On the Decomposition and Reconstruction of Discrete Structures: Characterization, Approximation and Computation

HABILITATION

Extended Abstract

Faculty 6 - Natural Sciences and Technology I
Mathematics and Computer Science
Saarland University

submitted by: JProf. Dr. Marc Hellmuth
September 28, 2015

Contents

1 List of Peer-Reviewed Journal Papers

We give here the full references of the published journal papers on which this cumulative habilitation thesis is based. Corresponding authors are marked with *.

1. *[HWL$^+$15]*
   M. Hellmuth*, N. Wiesecke, M. Lechner, H.P. Lenhof, M. Middendorf, and P. F. Stadler. Phylogenomics with paralogs. *Proceedings of the National Academy of Sciences*, 112(7):2058–2063, 2015
2. *[HM15]*
   M. Hellmuth* and T. Marc. On the Cartesian skeleton and the factorization of the strong product of digraphs. *J. Theor. Comp. Sci.*, 565(0):16–29, 2015
3. *[HIK15]*
   M. Hellmuth*, W. Imrich, and T. Kupka. Fast recognition of partial star products and quasi Cartesian products. *Ars Math. Cont.*, 9(2):233 – 252, 2015
4. *[HMOS15]*
   M. Hellmuth, T. Marc, L Ostermeier*, and P. F. Stadler. The relaxed square property. *Australas. J. Combin.*, 62(3):240–270, 2015
5. *[BCF$^+$15]*
   S. Berkemer, R. Chaves, A. Fritz, M. Hellmuth*, M. Hernandez-Rosales, and P. F. Stadler. Spiders can be recognized by counting their legs. *Math. Comput. Sci*, pages 1–5, 2015
6. *[HON14]*
   M. Hellmuth*, L. Ostermeier, and M. Noll. Strong products of hypergraphs: Unique prime factorization theorems and algorithms. *Discr. Appl. Math.*, 171:60–71, 2014
7. *[HRHWS14]*
   M. Hernandez-Rosales*, M. Hellmuth, N. Wieseke, and P. F. Stadler. Simulation of gene family histories. *BMC Bioinformatics*, 15(Suppl 3):A8, 2014
8. *[HOS14]*
   M. Hellmuth, L. Ostermeier*, and P. F. Stadler. Unique square property, equitable partitions, and product-like graphs. *Discr. Math.*, 320(0):92 – 103, 2014

9. *[HHRH+ 13]*
   M. Hellmuth, M. Hernandez-Rosales, K. T. Huber*, V. Moulton, P. F. Stadler, and N. Wieseke. Orthology relations, symbolic ultrametrics, and cographs. *J. Math. Biology*, 66(1-2):399–420, 2013
10. *[HIK13]*
   M. Hellmuth*, W. Imrich, and T. Kupka. Partial star products: A local covering approach for the recognition of approximate Cartesian product graphs. *Math. Comput. Sci*, 7(3):255–273, 2013
11. *[Hel13]*
   M. Hellmuth*. On the complexity of recognizing S-composite and S-prime graphs. *Discr. Appl. Math.*, 161(7-8):1006 – 1013, 2013
12. *[HOS12a]*
   M. Hellmuth*, L. Ostermeier, and P. F. Stadler. Diagonalized Cartesian products of S-prime graphs are S-prime. *Discr. Math.*, 312(1):74 – 80, 2012. Algebraic Graph Theory - A Volume Dedicated to Gert Sabidussi on the Occasion of His 80th Birthday
13. *[HOS12b]*
   M. Hellmuth, L. Ostermeier, and P. F. Stadler*. A survey on hypergraph products. *Math. Comput. Sci*, 6:1–32, 2012
14. *[OHS12]*
   L. Ostermeier*, M. Hellmuth, and P. F. Stadler. The Cartesian product of hypergraphs. *Journal of Graph Theory*, 70(2):180–196, 2012
15. *[HRHW+ 12]*
   M. Hernandez-Rosales*, M. Hellmuth, N. Wieseke, K. T. Huber, V. Moulton, and P. F. Stadler. From event-labeled gene trees to species trees. *BMC Bioinformatics*, 13(Suppl 19):S6, 2012
16. *[Hel11]*
   M. Hellmuth*. A local prime factor decomposition algorithm. *Discr. Math.*, 311(12):944–965, 2011
17. *[JHH+ 10]*
   S. Jänicke*, C. Heine, M. Hellmuth, P. F. Stadler, and G. Scheuermann. Visualization of graph products. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1082–1089, 2010

2 Decomposing and Reconstructing Discrete Structures

Introduction

A *relation* is one the most basic objects in science. Mathematical logic is expressed as a set of relations. The concept of function is defined as a special kind of binary relation. Grammar and language, in general, are a more or less formal and stringent system which relates signs and combinations thereof. Biologists, for instance, are interested in relations between genes or species and chemists want to know about relations between molecules. Thus, inferring, characterizing and computing relations between discrete objects is one of the most fundamental scientific tasks.

Binary relations $E \subseteq V \times V$ are equivalently represented as *graphs* (V, E). Every N-ary relation has a representation as *(ordered) set system* called *hypergraph* [Ber89]. In particular, hypergraphs generalize the notion of simple N-ary relations as they can describe *sets* of relations and might contain additional information as colors, labels or weights to emphasize the importance of certain elements contained in V or E, for instance. Graphs and hypergraphs are essential tools to establish models in life science [KHT09, DR88, GGS$^+$96, EMS89], computational engineering [YP96, KR04, KM08, KK08, AMA07, AMA08, KA15] or in studies on social networks [OR02, TP11, AHS10]. It is therefore of central interest to understand the underlying structure of such discrete systems, both from the theoretical as well as the practical point of view.

On the one hand, this contribution is concerned with the decomposition of graphs or hypergraphs to smaller elementary structures or “building blocks”. The knowledge of such building blocks leads to a simplified representation of the investigated structure. In particular, such building blocks help to understand the global underlying structure in a more systematic way and thus, reduce the structural complexity of the investigated discrete systems. On the other hand, we will focus on the inference and characterization of such small building blocks and the reconstruction of larger discrete objects from their elementary local substructures.

Part I of this thesis is concerned with products of set systems. Products are a common way in mathematics of constructing larger objects from smaller building blocks which are called *factors*. Product of graphs and hypergraphs have been investigated since the 1960’s. For graphs, hypergraphs, and related set systems not only one, but a few types of products can be defined. Different types of products are important for different types of models and related questions one might ask [DR88, GGS$^+$96, KR04, KM08, KK08, KA15]. These products often (but not always) satisfy a unique prime factor property. It is of key interest then to understand the structure of a large object by decomposing it into its prime factors. We will be concerned with prime factor theorems, to explore their limitations, and to devise efficient algorithms for set systems, as directed or undirected graphs and hypergraphs. Prime factorizations are closely related to so-called product relations on the edge sets (of graphs and hypergraphs) that identify layers or fibers corresponding to copies of the factors.

Part II of this contribution is concerned with product-like structures, of which we distinguish two types. *Approximate products* are obtained from product structures by means of small perturbations and relate to the practical question of recognizing product structures from noisy data. *Relaxed products*, in contrast, are defined as set systems that admit non-trivial “edge colorings” that generalize the properties of product relations in a way that still defines fiber structures. While approximate products often can be recognized only by heuristic algorithms, it is possible to devise exact efficient algorithms for relaxed product structures.

Part III focuses on the characterization of relations between biological objects such as genes and, in particular, the inference of small building blocks from those relations. Based on these results we will focus on the design of efficient algorithms and methods that allow to use the inferred building blocks for the reconstruction of larger structures, e.g. so-called supertrees that represent the evolutionary history of genes or species.

The thesis is outlined as follows. For each *Part I, II* and *III* we start with a section *Preliminaries*, in which we briefly summarize known results and the state of the art. In the respective sections *Important Basic Concepts*, we will provide necessary key definitions and ideas that we need in the subsequent parts of this thesis. We highlight established results that are part of this thesis in the sections *Outline of Central Results* and finish this chapter with a short *Summary*. More elaborated discussions on each of the established results can then be found in Chapter 3. *(not part of this extended abstract) Review of Own Contributions*.

In order to keep this work simple we omit many of the (quite technical) definitions, but refer the interested reader to [HIK11] for definitions concerning graph products, to [HOS12b] for hypergraph (product) definitions and to [SS03] for definitions of evolutionary trees and their corresponding building blocks. Detailed discussion and definitions can additionally be found in the respective attached publications. Nevertheless, since graphs and hypergraphs are the relevant objects in this contribution, we give here the formal definition.

Definition. A graph G is a tuple (V, E) with a non-empty vertex set $V(G) = V$ and edge set $E(G) = E$ that is either (i) a subset of $\binom{V}{2}$ or (ii) a subset of $V \times V$. A hypergraph H is a tuple (V, E) with a non-empty vertex set V and edge set E where each edge $e \in E$ is either (i) a subset of V or (ii) of the form $(t(e), h(e))$ with $t(e), h(e) \subseteq V$. In Case (i) the (hyper)graph is called *undirected* and in Case (ii) *directed*. For simplicity, we will often refer to both graphs and hypergraphs as *set systems* and use the shortcut *di-(hyper)graph* for *directed (hyper)graphs*.

