Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 68 (2002)

Complexity of the Exact Domatic Number Problem and of the Exact Conveyor Flow Shop Problem*

Tobias Riege† and Jörg Rothe‡
Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany

March 23, 2004

Abstract

We prove that the exact versions of the domatic number problem are complete for the levels of the boolean hierarchy over NP. The domatic number problem, which arises in the area of computer networks, is the problem of partitioning a given graph into a maximum number of disjoint dominating sets. This number is called the domatic number of the graph. We prove that the problem of determining whether or not the domatic number of a given graph is *exactly* one of k given values is complete for $\mathrm{BH}_{2k}(\mathrm{NP})$, the $2k$th level of the boolean hierarchy over NP. In particular, for $k = 1$, it is DP-complete to determine whether or not the domatic number of a given graph equals exactly a given integer. Note that $\mathrm{DP} = \mathrm{BH}_2(\mathrm{NP})$. We obtain similar results for the exact versions of generalized dominating set problems and of the conveyor flow shop problem. Our reductions apply Wagner's conditions sufficient to prove hardness for the levels of the boolean hierarchy over NP.

Key words: *Computational complexity; completeness; domatic number problem; conveyor flow shop problem; boolean hierarchy*

*Supported in part by the German Science Foundation (DFG) under grant RO 1202/9-1. An extended abstract of this paper appears in the proceedings of the *First International Conference on Information & Communication Technologies: From Theory to Applications* (ICTTA'04), Damascus, Syria, April 2004.
†Email: `riege@cs.uni-duesseldorf.de`.
‡Email: `rothe@cs.uni-duesseldorf.de`.

ISSN 1433-8092

1 Introduction and Motivation

1.1 Two Scenarios Motivating the Domatic Number Problem

A dominating set in an undirected graph G is a subset D of the vertex set $V(G)$ such that every vertex of $V(G)$ either belongs to D or is adjacent to some vertex in D. The domatic number problem is the problem of partitioning the vertex set $V(G)$ into a maximum number of disjoint dominating sets. This number, denoted by $\delta(G)$, is called the domatic number of G. The domatic number problem arises in various areas and scenarios. In particular, this problem is related to the task of distributing resources in a computer network, and also to the task of locating facilities in a communication network.

Scenario 1: Suppose, for example, that resources are to be allocated in a computer network such that expensive services are quickly accessible in the immediate neighborhood of each vertex. If every vertex has only a limited capacity, then there is a bound on the number of resources that can be supported. In particular, if every vertex can serve a single resource only, then the maximum number of resources that can be supported equals the domatic number of the network graph.

Scenario 2: In the communication network scenario, n cities are linked via communication channels. A transmitting group is a subset of those cities that are able to transmit messages to every city in the network. Such a transmitting group is nothing else than a dominating set in the network graph, and the domatic number of this graph is the maximum number of disjoint transmitting groups in the network.

1.2 Some Background and Motivation from Complexity Theory

Motivated by the scenarios given above, the domatic number problem has been thoroughly investigated. Its decision version, denoted by DNP, asks whether or not $\delta(G) \geq k$, for a given graph G and a positive integer k. This problem is known to be NP-complete (cf. [GJ79]), and it remains NP-complete even if the given graph belongs to certain special classes of perfect graphs including chordal and bipartite graphs; see the references in Section 2. Feige et al. [FHK00] established nearly optimal approximation algorithms for the domatic number.

Expensive resources should not be wasted. Given a graph G and a positive integer i, how hard is it to determine whether or not $\delta(G)$ equals i *exactly*? Of course, a binary search using logarithmically many questions to DNP would do the job and would prove this problem to be contained in $\mathrm{P}_{||}^{\mathrm{NP}}$, the class of problems solvable in deterministic polynomial time via parallel (a.k.a. "nonadaptive" or "truth-table") access to NP. Can this obvious upper bound be improved? Can we find a better upper bound and a matching lower bound so that this problem is classified according to its computational complexity?

In this paper, we provide a variety of such completeness results that pinpoint the precise complexity of *exact generalized dominating set problems*, including the just-mentioned exact domatic number problem. Motivated by such *exact versions of* NP-*complete optimization problems*, Papadimitriou and Yannakakis introduced in their seminal paper [PY84] the class DP, which consists of the differences of any two NP sets. They also studied various other important classes of problems that belong to DP, including *facet problems*, *unique solution problems*, and *critical problems*, and they proved many of them complete for DP.

As an example for a DP-complete critical graph problem, we mention one specific colorability problem on graphs. A graph G is said to be k*-colorable* if its vertices can be colored with no more than k colors such that no two adjacent vertices receive the same color. The *chromatic number of* G, denoted by $\chi(G)$, is defined to be the smallest k such that G is k-colorable. In particular, the 3-colorability problem, one of the standard NP-complete problems (cf. [GJ79]), is defined by

$$\texttt{3-Colorability} = \{G \mid G \text{ is a graph with } \chi(G) \leq 3\}.$$

Cai and Meyer [CM87] showed that `Minimal-3-Uncolorability` is DP-complete, a critical graph problem that asks whether a given graph is not 3-colorable, but deleting any of its vertices makes it 3-colorable.

As an example for a DP-complete exact graph problem, we mention one further specific colorability problem on graphs. Wagner [Wag87] showed that for any fixed integer $i \geq 7$, it is DP-complete to determine whether or not $\chi(G)$ equals i exactly, for a given graph G. Recently, Rothe optimally strengthened Wagner's result by showing that it is DP-complete to determine whether or not $\chi(G) = 4$, yet the problem of determining whether or not $\chi(G) = 3$ is in NP and thus very unlikely to be DP-complete [Rot03].

More generally, given a graph G and a list $M_k = \{i_1, i_2, \dots, i_k\}$ of k positive integers, how hard is it to determine whether or not $\delta(G)$ equals some i_j exactly? Generalizing DP, Cai et al. [CGH$^+$88,CGH$^+$89] introduced and studied $\mathrm{BH}(\mathrm{NP}) = \bigcup_{k \geq 1} \mathrm{BH}_k(\mathrm{NP})$, the boolean hierarchy over NP; see Definition 3 in Section 2. Note that DP is the second level of this hierarchy. Wagner [Wag87] identified a set of conditions sufficient to prove $\mathrm{BH}_k(\mathrm{NP})$-hardness for each k, and he applied his sufficient conditions to prove a host of exact versions of NP-complete optimization problems complete for the levels of the boolean hierarchy. In particular, Wagner [Wag87] proved that the problem of determining whether or not the chromatic number of a given graph is exactly one of k given values is complete for $\mathrm{BH}_{2k}(\mathrm{NP})$. Also this more general result of Wagner was improved optimally in [Rot03]: $\mathrm{BH}_{2k}(\mathrm{NP})$-completeness of these exact chromatic number problems for given k-element sets is achieved using k-tuples whose components indicate the smallest number of colors possible.

Wagner's technique was also useful in proving certain natural problems complete for $\mathrm{P}_{||}^{\mathrm{NP}}$. For example, the winner problem for Carroll elections [HHR97a,HHR97b] and for Young elections [RSV03] as well as the problem of determining when certain graph heuristics work well [HR98,HRS02] each are complete for $\mathrm{P}_{||}^{\mathrm{NP}}$.

1.3 Outline and Context of our Results

This paper is organized as follows. Section 2 introduces the graph-theoretical notation used and provides the necessary background from complexity theory. In addition, we present some results and proof techniques to be applied later on.

Section 3 introduces a uniform approach proposed by Heggernes and Telle [HT98] that defines graph problems by partitioning the vertex set of a graph into generalized dominating sets. These generalized dominating set problems are parameterized by two sets of nonnegative integers, σ and ρ, restricting the number of neighbors for each vertex in the partition. Using this uniform approach, a great variety of standard graph problems, including various domatic number and graph colorability problems, can be characterized by such (k, σ, ρ)-partitions for a given parameter k; Table I in [HT98] provides an extensive list containing 13 well-known graph problems in standard terminology and their characterization by (k, σ, ρ)-partitions. We adopt Heggernes and Telle's approach and expand it by defining the exact versions of their generalized dominating set problems. We also show in this section some easy properties of the problems defined.

In Section 4, we study these exact generalized dominating set problems in more depth. The main results of this paper are presented in Sections 4.2 and 4.3: We establish DP-completeness results for a variety of such exact generalized dominating set problems. In particular, we prove in Section 4.2.1 that for any fixed integer $i \geq 5$, it is DP-complete to determine whether or not the domatic number of a given graph is exactly i. In contrast, the problem of deciding whether or not $\delta(G) = 2$, for some given graph G, is coNP-complete.

An overview of all the results from Section 4 is given in Section 4.1. In Section 4.4, we observe that the results of Sections 4.2 and 4.3 can be generalized to completeness results in the higher levels of the boolean hierarchy over NP. This generalization applies Wagner's technique [Wag87] mentioned above. In particular, we prove that determining whether or not the domatic number of a given graph equals exactly one of k given values is complete for $\mathrm{BH}_{2k}(\mathrm{NP})$, thus expanding the list of problems known to be complete for the levels of the boolean hierarchy over NP.

The boolean hierarchy over NP has been thoroughly investigated. For example, a large number of definitions are known to be equivalent [CGH^{+}88,KSW87,HR97], see also [Hau14]. It is known that if the boolean hierarchy collapses to some finite level, then so does the polynomial hierarchy [Kad88,CK96,BCO93]. Hemaspaandra, Hempel, and Wechsung studied the question of whether and to what extent the order matters in which various oracle sets from the boolean hierarchy are accessed [HHW99]. Boolean hierarchies over classes other than NP were intensely investigated as well: Gundermann, Nasser, and Wechsung [GNW90] and Beigel, Chang, and Ogihara [BCO93] studied boolean hierarchies over counting classes, Bertoni et al. [BBJ$^+$89] studied boolean hierarchies over the class RP ("random polynomial time," see [Adl78]), and Hemaspaandra and Rothe [HR97] studied the boolean hierarchy over UP ("unambigous polynomial time," introduced by Valiant [Val76]) and over any set class closed under intersection.

Section 4.5 raises the DP- and $\mathrm{BH}_{2k}(\mathrm{NP})$-completeness results as yet obtained even higher: We prove several variants of the domatic number problem complete for $\mathrm{P}^{\mathrm{NP}}_{||}$, namely `DNP-Odd`, `DNP-Equ`, and `DNP-Geq`. Thus, we expand the list of problems known to be complete for this central complexity class. `DNP-Odd` asks whether or not the domatic number of a given graph is an odd number. `DNP-Equ` asks whether or not the domatic numbers of two given graphs are equal, and `DNP-Geq` asks, given the graphs G and H, whether or not $\delta(G) \geq \delta(H)$ is true. While these problems may not appear to be overly natural, they might serve as good starting points for reductions showing the $\mathrm{P}^{\mathrm{NP}}_{||}$-completeness of other, more natural problems. For example, the quite natural winner problem for Carroll elections was shown to be $\mathrm{P}^{\mathrm{NP}}_{||}$-complete via a reduction from a problem dubbed `TwoElectionRanking` in [HHR97a], which is analogous in structure to the problem `DNP-Geq`. Similarly, the $\mathrm{P}^{\mathrm{NP}}_{||}$-completeness of the quite natural winner problem for Young elections was proven via a reduction from the problem `Maximum Set Packing Compare` in [RSV03]. Finally, the $\mathrm{P}^{\mathrm{NP}}_{||}$-completeness of certain problems related to heuristics for finding a minimum vertex cover [HRS02] or a maxium independent set [HR98] in a graph are shown via reductions from the analogs of `DNP-Geq` and `DNP-Equ` for the vertex cover problem and the independent set problem, respectively.

$\mathrm{P}^{\mathrm{NP}}_{||}$ was introduced by Papadimitriou and Zachos [PZ83] and was intensely studied in a wide variety of contexts. For example, among many other characterizations, $\mathrm{P}^{\mathrm{NP}}_{||}$ is known to be equal to $\mathrm{P}^{\mathrm{NP}[\mathcal{O}(\log)]}$, the class of problems solvable in deterministic polynomial time by logarithmically many Turing queries to an NP oracle; see [Hem89,Wag90,BH91, KSW87]. Furthermore, it is known that if NP contains some $\mathrm{P}^{\mathrm{NP}}_{||}$-hard problem, then the polynomial hierarchy collapses to NP. Kadin [Kad89] proved that if NP has sparse Turing-hard sets, then the polynomial hierarchy collapses to $\mathrm{P}^{\mathrm{NP}}_{||}$. Krentel [Kre88] studied $\mathrm{P}^{\mathrm{NP}}_{||}$ and other levels of the polynomial hierarchy that are relevant for certain optimization problems, see also [GRW01,GRW02]. Ogihara studied the truth-table and log-Turing reducibilities in a general setting; his results in particular apply to $\mathrm{P}^{\mathrm{NP}}_{||}$ and related classes [Ogi94]. In [Ogi96], he investigated the function analogs of $\mathrm{P}^{\mathrm{NP}}_{||}$, see also [JT95, BKT94]. Hemaspaandra and Wechsung [HW91] characterized $\mathrm{P}^{\mathrm{NP}}_{||}$ and related classes in terms of Kolmogorov complexity. Finally, $\mathrm{P}^{\mathrm{NP}}_{||}$ is central to the study of the query and the truth-table hierarchies over NP (see, e.g., [KSW87,Hem89,Wag90,BH91,Bei91,Ko91, BCO93]), to the optimal placement of PP ("probabilistic polynomial time," defined by Gill [Gil77]) in the polynomial hierarchy [BHW91,Bei94], to the study of the low hierarchy and the extended low hierarchies [AH92,Ko89,LS95], and to many other topics.

In Section 5, we study the exact conveyor flow shop problem that we also prove complete for the levels of the boolean hierarchy over NP. The conveyor flow shop problem, which arises in real-world applications in the wholesale business, where warehouses are supplied with goods from a central storehouse, was introduced and intensely studied by Espelage and Wanke [EW00]. The present paper is the first to study the exact version of this natural problem, which we find intriguing mainly due to its applications in practice. For further results on this problem, we refer to [EW00,Esp01,EW01,EW03].

Finally, we conclude this paper with a number of open problems in Section 6.

