212, 1987.

[58] B. Zmazek and J. Žerovnik. Weak reconstruction of strong product graphs. *Discrete Math.*, 307:641–649, 2007.

List of Figures

Curriculum Vitae

Personal Information

Name:	Marc Hellmuth
Date of birth:	June 25, 1980
Place of birth:	Nordhausen, Germany

Education

since 2009	Scientific Assistant at the Max-Planck-Institute for Mathematics in the Sciences, Leipzig, Germany
2007 – 2009	Scientific Assistant at the Interdisciplinary Center for Bioinformatics, Faculty of Mathematics and Computer Science University Leipzig, Germany
2001 – 2007	Study of Economathematics (University Leipzig, Germany); Degree: Diploma
2000 – 2001	Study of Computer Science (University Leipzig, Germany)
1991 – 1999	German secondary school in Nordhausen; Degree: Diploma qualifying for university admission or matriculation

Scientific Cooperations

Prof. Dr. Wilfried Imrich & Dr. Werner Klöckl at the University of Leoben, Leoben (Austria).
Prof. Dr. Josef Leydold at the Vienna University of Economics and Business, Vienna (Austria).
Prof. Dr. Daniel Merkle at the University of Southern Denmark, IMADA, SDU Odense (Denmark)
Prof. Dr. Gerik Scheuermann & Dipl.-Inf. Christian Heine at the University Leipzig (Germany)

(Programming) Languages

C, C++, Pascal, Java, PHP, HTML, LaTeX
German, English and French

Publications

1. M. Hellmuth, D. Merkle, M. Middendorf,
   Extended Shapes for the Combinatorial Design of RNA Sequences,
   Int. J. of Computational Biology and Drug Design, **2**: 4, 371 - 384, 2009
2. M. Hellmuth, W. Imrich, W. Klöckl, P.F. Stadler,
   Local algorithms for the prime factorization of strong product graphs,
   Math. Comput. Sci, 2009, **2**: 4, 653-682, 2009
3. P.J. Ostermeier, M. Hellmuth, K. Klemm, J. Leydold, P.F. Stadler,
   A Note on Quasi-Robust Cycle Bases,
   Ars Math. Contemporanea, **2**: 2, 231-240, 2009
4. T. Ingalls, G. Martius, M. Hellmuth, M. Marz, S.J. Prohaska,
   Converting DNA to Music: ComposAlign,
   German Conference on Bioinformatics 2009:Lecture Notes in Informatics, 93-103, 2009
5. M. Hellmuth, D. Merkle, M. Middendorf,
   On the Design of RNA Sequences for Realizing Extended Shapes,
   Bioinformatics, Systems Biology and Intelligent Computing, IJCBS '09, 167-173, 2009
6. T. Biyikoglu, M. Hellmuth, J. Leydold,
   Largest Eigenvalues of the Discrete p-Laplacian of Trees with Degree Sequences,
   Elec. J. Lin. Alg, **18**: 202-210, 2009.
7. M. Hellmuth, W. Imrich, W. Klc̈kl, P.F. Stadler,
   Approximate graph products,

European Journal of Combinatorics, **30**: 5, Part Special Issue on Metric Graph Theory, 1119-1133, 2009.

8. T. Biyikoglu, M. Hellmuth, J. Leydold,
   Largest Laplacian Eigenvalue and Degree Sequences of Trees,
   Research Report Series / Department of Statistics and Mathematics, **64**: 2008.
9. M. Hellmuth, L. Gringmann, P.F. Stadler
   Diagonalized Cartesian Products of S-prime graphs are S-prime,
   (submitted to Discrete Mathematics, 11/2009)

Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialen oder erbrachten Dienstleistungen als solche gekennzeichnet.

Marc Hellmuth Leipzig, den 04. März 2010