Part I: Exact Product Structures

Preliminaries

Graph products are natural structures in discrete mathematics [IK00, HIK11] that arise in a variety of different contexts, from computer science [AMA07, JHH$^+$10, SSDN02, MS90] and computational engineering [KK08, KR04, KM08] to theoretical biology [FS98a, FS98b, CKS00, SSWF01, WS03].

A prominent example of application is given by Wagner and Stadler [SSWF01, WS03, SS04]. The authors provided a concept concerning the topological theory of the relationships between genotypes and phenotypes and, in particular, the differentiation of so-called “characters” (traits or *Merkmale*). 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 observable of course often are instantiations of characters. The important biological distinction is *whether* such measurable attributes (or combinations thereof) form a “factor” in the so-called phenotypespace 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 [Lew78]. 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. Given the information of the underlying so-called genotype space one can infer edges between the members of $\mathbb{X}$ resulting in the graph $G(\mathbb{X})$, called phenotype space. It was shown in [WS03, SS04] that the phenotype space $G(\mathbb{X})$ has (at least locally) a product structure and factors of induced subgraphs of $G(\mathbb{X})$ correspond to the biological meaningful traits, see Figure 2.1 for an illustrative example.

Graph products are defined as graphs whose vertex set is the Cartesian set product of the vertex sets of its factors. Requiring that the edge set of the product is locally constructed from the edge set of the factors, it was shown already in 1975 that there are 256 possible products [II75], of which only 6 are commutative, associative and have a unit. Four products are known as the *standard graph products* [IK00, HIK11]: the *Cartesian* product $\Box$, the *direct* product $\times$, the *strong* product $\boxtimes$, and the *lexicographic* product $\circ$. These standard graph products are the only ones that preserve the salient structure of their factors and behave in an algebraically reasonable way, i.e. among others, the product of simple graphs results in a simple graph and at least one of the projections of a product onto its factors is a so-called weak homomorphism (edges are mapped to edges or to vertices).

We continue to define the four standard products. Let G_1 and G_2 be two arbitrary graphs. In all of these products the vertex set $V(G_1 \circledast G_2)$ is defined as the Cartesian set product $V(G_1) \times V(G_2)$, $\circledast \in \{\Box,\ \times,\ \boxtimes,\ \circ\}$. Two vertices with coordinates (x_1, x_2), (y_1, y_2) are adjacent in the

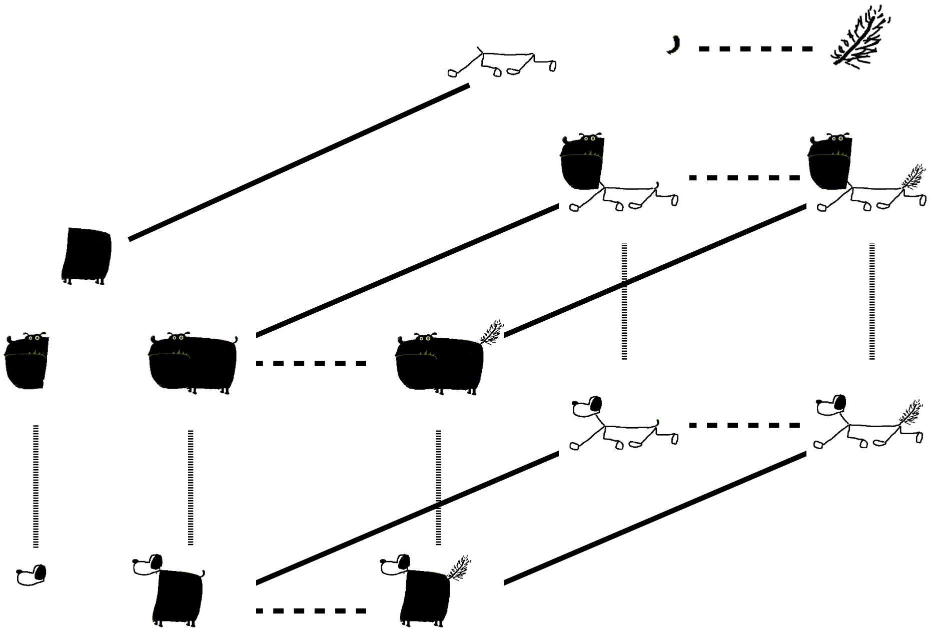

Figure 2.1: Shown is a notional phenotype space $G(\mathbb{X})$ for eight different “animals” contained in $\mathbb{X}$. The prime factorization of $G(\mathbb{X})$ w.r.t. the Cartesian product leads to three factors. The copies of these factors are highlighted in $G(\mathbb{X})$ by different line-styles. The factors represent the independent characters “face”, “body” and “tail”, respectively. The idea for this figure is taken from [SS04], where a similar picture with worm-like “animals” can be found.

Cartesian product $G_1 \Box G_2$, i.e. linked by an edge, if one of the following conditions is satisfied:

(C1) $(x_1, y_1) \in E(G_1)$ and $x_2 = y_2$; or (C2) $(x_2, y_2) \in E(G_2)$ and $x_1 = y_1$.

Two vertices (x_1, x_2), (y_1, y_2) are adjacent in the *direct product* $G_1 \times G_2$, if the following condition holds:

(D1) $(x_1, y_1) \in E(G_1)$ and $(x_2, y_2) \in E(G_2)$.

Two vertices with coordinates (x_1, x_2), (y_1, y_2) are adjacent in the *strong product* $G_1 \boxtimes G_2$, if one of the Conditions (C1), (C2) or (D1) is satisfied. Consequently, the edges of a strong product that satisfy (C1) or (C2) are called *Cartesian* edges, the others are the *non-Cartesian* edges. In the *lexicographic product* $G_1 \circ G_2$ two vertices (x_1, x_2), (y_1, y_2) are adjacent if $(x_1, y_1) \in E(G_1)$ or they satisfy (C2). For an illustrative example see Fig. 2.2.

The structural features of graph products have been extensively studied over the last decades. The unit element for the Cartesian, the strong and the lexicographic product is the single vertex graph K_1, for the direct product the unit is the single vertex graph with a loop $\mathcal{L}K_1$. A graph G is

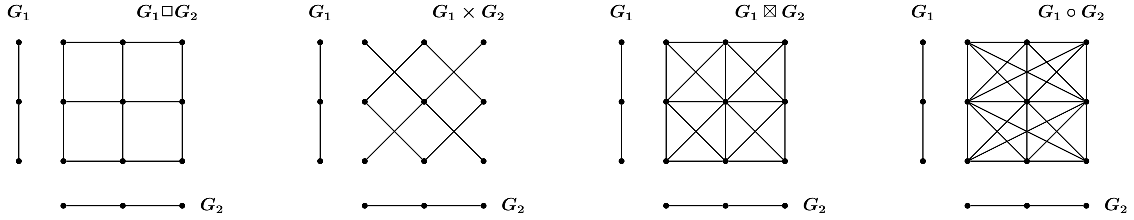

Figure 2.2: The four standard products: the *Cartesian* product $\Box$, the *direct* product $\times$, the *strong* product $\boxtimes$, and the *lexicographic* product $\circ$.

prime with respect to a product $\circledast$ if it cannot be written as the product of two nontrivial graphs, i.e., the identity $G = G_1 \circledast G_2$ implies that $G_1 \cong U$ or $G_2 \cong U$, where U is the unit w.r.t. $\circledast$. The representation of a graph G as a product $G_1 \circledast G_2 \circledast \cdots \circledast G_k$ of prime graphs is called a *prime factor decomposition (PFD)* of G.

In many cases, there is a unique PFD (up to isomorphism and the order of the factors). Sabidussi's seminal paper [Sab59] established this for the Cartesian product of finite undirected graphs. Analogous results hold for infinite graphs [Mil70, Imr71]. Subsequent research concentrated on structural results for the Cartesian, strong, direct, and lexicographic product, establishing certain connectedness properties, as well as sufficient and often necessary conditions for uniqueness of the PFD [HIK11].

Efficient algorithms for many graph factorization problems are known, starting with the construction of a polynomial time solution for the Cartesian product of undirected graphs [FHS85]. Feigenbaum and Schäffer [FS92, FS86] also solved the problem for the strong product (polynomial) and the lexicographic product (complexity of the isomorphism problem for graphs), both for undirected graphs. The first polynomial algorithm for the direct product is due to Imrich [Imr98]. Improved algorithms for the Cartesian product of undirected graphs have been published by several authors, e.g. Winkler [Win87], Feder [Fed92] or Aurenhammer et al. [AHI92], culminating in the $O(|E| + |V|)$ space and time algorithm of Imrich and Peterin [IP07]. Improvements for the strong product of undirected graphs were achieved by Hammack and Imrich [HI09] and Hellmuth [Hel11]. Several monographs cover the topic in substantial detail and serve as standard references [IK00, IKD08, HIK11].

Products of undirected graphs are much better understood than products of *directed* graphs. R. McKenzie [McK71] proved that directed graphs have unique PFDs w.r.t. the direct and the strong product requiring strong conditions on connectedness. For some other restricted graph classes (requiring again connectivity and certain so-called thinness conditions) Imrich and Klöckl [IK07, IK10] were able to establish unique PFD results and give a first polynomial-time algorithm for the direct product of directed graphs. For the strong product of directed graphs only a weak result is known [AB12]. The authors give a polynomial time algorithm that decomposes the strong product of two connected directed triangle-free factors, if at least one factor has a vertex of degree one. The PFD of directed graphs w.r.t. the Cartesian product is unique and can be computed in linear time [Fei86a, CTL13]. The results for the lexicographic product, however, include only factorization theorems that show the relation to factorizations of so-called transpositions of other graphs [HIK11, Theorem 32.8]. In general, determining whether a connected graph is a lexicographic product is just as difficult as the graph isomorphism problem [FS86, HIK11].