2 Preliminaries and Notation

We start by introducing some graph-theoretical notation. For any graph G, $V(G)$ denotes the vertex set of G, and $E(G)$ denotes the edge set of G. All graphs in this paper are undirected, simple graphs. That is, edges are unordered pairs of vertices, and there are neither multiple nor reflexive edges (i.e., for any two vertices u and v, there is at most one edge of the form $\{u, v\}$, and there is no edge of the form $\{u, u\}$). Also, all graphs considered do not have isolated vertices, yet they need not be connected in general.

For any vertex $v \in V(G)$, the *degree of* v (denoted by $\mathit{deg}_G(v)$) is the number of vertices adjacent to v in G; if G is clear from the context, we omit the subscript and simply write $\mathit{deg}(v)$. Let $\mathit{max\text{-}deg}(G) = \max_{v \in V(G)} \mathit{deg}(v)$ denote the maximum degree of the vertices of graph G, and let $\mathit{min\text{-}deg}(G) = \min_{v \in V(G)} \mathit{deg}(v)$ denote the minimum degree of the vertices of graph G. The *neighborhood of a vertex v in G* is the set of all vertices adjacent to v, i.e., $N(v) = \{w \in V(G) \mid \{v, w\} \in E(G)\}$. A *partition of* $V(G)$ into k pairwise disjoint subsets $V_1, V_2, \ldots, V_k$ satisfies $V(G) = \bigcup_{i=1}^{k} V_i$ and $V_i \cap V_j = \emptyset$ for $1 \leq i < j \leq k$. For some of the reductions presented in this paper, we need the following operations on graphs.

Definition 1 *The* join operation on graphs*, denoted by $\oplus$, is defined as follows: Given two disjoint graphs A and B, their join $A \oplus B$ is the graph with vertex set $V(A \oplus B) = V(A) \cup V(B)$ and edge set $E(A \oplus B) = E(A) \cup E(B) \cup \{\{a, b\} \mid a \in V(A)$ and $b \in V(B)\}$.*

The disjoint union of any two graphs A and B *is defined as the graph $A \cup B$ with vertex set $V(A) \cup V(B)$ und edge set $E(A) \cup E(B)$.*

Note that $\oplus$ is an associative operation on graphs and $\chi(A \oplus B) = \chi(A) + \chi(B)$. We now define the domatic number problem.

Definition 2 *For any graph G, a* dominating set of G *is a subset $D \subseteq V(G)$ such that for each vertex $u \in V(G) - D$, there exists a vertex $v \in D$ with $\{u, v\} \in E$. The* domatic number of G*, denoted by $\delta(G)$, is the maximum number of disjoint dominating sets. Define the decision version of the* domatic number problem *by*

$$\mathtt{DNP} = \{\langle G, k\rangle \mid G \textit{ is a graph and } k \textit{ is a positive integer such that } \delta(G) \geq k\}.$$

Note that $\delta(G) \leq \textit{min-deg}(G) + 1$ for each graph G. For fixed $k \geq 3$, DNP is known to be NP-complete (cf. [GJ79]), and it remains NP-complete for circular-arc graphs [Bon85], for split graphs (thus, in particular, for chordal and co-chordal graphs) [KS94], and for bipartite graphs (thus, in particular, for comparability graphs) [KS94]. In contrast, DNP is known to be polynomial-time solvable for certain other graph classes, including strongly chordal graphs (thus, in particular, for interval graphs and path graphs) [Far84] and proper circular-arc graphs [Bon85]. For graph-theoretical notions and special graph classes not defined in this extended abstract, we refer to the monograph by Brandstädt et al. [BLS99], a follow-up to the classic text by Golumbic [Gol80].

Feige et al. [FHK00] show that every graph G with n vertices has a domatic partition with $(1 - o(1))(\textit{min-deg}(G) + 1)/\ln n$ sets that can be found in polynomial time, which implies a $(1 - o(1)) \ln n$ approximation algorithm for the domatic number $\delta(G)$. This is a tight bound, since they also show that, for any fixed constant $\varepsilon > 0$, the domatic number cannot be approximated within a factor of $(1 - \varepsilon) \ln n$, unless NP $\subseteq$ DTIME$(n^{\log \log n})$. Finally, Feige et al. [FHK00] give a refined algorithm that yields a domatic partition of $\Omega(\delta(G)/\ln \textit{max-deg}(G))$, which implies a $\mathcal{O}(\ln \textit{max-deg}(G))$ approximation algorithm for the domatic number $\delta(G)$. For more results on the domatic number problem, see [FHK00, KS94] and the references therein.

We assume that the reader is familiar with standard complexity-theoretic notions and notation. For more background, we refer to any standard textbook on computational complexity theory such as Papadimitriou's book [Pap94]. All completeness results in this paper are with respect to the polynomial-time many-one reducibility, denoted by $\leq_{\mathrm{m}}^{\mathrm{p}}$. For sets A and B, define $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ if and only if there is a polynomial-time computable function f such that for each $x \in \Sigma^*$, $x \in A$ if and only if $f(x) \in B$. A set B is $\mathcal{C}$-hard for a complexity class $\mathcal{C}$ if and only if $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ for each $A \in \mathcal{C}$. A set B is $\mathcal{C}$-complete if and only if B is $\mathcal{C}$-hard and $B \in \mathcal{C}$.

To define the boolean hierarchy over NP, we use the symbols $\wedge$ and $\vee$, respectively, to denote the *complex intersection* and the *complex union* of set classes. That is, for classes $\mathcal{C}$ and $\mathcal{D}$ of sets, define

$$\begin{aligned} \mathcal{C} \wedge \mathcal{D} &= \{A \cap B \mid A \in \mathcal{C} \text{ and } B \in \mathcal{D}\}; \\ \mathcal{C} \vee \mathcal{D} &= \{A \cup B \mid A \in \mathcal{C} \text{ and } B \in \mathcal{D}\}. \end{aligned}$$

Definition 3 (Cai et al.) *The* boolean hierarchy over NP *is inductively defined by:*

$$\begin{aligned} \mathrm{BH}_1(\mathrm{NP}) &= \mathrm{NP}, \\ \mathrm{BH}_2(\mathrm{NP}) &= \mathrm{NP} \wedge \mathrm{coNP}, \\ \mathrm{BH}_k(\mathrm{NP}) &= \mathrm{BH}_{k-2}(\mathrm{NP}) \vee \mathrm{BH}_2(\mathrm{NP}) \quad \textit{for } k \geq 3, \textit{ and} \\ \mathrm{BH}(\mathrm{NP}) &= \bigcup_{k \geq 1} \mathrm{BH}_k(\mathrm{NP}). \end{aligned}$$

Note that $\mathrm{DP} = \mathrm{BH}_2(\mathrm{NP})$. In his seminal paper [Wag87], Wagner provided a set of conditions sufficient to prove hardness results for the levels of the boolean hierarchy over NP and for other complexity classes. His sufficient conditions were successfully applied to classify the complexity of a variety of natural, important problems, see, e.g., [Wag87,HHR97a,HHR97b,HR98,Rot03,HRS02,RSV03]. Below, we state one of Wagner's sufficient conditions that is relevant for this paper; see Theorem 5.1(3) in [Wag87].

Lemma 4 (Wagner) *Let A be some* NP*-complete problem, let B be an arbitrary problem, and let $k \geq 1$ be fixed. If there exists a polynomial-time computable function f such that the equivalence*

$$||\{i \mid x_i \in A\}|| \text{ is odd} \iff f(x_1, x_2, \ldots, x_{2k}) \in B \tag{2.1}$$

is true for all strings $x_1, x_2, \ldots, x_{2k} \in \Sigma^*$ satisfying that for each j with $1 \leq j < 2k$, $x_{j+1} \in A$ implies $x_j \in A$, then B is $\mathrm{BH}_{2k}(\mathrm{NP})$*-hard.*

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$ denote the set of nonnegative integers, and let $\mathbb{N}^+ = \{1, 2, 3, \ldots\}$ denote the set of positive integers. We now define the exact versions of the domatic number problem, parameterized by k-element sets $M_k \subseteq \mathbb{N}$ of noncontiguous integers.

Definition 5 *Given any set $M_k \subseteq \mathbb{N}$ containing k noncontiguous integers, define the problem*

$$\texttt{Exact-}M_k\texttt{-DNP} = \{G \mid G \text{ is a graph and } \delta(G) \in M_k\}.$$

In particular, for each singleton $M_1 = \{t\}$, we write $\texttt{Exact-}t\texttt{-DNP} = \{G \mid \delta(G) = t\}$.

Note that if some elements of M_k were contiguous, one might encode problems of lower complexity. For instance, if M_k happens to be just one interval of k contiguous integers, $\texttt{Exact-}M_k\texttt{-DNP}$ in fact is contained in DP, whereas $\texttt{Exact-}M_k\texttt{-DNP}$ will be shown to be $\mathrm{BH}_{2k}(\mathrm{NP})$-complete in Theorem 26 if M_k is a set of k sufficiently large noncontiguous integers.

To apply Wagner's sufficient condition from Lemma 4 in the proof of the main result of this paper, Theorem 13 in Section 4.2.1, we need the following lemma due to Kaplan and Shamir [KS94] that gives a reduction from 3-`Colorability` to `DNP` with useful properties. Since Kaplan and Shamir's construction will be used explicitly in the proofs of Theorems 13 and 26, we present it below.

Lemma 6 (Kaplan and Shamir) *There exists a polynomial-time many-one reduction g from* 3-`Colorability` *to* `DNP` *with the following properties:*

$$G \in 3\text{-}\texttt{Colorability} \implies \delta(g(G)) = 3; \tag{2.2}$$

$$G \notin 3\text{-}\texttt{Colorability} \implies \delta(g(G)) = 2. \tag{2.3}$$

Proof. The reduction g maps any given graph G to a graph H such that the implications (2.2) and (2.3) are satisfied. Since it can be tested in polynomial time whether or not a given graph is 2-colorable, we may assume, without loss of generality, that G is not 2-colorable. Recall that we also assume that G has no isolated vertices; note that the domatic number of any graph is always at least 2 if it has no isolated vertices (cf. [GJ79]). Graph H is constructed from G by creating $||E(G)||$ new vertices, one on each edge of G, and by adding new edges such that the original vertices of G form a clique. Thus, every edge of G induces a triangle in H, and every pair of nonadjacent vertices in G is connected by an edge in H. The proofs of upcoming Theorems 13 and 26 explicitly use this construction and such triangles, see Figure 1.

Let $V(G) = \{v_1, v_2, \ldots, v_n\}$. Formally, define the vertex set and the edge set of H by:

$$\begin{aligned} V(H) &= V(G) \cup \{u_{i,j} \mid \{v_i, v_j\} \in E(G)\}; \\ E(H) &= \{\{v_i, u_{i,j}\} \mid \{v_i, v_j\} \in E(G)\} \cup \{\{v_j, u_{i,j}\} \mid \{v_i, v_j\} \in E(G)\} \\ &\quad \cup \{\{v_i, v_j\} \mid 1 \leq i, j \leq n \text{ and } i \neq j\}\}. \end{aligned}$$

Since, by construction, *min-deg*$(H) = 2$ and H has no isolated vertices, the inequality $\delta(H) \leq$ *min-deg*$(H) + 1$ implies that $2 \leq \delta(H) \leq 3$.

Suppose $G \in$ 3-`Colorability`. Let C_1, C_2, and C_3 be the three color classes of G, i.e., $C_k = \{v_i \in V(G) \mid v_i \text{ is colored by color } k\}$, for $k \in \{1, 2, 3\}$. Form a partition of $V(H)$ by $\hat{C}_k = C_k \cup \{u_{i,j} \mid v_i \notin C_k \text{ and } v_j \notin C_k\}$, for $k \in \{1, 2, 3\}$. Since for each k, $\hat{C}_k \cap V(G) \neq \emptyset$ and $V(G)$ induces a clique in H, every $\hat{C}_k$ dominates $V(G)$ in H. Also, every triangle $\{v_i, u_{i,j}, v_j\}$ contains one element from each $\hat{C}_k$, so every $\hat{C}_k$ also dominates $\{u_{i,j} \mid \{v_i, v_j\} \in E(G)\}$ in H. Hence, $\delta(H) = 3$, which proves the implication (2.2).

Conversely, suppose $\delta(H) = 3$. Given a partition of $V(H)$ into three dominating sets, $\hat{C}_1$, $\hat{C}_2$, and $\hat{C}_3$, color the vertices in $\hat{C}_k$ by color k. Every triangle $\{v_i, u_{i,j}, v_j\}$ is 3-colored, which implies that this coloring on $V(G)$ induces a legal 3-coloring of G; so $G \in$ 3-`Colorability`. Hence, $\chi(G) = 3$ if and only if $\delta(H) = 3$. Since $2 \leq \delta(H) \leq 3$, the implication (2.3) follows. ∎

We now define two well-known problems that will be used later in our reductions.

Definition 7 *Let $X = \{x_1, x_2, \ldots, x_n\}$ be a finite set of variables.*

- 1-3-`SAT` ("one-in-three satisfiability"): *Let H be a boolean formula consisting of a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ of m sets of literals over X such that each S_i has exactly three members. H is in* 1-3-`SAT` *if and only if there exists a subset T of the literals over X with $||T \cap S_i|| = 1$ for each i, $1 \leq i \leq m$.*

- `NAE-3-SAT` ("not-all-equal satisfiability"): *Let H be a boolean formula consisting of a collection $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ of m clauses over X such that each c_i contains exactly three literals. H is in* `NAE-3-SAT` *if and only if there exists a truth assignment* *for X that satisfies all clauses in $\mathcal{C}$ and such that in none of the clauses, all literals are true.*

Both problems were shown to be NP-complete by Schaefer [Sch78]. Note that 1-3-`SAT` remains NP-complete even if all literals are positive.

3 A General Framework for Dominating Set Problems

Heggernes and Telle [HT98] proposed a general, uniform approach to define graph problems by partitioning the vertex set of a graph into generalized dominating sets. Generalized dominating sets are parameterized by two sets of nonnegative integers, σ and ρ, which restrict the number of neighbors for each vertex in the partition. We adopt this approach in defining the exact versions of such generalized dominating set problems. Their computational complexity will be studied in Section 4.

We now define the notions of (σ, ρ)-sets and (k, σ, ρ)-partitions introduced by Heggernes and Telle [HT98].