Hypergraphs are a natural generalization of graphs in which “edges” may consist of more than two vertices. Products of hypergraphs have interesting application e.g. in structural mechanics [KA15]. Here, the author derived so-called adjacency functions of hypergraph products, as a generalization of “static” graph products. These functions are used to model so-called “dynamic” graph products. From application point of view, the dynamic graph products can be used in modeling and configuration processing of dynamic regular structures.

Requirements that edges in a product depend on the adjacencies in the factors, can be easily generalized to hypergraphs. For hypergraphs, however, this leads only to a restriction on edges but does not provide a complete recipe for the construction of the edge set of the product. Hence, in contrast to graphs a surprising diversity of different definitions of products has been discussed in the literature. Most constructions can be seen as different generalizations of the four standard graph products.

The literature on hypergraph products is mostly concerned with the propagation of invariants. Comparably little is known about prime factorization theorems, algorithmic considerations are large lacking. For the so-called costrong product and the square product of hypergraphs, unique PFD results are known [GI71, Dör82]. As in the case of graphs, results for the lexicographic product, include only factorization theorems for very restricted hypergraph classes [GI71]. A first polynomial time algorithm to compute the PFD of connected undirected hypergraphs w.r.t. Cartesian product is established by Bretto and Silvestre [BS10].

Important Basic Concepts

We summarize here important concepts that we need in the following discussion and summary of the results. In this contribution, we focus mainly on the Cartesian and strong product of (un)directed graphs and hypergraphs.

Product Relation σ. Given a (hyper)graph $G = (V, E)$. The main idea for (modern) proofs of PFD theorems and algorithmic approaches for some product $\circledast \in \{\Box, \boxtimes\}$ rely on characterizations and computation of the so-called (finest) product relation $\sigma_\circledast$ defined on the edge set. The key property of $\sigma_\circledast$ is that connected components of the subgraphs induced by the classes of $\sigma_\circledast$ are precisely the copies of the factors, called *layers* or *fibers* [Sab59, HIK11]. In other words, $(e, f) \in \sigma_\circledast$ if and only if $e, f \in E$ belong to the copy of the same prime factor w.r.t. $\circledast$. Hence, it is desired to compute a product relation $\sigma_\circledast$ to determine the prime factors w.r.t. $\circledast$ and to prove when such a finest product relation $\sigma_\circledast$ is uniquely determined.

The Relation δ and Square Property. A classical result of the theory of Cartesian graph products establishes that $\sigma_\Box$ can be derived from another relation of the edge set: $\sigma_\Box = \mathcal{C}(\delta)$, where $\mathcal{C}(\delta)$ denotes the convex closure of the so-called *square property relation* δ. Two edges e and f are in relation δ if one of the following holds: (i) $e = f$, (ii) e and f are adjacent and there is no (unique) chordless square containing both e and f, or (iii) e and f are opposite edges of a chordless square. Clearly, the relation δ is reflexive and symmetric but not necessarily transitive. However, the transitive closure δ^* is an equivalence relation on E that contains δ. We say that any equivalence relation R containing δ has the *square property*. In fact, the first polynomial time algorithm to compute the Cartesian prime factorization of a graph explicitly constructs $\sigma_\Box$ starting from the relation δ [FJS85].

The Cartesian Skeleton $\mathbb{S}$. A further important concept to compute $\sigma_\boxtimes$ is the definition of the Cartesian skeleton. Note, $G = H \boxtimes K$ contains as subgraph the Cartesian product $H \Box K$. The Cartesian skeleton $\mathbb{S}(G)$ of a (hyper)graph $G = (V, E)$ has as vertex set V and contains "sufficiently many" of the Cartesian edges of G. In particular, for a connected graph $G = H \boxtimes K$ the Cartesian skeleton $\mathbb{S}(G)$ is desired to be a connected spanning subgraph of G that should fulfill $\mathbb{S}(G) = \mathbb{S}(H \boxtimes K) = \mathbb{S}(H) \Box \mathbb{S}(K) \subseteq H \Box K$. In other words, the Cartesian skeleton $\mathbb{S}(G)$ of the strong product graph G is the Cartesian product of the Cartesian skeletons $\mathbb{S}(H)$ and $\mathbb{S}(K)$ of its factors H, K. The prime factors of G w.r.t. the strong product are then constructed by utilizing the information of the *Cartesian* prime factors of $\mathbb{S}(G)$.

Thinness. In general, the Cartesian skeleton is only uniquely determined whenever the input (hyper)graph is *thin*. For undirected graphs, thinness is defined in terms of the (closed) neighborhood $N[v] = N(v) \cup \{v\}$ of a vertex and the associated relation $S = \{(x, y) \mid N[x] = N[y]\}$. Thinness means that S is trivial. In other words, an undirected graph G is thin if each of its vertex $v \in V(G)$ is uniquely distinguished from other vertices by its neighborhood $N[v]$. The quotient graph G/S is thin and the Cartesian skeleton is uniquely determined in G/S. For non-thin graphs the $\boxtimes$-PFD is initially constructed for the thin quotient graph G/S of G, and then lifted back to the input G, which results in a unique $\boxtimes$-PFD for G (with a possibly non-unique coordinatization).
For directed graphs, one defines thinness in terms of *in*- and *out-neighborhoods*: A directed graph G is thin, if for any distinct vertices $v, w \in V(G)$ the out-neighborhoods are distinct $N^+[v] \neq N^+[w]$ or the in-neighborhoods are distinct $N^-[v] \neq N^-[w]$. In other words, a di-graph is thin, if any two distinct vertices can be distinguished by their in- or out-neighborhoods. As a consequence, a di-graph is thin, if for the respective equivalence relation S (with equivalence classes comprising the vertices with identical in- and out-neighborhoods) all classes consist of a single vertex.
Similarly, the term thinness is defined for undirected and directed hypergraphs.

Outline of Central Results

In what follows, we briefly summarize established results that are part of this thesis concerning exact product structures. A more detailed discussion of these results can be found in Chapter 3. *(not part of this extended abstract)*.

There is a strong interest for automatic graph-product visualization tools since drawings of those products can effectively communicate the quality of results concerning products or product-like structures by emphasizing the regularity of graph structures through regularity of the layout. To this end, we developed in [JHH^{+}10] efficient algorithms for drawing product graphs including the four standard products. The designed algorithm were integrated in the classical graph layout program `TopoLayout` [AMA07, AMA08]. These algorithms provide the *first* collection of methods to produce high-quality aesthetic pictures of product graphs.

In [Hel11], I established one of the currently fastest algorithms to determine the prime factors of finite undirected graphs w.r.t. to the strong product. The approach is working on a local level in the sense that at a given time, only neighborhoods, edge-neighborhoods, or enlarged edge neighborhoods are factored instead of the entire global graph. In other words, the given graph is covered by factorizable small patches which are then stepwisely extended to regions with a product structures. This method opened a complete new way of finding prime factors of graphs and inspired new ideas and the development of algorithms for the recognition of other products or product-like structures, as we shall discuss in the following section.

In [Hel13] and [HOS12a] we investigated the structure and the complexity of determining so-called S-prime graphs. S-prime graphs have the special property that whenever they are a subgraph of a non-trivial Cartesian product graph, then they must be a subgraph of one of the factors. In particular, those graphs are heavily used in important subroutines for the local $\boxtimes$-PFD algorithm [Hel11] and are crucial to prove correctness results for this PFD algorithm. We provided a new infinite class of S-prime graphs. Surprisingly, this class includes all n-dimensional hypercubes ($n > 1$) where two diametrical vertices are linked by an edge. Furthermore, I showed that determining if a graph is *not* S-prime is NP-complete and thus, determining whether a graph is S-prime is CoNP-complete. A couple of corollaries where derived, including the problem of determining whether a graph is a nontrivial subgraph of a Hamming graph with two factors, which is shown to be NP-complete, as well.

Although many algorithms for the standard graph products have been devised for undirected graphs, no known methods to compute the PFD of directed graphs with respect to the direct and the strong product in general had been established. Such algorithms are quite relevant for solving problems e.g. in biology [WS03]. In [HM15], we focused on the algorithmic aspects for determining the PFD of directed graphs with respect to the strong product and gave a complete new construction of the underlying Cartesian skeleton of directed graphs, resulting in the *first* polynomial-time algorithm for the computation of the PFD w.r.t. the strong product of arbitrary connected di-graphs. Note, since the strong product is just a special case of the direct product, the established results lay the foundation for further research on direct products.

For a first step to generalize PFD properties and algorithms to hypergraphs, we studied the surprising diversity of different definitions of products that has been discussed in the literature, so-far and provided a comprehensive survey on hypergraph products in [HOS12b].

In [OHS12] we showed that every (directed) hypergraph which might have infinitely many vertices has a unique prime factorization w.r.t. the (weak) Cartesian product, even if there are infinitely many factors. To this end, we generalized the so-called square property for graphs to the grid property in hypergraphs that has to be satisfied among edges that characterize the copies of the factors. This relation enabled us to prove the uniqueness of the PFD, even for infinite hypergraphs, which is basically one of the most general result one can derive for the Cartesian product of hypergraphs.

The normal (hypergraph) product is a natural generalization of the strong graph product. In [HON14] we showed that the PFD of hypergraphs w.r.t. to the normal product is unique in the class of connected finite and thin hypergraphs and gave the first polynomial time algorithm to determine this PFD. To this end, we generalized the Cartesian skeleton for graphs to hypergraphs and provided simple and transparent polynomial-time algorithm for their computation.

Part II: Product-like Structures

Preliminaries

The practical use of exact products is very limited, however, because real world graphs rarely have a perfect product structure. The product structure itself is very fragile in the sense that a single edge or vertex edit yields a perturbed product that is, in particular, prime [Fei86b, ZŽ07]. Nonetheless, the recognition of product-like structures is of considerable practical interest e.g. in computer science [AMA07, JHH$^+$10, SSDN02, MS90], computational engineering [KK08, KR04, KM08] or theoretical biology [FS98a, FS98b, CKS00, SSWF01, WS03].

We will distinguish two types: *Approximate products* are obtained from product structures by means of small perturbations and relate to the practical question of recognizing product structures from noisy data. *Relaxed products*, in contrast, are defined as set systems that admit non-trivial “edge colorings” that generalize the properties of product relations in a way that meaningful “copies” of the factors can still be defined.

APPROXIMATE PRODUCTS

A graph G, or a set system in general, is an approximate product if there is a set system G' that (i) has a non-trivial prime factor decomposition w.r.t. to the product of interest, and (ii) G is similar to G'. The latter condition can be expressed e.g. in terms of editing distance between G

and G' that may allow G and G' to differ in both their edge set and their vertex set. This kind of structures was introduced in [HIKS09] for graphs.

The recognition of Cartesian and strong product graphs from which one or several vertices have been deleted was first considered by Imrich, Žerovnik, and Zmazek showing that unique reconstruction is always possible for single vertex deletions [IŽ96, ZŽ07], while k-vertex deletions can be corrected provided the products have at least $k+1$ factors with a least $k+1$ vertices each [IZv01, IŽZ03]. A polynomial-time algorithm for the reconstruction of one-vertex-deleted Cartesian product graphs is given in [Hv99]. On the other hand, even in the simple case of the Cartesian product, Feigenbaum and Haddad [FH89] showed that minimal factorizable extensions and maximal factorizable subgraphs are not unique and that finding them is NP-hard in general.

However, despite its central importance for application only very rare results of general methods to recover perturbed product structures have been established, so-far.

RELAXED PRODUCTS

Complementary to our work on approximate products, which dealt with local editing-like perturbations, we explored in our further work the consequences of systematically relaxing the conditions on the product relation that characterizes product graphs. Remind, the Cartesian product relation $\sigma_\Box$ can be computed by means of the relation δ, that is, the convex closure $\mathcal{C}(\delta)$ is $\sigma_\Box$. Moreover, any equivalence relation R with $\delta \subseteq R$ satisfies the *square property*, i.e., any two incident edges that are contained in different equivalence classes span exactly one square in G, this square is chordless and the opposite edges of any square in G are in the same equivalence class. Not insisting on the conditions as *unique* or *chordless* squares leads to relaxations of such a relation, and eventually, to relaxed product structures.

In fact, a slight modification of the relation δ turns out to play a fundamental role for the characterization of so-called graph bundles [ZŽ02] and forms the basis of efficient algorithms to recognize Cartesian graph bundles [IPv97, Ž00, Zv02, ZŽ02] that can be considered as Moebius strip like graphs. Graph bundles [PSTV83], the combinatorial analog of the topological notion of a fiber bundle [Hus93], are a common generalization of both Cartesian products [HIK11] and covering graphs [AFS91]. To determine graph bundles one searches for relations that satisfy the *unique square property*: An equivalence relation has the unique square property if any two adjacent edges e and f from distinct equivalence classes span a unique chordless square with opposite edges in the same equivalence class. In contrast to the square property, which states that adjacent edges of different classes span *exactly* one chordless square, for a relation R that has the unique square property adjacent edges of different classes may span more than one chordless square. In the latter case one and only one of these squares has opposite edges in the same equivalence class. Any such relation with unique square property that is in addition weakly 2-convex yields the structural properties of a graph bundle [ZŽ02]. Moreover, every Cartesian graph bundle over a triangle-free simple base can be characterized by the transitive closure δ^* of

δ, which satisfies the unique square property [IPv97].

However, explicit relaxations of product relations have not been considered so-far. In this contribution, we will establish general results on such relaxed product structure.

Important Basic Concepts

The used concepts are essentially the same as in the letter section "*Part I: Exact Product Structures*" and we refer, therefore, to this section.

Outline of Central Results

In what follows, we briefly summarize established results that are part of this thesis concerning product-like structures. A more detailed discussion of these results can be found in Chapter 3. *(not part of this extended abstract)*.

In [Hel11], we designed the first heuristic method for approximate strong products of undirected graphs. This method covers a given graph by factorizable small patches and attempt to stepwisely extend regions with product structures. Hence, local regions of the strong product that are disturbed can be circumvented and the remaining factored part of the graph yields then valuable important information of the underlying product structure in terms of a computed edge-coloring that reflects the copies of the underlying factors. The "putative" factors can then easily be derived. This method pioneered the way for the recognition of approximate (strong) products and, in fact, many findings were re-used to establish methods for Cartesian product-like structure [HIK13, HIK15].

In [HIK13] and [HIK15], we followed again a local strategy to design algorithms that can deal with perturbed Cartesian product structures. It should be mentioned that the problem of finding factorizable patches in Cartesian products is more challenging than in strong product graphs. In order to develop such a heuristic, we introduced the new notion of a partial star product (PSP), which is based on the square property relation δ. These PSP's are the local subgraphs that we need in order to infer local factorization. Importantly, this heuristic does not only compute an edge coloring that reflects the copies of the factors, but an embedding of the graph into a Cartesian product. This allows, to understand the underlying full product structure in detail, since coordinates on perturbed product structures can be inferred. While we established the necessary proofs and an basic algorithm in [HIK13], we improved and generalized these results in [HIK15] to an $O(|E|)$-time algorithm for graphs with bounded maximum degree and showed, how this algorithm can be parallelized.

Equivalence relations on the edge set of a graph that satisfy restrictive conditions on chordless squares play a crucial role in the theory of Cartesian graph products and so-called twisted products or bundles. We considered in [HOS14] systems of axioms that generalize the square property, called unique-square-property (USP) relation. We showed that USP-relations, induce in a natural way equitable partitions on the vertex set of G. As it turned out, the quotient graphs w.r.t. these equitable partitions exhibit a natural, rich product structure even when G itself is prime. In other words, graphs with non-trivial USP-relation can in a certain manner also be considered as approximate products, although their structural features are much more conserved then in arbitrary disturbed products.

In [HMOS15], we considered a further generalization of the square property, called *relaxed square property (RSP)* by omitting the requirement for unique and chordless squares. This leads to the complete new concept of *relaxed product structures*. We showed that for each of the products $\circledast \in \{\Box, \boxtimes, \times\}$ a relation R with relaxed square property is known for the product, if and only if such a relation R_i is known for each of its factors G_i. Moreover, we provided a polynomial-time algorithm to determine for a large class of graphs, a finest relaxed product relation. We showed that the class of graphs with non-trivial RSP-relations in particular includes so-called graph bundles. Furthermore, we observed the surprising relation between RSP-relations and so-called covering graph constructions.

Part III: Phylogenetic Building Blocks and Reconstructing Evolutionary Histories

Preliminaries

Phylogenetic Reconstructions are concerned with the study of the evolutionary history of groups of systematic biological units, e.g. genes or species. The objective is the assembling of so-called phylogenetic trees or networks that represent a hypothesis about the evolutionary ancestry of a set of genes, species or other taxa, see Figure 2.3 for an example.

Genes are passed from generation to generation to the offspring. Some of those genes are frequently duplicated, mutate or get lost - a mechanism that also ensures that new species can evolve. The central problem is as follows: To determine the entire evolutionary history of the species, one can only use genomic data of living species or, in some rare exceptions, of extinct species, such as the Neanderthals [P^{+}14]. The building blocks we use here are so-called triples, that give information of the local topology of the tree we want to reconstruct. However, determining building blocks to infer and reconstruct the evolutionary history is difficult, as in practice they must be inferred through several working stages starting from the given genetic material.