Definition 8 (Heggernes and Telle) *Let G be a given graph, let $\sigma \subseteq \mathbb{N}$ and $\rho \subseteq \mathbb{N}$ be given sets, and let $k \in \mathbb{N}^{+}$.*

1. *A subset $U \subseteq V(G)$ of the vertices of G is said to be a (σ, ρ)-*set* if and only if for each $u \in U$, $||N(u) \cap U|| \in \sigma$, and for each $u \notin U$, $||N(u) \cap U|| \in \rho$.*
2. *A (k, σ, ρ)-*partition of G *is a partition of $V(G)$ into k pairwise disjoint subsets $V_1, V_2, \ldots, V_k$ such that V_i is a (σ, ρ)-set for each i, $1 \leq i \leq k$.*
3. *Define the problem*

$$(k, \sigma, \rho)\text{-}\mathtt{Partition} \;=\; \{G \mid G \textit{ is a graph that has a } (k, \sigma, \rho)\textit{-partition}\}.$$

Heggernes and Telle [HT98] examined the (k, σ, ρ)-partitions of graphs for the parameters σ and ρ chosen among $\{0\}$, $\{1\}$, $\{0, 1\}$, $\mathbb{N}$, and $\mathbb{N}^{+}$. In particular, they determined the precise cut-off points between tractability and intractability for these problems. That is, they determined the precise value of k for which the resulting (k, σ, ρ)-`Partition` problem is NP-complete, yet it can be decided in polynomial time whether or not a given graph has a $(k-1, \sigma, \rho)$-partition. An overview of their (and previously known) results is given in Table 1.

For example, $(3, \mathbb{N}, \mathbb{N}^{+})$-`Partition` is nothing else than the NP-complete domatic number problem: Given a graph G, decide whether or not G can be partitioned into three dominating sets. In contrast, $(2, \mathbb{N}, \mathbb{N}^{+})$-`Partition` is in P, and therefore the corresponding entry in Table 1 is 3 for $\sigma = \mathbb{N}$ and $\rho = \mathbb{N}^{+}$. A value of ∞ in Table 1 means that this problem is efficiently solvable for all values of k. The value of $\rho = \{0\}$

ρ / σ	$\mathbb{N}$	$\mathbb{N}^+$	$\{1\}$	$\{0,1\}$
$\mathbb{N}$	∞^-	3^+	2	∞^-
$\mathbb{N}^+$	∞^-	2^+	2	∞^-
$\{1\}$	2^-	2	3	3^-
$\{0,1\}$	2^-	2	3	3^-
$\{0\}$	3^-	3	4	4^-

Table 1: NP-completeness for the problems (k, σ, ρ)-`Partition`.

is not considered, since all graphs have a $(k, \sigma, \{0\})$-partition if and only if they have the trivial partition into k disjoint $(\sigma, \{0\})$-sets $V_1 = V(G)$ and $V_i = \emptyset$, for each $i \in \{2, \ldots, k\}$.

Definition 9 *Let σ and ρ be sets that are chosen among $\mathbb{N}$, $\mathbb{N}^+$, $\{0\}$, $\{0,1\}$, and $\{1\}$, and let $k \in \mathbb{N}^+$. We say that (k, σ, ρ)-`Partition` is a* minimum problem *if and only if (k, σ, ρ)-`Partition` $\subseteq$ $(k+1, \sigma, \rho)$-`Partition` for each $k \geq 1$, and we say that (k, σ, ρ)-`Partition` is a* maximum problem *if and only if $(k+1, \sigma, \rho)$-`Partition` $\subseteq$ (k, σ, ρ)-`Partition` for each $k \geq 1$.*

The problems in Table 1 that are marked by a "+" are maximum problems, and the problems that are marked by a "−" are minimum problems in the above sense. These properties are stated in the following fact.

Fact 10 *1. For each $k \geq 1$, for each $\sigma \in \{\mathbb{N}, \mathbb{N}^+, \{0\}, \{0,1\}, \{1\}\}$, and for each $\rho \in \{\mathbb{N}, \{0,1\}\}$, it holds that (k, σ, ρ)-`Partition` $\subseteq (k+1, \sigma, \rho)$-`Partition`.*

2. For each $k \geq 1$ and for each $\sigma \in \{\mathbb{N}, \mathbb{N}^+\}$, it holds that $(k+1, \sigma, \mathbb{N}^+)$-`Partition` $\subseteq$ $(k, \sigma, \mathbb{N}^+)$-`Partition`.

Proof. To see that all (k, σ, ρ)-`Partition` problems with $\rho = \mathbb{N}$ are minimum problems, note that we obtain a $(k+1, \sigma, \mathbb{N})$-partition from a $(k, \sigma, \mathbb{N})$-partition by simply adding the empty set $V_{k+1} = \emptyset$. The proof for the case $\rho = \{0,1\}$ is analogous.

To prove that the (k, σ, ρ)-`Partition` problems with $\rho = \mathbb{N}^+$ are maximum problems, note that once we have found a $(k+1, \sigma, \mathbb{N}^+)$-partition into $k+1$ pairwise disjoint sets $V_1, V_2, \ldots, V_{k+1}$, the sets $V_1, V_2, \ldots, V_{k-1}, \tilde{V}_k$ with $\tilde{V}_k = V_k \cup V_{k+1}$ are a $(k, \sigma, \mathbb{N}^+)$-partition as well. ∎

Observe that those problems in Table 1 that are marked neither by a "+" nor by a "−" are neither maximum nor minimum problems in the sense defined above. That is, we have neither $(k+1, \sigma, \rho)$-`Partition` $\subseteq$ (k, σ, ρ)-`Partition` nor (k, σ, ρ)-`Partition` $\subseteq$ $(k+1, \sigma, \rho)$-`Partition`, since for each $k \geq 1$, there exist graphs G such that G is in (k, σ, ρ)-`Partition` but G is not in (ℓ, σ, ρ)-`Partition` for any $\ell \geq 1$ with $\ell \neq k$.

For example, consider $(k, \{1\}, \{1\})$-`Partition`. By definition, this problem contains all graphs G that can be partitioned into k subsets $V_1, V_2, \ldots, V_k$ such that, for each i, if $v \in V_i$ then $||N(v) \cap V_i|| = 1$, and if $v \notin V_i$ then $||N(v) \cap V_i|| = 1$. It follows that every graph in $(k, \{1\}, \{1\})$-`Partition` must be k-regular; that is, every vertex has degree k. Hence, for all $k \geq 1$, $(k, \{1\}, \{1\})$-`Partition` and $(k+1, \{1\}, \{1\})$-`Partition` are disjoint, so neither $(k, \{1\}, \{1\})$-`Partition` $\subseteq$ $(k+1, \{1\}, \{1\})$-`Partition` nor $(k+1, \{1\}, \{1\})$-`Partition` $\subseteq (k, \{1\}, \{1\})$-`Partition`.

In the case of $(k, \{0\}, \mathbb{N}^{+})$-`Partition`, the complete graph K_n with n vertices is in $(n, \{0\}, \mathbb{N}^{+})$-`Partition` but not in $(k, \{0\}, \mathbb{N}^{+})$-`Partition` for any $k \geq 1$ with $k \neq n$. Almost the same argument applies to the case $\sigma = \mathbb{N}$ and $\rho = \{1\}$, except that now K_n is in $(k, \mathbb{N}, \{1\})$-`Partition` for $k \in \{1, n\}$ but not in $(\ell, \mathbb{N}, \{1\})$-`Partition` for any $\ell \geq 1$ with $\ell \notin \{1, n\}$. Similar arguments work in the other cases.

Therefore, when defining the exact versions of generalized dominating set problems, we confine ourselves to those (k, σ, ρ)-`Partition` problems that are minimum or maximum problems in the above sense. For a maximum problem, its exact version asks whether $G \in (k, \sigma, \rho)$-`Partition` but $G \notin (k+1, \sigma, \rho)$-`Partition`, and for a minimum problem, its exact version asks whether $G \in (k, \sigma, \rho)$-`Partition` but $G \notin (k-1, \sigma, \rho)$-`Partition`.

Definition 11 *Let σ and ρ be sets that are chosen among $\mathbb{N}$, $\mathbb{N}^{+}$, $\{0\}$, $\{0,1\}$, and $\{1\}$, and let $k \in \mathbb{N}^{+}$. Define the* exact version of (k, σ, ρ)-`Partition` *by*

$$\texttt{Exact-}(k,\sigma,\rho)\texttt{-Partition} = \begin{cases} (k,\sigma,\rho)\texttt{-Partition} \cap \overline{(k-1,\sigma,\rho)\texttt{-Partition}} \\ \qquad \text{if } k \geq 2 \text{ and } (k,\sigma,\rho)\texttt{-Partition} \\ \qquad \text{is a minimum problem} \\ (k,\sigma,\rho)\texttt{-Partition} \cap \overline{(k+1,\sigma,\rho)\texttt{-Partition}} \\ \qquad \text{if } k \geq 1 \text{ and } (k,\sigma,\rho)\texttt{-Partition} \\ \qquad \text{is a maximum problem.} \end{cases}$$

For example, the problem $(k, \{0\}, \mathbb{N})$-`Partition` is equal to the k-colorability problem, which is a minimization problem: Given a graph G, find a partition into at most k color classes such that any two adjacent vertices belong to distinct color classes. In contrast, $(k, \mathbb{N}, \mathbb{N}^{+})$-`Partition` is equal to `DNP`, the domatic number problem, which is a maximization problem.

Clearly, since (k, σ, ρ)-`Partition` is in NP, the problems defined in Definition 11 above are contained in DP. This fact is needed for the DP-completeness results in Section 4.

Fact 12 `Exact-`(k, σ, ρ)-`Partition` *is in* DP.

4 Exact Generalized Dominating Set Problems

4.1 Overview of the Results

In this section, we prove DP-completeness for a number of problems defined in Section 3. Our results from Sections 4.2 and 4.3 are summarized in Table 2.

σ \ ρ	$\mathbb{N}$	$\mathbb{N}^+$
$\mathbb{N}$	∞	5^*
$\mathbb{N}^+$	∞	3^*
$\{1\}$	5^*	–
$\{0,1\}$	5^*	–
$\{0\}$	4	–

Table 2: DP-completeness for the problems $\texttt{Exact-}(k,\sigma,\rho)\texttt{-Partition}$.

The numbers in Table 2 indicate the best DP-completeness results currently known for the exact versions of generalized dominating set problems, where the results from this paper are marked by an asterisk.[1] That is, they give the best value of k for which the problem $\texttt{Exact-}(k,\sigma,\rho)\texttt{-Partition}$ is known to be DP-complete. In some cases this value is not yet optimal. For example, $\texttt{Exact-}(5,\mathbb{N},\mathbb{N}^+)\texttt{-Partition}$ is known to be DP-complete and $\texttt{Exact-}(2,\mathbb{N},\mathbb{N}^+)\texttt{-Partition}$ is known to be coNP-complete. What about $\texttt{Exact-}(3,\mathbb{N},\mathbb{N}^+)\texttt{-Partition}$ and $\texttt{Exact-}(4,\mathbb{N},\mathbb{N}^+)\texttt{-Partition}$? Only the DP-completeness of $\texttt{Exact-}(4,\{0\},\mathbb{N})\texttt{-Partition}$ is known to be optimal [Rot03].

The results stated in Table 2 can easily be extended to more general results involving slightly more general problems complete in the higher levels of the boolean hierarchy and in the class $\mathrm{P}_{||}^{\mathrm{NP}}$, respectively. These results are presented in Sections 4.4 and 4.5.

4.2 The Case $\rho = \mathbb{N}^+$

For $\rho = \mathbb{N}^+$, we consider the cases $\sigma = \mathbb{N}$ and $\sigma = \mathbb{N}^+$ only. The corresponding two problems are the only maximum problems in Table 1.

Recall that since $(k,\mathbb{N},\mathbb{N}^+)\texttt{-Partition}$ and $(k,\mathbb{N}^+,\mathbb{N}^+)\texttt{-Partition}$ are maximum problems, their exact versions are defined as follows:

$$\texttt{Exact-}(k,\sigma,\mathbb{N}^+)\texttt{-Partition} = \left\{ G \;\middle|\; \begin{array}{l} G \in (k,\sigma,\mathbb{N}^+)\texttt{-Partition and} \\ G \notin (k+1,\sigma,\mathbb{N}^+)\texttt{-Partition} \end{array} \right\},$$

where $\sigma \in \{\mathbb{N},\mathbb{N}^+\}$.

[1] Again, a value of ∞ in Table 2 means that this problem is efficiently solvable for all values of k.

4.2.1 The Case $\sigma = \mathbb{N}$ and $\rho = \mathbb{N}^+$

Recall that the problem $(k, \mathbb{N}, \mathbb{N}^+)$-`Partition` is equal to `DNP`, the domatic number problem. Consequently, its exact version `Exact`-$(k, \mathbb{N}, \mathbb{N}^+)$-`Partition` is just the problem `Exact`-k-`DNP`.

Theorem 13 *For each $i \geq 5$,* `Exact`-i-`DNP` *is* DP-*complete.*

Proof. It is enough to prove the theorem for $i = 5$. By Fact 12, `Exact`-5-`DNP` is contained in DP. The proof that `Exact`-5-`DNP` is DP-hard draws on Lemma 4 with $k = 1$ being fixed, with 3-`Colorability` being the NP-complete set A, and with `Exact`-5-`DNP` being the set B from this lemma.

Fix any two graphs, G_1 and G_2, satisfying that if G_2 is in 3-`Colorability`, then so is G_1. Without loss of generality, we assume that none of these two graphs is 2-colorable, nor does it contain isolated vertices. Moreover, we may assume that $\chi(G_j) \leq 4$ for each $j \in \{1, 2\}$, without loss of generality, since the standard reduction from 3-`SAT` to 3-`Colorability` (cf. [GJ79]) maps each satisfiable formula to a graph G with $\chi(G) = 3$, and it maps each unsatisfiable formula to a graph G with $\chi(G) = 4$.