Crucial for the evolutionary reconstruction of species history is the knowledge of the relationship between the respective genes. Genes that share a common origin (*homologs*) are divided into three classes, namely *orthologs*, *paralogs* and *xenologs* [Fit00]. Two homologous genes are *orthologous* if at their most recent point of origin the ancestral gene is transmitted to two daughter lineages; a *speciation* event happened. They are *paralogous* if the ancestor gene at their

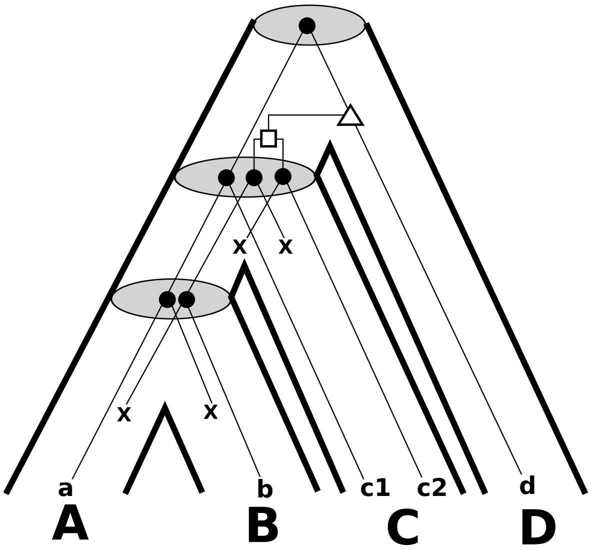

Figure 2.3: Shown is a gene tree T on the gene set $\mathcal{G} = \{a, b, c1, c2, d\}$ evolving along the species tree S (the tube-like tree) on the species $\mathcal{S} = \{A, B, C, D\}$. In this example, the gene tree T is reconciled with (or equivalently, embedded into) the species tree. The events on the gene tree are HGT-event ($\triangle$), duplication event ($\square$) and speciation event ($\bullet$). The inner vertices of the species tree represent always speciation events. Certain gene losses (**x**) are also reflected. In practice, however, only the information of observable genes and species, that is in this case, only the information of the sets $\mathcal{G}$ and $\mathcal{S}$ can be used in order to infer the gene tree (without losses), the species tree and the reconciliation between the two. The original figure appeared in [HWL$^+$15, Supplemental Information].

most recent point of origin was duplicated within a single ancestral genome; a *duplication* event happened. Horizontal gene transfer (HGT) refers to the transfer of genes between organisms in a manner other than traditional reproduction and across different species; if such an event happened at the most recent point of origin of two genes, then they are called *xenologous*.

Intriguingly, there are practical sequence-based methods that allow to distinguish orthologs and paralogs with acceptable accuracy without constructing either gene or species trees [AD09]. Orthologous genes are present as similar DNA-segments in different species and trace back to a common gene-ancestor where a speciation event happened. Many orthology inference tools have become available over the last decade. To name only a few, `COG` [TGNK00], `OMA` [SDG07, ASGD11], `eggNOG` [JJK$^+$08], `OrthoMCL` [LSR03, CMSR06], `InParanoid` [ÖSF$^+$10], `Roundup 2.0` [DCJ$^+$12], `EGM2` [MWS$^+$12] or `ProteinOrtho` [LFS$^+$11] and its extension `PoFF` [LHRD$^+$14], see [KWMK11, DAGD13] for a recent review. These methods have in common that the output is a set of (pairs of) putative orthologous genes. In addition, orthology detection tools often report some weight or confidence value $w(x, y)$ for x and y to be orthologs or not.

To infer the evolutionary history of species, it has been common practice, to employ alignments of protein or DNA sequences, whose evolutionary history is believed to be congruent to that of the respective species. This property can be ensured most easily in the absence of gene duplications and horizontal gene transfer. Phylogenetic studies judiciously select families of genes that rarely exhibit duplications (such as rRNAs, most ribosomal proteins, and many of the housekeeping enzymes). In phylogenomics, elaborate automatic pipelines such as `HaMStR` [ESvH09] are used to filter genome-wide data sets to at least deplete sequences with detectable paralogs (homologs in the same species).

However, this way of reconstruction amounts to ignoring the information of paralogous and xenologous genes. Paralogs and xenologs are treated as a dangerous nuisance that has to be detected and removed. Hence, the set of available genes from which valuable information about the evolutionary history can be derived is strongly restricted. Moreover, in the presence of gene duplications it becomes necessary to distinguish between the evolutionary history of genes (*gene trees*) and the evolutionary history of the species (*species trees*) in which these genes reside. The supposed history of genes is usually visualized by a rooted tree $T = (V, E)$, i.e., an connected cycle-free graph with one distinguished vertex $\rho \in V$ as the root [SS03]. As a good approximation, one can also suppose that the evolutionary history of species can be depicted as such a rooted tree, although evolutionary events such as HGT or hybridization can locally disturb the acyclic structure. The root serves as the common ancestor of all considered genes and species, respectively, and implies a partial order on the tree to distinguish between the so-called lowest common ancestors (lca) of certain leaves and “points in time” of evolution. While inner vertices of species trees always correspond to speciation events in the gene trees, the inner vertices of a gene tree are in addition labeled to distinguish between the particular events that happened at this point of evolution. In other words, the event on the lowest common ancestor $\mathrm{lca}(x, y)$ of two genes x, y in the corresponding gene tree states whether x and y are orthologous (speciation event) or if they share another kind of homologous relationship such as paralogy (duplication event) or xenology (HGT), see Fitch [Fit00]. An illustrative example is given in Figure 2.3.

Hence, one might ask if one can design methods that allow to use more than the information provided by orthologous genes only. In this part of the contribution, we will show how to derive reliable species trees from genomic data including the knowledge of orthologous and paralogous genes. Hence, this methods allows to use significantly larger sets of possible genes and thus, strikingly more information for more precise computations of the evolutionary history of the species.

Important Basic Concepts

We summarize here important concepts that we need in the following discussion and summary of the results.

Trees. A *(phylogenetic) tree (on L)* is a rooted connected cycle-free graph $T = (V, E)$ with leaf set $L \subseteq V$ such that no inner vertex $v \in V^0 := V \setminus L$ has outdegree one and whose root $\rho \in V$ has indegree zero. A phylogenetic tree on a set of genes $\mathcal{G}$ is called *gene tree* and on a set of species $\mathcal{S}$ *species tree*. An inner vertex y is an ancestor of $x \in V$, in symbols $x \preceq_T y$, if y lies on the unique path connecting x with the root ρ of T. The first inner vertex $\mathrm{lca}_T(x, y)$ that lies on both unique paths from two leaves x, resp. y to the root is called *lowest common ancestor* of x and y. If the context is clear, we will sometimes write $\mathrm{lca}(x, y)$ instead of $\mathrm{lca}_T(x, y)$.
While inner vertices of species trees always correspond to speciation events, the inner vertices of a gene tree are additionally labeled with $t : V^0 \to M$ to distinguish between the particular events that happened at this point of evolution. Hence, for orthologs $x, y \in \mathcal{G}$ it must hold that $t(\mathrm{lca}_T(x, y)) =$ speciation. Other kind of homologous relationship such as paralogy (duplication event) or xenology (HGT), see Fitch [Fit00], can also be inferred by such event-labelings t.

The Orthology Relation Θ. The output of orthology inference tools is a set of (pairs of) putative orthologous genes. This gives rise to a symmetric, irreflexive binary relation

$$\begin{aligned}\widehat{\Theta} &= \{(x, y) \mid x, y \text{ are estimated as orthologous}\}\\ &= \{(x, y) \mid \mathrm{lca}(x, y) = \text{ speciation event (in the respective gene tree)}\}.\end{aligned}$$

This relation $\widehat{\Theta}$ is an estimate of the true orthology relation Θ. The latter allows to infer edges in the orthology graph $G_{\widehat{\Theta}}$. This graph $G_{\widehat{\Theta}}$ has as vertex set the set of considered genes $\mathcal{G}$ and edges $(x, y) \in E(G_{\widehat{\theta}})$ if $(x, y) \in \widehat{\Theta}$. In addition, reported confidence values can be used as weights $w(x, y)$ for edges and non-edges (x, y) of $G_{\widehat{\theta}}$.

Rooted Triples. Rooted triples, i.e., rooted binary trees on three leaves, are a key concept in the theory of supertrees [SS03, BE04, DHK$^+$12]. A rooted triple $r =$ (xy|z) is *displayed* by a rooted tree T with leaf set L if (i) $x, y, z \in L$ and (ii) the path from x to y does not intersect the path from z to the root ρ_T of T. Thus $\mathrm{lca}_T(x, y) \prec_T \mathrm{lca}_T(x, y, z)$. Given a triple set R, the polynomial-time algorithm `BUILD` [ASSU81] either constructs a phylogenetic tree T displaying all triples in R or recognizes that R is inconsistent. The problem of finding a phylogenetic tree with the smallest possible number of vertices that is consistent with every rooted triple in R, i.e., a *least resolved* tree, is an NP-hard problem [JLL12]. If R is inconsistent, the problem of determining a maximum consistent subset of an inconsistent set of triples is NP-hard and also APX-hard, see [BGHK10, vIKM09].

Reconciliation. A phylogenetic tree $S = (W, F)$ with leaf set $\mathcal{S}$ (the species) is a species tree for an event-labeled gene tree $T = (V, E)$ with leaf set $\mathcal{G}$ (the respective genes contained in the species of $\mathcal{S}$) if there is a so-called reconciliation map $\mu : V \to W \cup F$ that maps genes $a \in \mathcal{G}$ to its respective species $\mu(a) \in \mathcal{S}$ such that, roughly spoken, the ancestor relation of the species in S is implied by the ancestor relation of the genes in T. Such maps associate inner vertices of the gene tree T corresponding to speciation events with inner vertices of the species tree S, while vertices of T that map to edges of S are other events, such as duplications. In other words, a phylogenetic tree S is a species tree for some gene tree T if there is a valid embedding of T in S so that the evolutionary history of the genes and the respective species agree. The latter definition is quite technical and we omit it here, but refer to [HRHW^{+}12] for the complete definition for such tree reconciliation maps in the absence of HGT.