We now define a polynomial-time computable function f that maps the graphs G_1 and G_2 to a graph $H = f(G_1, G_2)$ such that the equivalence from Lemma 4 is satisfied. Applying the Lemma 6 reduction g from 3-`Colorability` to `DNP`, we obtain two graphs, $H_1 = g(G_1)$ and $H_2 = g(G_2)$, each satisfying the implications from Lemma 6. Hence, both $\delta(H_1)$ and $\delta(H_2)$ is in $\{2, 3\}$, and $\delta(H_2) = 3$ implies $\delta(H_1) = 3$. The graph H is constructed from the graphs H_1 and H_2 such that

$$\delta(H) = \delta(H_1) + \delta(H_2), \tag{4.4}$$

which implies that f satisfies Equation (2.1) from Lemma 4:

$$\begin{aligned}
& G_1 \in \texttt{3-Colorability} \text{ and } G_2 \notin \texttt{3-Colorability} \\
\iff\ & \delta(H_1) = 3 \text{ and } \delta(H_2) = 2 \\
\iff\ & \delta(H) = \delta(H_1) + \delta(H_2) = 5 \\
\iff\ & f(G_1, G_2) = H \in \texttt{Exact-5-DNP}.
\end{aligned}$$

Applying Lemma 4 with $k = 1$, it follows that `Exact`-5-`DNP` is DP-complete.

We now prove Equation (4.4). Note that the analogous property for the chromatic number (i.e., $\chi(H) = \chi(H_1) + \chi(H_2)$) is easy to achieve by simply joining the graphs H_1 and H_2 ([Wag87], see also [Rot03]). However, for the domatic number, the construction is more complicated. Construct a gadget connecting H_1 and H_2 as follows. Recalling the construction from Lemma 6, for each edge $\{v_i, v_j\}$, a new vertex $u_{i,j}$ and two new edges, $\{v_i, u_{i,j}\}$ and $\{u_{i,j}, v_j\}$, are created. Further edges are added such that the original vertices in G form a clique. Thus, every edge of G induces a triangle in $H = g(G)$, and every pair of nonadjacent vertices in G is connected by an edge in H. Let T_1 with $V(T_1) = \{v_q, u_{q,r}, v_r\}$ be any fixed triangle in H_1, and let T_2 with $V(T_2) = \{v_s, u_{s,t}, v_t\}$ be any fixed triangle in H_2. Connect T_1 and T_2 using the gadget shown in Figure 1, where $a_1, a_2, \dots, a_6$ are new vertices. Using pairwise disjoint copies of the gadget from Figure 1, connect each pair of triangles from H_1 and H_2 and call the resulting graph H. Note that f is polynomial-time computable.

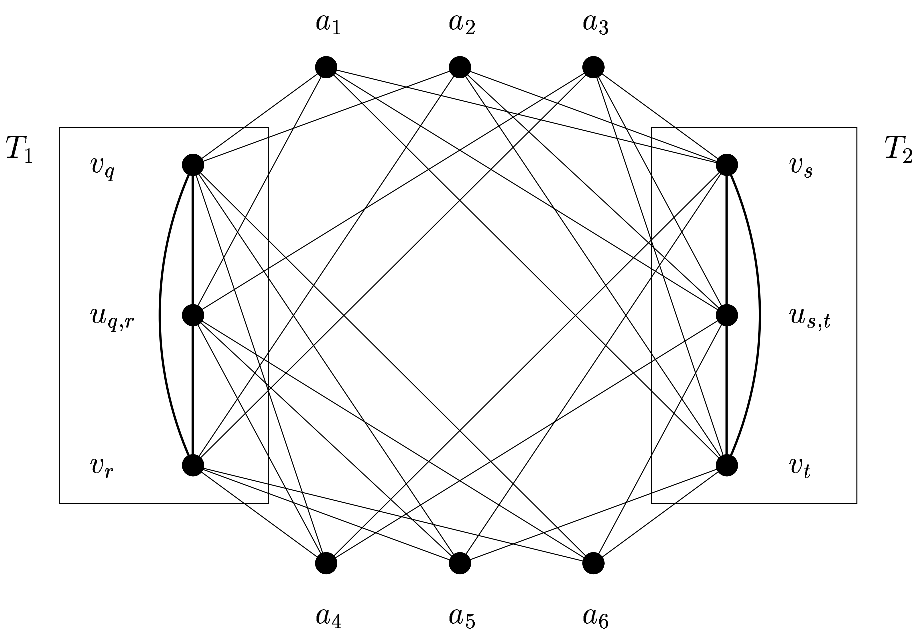

Figure 1: Gadget connecting two triangles T_1 and T_2.

Since *deg*$(a_i) = 5$ for each gadget vertex a_i, we have $\delta(H) \leq 6$, regardless of whether the domatic numbers of H_1 and H_2 are 2 or 3. We now show that $\delta(H) = \delta(H_1) + \delta(H_2)$. Let $D_1, D_2, \dots, D_{\delta(H_1)}$ be $\delta(H_1)$ pairwise disjoint sets dominating H_1, and let $D_{\delta(H_1)+1}$, $D_{\delta(H_1)+2}$, …, $D_{\delta(H_1)+\delta(H_2)}$ be $\delta(H_2)$ pairwise disjoint sets dominating H_2. Distinguish the following three cases.

Case 1: $\boldsymbol{\delta(H_1) = \delta(H_2) = 3}$. Consider any fixed D_j, where $1 \leq j \leq 3$. Since D_j dominates H_1, every triangle T_1 of H_1 has exactly one vertex in D_j. Fix T_1, and suppose $V(T_1) = \{v_q, u_{q,r}, v_r\}$ and, say, $V(T_1) \cap D_j = \{v_q\}$; the other cases are analogous. For each triangle T_2 of H_2, say T_2 with $V(T_2) = \{v_s, u_{s,t}, v_t\}$, let $a_1^{T_2}, a_2^{T_2}, \dots, a_6^{T_2}$ be the gadget vertices connecting T_1 and T_2 as in Figure 1. Note that exactly one of these gadget vertices, $a_3^{T_2}$, is not adjacent to v_q. For each triangle T_2, add the missing gadget vertex to D_j, and define $\hat{D}_j = D_j \cup$

$\{a_3^{T_2} \mid T_2$ is a triangle of $H_2\}$. Since every vertex of H_2 is contained in some triangle T_2 of H_2 and since $a_3^{T_2}$ is adjacent to each vertex in T_2, $\hat{D}_j$ dominates H_2. Also, $\hat{D}_j \supseteq D_j$ dominates H_1, and since v_q is adjacent to each $a_i^{T_2}$ except $a_3^{T_2}$ for each triangle T_2 of H_2, $\hat{D}_j$ dominates every gadget vertex of H. Hence, $\hat{D}_j$ dominates H. By a symmetric argument, every set D_j, where $4 \leq j \leq 6$, dominating H_2 can be extended to a set $\hat{D}_j$ dominating the entire graph H. By construction, the sets $\hat{D}_j$ with $1 \leq j \leq 6$ are pairwise disjoint. Hence, $\delta(H) = 6 = \delta(H_1) + \delta(H_2)$.

Case 2: $\boldsymbol{\delta(H_1) = 3}$ **and** $\boldsymbol{\delta(H_2) = 2}$. As in Case 1, we can add appropriate gadget vertices to the five given sets $D_1, D_2, \ldots, D_5$ to obtain five pairwise disjoint sets $\hat{D}_1, \hat{D}_2, \ldots, \hat{D}_5$ such that each $\hat{D}_i$ dominates the entire graph H. It follows that $5 \leq \delta(H) \leq 6$. It remains to show that $\delta(H) \neq 6$. For a contradiction, suppose that $\delta(H) = 6$. Look at Figure 1 showing the gadget between any two triangles T_1 and T_2 belonging to H_1 and H_2, respectively. Fix T_1 with $V(T_1) = \{v_q, u_{q,r}, v_r\}$. The only way (except for renaming the dominating sets) to partition the graph H into six dominating sets, say $E_1, E_2, \ldots, E_6$, is to assign to the sets E_i the vertices of T_1, of H_2, and of the gadgets connected with T_1 as follows:

- E_1 contains v_q and the set $\{a_3^{T_2} \mid T_2$ is a triangle in $H_2\}$,
- E_2 contains $u_{q,r}$ and the set $\{a_2^{T_2} \mid T_2$ is a triangle in $H_2\}$,
- E_3 contains v_r and the set $\{a_1^{T_2} \mid T_2$ is a triangle in $H_2\}$,
- E_4 contains $v_s \in T_2$, for each triangle T_2 of H_2, and the set
$$\{a_6^{T_2} \mid T_2 \text{ is a triangle in } H_2\},$$
- E_5 contains $u_{s,t} \in T_2$, for each triangle T_2 of H_2, and the set
$$\{a_5^{T_2} \mid T_2 \text{ is a triangle in } H_2\},$$
- E_6 contains $v_t \in T_2$, for each triangle T_2 of H_2, and the set
$$\{a_4^{T_2} \mid T_2 \text{ is a triangle in } H_2\}.$$

Hence, all vertices from H_2 must be assigned to the three dominating sets E_4, E_5, and E_6, which induces a partition of H_2 into three dominating sets. This contradicts the case assumption that $\delta(H_2) = 2$. Hence, $\delta(H) = 5 = \delta(H_1) + \delta(H_2)$.

Case 3: $\boldsymbol{\delta(H_1) = \delta(H_2) = 2}$. As in the previous two cases, we can add appropriate gadget vertices to the four given sets D_1, D_2, D_3, and D_4 to obtain a partition of $V(H)$ into four sets $\hat{D}_1$, $\hat{D}_2$, $\hat{D}_3$, and $\hat{D}_4$ such that each $\hat{D}_i$ dominates the entire graph H. It follows that $4 \leq \delta(H) \leq 6$. By the same arguments as in Case 2,

$\delta(H) \neq 6$. It remains to show that $\delta(H) \neq 5$. For a contradiction, suppose that $\delta(H) = 5$. Look at Figure 1 showing the gadget between any two triangles T_1 and T_2 belonging to H_1 and H_2, respectively. Suppose H is partitioned into five dominant sets $E_1, E_2, \ldots, E_5$.

First, we show that neither T_1 nor T_2 can have two vertices belonging to the same dominating set. Suppose otherwise, and let, for example, v_q and $u_{q,r}$ be both in E_1, and let v_r be in E_2; all other cases are treated analogously. This implies that the vertices v_s, $u_{s,t}$, and v_t in T_2 must be assigned to the other three dominating sets, E_3, E_4, and E_5, since otherwise one of the sets E_i would not dominate all gadget vertices a_j, $1 \leq j \leq 6$. Since T_1 is connected with each triangle of H_2 via some gadget, the same argument shows that $V(H_2)$ can be partitioned into three dominating sets, which contradicts the assumption that $\delta(H_2) = 2$.

Hence, the vertices of T_1 are assigned to three different dominating sets, say E_1, E_2, and E_3. Then, every triangle T_2 of H_2 must have one of its vertices in E_4, one in E_5, and one in either one of E_1, E_2, and E_3. Again, this induces a partition of H_2 into three dominating sets, which contradicts the assumption that $\delta(H_2) = 2$. It follows that $\delta(H) \neq 5$, so $\delta(H) = 4 = \delta(H_1) + \delta(H_2)$.

By construction, $\delta(H_2) = 3$ implies $\delta(H_1) = 3$, and thus the case "$\delta(H_1) = 2$ and $\delta(H_2) = 3$" cannot occur. The case distinction is complete, which proves Equation (4.4) and the theorem. ∎

In contrast to Theorem 13, Exact-2-DNP is in coNP (and even coNP-complete) and thus cannot be DP-complete unless the boolean hierarchy over NP collapses.

Theorem 14 Exact-2-DNP *is* coNP-*complete.*

Proof. The problem Exact-2-DNP can be written as

$$\texttt{Exact-2-DNP} = \{G \mid \delta(G) \leq 2\} \cap \{G \mid \delta(G) \geq 2\}.$$

Since every graph without isolated vertices has a domatic number of at least 2 (cf. [GJ79]), the set $\{G \mid \delta(G) \geq 2\}$ is in P. On the other hand, the set $\{G \mid \delta(G) \leq 2\}$ is in coNP, so Exact-2-DNP is also in coNP and, thus, cannot be DP-complete unless the boolean hierarchy over NP collapses to its first level. Note that the coNP-hardness of Exact-2-DNP follows immediately via the Lemma 6 reduction g from 3-Colorability to DNP. ∎

4.2.2 The Case $\sigma = \mathbb{N}^+$ and $\rho = \mathbb{N}^+$

Definition 15 *For every graph G, define the maximum value k for which G has a $(k, \mathbb{N}^+, \mathbb{N}^+)$-partition as follows:*

$$\gamma(G) = \max\{k \in \mathbb{N}^+ \mid G \in (k, \mathbb{N}^+, \mathbb{N}^+)\text{-}\texttt{Partition}\}.$$

Theorem 16 *For each $i \geq 3$,* $\texttt{Exact-}(i, \mathbb{N}^+, \mathbb{N}^+)\texttt{-Partition}$ *is* DP-*complete.*

Proof. Again, it is enough to prove the theorem for the case $i = 3$. By Fact 12, $\texttt{Exact-}(3, \mathbb{N}^+, \mathbb{N}^+)\texttt{-Partition}$ is contained in DP. We now prove that $\texttt{Exact-5-DNP}$ is DP-hard.

Heggernes and Telle [HT98] presented a reduction from the problem $\texttt{NAE-3-SAT}$ to the problem $(2, \mathbb{N}^+, \mathbb{N}^+)\texttt{-Partition}$ to prove the latter problem NP-complete. We modify their reduction as follows. Let two boolean formulas $H_1 = (X, \hat{C})$ and $H_2 = (Y, \hat{D})$ be given, with disjoint variable sets, $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_r\}$, and with disjoint clause sets, $\hat{C} = \{c_1, c_2, \ldots, c_m\}$ and $\hat{D} = \{d_1, d_2, \ldots, d_s\}$. If the variable sets consist of less than two variables, we put additional variables into the sets. Moreover, we may assume, without loss of generality, that every literal appears in at least one clause, since otherwise we can easily alter the given formulas H_1 and H_2, without changing membership in $\texttt{NAE-3-SAT}$, so that they are of this form.