Outline of Central Results

In what follows, we briefly summarize established results that are part of this thesis concerning phylogenetic reconstruction. A more detailed discussion of these results can be found in Chapter 3. *(not part of this extended abstract)*.

In [HHRH^{+}13] we characterized "valid" orthology relations Θ, i.e., orthology relations for which we can ensure that there is an event-labeled phylogenetic tree T so that the event corresponding to the least common ancestor $\mathrm{lca}(x, y)$ of x and y in T is a speciation if and only if $(x, y) \in \Theta$. Somewhat surprisingly, such an orthology relation Θ is valid if and only if the orthology graph G_Θ does not contain an induced path on four vertices. The latter kind of graphs are also known as P_4-free graph or cograph. We generalized this result and provided a characterization of relations that might correspond to further additional events such as horizontal gene transfer. This, in particular, is the first mathematical graph theoretical characterization of "validly computed" relations between genes that might have important impact on methods to clean up estimated noisy data corresponding to such relations. Additionally, we provided a polynomial-time algorithm, called `BOTTOM-UP`, that for a given map $\vartheta : \mathcal{G} \times \mathcal{G} \to M$ constructs an event-labeled gene tree for ϑ (if one exists), where ϑ is a map that assigns to pairs of distinct genes $(x, y) \in \mathcal{G} \times \mathcal{G}$ the corresponding evolutionary relationship. We also showed that the classical algorithm `BUILD` can be used to derive the representing tree of Θ. To this end, we showed how to infer small building blocks – specific triples – directly from Θ in order to determine whether there is tree representation for those sets of relations.

In [HRHW^{+}12] we addressed the *tree reconciliation problem* and answered the following question: What conditions have to be satisfied so that there exists a species tree for an event-labeled gene tree with speciation and duplication events? To this end, we characterized gene trees for which there is a species tree by means of particular small building blocks, so-called species triples. These species triples can be inferred directly from the gene tree and give important information of the local topology of the species tree. Additionally, we showed that the computation of the species tree S and a valid reconciliation map, i.e., an embedding of the gene tree T into the species tree S, can be found in polynomial time, if one exists.

In order to test upcoming algorithms and heuristics for phylogenetic reconstruction, we provided a simple model and freely-available software for the simulation of gene family histories

[HRHWS14], and thus, for generating gene trees that are close to “real world” gene trees. This model can be used to generate test data and provides evaluation procedures to test, compare, and benchmark the performance and results of established methods that infer gene trees and species trees.

For a first step towards establishing heuristics for editing an estimated orthology relation we investigated the structure of so-called spider graphs in [BCF$^+$15]. Editing a spider graph to a closest valid orthology relation, that is, a P_4-free graph, can be done in polynomial time [LWGC12]. In order to easily detect spider graphs, we gave a full characterization of spiders in terms of their degrees and respective degree sequences. This allows an easy and efficient check whether a graph is a spider. Additionally, the degree sequence of arbitrary graphs can easily be tested for inconsistencies compared to a closest valid spider degree sequence.

In our seminal contribution [HWL$^+$15] published in the *Proceedings of the National Academy of Sciences*, we used the latter established results and showed that in practice it is possible to derive highly accurate species trees from genomic material including orthologous, paralogous and xenologous genes. This in particular, heavily improves existing methods that rely on the information of 1:1 orthologs only and that treat paralogs and xenologs as a dangerous nuisance that has to be detected and removed. Hence, the set of available genes that we can use is significantly larger as the set of allowed genes in existing methods. In other words, our method allows to use larger gene sets from which more valuable information about the evolutionary history can be derived and hence, more reliable species trees can be inferred compared to other approaches. To this end, we had to solve three intertwined NP-hard combinatorial optimization problems. We showed that these problems are nevertheless tractable in practice by formulating them as Integer Linear Programs (ILP) that additionally allow to compute exact solutions. These ILP formulations are implemented in the freely-available software `ParaPhylo` using `IBM ILOG CPLEX`$^{\text{TM}}$ Optimizer 12.6. We showed that we can get highly accurate solutions for both, artificial benchmark data sets and real-life data sets, comprising genome-scale gene sets for dozens of species, even in the massive presence of events as horizontal gene transfer that usually lead to noisy data and conflicting inferred triple sets.

Summary

This thesis is concerned with the decomposition of graphs or hypergraphs to smaller elementary structures such as prime factors or triples and the reconstruction of set systems from their building blocks such as (approximate) products or supertrees.

In Part I and II, we generalized existing results including unique PFD theorems and algorithms for graphs to general set system containing even infinite directed hypergraphs having infinitely many factors. This includes the first PFD algorithms for directed graphs [HM15] and undirected hypergraphs [HON14] w.r.t. generalizations of the strong graph product, as well as unique PFD theorems for the Cartesian product of infinite hypergraphs [OHS12]. Moreover, we established a local $\boxtimes$-PFD for undirected graphs that offers a complete new way to determine the prime factors and laid the basis for further design of algorithms for approximate products [Hel11]. We were able to design the first general algorithms for the recognition of approximate products w.r.t. the Cartesian [HIK13, HIK15] and the strong product [Hel11]. We provided an infinite class of Cartesian prime graphs, so-called S-prime graphs [HOS12a], and gave several (co)NP-hardness results concerning the recognition of S-prime graphs and subgraphs of Hamming graphs [Hel13]. Additionally, we provided automatic visualization tools for product graphs [JHH$^+$10] and a comprehensive survey on hypergraph products [HOS12b]. Moreover, we have generalized the notion of products to relaxed product structures, by relaxing the axiom set that describe product relations [HOS14, HMOS15]. We showed that graphs having non-trivial relaxed relations exhibit a rich product structure. Taken all latter results together, we have, on the one hand, largely contributed to understand graph products and their generalizations to set-systems (as directed or infinite hypergraphs) and, on the other hand, established an entire new line of research in computational graph theory, namely "product-like set systems".

In Part III, we have established important results concerning the inference of phylogenetic building blocks (triples) and the reconstruction of trees that display the evolutionary history of underlying genes and species [BCF$^+$15, HRHW$^+$12, HHRH$^+$13, HRHWS14, HWL$^+$15]. While recent research focused on approaches that only involved pairwise orthologous genes to find respective species trees, and hence, ignored any information provided by possible paralogs or xenologs, we have demonstrated that the distribution of paralogs in large gene families contains in itself sufficient phylogenetic signal to infer fully resolved species phylogenies. This source of phylogenetic information is independent of information contained in orthologous sequences and is resilient against horizontal gene transfer. An important consequence is that phylogenomics data sets need not be restricted to 1:1 orthologs. To this end, we characterized "valid" orthology relations and generalizations of it [HHRH$^+$13]; we addressed the tree reconciliation problem and characterized gene trees for which there exists a correct species tree in terms of specific gene triples [HRHW$^+$12]; we showed how to reconstruct those species trees from inferred building blocks (specific species triples) and provided the freely available software tool `ParaPhylo` [HWL$^+$15]. `ParaPhylo` allows to compute exact solutions of problems concerning the inference

of the respective building blocks as well as the reconstruction of the evolutionary history of the species. The latter results, in particular, established a new line of research, namely “phylogenomics *with* paralogs” and opened a new viewpoint on how to solve an old problem, since these methods allow to attack the problem of inferring species trees taking into account the additional information of paralogs. We can suspect that our results will have far-reaching consequences for the area of phylogenetics and orthology detection.

Bibliography

[AB12] S. Afzal and C Brand. Recognizing triangulated Cartesian graph products. *Discrete Mathematics*, 312(1):188–193, 2012.

[AD09] A M Altenhoff and C. Dessimoz. Phylogenetic and functional assessment of orthologs inference projects and methods. *PLoS Comput Biol.*, 5:e1000262, 2009.

[AFS91] J. Abello, M. R. Fellows, and J. C. Stillwell. On the complexity and combinatorics of covering finite complexes. *Australas. J. Combin.*, 4:103–112, 1991.

[AHI92] F. Aurenhammer, J. Hagauer, and W. Imrich. Cartesian graph factorization at logarithmic cost per edge. *Comput. Complexity*, 2:331–349, 1992.

[AHS10] Ajith Abraham, Aboul-Ella Hassanien, and Vaclav Snašel, editors. *Computational Social Network Analysis: Trends, Tools and Research Advances*. Computer Communications and Networks. Springer, 2010.

[AMA07] D. Archambault, T. Munzner, and D. Auber. TopoLayout: Multilevel graph layout by topological features. *IEEE Trans. Vis. Comput. Graphics*, 13(2):305–317, 2007.

[AMA08] D. Archambault, T. Munzner, and D. Auber. Grouseflocks: Steerable exploration of graph hierarchy space. *IEEE Trans. Vis. Comput. Graphics*, 14(4):900–913, 2008.

[ASGD11] Adrian M. Altenhoff, Adrian Schneider, Gaston H. Gonnet, and Christophe Dessimoz. OMA 2011: orthology inference among 1000 complete genomes. *Nucleic Acids Research*, 39(suppl 1):D289–D294, 2011.

[ASSU81] A. V. Aho, Y. Sagiv, T. G. Szymanski, and J. D. Ullman. Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. *SIAM J. Comput.*, 10:405–421, 1981.