For any clause $c = (x \vee y \vee z)$, define $\check{c} = (\overline{x} \vee \overline{y} \vee \overline{z})$, where $\overline{x}$, $\overline{y}$, and $\overline{z}$, respectively, denotes the negation of the literal x, y, and z. Define $\check{C} = \{\check{c}_1, \check{c}_2, \ldots, \check{c}_m\}$ and $\check{D} = \{\check{d}_1, \check{d}_2, \ldots, \check{d}_s\}$, and define $C = \hat{C} \cup \check{C}$ and $D = \hat{D} \cup \check{D}$. Note that due to the not-all-equal property, we have:

$$\begin{aligned} (X, C) \in \texttt{NAE-3-SAT} &\iff (X, \hat{C}) \in \texttt{NAE-3-SAT} \\ &\iff (X, \check{C}) \in \texttt{NAE-3-SAT} \end{aligned}$$

and

$$\begin{aligned} (Y, D) \in \texttt{NAE-3-SAT} &\iff (Y, \hat{D}) \in \texttt{NAE-3-SAT} \\ &\iff (Y, \check{D}) \in \texttt{NAE-3-SAT}. \end{aligned}$$

We apply Lemma 4 with $k = 1$ being fixed, with $\texttt{NAE-3-SAT}$ being the NP-complete problem A, and with $\texttt{Exact-}(3, \mathbb{N}^+, \mathbb{N}^+)\texttt{-Partition}$ being the set B from this lemma. Let H_1 and H_2 be such that $H_2 \in \texttt{NAE-3-SAT}$ implies $H_1 \in \texttt{NAE-3-SAT}$. Our polynomial-time reduction f transforms H_1 and H_2 into a graph $G = f(H_1, H_2)$ with the property:

$$(H_1 \in \texttt{NAE-3-SAT} \wedge H_2 \notin \texttt{NAE-3-SAT}) \iff \gamma(G) = 3. \tag{4.5}$$

The reduction f is defined as follows. For H_1, we create an 8-clique A_8 with vertices a_1, a_2, …, a_8. We do the same for H_2, creating an 8-clique B_8 with vertices b_1, b_2, …, b_8. For each i with $1 \leq i \leq n$, we create two vertices, x_i and $\overline{x}_i$, for the variable x_i. For each j with $1 \leq j \leq r$, we create two vertices, y_j and $\overline{y}_j$, for the variable y_j. Every vertex x_i and $\overline{x}_i$ is connected to both a_1 and a_2, and every vertex y_j and $\overline{y}_j$ is connected to both b_1 and b_2. For each pair of variables $\{x_i, y_j\}$, we create one vertex $u_{i,j}$ that is connected to the four vertices x_i, $\overline{x}_i$, y_j, and $\overline{y}_j$. Finally, for each clause $c_i \in C$ and $d_j \in D$ with $1 \leq i \leq m$ and

$1 \leq j \leq s$, we create the two vertices c_i and d_j. Each such clause vertex is connected to the vertices representing the literals in that clause. Additionally, every vertex c_i is connected to both a_1 and a_2, and every vertex d_j is connected to both b_1 and b_2. This completes the construction of the graph $G = f(H_1, H_2)$.

Figure 2 shows the graph G resulting from the reduction f applied to the two formulas

$$\begin{aligned} H_1 &= (x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee x_3) \quad \text{and} \\ H_2 &= (y_1 \vee y_2 \vee y_3) \wedge (\overline{y}_1 \vee \overline{y}_2 \vee \overline{y}_3). \end{aligned}$$

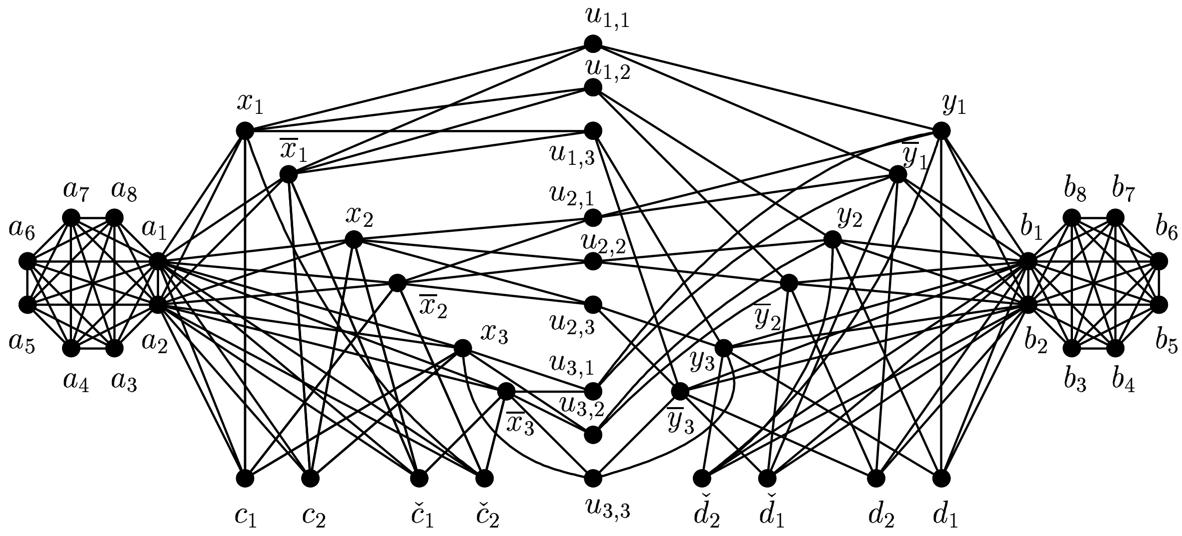

Figure 2: `Exact-`$(3, \mathbb{N}^+, \mathbb{N}^+)$`-Partition` is DP-complete: Graph $G = f(H_1, H_2)$.

Note that $\gamma(G) \leq 4$, since the degree of each $u_{i,j}$ is four. We have three cases to distinguish.

Case 1: $\boldsymbol{H_1} \in$ `NAE-3-SAT` **and** $\boldsymbol{H_2} \in$ `NAE-3-SAT`. Let t be a truth assignment satisfying H_1, and let $\tilde{t}$ be a truth assignment satisfying H_2. We can partition G into four $(\mathbb{N}^+, \mathbb{N}^+)$-sets V_1, V_2, V_3, and V_4 as follows:

$$\begin{aligned} V_1 &= \hat{C} \cup \check{C} \cup \{a_5, a_6\} \cup \{b_1, b_3\} \cup \{x \mid x \text{ is a literal over } X \text{ and } t(x) = \text{true}\}, \\ V_2 &= \{u_{i,j} \mid (1 \leq i \leq n-1 \wedge j = 1) \vee (i = n \wedge 2 \leq j \leq r)\} \cup \{a_7, a_8\} \cup \{b_2, b_4\} \\ &\quad \cup \{x \mid x \text{ is a literal over } X \text{ and } t(x) = \text{false}\}, \\ V_3 &= \hat{D} \cup \check{D} \cup \{a_1, a_3\} \cup \{b_5, b_6\} \cup \{y \mid y \text{ is a literal over } Y \text{ and } \tilde{t}(y) = \text{true}\}, \\ V_4 &= \{u_{i,j} \mid (i = n \wedge j = r) \vee (1 \leq i \leq n-1 \wedge 2 \leq j \leq r)\} \cup \{a_2, a_4\} \cup \{b_7, b_8\} \\ &\quad \cup \{y \mid y \text{ is a literal over } Y \text{ and } \tilde{t}(y) = \text{false}\}. \end{aligned}$$

Thus, $\gamma(G) \geq 4$. Since $\gamma(G) \leq 4$, it follows that $\gamma(G) = 4$ in this case.

Case 2: $\boldsymbol{H_1} \in$ `NAE-3-SAT` **and** $\boldsymbol{H_2} \notin$ `NAE-3-SAT`. Let t be a truth assignment satisfying H_1. We can partition G into three $(\mathbb{N}^+, \mathbb{N}^+)$-sets V_1, V_2, and V_3 as follows:

$$\begin{aligned} V_1 &= \hat{C} \cup \check{C} \cup \{a_5, a_6\} \cup \{b_1, b_3\} \cup \{x \mid x \text{ is a literal over } X \text{ and } t(x) = \text{true}\}, \\ V_2 &= \{u_{i,j} \mid 1 \leq i \leq n \wedge 1 \leq j \leq r\} \cup \{a_7, a_8\} \cup \{b_2, b_4\} \\ &\quad \cup \{x \mid x \text{ is a literal over } X \text{ and } t(x) = \text{false}\}, \\ V_3 &= \hat{D} \cup \check{D} \cup \{a_1, a_2, a_3, a_4\} \cup \{b_5, b_6, b_7, b_8\} \cup \{y \mid y \text{ is a literal over } Y\}. \end{aligned}$$

Thus, $3 \leq \gamma(G) \leq 4$. For a contradiction, suppose that $\gamma(G) = 4$, with a partition of G into four $(\mathbb{N}^+, \mathbb{N}^+)$-sets, say U_1, U_2, U_3, and U_4. Vertex $u_{1,1}$ is adjacent to exactly four vertices, namely to x_1, $\overline{x}_1$, y_1 and $\overline{y}_1$. These four vertices must then be in four distinct sets of the partition. Without loss of generality, suppose that $x_1 \in U_1$, $\overline{x}_1 \in U_2$, $y_1 \in U_3$, and $\overline{y}_1 \in U_4$. For each j with $2 \leq j \leq r$, the vertices y_j and $\overline{y}_j$ are connected to x_1 and $\overline{x}_1$ via vertex $u_{1,j}$, so it follows that either $y_j \in U_3$ and $\overline{y}_j \in U_4$, or $y_j \in U_4$ and $\overline{y}_j \in U_3$.

Every clause vertex d_j, $1 \leq j \leq r$, is connected only to the vertices representing its literals and to the vertices b_1 and b_2, which therefore must be in the sets U_1 and U_2, respectively. Thus, every clause vertex d_j is connected to at least one literal vertex in U_3 and to at least one literal vertex in U_4. This describes a valid truth assignment for H_2 in the not-all-equal sense. This is a contradiction to the case assumption $H_2 \notin$ `NAE-3-SAT`.

Case 3: $\boldsymbol{H_1} \notin$ `NAE-3-SAT` **and** $\boldsymbol{H_2} \notin$ `NAE-3-SAT`. A valid partition of G into two $(\mathbb{N}^+, \mathbb{N}^+)$-sets is:

$$\begin{aligned} V_1 &= \{u_{i,j} \mid 1 \leq i \leq n \wedge 1 \leq j \leq r\} \cup \{x_i \mid 1 \leq i \leq n\} \cup \{y_j \mid 1 \leq j \leq r\} \\ &\quad \cup \{a_1, a_3, a_5, a_7\} \cup \{b_1, b_3, b_5, b_7\}, \\ V_2 &= \hat{C} \cup \check{C} \cup \hat{D} \cup \check{D} \cup \{\overline{x}_i \mid 1 \leq i \leq n\} \cup \{\overline{y}_j \mid 1 \leq j \leq r\} \\ &\quad \cup \{a_2, a_4, a_6, a_7\} \cup \{b_2, b_4, b_6, b_8\}. \end{aligned}$$

Thus, $2 \leq \gamma(G) \leq 4$. By the same argument as in Case 2, $\gamma(G) \neq 4$. For a contradiction, suppose that $\gamma(G) = 3$, with a partition of G into three $(\mathbb{N}^+, \mathbb{N}^+)$-sets, say U_1, U_2, and U_3. Without loss of generality, assume that x_1 and $\overline{x}_1$ belong to distinct U_i sets,[2] say $x_1 \in U_1$ and $\overline{x}_1 \in U_2$.

[2] If x_1 and $\overline{x}_1$ both belong to the same set U_i, then each y_j and $\overline{y}_j$ must belong to distinct sets U_k and U_ℓ, $k \neq \ell$, since $u_{1,j}$ is connected with x_1, $\overline{x}_1$, y_j, and $\overline{y}_j$. Thus, a symmetric argument works for y_j and $\overline{y}_j$ in this case.

It follows that for each j with $1 \le j \le r$, at least one of y_j or $\overline{y}_j$ has to be in U_3. If both vertices are in U_3, then we have:

$$(\forall i : 1 \le i \le n)\,[\text{either } x_i \in U_1 \text{ and } \overline{x}_i \in U_2, \text{ or } x_i \in U_2 \text{ and } \overline{x}_i \in U_1]. \tag{4.6}$$

Since $H_1 \notin \mathtt{NAE\text{-}3\text{-}SAT}$, for each truth assignment t for H_1, there exists a clause $c_i \in \hat{C}$ such that $c_i = (x \vee y \vee z)$ and the literals x, y, and z are either simultaneously true or simultaneously false under t. Note that for the corresponding clause $\check{c}_i \in \check{C}$, which contains the negations of x, y, and z, the truth value of its literals is flipped under t. That is, $t(\overline{x}) = 1 - t(x)$, $t(\overline{y}) = 1 - t(y)$, and $t(\overline{z}) = 1 - t(z)$. Since the corresponding clause vertex c_i is adjacent to x, y, z, a_1, and a_2, it follows that x, y, and z are in the same set of the partition, say in U_1. Hence, either $a_1 \in U_2$ and $a_2 \in U_3$, or $a_1 \in U_3$ and $a_2 \in U_2$. Similarly, since the clause vertex $\check{c}_i$ is adjacent to $\overline{x}$, $\overline{y}$, $\overline{z}$, a_1, and a_2, the vertices $\overline{x}$, $\overline{y}$, $\overline{z}$ are in the same set of the partition that must be distinct from U_1. Let U_2, say, be this set. It follows that either $a_1 \in U_1$ and $a_2 \in U_3$, or $a_1 \in U_3$ and $a_2 \in U_1$, which is a contradiction.

Each of the remaining subcases can be reduced to (4.6), and the above contradiction follows. Hence, $\gamma(G) = 2$.

By construction, the case "$H_1 \notin \mathtt{NAE\text{-}3\text{-}SAT}$ and $H_2 \in \mathtt{NAE\text{-}3\text{-}SAT}$" cannot occur, since it contradicts our assumption that $H_2 \in \mathtt{NAE\text{-}3\text{-}SAT}$ implies $H_1 \in \mathtt{NAE\text{-}3\text{-}SAT}$. The case distinction is complete. Thus, we obtain:

$$\begin{aligned} ||\{i \mid H_i \in \mathtt{NAE\text{-}3\text{-}SAT}\}|| \text{ is odd} &\iff H_1 \in \mathtt{NAE\text{-}3\text{-}SAT} \wedge H_2 \notin \mathtt{NAE\text{-}3\text{-}SAT} \\ &\iff \gamma(G) = 3, \end{aligned}$$

which proves Equation (4.5). Thus, Equation (2.1) of Lemma 4 is fulfilled, and it follows that $\mathtt{Exact}\text{-}(3, \mathbb{N}^+, \mathbb{N}^+)\text{-}\mathtt{Partition}$ is DP-complete. ∎

In contrast to Theorem 16, $\mathtt{Exact}\text{-}(1, \mathbb{N}^+, \mathbb{N}^+)\text{-}\mathtt{Partition}$ is in coNP (and even coNP-complete) and thus cannot be DP-complete unless the boolean hierarchy over NP collapses.