[BCF$^+$15] S. Berkemer, R. Chaves, A. Fritz, M. Hellmuth*, M. Hernandez-Rosales, and P. F. Stadler. Spiders can be recognized by counting their legs. *Math. Comput. Sci*, pages 1–5, 2015.

[BE04] O.R.P Bininda-Emonds. *Phylogenetic Supertrees*. Kluwer Academic Press, Dordrecht, The Netherlands, 2004.

[Ber89] C. Berge. *Hypergraphs: Combinatorics of finite sets*, volume 45. North-Holland, Amsterdam, 1989.

[BGHK10] J. Byrka, P. Gawrychowski, K. T. Huber, and S. Kelk. Worst-case optimal approximation algorithms for maximizing triplet consistency within phylogenetic networks. *J. Discr. Alg.*, 8:65–75, 2010.

[BS10] Alain Bretto and Yannick Silvestre. Factorization of cartesian products of hypergraphs. In My Thai and Sartaj Sahni, editors, *Computing and Combinatorics*, volume 6196 of *Lecture Notes in Computer Science*, pages 173–181. Springer Berlin / Heidelberg, 2010.

[CKS00] J. Cupal, S. Kopp, and P. F. Stadler. RNA shape space topology. *Artificial Life*, 6:3–23, 2000.

[CMSR06] Feng Chen, Aaron J. Mackey, Christian J. Stoeckert, and David S. Roos. Orthomcl-db: querying a comprehensive multi-species collection of ortholog groups. *Nucleic Acids Research*, 34(suppl 1):D363–D368, 2006.

[CTL13] Christophe Crespelle, Eric Thierry, and Thomas Lambert. A linear-time algorithm for computing the prime decomposition of a directed graph with regard to the cartesian product. In Ding-Zhu Du and Guochuan Zhang, editors, *Computing and Combinatorics*, volume 7936 of *Lecture Notes in Computer Science*, pages 469–480. Springer Berlin Heidelberg, 2013.

[DAGD13] Daniel A. Dalquen, Adrian M. Altenhoff, Gaston H. Gonnet, and Christophe Dessimoz. The impact of gene duplication, insertion, deletion, lateral gene transfer and sequencing error on orthology inference: A simulation study. *PLoS ONE*, 8(2):e56925, 02 2013.

[DCJ$^+$12] Todd F. DeLuca, Jike Cui, Jae-Yoon Jung, Kristian Che St. Gabriel, and Dennis P. Wall. Roundup 2.0: Enabling comparative genomics for over 1800 genomes. *Bioinformatics*, 28(5):715–716, 2012.

[DHK$^+$12] A.W.M. Dress, K.T. Huber, J. Koolen, V. Moulton, and A. Spillner. *Basic phylogenetic combinatorics*. Cambridge University Press, 2012.

[Dör82] W. Dörfler. On the direct product of hypergraphs. *Ars Comb.*, 14:67–78, 1982.

[DR88] 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.

[EMS89] M. Eigen, J. McCaskill, and P. Schuster. The molecular quasispecies. *Adv. Chem. Phys.*, 75:149–263, 1989.

[ESvH09] I. Ebersberger, S. Strauss, and A. von Haeseler. HaMStR: Profile hidden markov model based search for orthologs in ESTs. *BMC Evol. Biol.*, 9:157, 2009.

[Fed92] T. Feder. Product graph representations. *J. Graph Theory*, 16:467–488, 1992.

[Fei86a] J. Feigenbaum. Directed cartesian-product graphs have unique factorizations that can be computed in polynomial time. *Discrete Applied Mathematics*, 15(1):105 – 110, 1986.

[Fei86b] J. Feigenbaum. Product graphs: some algorithmic and combinatorial results. Technical Report STAN-CS-86-1121, Stanford University, Computer Science, 1986. PhD Thesis.

[FH89] J. Feigenbaum and R. A. Haddad. On factorable extensions and subgraphs of prime graphs. *SIAM J. Discrete Math.*, 2:197–218, 1989.

[FHS85] J. Feigenbaum, J. Hershberger, and A. A. Schäffer. A polynomial time algorithm for finding the prime factors of Cartesian-product graphs. *Discrete Appl. Math.*, 12:123–138, 1985.

[Fit00] W.M. Fitch. Homology: a personal view on some of the problems. *Trends Genet.*, 16:227–231, 2000.

[FJS85] 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.

[FS86] J. Feigenbaum and A. A. Schäffer. Recognizing composite graphs is equivalent to testing graph isomorphism. *SIAM J. Comput.*, 15:619–627, 1986.

[FS92] J. Feigenbaum and A. A. Schäffer. Finding the prime factors of strong direct product graphs in polynomial time. *Discrete Math.*, 109:77–102, 1992.

[FS98a] W. Fontana and P. Schuster. Continuity in Evolution: On the Nature of Transitions. *Science*, 280:1451–1455, 1998.

[FS98b] W. Fontana and P. Schuster. Shaping Space: The Possible and the Attainable in RNA Genotype-Phenotype Mapping. *J. Theor. Biol.*, 194:491–515, 1998.

[GGS^{+}96] 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.

[GI71] G. Gaszt and W. Imrich. On the lexicographic and costrong product of set systems. *Aequationes Mathematicae*, 6:319–320, 1971.

[Hel11] M. Hellmuth*. A local prime factor decomposition algorithm. *Discr. Math.*, 311(12):944–965, 2011.

[Hel13] M. Hellmuth*. On the complexity of recognizing S-composite and S-prime graphs. *Discr. Appl. Math.*, 161(7-8):1006 – 1013, 2013.

[HHRH$^+$13] M. Hellmuth, M. Hernandez-Rosales, K. T. Huber*, V. Moulton, P. F. Stadler, and N. Wieseke. Orthology relations, symbolic ultrametrics, and cographs. *J. Math. Biology*, 66(1-2):399–420, 2013.

[HI09] R. Hammack and W. Imrich. On Cartesian skeletons of graphs. *Ars Math. Contemp.*, 2(2):191–205, 2009.

[HIK11] R. Hammack, W. Imrich, and S. Klavžar. *Handbook of Product Graphs*. Discrete Mathematics and its Applications. CRC Press, 2nd edition, 2011.

[HIK13] M. Hellmuth*, W. Imrich, and T. Kupka. Partial star products: A local covering approach for the recognition of approximate Cartesian product graphs. *Math. Comput. Sci*, 7(3):255–273, 2013.

[HIK15] M. Hellmuth*, W. Imrich, and T. Kupka. Fast recognition of partial star products and quasi Cartesian products. *Ars Math. Cont.*, 9(2):233 – 252, 2015.

[HIKS09] M. Hellmuth, W. Imrich, W. Klöckl, and P. F. Stadler. Approximate graph products. *European J. Combin.*, 30(5):1119 – 1133, 2009.

[HM15] M. Hellmuth* and T. Marc. On the Cartesian skeleton and the factorization of the strong product of digraphs. *J. Theor. Comp. Sci.*, 565(0):16–29, 2015.

[HMOS15] M. Hellmuth, T. Marc, L Ostermeier*, and P. F. Stadler. The relaxed square property. *Australas. J. Combin.*, 62(3):240–270, 2015.

[HON14] M. Hellmuth*, L. Ostermeier, and M. Noll. Strong products of hypergraphs: Unique prime factorization theorems and algorithms. *Discr. Appl. Math.*, 171:60–71, 2014.

[HOS12a] M. Hellmuth*, L. Ostermeier, and P. F. Stadler. Diagonalized Cartesian products of S-prime graphs are S-prime. *Discr. Math.*, 312(1):74 – 80, 2012. Algebraic Graph Theory - A Volume Dedicated to Gert Sabidussi on the Occasion of His 80th Birthday.

[HOS12b] M. Hellmuth, L. Ostermeier, and P. F. Stadler*. A survey on hypergraph products. *Math. Comput. Sci*, 6:1–32, 2012.

[HOS14] M. Hellmuth, L. Ostermeier*, and P. F. Stadler. Unique square property, equitable partitions, and product-like graphs. *Discr. Math.*, 320(0):92 – 103, 2014.

[HRHW$^+$12] M. Hernandez-Rosales*, M. Hellmuth, N. Wieseke, K. T. Huber, V. Moulton, and P. F. Stadler. From event-labeled gene trees to species trees. *BMC Bioinformatics*, 13(Suppl 19):S6, 2012.

[HRHWS14] M. Hernandez-Rosales*, M. Hellmuth, N. Wieseke, and P. F. Stadler. Simulation of gene family histories. *BMC Bioinformatics*, 15(Suppl 3):A8, 2014.

[Hus93] D. Husemoller. *Fibre Bundles*. Springer, Heidelberg, 3rd edition, 1993.

[Hv99] J. Hagauer and J. Žerovnik. An algorithm for the weak reconstruction of cartesian-product graphs. *J. Combin. Inf. Syst. Sci.*, 24:97–103, 1999.

[HWL+15] M. Hellmuth*, N. Wiesecke, M. Lechner, H.P. Lenhof, M. Middendorf, and P. F. Stadler. Phylogenomics with paralogs. *Proceedings of the National Academy of Sciences*, 112(7):2058–2063, 2015.

[II75] W. Imrich and H. Izbicki. Associative products of graphs. *Monatshefte für Mathematik*, 80(4):277–281, 1975.

[IK00] W. Imrich and S Klavžar. *Product graphs*. Wiley-Intersci. Ser. Discrete Math. Optim. Wiley-Interscience, New York, 2000.