Theorem 17 $\mathtt{Exact}\text{-}(1, \mathbb{N}^+, \mathbb{N}^+)\text{-}\mathtt{Partition}$ *is* coNP-*complete.*

Proof. $\mathtt{Exact}\text{-}(1, \mathbb{N}^+, \mathbb{N}^+)\text{-}\mathtt{Partition}$ is in coNP, since it can be written as

$$\mathtt{Exact}\text{-}(1, \mathbb{N}^+, \mathbb{N}^+)\text{-}\mathtt{Partition} = A \cap \overline{B}$$

with $A = (1, \mathbb{N}^+, \mathbb{N}^+)\text{-}\mathtt{Partition}$ being in P and with $B = (2, \mathbb{N}^+, \mathbb{N}^+)\text{-}\mathtt{Partition}$ being in NP. Note that the coNP-hardness of $\mathtt{Exact}\text{-}(1, \mathbb{N}^+, \mathbb{N}^+)\text{-}\mathtt{Partition}$ follows immediately via the original reduction from $\mathtt{NAE\text{-}3\text{-}SAT}$ to $(2, \mathbb{N}^+, \mathbb{N}^+)\text{-}\mathtt{Partition}$ presented in [HT98]. ∎

4.3 The Case $\rho = \mathbb{N}$

In this section, we are concerned with the minimum problems $\texttt{Exact-}(k, \sigma, \mathbb{N})\texttt{-Partition}$, where σ is chosen from $\{\mathbb{N}, \mathbb{N}^+, \{0\}, \{0,1\}, \{1\}\}$. Depending on the value of $k \geq 2$, we ask how hard it is to decide whether a given graph G has a $(k, \sigma, \mathbb{N})$-partition but not a $(k-1, \sigma, \mathbb{N})$-partition.

4.3.1 The Cases $\sigma \in \{\mathbb{N}, \mathbb{N}^+\}$ and $\rho = \mathbb{N}$

These cases are trivial, since $(k, \mathbb{N}, \mathbb{N})\texttt{-Partition}$ and $(k, \mathbb{N}^+, \mathbb{N})\texttt{-Partition}$ are in P for each $k \geq 1$, which outright implies that the problems $\texttt{Exact-}(k, \mathbb{N}, \mathbb{N})\texttt{-Partition}$ and $\texttt{Exact-}(k, \mathbb{N}^+, \mathbb{N})\texttt{-Partition}$ are in P as well.

4.3.2 The Case $\sigma = \{0\}$ and $\rho = \mathbb{N}$

Recall that the problem $(k, \{0\}, \mathbb{N})\texttt{-Partition}$ is equal to the k-colorability problem defined in Section 2. The question about the complexity of the exact versions of this problem was first addressed by Wagner [Wag87] and optimally solved by Rothe [Rot03].

Theorem 18 (Rothe) $\texttt{Exact-}(4, \{0\}, \mathbb{N})\texttt{-Partition}$ *is* DP-*complete.*

In contrast to Theorem 18, $\texttt{Exact-}(3, \{0\}, \mathbb{N})\texttt{-Partition}$ is in NP (and even NP-complete) and thus cannot be DP-complete unless the boolean hierarchy over NP collapses.

Theorem 19 $\texttt{Exact-}(3, \{0\}, \mathbb{N})\texttt{-Partition}$ *is* NP-*complete.*

4.3.3 The Case $\sigma = \{0, 1\}$ and $\rho = \mathbb{N}$

Definition 20 *For every graph G, define the minimum value of k for which G has a $(k, \{0,1\}, \mathbb{N})$-partition as follows:*

$$\alpha(G) \quad = \quad \min\{k \in \mathbb{N}^+ \mid G \in (k, \{0,1\}, \mathbb{N})\texttt{-Partition}\}.$$

Theorem 21 *For each $i \geq 5$,* $\texttt{Exact-}(i, \{0,1\}, \mathbb{N})\texttt{-Partition}$ *is* DP-*complete.*

Proof. Again, it is enough to prove the theorem for the case $i = 5$. By Fact 12, $\texttt{Exact-}(5, \{0,1\}, \mathbb{N})\texttt{-Partition}$ is contained in DP. So it remains to prove DP-hardness. Again, we apply Wagner's Lemma 4 with $k = 1$ being fixed, with 1-3-$\texttt{SAT}$ being the NP-complete problem A, and with $\texttt{Exact-}(5, \{0,1\}, \mathbb{N})\texttt{-Partition}$ being the set B from this lemma.

In their paper [HT98], Heggernes and Telle presented a $\leq_m^p$-reduction f from 1-3-$\texttt{SAT}$ to $(2, \{0,1\}, \mathbb{N})\texttt{-Partition}$ with the following properties:

$$\begin{aligned} H \in \text{1-3-}\texttt{SAT} &\implies \alpha(f(H)) = 2 \\ H \notin \text{1-3-}\texttt{SAT} &\implies \alpha(f(H)) = 3. \end{aligned}$$

In short, reduction f works as follows. Let H be any given boolean formula that consists of a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ of m sets of literals over $X = \{x_1, x_2, \ldots, x_n\}$. Without loss of generality, we may assume that all literals in H are positive; recall the remark right after Definition 7. Reduction f maps H to a graph G as follows. For each set $S_i = \{x, y, z\}$, there is a 4-clique C_i in G induced by the vertices x_i, y_i, z_i, and a_i. For each literal x, there is an edge e_x in G. For each S_i in which x occurs, both endpoints of e_x are connected to the vertex x_i in C_i corresponding to $x \in S_i$. Finally, there is yet another 4-clique induced by the vertices s, t_1, t_2, and t_3. For each i with $1 \leq i \leq m$, vertex s is connected to a_i. This completes the reduction f. Figure 3 shows the graph G resulting from the reduction f applied to the formula $H = (x \vee y \vee z) \wedge (v \vee w \vee x) \wedge (u \vee w \vee z)$.

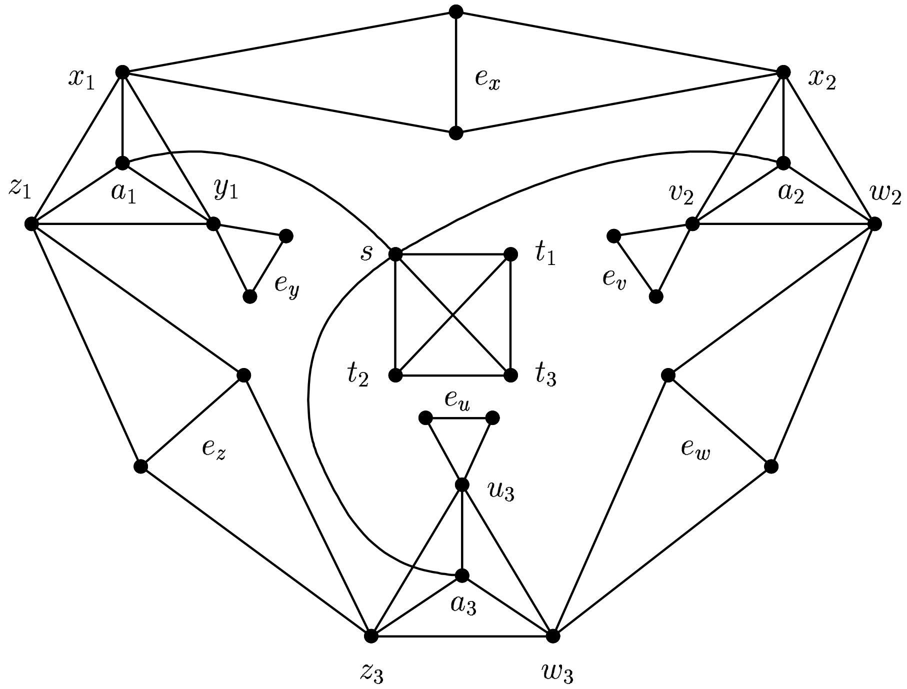

Figure 3: Heggernes and Telle's reduction f from 1-3-`SAT` to $(2, \{0, 1\}, \mathbb{N})$-`Partition`.

In order to apply Lemma 4, we need to find a reduction g satisfying

$$(H_1 \in \text{1-3-}\mathtt{SAT} \wedge H_2 \notin \text{1-3-}\mathtt{SAT}) \iff \alpha(g(H_1, H_2)) = 5 \tag{4.7}$$

for any two given instances H_1 and H_2 such that $H_2 \in$ 1-3-`SAT` implies $H_1 \in$ 1-3-`SAT`.

Reduction g is constructed from f as follows. Let $G_{1,1}$ and $G_{1,2}$ be two disjoint copies of the graph $f(H_1)$, and let $G_{2,1}$ and $G_{2,2}$ be two disjoint copies of the graph $f(H_2)$. Define G_i to be the disjoint union of $G_{i,1}$ and $G_{i,2}$, for $i \in \{1, 2\}$. Define the graph $G = g(H_1, H_2)$

to be the join of G_1 and G_2; see Definition 1. That is,

$$g(H_1, H_2) = G = G_1 \oplus G_2 = (G_{1,1} \cup G_{1,2}) \oplus (G_{2,1} \cup G_{2,2}).$$

Figure 4 shows the graph G resulting from the reduction g applied to the formulas

$$\begin{aligned} H_1 &= (x \vee y \vee z) \wedge (v \vee w \vee x) \wedge (u \vee w \vee z) \quad \text{and} \\ H_2 &= (c \vee d \vee e) \wedge (e \vee f \vee g) \wedge (g \vee h \vee i) \wedge (i \vee j \vee c). \end{aligned}$$

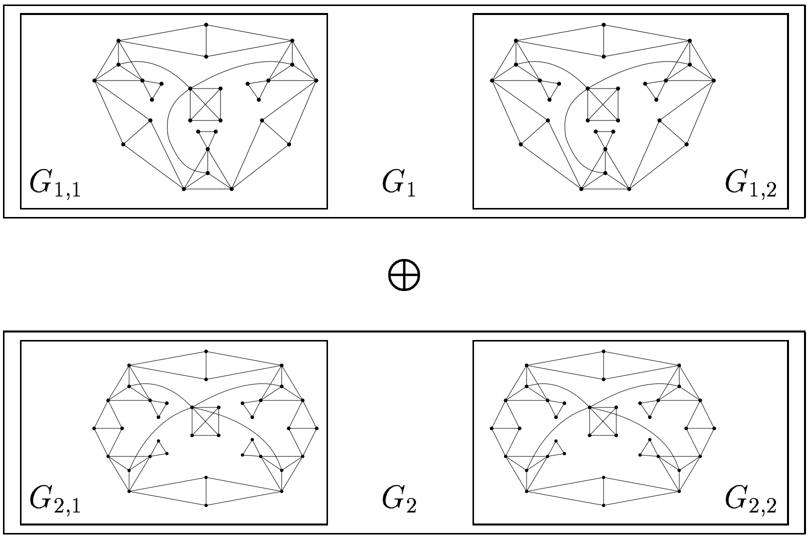

Figure 4: `Exact-`$(5, \{0,1\}, \mathbb{N})$`-Partition` is DP-complete: Graph $G = g(H_1, H_2)$.

Let $a = \alpha(G_{1,1}) = \alpha(G_{1,2})$ and $b = \alpha(G_{2,1}) = \alpha(G_{2,2})$. Clearly, $\alpha(G_1) = a$, $\alpha(G_2) = b$, and $\alpha(G) \leq a+b$. Simply partition G the same way as graphs G_1 and G_2 were partitioned before. Note that we obtain 8-cliques in G as a result of joining pairs of 4-cliques from G_1 and G_2. Thus, $\alpha(G) \geq 4$, since an 8-clique has to be partitioned into at least four disjoint $(\{0,1\}, \mathbb{N})$-sets.

To prove that $\alpha(G) = \alpha(G_1) + \alpha(G_2) = a + b$, let $k = \alpha(G)$. Thus, we know $4 \leq k \leq a + b$. For a contradiction, suppose that $k < a + b$. Distinguish the following cases.

Case 1: $\boldsymbol{a = b = 2}$. Then $k < 4$ is a contradiction to $k \geq 4$.

Case 2: $\boldsymbol{a = 2}$ **and** $\boldsymbol{b = 3}$. Then $k = 4 < 5 = a+b$. One of the four disjoint $(\{0,1\}, \mathbb{N})$-sets consists of at least one vertex u in G_1 and one vertex v in G_2. (Otherwise, it would induce a partition of less than two $(\{0,1\}, \mathbb{N})$-sets in G_1 or of less than three $(\{0,1\}, \mathbb{N})$-sets in G_2, which contradicts our assumption $a = 2$ and $b = 3$.) Suppose that this set is V_1. Then, since $\sigma = \{0, 1\}$ and since u is adjacent to every vertex in G_2 and v is adjacent to every vertex in G_1, we have $V_1 = \{u, v\}$. But there is no way to assign the 8-cliques, which do not contain u or v, to the remaining three $(\{0, 1\}, \mathbb{N})$-sets in order to obtain a $(4, \{0, 1\}, \mathbb{N})$-partition for G. This is a contradiction, and our assumption $k < a + b = 5$ does not hold. Thus, $k = 5$.

Case 3: $\boldsymbol{a = 3}$ **and** $\boldsymbol{b = 2}$. This case cannot occur, since we have to prove Equation (4.7) only for instances H_1 and H_2 such that $H_2 \in$ 1-3-`SAT` implies $H_1 \in$ 1-3-`SAT`.

Case 4: $\boldsymbol{a = b = 3}$. By the same argument used in Case 2, $k = 4$ does not hold. Suppose $k = 5$. As seen before, one of the sets in the partition must contain exactly one vertex u from G_1 and exactly one vertex v from G_2. Let $V_1 = \{u, v\}$ be this set. There are four sets left for the partition, say V_2, V_3, V_4, and V_5. Every set V_i can have only vertices from either G_1 or G_2. This means that two of these sets cover all vertices in G_1 except for u. Vertex u is either in $G_{1,1}$ or in $G_{1,2}$, which implies that one of these induced subgraphs ($G_{1,1}$ or $G_{1,2}$) has a $(2, \{0, 1\}, \mathbb{N})$-partition. This is a contradiction to $a = 3$. Thus, $k = 6$.

Thus, $\alpha(G) = \alpha(G_1) + \alpha(G_2)$, which implies Equation (4.7) and thus fulfills Equation (2.1) of Lemma 4:

$$\begin{aligned} ||\{i \mid H_i \in \text{1-3-}\mathtt{SAT}\}|| \text{ is odd} &\iff H_1 \in \text{1-3-}\mathtt{SAT} \wedge H_2 \notin \text{1-3-}\mathtt{SAT} \\ &\iff \alpha(G_1) = 2 \wedge \alpha(G_2) = 3 \\ &\iff \alpha(G) = 5. \end{aligned}$$

By Lemma 4, `Exact-`$(5, \{0, 1\}, \mathbb{N})$`-Partition` is DP-complete. ∎

In contrast to Theorem 21, `Exact-`$(2, \{0, 1\}, \mathbb{N})$`-Partition` is in NP (and even NP-complete) and thus cannot be DP-complete unless the boolean hierarchy over NP collapses.

Theorem 22 `Exact-`$(2, \{0, 1\}, \mathbb{N})$`-Partition` *is* NP-*complete.*

Proof. `Exact-`$(2, \{0, 1\}, \mathbb{N})$`-Partition` is in NP, since it can be written as

$$\mathtt{Exact}\text{-}(2, \{0, 1\}, \mathbb{N})\text{-}\mathtt{Partition} = A \cap \overline{B}$$

with $A = (2, \{0, 1\}, \mathbb{N})$`-Partition` being in NP and with $B = (1, \{0, 1\}, \mathbb{N})$`-Partition` being in P. NP-hardness follows immediately via the reduction f defined in the proof of Theorem 21, see Figure 3:

$$H \in \text{1-3-}\mathtt{SAT} \iff f(H) \in \mathtt{Exact}\text{-}(2, \{0, 1\}, \mathbb{N})\text{-}\mathtt{Partition}.$$

Thus, `Exact-`$(2, \{0, 1\}, \mathbb{N})$`-Partition` is NP-complete. ∎

4.3.4 The Case $\sigma = \{1\}$ and $\rho = \mathbb{N}$

Definition 23 *For every graph G, define the minimum value k for which G has a $(k, \{1\}, \mathbb{N})$-partition as follows:*

$$\beta(G) = \min\{k \in \mathbb{N}^+ \mid G \in (k, \{1\}, \mathbb{N})\text{-}\mathtt{Partition}\}.$$

Theorem 24 *For each $i \geq 5$,* $\mathtt{Exact}$-$(i, \{1\}, \mathbb{N})$-$\mathtt{Partition}$ *is* DP-*complete.*

Proof. Clearly, $\alpha(G) \leq \beta(G)$ for all graphs G. Conversely, we show that $\alpha(G) \geq \beta(G)$. It is enough to do so for all graphs $G = f(H)$ resulting from any given instance H of 1-3-$\mathtt{SAT}$ via the reduction f in Theorem 21. If $H \in$ 1-3-$\mathtt{SAT}$, we have $\alpha(G) = 2$. Using the same partition, we even get two $(\{1\}, \mathbb{N})$-sets for G. Every vertex of G has exactly one neighbor, which is in the same set of the partition as the vertex itself. If $S \notin$ 1-3-$\mathtt{SAT}$, then $\alpha(G) = 3$. We can then partition G into three $(\{1\}, \mathbb{N})$-sets: V_1 consists of the vertices s and t_1 plus the endpoints of each edge e_x. V_2 consists of t_2 and t_3, every vertex a_i, and one more vertex in the 4-clique C_i, for each i with $1 \leq i \leq 2m$. The two remaining vertices in each C_i are then put into the set V_3. Hence, $\alpha(G) = \beta(G)$. The rest of the proof is analogous to the proof of Theorem 21. ∎

In contrast to Theorem 24, $\mathtt{Exact}$-$(2, \{1\}, \mathbb{N})$-$\mathtt{Partition}$ is in NP (and even NP-complete) and thus cannot be DP-complete unless the boolean hierarchy over NP collapses. The proof follows from the proofs of Theorems 22 and 24 and is omitted here.

Theorem 25 $\mathtt{Exact}$-$(2, \{1\}, \mathbb{N})$-$\mathtt{Partition}$ *is* NP-*complete.*

4.4 Completeness in the Higher Levels of the Boolean Hierarchy

In this section, we show that the results of the previous two subsections can be generalized to higher levels of the boolean hierarchy over NP. We exemplify this observation only for the case of Theorem 13. Using the techniques of Wagner [Wag87], it is a matter of routine to obtain the analogous results for the other exact generalized dominating set problems.

For each fixed set M_k containing k noncontiguous integers not smaller than $4k+1$, we show that $\mathtt{Exact}$-M_k-$\mathtt{DNP}$ is complete for $\mathrm{BH}_{2k}(\mathrm{NP})$, the $2k$th level of the boolean hierarchy over NP. Note that the special case of $k = 1$ in Theorem 26 yields Theorem 13. Note also that the specific set M_k defined in Theorem 26 gives the smallest k noncontiguous numbers for which $\mathrm{BH}_{2k}(\mathrm{NP})$-completeness of $\mathtt{Exact}$-M_k-$\mathtt{DNP}$ can be achieved by the proof method of Theorem 26. However, Theorem 26 may not be optimal yet; see the open questions in Section 6.

Theorem 26 *For fixed $k \geq 1$, let $M_k = \{4k+1, 4k+3, \ldots, 6k-1\}$. Then,* $\mathtt{Exact}$-M_k-$\mathtt{DNP}$ *is* $\mathrm{BH}_{2k}(\mathrm{NP})$-*complete.*

Proof. To show that $\texttt{Exact-}M_k\texttt{-DNP}$ is contained in $\mathrm{BH}_{2k}(\mathrm{NP})$, partition the problem into k subproblems: $\texttt{Exact-}M_k\texttt{-DNP} = \bigcup_{i \in M_k} \texttt{Exact-}i\texttt{-DNP}$. Every set $\texttt{Exact-}i\texttt{-DNP}$ can be rewritten as

$$\texttt{Exact-}i\texttt{-DNP} = \{G \mid \delta(G) \geq i\} \cap \{G \mid \delta(G) < i+1\}.$$

Clearly, the set $\{G \mid \delta(G) \geq i\}$ is in NP, and the set $\{G \mid \delta(G) < i+1\}$ is in coNP. It follows that $\texttt{Exact-}i\texttt{-DNP}$ is in DP, for each $i \in M_k$. By definition, $\texttt{Exact-}M_k\texttt{-DNP}$ is in $\mathrm{BH}_{2k}(\mathrm{NP})$.

The proof that $\texttt{Exact-}M_k\texttt{-DNP}$ is $\mathrm{BH}_{2k}(\mathrm{NP})$-hard straightforwardly generalizes the proof of Theorem 13. Again, we draw on Lemma 4 with $\texttt{3-Colorability}$ being the NP-complete set A and with $\texttt{Exact-}M_k\texttt{-DNP}$ being the set B from this lemma. Fix any $2k$ graphs $G_1, G_2, \ldots, G_{2k}$ satisfying that for each j with $1 \leq j < 2k$, if G_{j+1} is in $\texttt{3-Colorability}$, then so is G_j. Without loss of generality, we assume that none of these graphs G_j is 2-colorable, nor does it contain isolated vertices, and we assume that $\chi(G_j) \leq 4$ for each j. Applying the Lemma 6 reduction g from $\texttt{3-Colorability}$ to $\texttt{DNP}$, we obtain $2k$ graphs $H_j = g(G_j)$, $1 \leq j \leq 2k$, each satisfying the implications (2.2) and (2.3). Hence, for each j, $\delta(H_j) \in \{2, 3\}$, and $\delta(H_{j+1}) = 3$ implies $\delta(H_j) = 3$.

Now, generalize the construction of graph H in the proof of Theorem 13 as follows. For any fixed sequence $T_1, T_2, \ldots, T_{2k}$ of triangles, where T_i belongs to H_i, add $6k$ new gadget vertices $a_1, a_2, \ldots, a_{6k}$ and, for each i with $1 \leq i \leq 2k$, associate the three gadget vertices $a_{1+3(i-1)}$, $a_{2+3(i-1)}$, and a_{3i} with the triangle T_i. For each i with $1 \leq i \leq 2k$, connect T_i with every T_j, where $1 \leq j \leq 2k$ and $i \neq j$, via the same three gadget vertices $a_{1+3(i-1)}$, $a_{2+3(i-1)}$, and a_{3i} associated with T_i the same way T_1 and T_2 are connected in Figure 1 via the vertices a_1, a_2, and a_3.

It follows that $\mathit{deg}(a_i) = 6k - 1$ for each i, so $\delta(H) \leq 6k$. An argument analogous to the case distinction in the proof of Theorem 13 shows that $\delta(H) = \sum_{j=1}^{2k} \delta(H_j)$. Hence,

$$
\begin{aligned}
& \|\{i \mid G_i \in \texttt{3-Colorability}\}\| \text{ is odd} \\
\iff\; & (\exists i : 1 \leq i \leq k)\ [\chi(G_1) = \cdots = \chi(G_{2i-1}) = 3 \text{ and } \chi(G_{2i}) = \cdots = \chi(G_{2k}) = 4] \\
\iff\; & (\exists i : 1 \leq i \leq k)\ [\delta(H_1) = \cdots = \delta(H_{2i-1}) = 3 \text{ and } \delta(H_{2i}) = \cdots = \delta(H_{2k}) = 2] \\
\iff\; & (\exists i : 1 \leq i \leq k)\ \left[\delta(H) = \sum_{j=1}^{2k} \delta(H_j) = 3(2i-1) + 2(2k-2i+1)\right] \\
\iff\; & (\exists i : 1 \leq i \leq k)\ [\delta(H) = 4k + 2i - 1] \\
\iff\; & \delta(H) \in \{4k+1, 4k+3, \ldots, 6k-1\} \\
\iff\; & f(G_1, G_2, \ldots, G_{2k}) = H \in \texttt{Exact-}M_k\texttt{-DNP}.
\end{aligned}
$$

Thus, f satisfies Equation (2.1). By Lemma 4, $\texttt{Exact-}M_k\texttt{-DNP}$ is $\mathrm{BH}_{2k}(\mathrm{NP})$-complete. ∎

4.5 Domatic Number Problems Complete for Parallel Access to NP

In this section, we consider the problem of deciding whether or not the domatic number of a given graph is an odd integer, and the problem of comparing the domatic numbers of two given graphs. Applying the techniques of the previous section, we prove in Theorem 29 below that these variants of the domatic number problem are complete for $\mathrm{P}_{||}^{\mathrm{NP}}$, the class of problems that can be solved by a deterministic polynomial-time Turing machine making parallel (a.k.a. "nonadaptive" or "truth-table") queries to some NP oracle set. Other characterizations of $\mathrm{P}_{||}^{\mathrm{NP}}$ and further results related to this important class are listed in the introduction.

Definition 27 *Define the following variants of the domatic number problem:*

$$
\begin{aligned}
\texttt{DNP-Odd} &= \{G \mid G \textit{ is a graph such that } \delta(G) \textit{ is odd}\}; \\
\texttt{DNP-Equ} &= \{\langle G, H\rangle \mid G \textit{ and } H \textit{ are graphs such that } \delta(G) = \delta(H)\}; \\
\texttt{DNP-Geq} &= \{\langle G, H\rangle \mid G \textit{ and } H \textit{ are graphs such that } \delta(G) \geq \delta(H)\}.
\end{aligned}
$$

Wagner provided a sufficient condition for proving $\mathrm{P}_{||}^{\mathrm{NP}}$-hardness that is analogous to Lemma 4 except that in Lemma 28 the value of k is not fixed; see Theorem 5.2 in [Wag87]. The introduction gives a list of related $\mathrm{P}_{||}^{\mathrm{NP}}$-completeness results for which Wagner's technique was applied.

Lemma 28 (Wagner) *Let A be some* NP*-complete problem and B be an arbitrary problem. If there exists a polynomial-time computable function f such that the equivalence*

$$||\{i \mid x_i \in A\}|| \textit{ is odd} \iff f(x_1, x_2, \ldots, x_{2k}) \in B \tag{4.8}$$

is true for each $k \geq 1$ and for all strings $x_1, x_2, \ldots, x_{2k} \in \Sigma^*$ satisfying that for each j with $1 \leq j < 2k$, $x_{j+1} \in A$ implies $x_j \in A$, then B is $\mathrm{P}_{||}^{\mathrm{NP}}$-hard.

Theorem 29 `DNP-Odd`*,* `DNP-Equ`*, and* `DNP-Geq` *each are $\mathrm{P}_{||}^{\mathrm{NP}}$-complete.*

Proof. It is easy to see that each of the problems `DNP-Odd`, `DNP-Equ`, and `DNP-Geq` belongs to $\mathrm{P}_{||}^{\mathrm{NP}}$, since the domatic number of a given graph can be determined exactly by parallel queries to the NP oracle `DNP`. It remains to prove that each of these problems is $\mathrm{P}_{||}^{\mathrm{NP}}$-hard. For `DNP-Odd`, this follows immediately from the proof of Theorems 13 and 26, respectively, using Lemma 28.