[IK07] Wilfried Imrich and Werner Klöckl. Factoring directed graphs with respect to the cardinal product in polynomial time. *Dicussiones Mathematicae Graph Theory*, 27(3):593–601, 2007.

[IK10] Wilfried Imrich and Werner Klöckl. Factoring directed graphs with respect to the cardinal product in polynomial time ii. *Discuss. Math. Graph Theory*, 30(3):461–474, 2010.

[IKD08] W. Imrich, S. Klavžar, and F. R. Douglas. *Topics in Graph Theory: Graphs and Their Cartesian Product*. AK Peters, Ltd., Wellesley, MA, 2008.

[Imr71] W. Imrich. über das schwache kartesisches Produkt von Graphen. *J.Combin. Theory Ser B*, 11:1 – 6, 1971.

[Imr98] Wilfried Imrich. Factoring cardinal product graphs in polynomial time. *Discrete Math.*, 192:119–144, 1998.

[IP07] W. Imrich and I. Peterin. Recognizing Cartesian products in linear time. *Discrete Math.*, 307:472–482, 2007.

[IPv97] W. Imrich, T. Pisanski, and J. Žerovnik. Recognizing cartesian graph bundles. *Discr. Math*, 167-168:393–403, 1997.

[IŽ96] Wilfried Imrich and Janez Žerovnik. On the weak reconstruction of Cartesian-product graphs. *Discrete Math.*, 150(1-3):167–178, 1996. Selected papers in honour of Paul Erdős on the occasion of his 80th birthday (Keszthely, 1993).

[IZv01] Wilfried Imrich, Blaz Zmazek, and Janez Žerovnik. Weak k-reconstruction of cartesian product graphs. *Electronic Notes in Discrete Mathematics*, 10:297 – 300, 2001. Comb01, Euroconference on Combinatorics, Graph Theory and Applications.

[IŽZ03] W. Imrich, J. Žerovnik, and B. Zmazek. Weak k-reconstruction of cartesian products. *Discussiones Mathematicae Graph Theory*, 23:273–285, 2003.

[JHH$^+$10] S. Jänicke*, C. Heine, M. Hellmuth, P. F. Stadler, and G. Scheuermann. Visualization of graph products. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1082–1089, 2010.

[JJK$^+$08] Lars Juhl Jensen, Philippe Julien, Michael Kuhn, Christian von Mering, Jean Muller, Tobias Doerks, and Peer Bork. eggnog: automated construction and annotation of orthologous groups of genes. *Nucleic Acids Research*, 36(suppl 1):D250–D254, 2008.

[JLL12] J. Jansson, R.S. Lemence, and A. Lingas. The complexity of inferring a minimally resolved phylogenetic supertree. *SIAM J. Comput.*, 41:272–291, 2012.

[KA15] A. Kaveh and B. Alinejad. Hypergraph products for structural mechanics. *Advances in Engineering Software*, 80:72 – 81, 2015.

[KHT09] Steffen Klamt, Utz-Uwe Haus, and Fabian Theis. Hypergraphs and cellular networks. *PLoS Comput Biol*, 5(5):e1000385, 05 2009.

[KK08] A. Kaveh and K. Koohestani. Graph products for configuration processing of space structures. *Comput. Struct.*, 86(11-12):1219–1231, 2008.

[KM08] 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.

[KR04] A. Kaveh and H. Rahami. An efficient method for decomposition of regular structures using graph products. *Intern. J. Numer. Meth. Eng.*, 61(11):1797–1808, 2004.

[KWMK11] David M. Kristensen, Yuri I. Wolf, Arcady R. Mushegian, and Eugene V. Koonin. Computational methods for gene orthology inference. *Briefings in Bioinformatics*, 12(5):379–391, 2011.

[Lew78] R. C. Lewontin. Adaptation. *Sci. Am.*, 239:156–169, 1978.

[LFS$^+$11] Marcus Lechner, Sven Findeiß, Lydia Steiner, Manja Marz, Peter F. Stadler, and Sonja J. Prohaska. `Proteinortho`: detection of (co-)orthologs in large-scale analysis. *BMC Bioinformatics*, 12:124, 2011.

[LHRD$^+$14] Marcus Lechner, Maribel Hernandez-Rosales, D. Doerr, N. Wiesecke, A. Thevenin, J. Stoye, Roland K. Hartmann, Sonja J. Prohaska, and Peter F. Stadler. Orthology detection combining clustering and synteny for very large datasets. *PLoS ONE*, 9(8):e105015, 08 2014.

[LSR03] Li Li, Christian J Stoeckert, and David S Roos. OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome research*, 13(9):2178–2189, 2003.

[LWGC12] Yunlong Liu, Jianxin Wang, Jiong Guo, and Jianer Chen. Complexity and parameterized algorithms for cograph editing. *Theoretical Computer Science*, 461(0):45 – 54, 2012.

[McK71] R. McKenzie. Cardinal multiplication of structures with a reflexive relation. *Fund. Math.*, 70:59–101, 1971.

[Mil70] D.J. Miller. Weak cartesian product of graphs. *Colloq.Math.*, 21:55 – 74, 1970.

[MS90] B. Monien and H. Sudborough. Embedding one interconnection network in another. In G. Tinhofer, E. Mayr, H. Noltemeier, and M.M. Syslo, editors, *Computational Graph Theory*, volume 7 of *Computing Supplementum*, pages 257–282. Springer Vienna, 1990.

[MWS$^+$12] Khalid Mahmood, Geoffrey I. Webb, Jiangning Song, James C. Whisstock, and Arun S. Konagurthu. Efficient large-scale protein sequence comparison and gene matching to identify orthologs and co-orthologs. *Nucleic Acids Research*, 40(6):e44–e44, 2012.

[OHS12] L. Ostermeier*, M. Hellmuth, and P. F. Stadler. The Cartesian product of hypergraphs. *Journal of Graph Theory*, 70(2):180–196, 2012.

[OR02] Evelien Otte and Ronald Rousseau. Social network analysis: a powerful strategy, also for the information sciences. *Journal of Information Science*, 28(6):441–453, 2002.

[ÖSF$^+$10] Gabriel Östlund, Thomas Schmitt, Kristoffer Forslund, Tina Köstler, David N Messina, Sanjit Roopra, Oliver Frings, and Erik LL Sonnhammer. `InParanoid 7`: new algorithms and tools for eukaryotic orthology analysis. *Nucleic acids research*, 38(suppl 1):D196–D203, 2010.

[P$^+$14] K. Prüfer et al. The complete genome sequence of a neanderthal from the altai mountains. *Nature*, 505(7481):43–49, 2014.

[PSTV83] T Pisanski, J. Shawe-Taylor, and J. Vrabec. Edge-colorability of graph bundles. *J. Combinat. Theory Ser. B*, 35:12–19, 1983.

[Sab59] G. Sabidussi. Graph multiplication. *Mathematische Zeitschrift*, 72:446–457, 1959.

[SDG07] Adrian Schneider, Christophe Dessimoz, and Gaston H. Gonnet. Oma browserâĂŤ-exploring orthologous relations across 352 complete genomes. *Bioinformatics*, 23(16):2180–2182, 2007.

[SS03] Charles Semple and Mike Steel. *Phylogenetics*, volume 24 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, UK, 2003.

[SS04] B.M.R. Stadler and P. F. Stadler. The topology of evolutionary biology. In Ciobanu, editor, *Modeling in Molecular Biology*, Natural Computing Series, 267-286, 2004. Springer Verlag.

[SSDN02] Mario Schlosser, Michael Sintek, Stefan Decker, and Wolfgang Nejdl. Hypercup - hypercubes, ontologies and efficient search on p2p networks. In *LNCS*, pages 112–124. Springer, 2002.

[SSWF01] 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.

[TGNK00] Roman L. Tatusov, Michael Y. Galperin, Darren A. Natale, and Eugene V. Koonin. The cog database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Research*, 28(1):33–36, 2000.

[TP11] My T. Thai and Panos M. Pardalos. *Handbook of Optimization in Complex Networks: Communication and Social Networks Cover*, volume 58 of *Springer Optimization and its Applications*. Springer Science & Business Media, 2011.

[vIKM09] L. van Iersel, S. Kelk, and M. Mnich. Uniqueness, intractability and exact algorithms: reflections on level-k phylogenetic networks. *J. Bioinf. Comp. Biol.*, 7:597–623, 2009.

[Ž00] J. Žerovnik. On recognition of strong graph bundles. *Math. Slovaca*, 50:289–301, 2000.

[Win87] P.M. Winkler. Factoring a graph in polynomial time. *European J. Combin.*, 8:209–212, 1987.

[WS03] 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.

[YP96] Sang-Kyun Yun and Kyu Ho Park. Hierarchical hypercube networks (hhn) for massively parallel computers. *Journal of Parallel and Distributed Computing*, 37(2):194 – 199, 1996.

[Zv02] B. Zmazek and J. Žerovnik. Algorithm for recognizing cartesian graph bundles. *Discrete Appl. Math.*, 120:275–302, 2002.

[ZŽ02] B. Zmazek and J. Žerovnik. Unique square property and fundamental factorizations of graph bundles. *Discr. Math.*, 244:551–561, 2002.

[ZŽ07] B. Zmazek and J. Žerovnik. Weak reconstruction of strong product graphs. *Discrete Math.*, 307:641–649, 2007.