We now show that `DNP-Equ` is $\mathrm{P}_{||}^{\mathrm{NP}}$-hard by applying Lemma 28 with A being the NP-complete problem `3-Colorability` and B being `DNP-Equ`. Fix any $k \geq 1$, and let $G_1, G_2, \ldots, G_{2k}$ be any given sequence of graphs satisfying that for each j with $1 \leq j < 2k$, if G_{j+1} is 3-colorable, then so is G_j. Since $\mathrm{P}_{||}^{\mathrm{NP}}$ is closed under complement, Equation (4.8) from Lemma 28 can be replaced by

$$||\{i \mid G_i \in \texttt{3-Colorability}\}|| \text{ is even} \iff f(G_1, G_2, \ldots, G_{2k}) \in \texttt{DNP-Equ}. \tag{4.9}$$

As in the proof of Theorem 26, construct the graphs $H_1, H_2, \ldots, H_{2k}$ from the given graphs $G_1, G_2, \ldots, G_{2k}$ according to Lemma 6, where each $H_j = g(G_j)$ satisfies the implications (2.2) and (2.3). Let $\times$ denote the associative operation on graphs constructed in the proof of Theorem 26 to sum up the domatic numbers of the given graphs, and define the graphs:

$$\begin{aligned} G_{\mathtt{odd}} &= H_1 \times H_3 \times \cdots \times H_{2k-1}, \\ G_{\mathtt{even}} &= H_2 \times H_4 \times \cdots \times H_{2k}. \end{aligned}$$

We now prove Equation (4.9). From left to right we have:

$$\begin{aligned} & ||\{i \mid G_i \in 3\text{-}\mathtt{Colorability}\}|| \text{ is even} \\ \Longrightarrow\ & (\forall i : 1 \leq i \leq k)\ [\delta(H_{2i-1}) = \delta(H_{2i})] \\ \Longrightarrow\ & \sum_{1 \leq i \leq k} \delta(H_{2i-1}) = \sum_{1 \leq i \leq k} \delta(H_{2i}) \\ \Longrightarrow\ & \delta(G_{\mathtt{odd}}) = \delta(G_{\mathtt{even}}) \\ \Longrightarrow\ & \langle G_{\mathtt{odd}}, G_{\mathtt{even}} \rangle = f(G_1, G_2, \ldots, G_{2k}) \in \mathtt{DNP\text{-}Equ}. \end{aligned}$$

From right to left we have:

$$\begin{aligned} & ||\{i \mid G_i \in 3\text{-}\mathtt{Colorability}\}|| \text{ is odd} \\ \Longrightarrow\ & (\exists i : 1 \leq i \leq k)\ [\delta(H_{2i-1}) = 3 \wedge \delta(H_{2i}) = 2 \text{ and } \delta(H_{2j-1}) = \delta(H_{2j}) \text{ for } j \neq i] \\ \Longrightarrow\ & -1 + \sum_{1 \leq i \leq k} \delta(H_{2i-1}) = \sum_{1 \leq i \leq k} \delta(H_{2i}) \\ \Longrightarrow\ & \delta(G_{\mathtt{odd}}) - 1 = \delta(G_{\mathtt{even}}) \\ \Longrightarrow\ & \langle G_{\mathtt{odd}}, G_{\mathtt{even}} \rangle = f(G_1, G_2, \ldots, G_{2k}) \notin \mathtt{DNP\text{-}Equ}. \end{aligned}$$

Lemma 28 implies that `DNP-Equ` is $\mathrm{P}_{||}^{\mathrm{NP}}$-complete.

The above proof for `DNP-Equ` also gives $\mathrm{P}_{||}^{\mathrm{NP}}$-completeness for `DNP-Geq`. ∎

5 The Exact Conveyor Flow Shop Problem

5.1 NP-Completeness

The conveyor flow shop problem is a minimization problem arising in real-world applications in the wholesale business, where warehouses are supplied with goods from a central storehouse. Suppose you are given m machines, $P_1, P_2, \ldots, P_m$, and n jobs, $J_1, J_2, \ldots, J_n$. Conveyor belt systems are used to convey jobs from machine to machine at which they are to be processed in a "permutation flow shop" manner. That is, the jobs visit the machines in the fixed order $P_1, P_2, \ldots, P_m$, and the machines process the jobs in the fixed order $J_1, J_2, \ldots, J_n$. An $(n \times m)$ task matrix $\mathcal{M} = (\mu_{j,p})_{j,p}$ with $\mu_{j,p} \in \{0,1\}$ provides the information which job has to be processed at which machine: $\mu_{j,p} = 1$ if job J_j is to be processed at machine P_p, and $\mu_{j,p} = 0$ otherwise. Every machine can process at most one job at a time. There is one worker supervising the system. Every machine can process a job only if the worker is present, which means that the worker occasionally has to move from one machine to another. If the worker is currently not present at some machine, jobs can be queued in a buffer at this machine. The objective is to minimize the movement of the worker, where we assume the "unit distance" between any two machines, i.e., to measure the worker's movement, we simply count how many times he has switched machines until the complete task matrix has been processed.[3] Let $\Delta_{\min}(\mathcal{M})$ denote the minimum number of machine switches needed for the worker to completely process a given task matrix $\mathcal{M}$, where the minimum is taken over all possible orders in which the tasks in $\mathcal{M}$ can be processed. Define the decision version of the conveyor flow shop problem by

$$\texttt{CFSP} = \{\langle \mathcal{M}, k\rangle \mid \mathcal{M} \text{ is a task matrix and } k \text{ is a positive integer such that } \Delta_{\min}(\mathcal{M}) \leq k\}.$$

Espelage and Wanke [EW00,Esp01,EW01,EW03] introduced the problem CFSP defined above. They studied CFSP and variations thereof extensively; in particular, they showed that CFSP is NP-complete. In our proof of Theorem 33 we apply Lemma 30 below, that provides a reduction to CFSP having certain useful properties.

To show that CFSP is NP-complete, Espelage provided, in a rather involved 17 pages proof (see pp. 27–44 of [Esp01]), a reduction g from the 3-SAT problem to CFSP, via the intermediate problem of finding a "minimum valid block cover" of a given task matrix $\mathcal{M}$. In particular, finding a minimum block cover of $\mathcal{M}$ directly yields a minimum number of machine switches. Espelage's reduction can easily be modified so as to have certain useful properties, which we state in the following lemma. The details of this modification can be found in pp. 37–42 of [Rie02]. In particular, prior to the Espelage reduction, a reduction from the (unrestricted) satisfiability problem to 3-SAT is used that has the properties stated as Equations (5.10) and (5.11) below.

Lemma 30 (Espelage and Riege) *There exists a polynomial-time many-one reduction g that witnesses 3-SAT $\leq_{\mathrm{m}}^{\mathrm{p}}$ CFSP and satisfies, for each given boolean formula φ, the following properties:*

1. *$g(\varphi) = \langle \mathcal{M}_\varphi, z_\varphi \rangle$, where $\mathcal{M}_\varphi$ is a task matrix and $z_\varphi \in \mathbb{N}$ is an odd number.*
2. *$\Delta_{\min}(\mathcal{M}_\varphi) = z_\varphi + u_\varphi$, where u_φ denotes the minimum number of clauses of φ not satisfied under assignment t, where the minimum is taken over all assignments t of φ. Moreover, $u_\varphi = 0$ if $\varphi \in$ 3-SAT, and $u_\varphi = 1$ if $\varphi \notin$ 3-SAT.*

[3] We do not consider possible generalizations of the problem CFSP such as other distance functions, variable job sequences, more than one worker, etc. We refer to Espelage's thesis [Esp01] for results on such more general problems.

In particular, $\varphi \in$ 3-SAT *if and only if* $\Delta_{\min}(\mathcal{M}_\varphi)$ *is odd.*

5.2 Completeness in the Higher Levels of the Boolean Hierarchy

We are interested in the complexity of the exact versions of CFSP.

Definition 31 *For each* $k \geq 1$*, define the* exact version of the conveyor flow shop problem *by*

$$\texttt{Exact-}k\texttt{-CFSP} = \left\{ \langle \mathcal{M}, S_k \rangle \,\middle|\, \begin{array}{l} \mathcal{M} \textit{ is a task matrix and } S_k \subseteq \mathbb{N} \textit{ is a set of } k \\ \textit{noncontiguous integers with } \Delta_{\min}(\mathcal{M}) \in S_k \end{array} \right\}.$$

Since CFSP is in NP, the upper bound of the complexity of Exact-k-CFSP stated in Fact 32 follows immediately. Theorem 33 proves a matching lower bound.

Fact 32 *For each* $k \geq 1$, Exact-k-CFSP *is in* $\mathrm{BH}_{2k}(\mathrm{NP})$.

Theorem 33 *For each* $k \geq 1$, Exact-k-CFSP *is* $\mathrm{BH}_{2k}(\mathrm{NP})$*-complete.*

Proof. By Fact 32, Exact-k-CFSP is contained in $\mathrm{BH}_{2k}(\mathrm{NP})$ for each k. To prove $\mathrm{BH}_{2k}(\mathrm{NP})$-hardness of Exact-$k$-CFSP, we again apply Lemma 4, with some fixed NP-complete problem A and with Exact-k-CFSP being the problem B from this lemma. The reduction f satisfying Equation (2.1) from Lemma 4 is defined by using two polynomial-time many-one reductions, g and h.

We now define the reductions g and h. Fix the NP-complete problem A. Let $x_1, x_2, \ldots, x_{2k}$ be strings in Σ^* satisfying that $c_A(x_1) \geq c_A(x_2) \geq \cdots \geq c_A(x_{2k})$, where c_A denotes the characteristic function of A, i.e., $c_A(x) = 1$ if $x \in A$, and $c_A(x) = 0$ if $x \notin A$. Wagner [Wag87] observed that the standard reduction (cf. [GJ79]) from the (unrestricted) satisfiability problem to 3-SAT can be easily modified so as to yield a reduction h from A to 3-SAT (via the intermediate satisfiability problem) such that, for each $x \in \Sigma^*$, the boolean formula $\varphi = h(x)$ satisfies the following properties:

$$x \in A \implies s_\varphi = m_\varphi; \tag{5.10}$$
$$x \notin A \implies s_\varphi = m_\varphi - 1, \tag{5.11}$$

where $s_\varphi = \max_t\{\ell \mid \ell$ clauses of φ are satisfied under assignment $t\}$, and m_φ denotes the number of clauses of φ. Moreover, m_φ is always odd.

Let $\varphi_1, \varphi_2, \ldots, \varphi_{2k}$ be the boolean formulas after applying reduction h to each given $x_i \in \Sigma^*$, i.e., $\varphi_i = h(x_i)$ for each i. For $i \in \{1, 2, \ldots, 2k\}$, let $m_i = m_{\varphi_i}$ be the number of clauses in φ_i, and let $s_i = s_{\varphi_i}$ denote the maximum number of satisfiable clauses of φ_i, where the maximum is taken over all assignments of φ_i. For each i, apply the Lemma 30 reduction g from 3-SAT to CFSP to obtain $2k$ pairs $\langle \mathcal{M}_i, z_i \rangle = g(\varphi_i)$, where each $\mathcal{M}_i = \mathcal{M}_{\varphi_i}$ is a task matrix and each $z_i = z_{\varphi_i}$ is the odd number corresponding to φ_i according to Lemma 30. Use these $2k$ task matrices to form a new task matrix:

$$\mathcal{M} = \begin{pmatrix} \mathcal{M}_1 & 0 & \cdots & 0 \\ 0 & \mathcal{M}_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \mathcal{M}_{2k} \end{pmatrix}.$$

Every task of some matrix $\mathcal{M}_i$, where $1 \leq i \leq 2k$, can be processed only if all tasks of the matrices $\mathcal{M}_j$ with $j < i$ have already been processed; see [Esp01,Rie02] for arguments as to why this is true. This implies that

$$\Delta_{\text{min}}(\mathcal{M}) = \sum_{i=1}^{2k} \Delta_{\text{min}}(\mathcal{M}_i).$$

Let $z = \sum_{i=1}^{2k} z_i$; note that z is even. Define the set $S_k = \{z+1, z+3, \ldots, z+2k-1\}$, and define the reduction f by $f(x_1, x_2, \ldots, x_{2k}) = \langle \mathcal{M}, S_k \rangle$. Clearly, f is polynomial-time computable.

Let $u_i = u_{\varphi_i} = \min_t\{\ell \mid \ell \text{ clauses of } \varphi_i \text{ are not satisfied under assignment } t\}$. Equations (5.10) and (5.11) then imply that for each i:

$$u_i = m_i - s_i = \begin{cases} 0 & \text{if } x_i \in A \\ 1 & \text{if } x_i \notin A. \end{cases}$$

Recall that, by Lemma 30, we have $\Delta_{\text{min}}(\mathcal{M}_i) = z_i + u_i$. Hence,

$$\begin{aligned}
&||\{i \mid x_i \in A\}|| \text{ is odd} \\
&\iff (\exists i : 1 \leq i \leq k)\ [x_1, \ldots, x_{2i-1} \in A \text{ and } x_{2i}, \ldots, x_{2k} \notin A] \\
&\iff (\exists i : 1 \leq i \leq k)\ [s_1 = m_1, \ldots, s_{2i-1} = m_{2i-1} \text{ and } s_{2i} = m_{2i} - 1, \ldots, s_{2k} = m_{2k} - 1] \\
&\iff (\exists i : 1 \leq i \leq k)\ [\Delta_{\text{min}}(\mathcal{M}_1) = z_1, \ldots, \Delta_{\text{min}}(\mathcal{M}_{2i-1}) = z_{2i-1} \text{ and} \\
&\qquad\qquad \Delta_{\text{min}}(\mathcal{M}_{2i}) = z_{2i} + 1, \ldots, \Delta_{\text{min}}(\mathcal{M}_{2k}) = z_{2k} + 1] \\
&\iff (\exists i : 1 \leq i \leq k)\ \left[\Delta_{\text{min}}(\mathcal{M}) = \sum_{j=1}^{2k} \Delta_{\text{min}}(\mathcal{M}_j) = \left(\sum_{j=1}^{2k} z_j\right) + 2k - 2i + 1\right] \\
&\iff \Delta_{\text{min}}(\mathcal{M}) \in S_k = \{z+1, z+3, \ldots, z+2k-1\} \\
&\iff f(x_1, x_2, \ldots, x_{2k}) = \langle \mathcal{M}, S_k \rangle \in \texttt{Exact-}k\texttt{-CFSP}.
\end{aligned}$$

Thus, f satisfies Equation (2.1). By Lemma 4, Exact-k-CFSP is $\mathrm{BH}_{2k}(\mathrm{NP})$-complete. ∎

For the special case of $k = 1$, Theorem 33 gives the following corollary.

Corollary 34 Exact-1-CFSP *is* DP-*complete.*

6 Conclusions and Open Questions

In this paper, we have shown that the exact versions of the domatic number problem and of the conveyor flow shop problem are complete for the levels of the boolean hierarchy over NP. Our main results are proven in Section 4 in which we have studied the exact versions of generalized dominating set problems. Based on Heggernes and Telle's uniform approach to define graph problems by partitioning the vertex set of a graph into generalized dominating sets [HT98], we have considered problems of the form $\texttt{Exact-}(k, \sigma, \rho)\texttt{-Partition}$, where the parameters σ and ρ specify the number of neighbors that are allowed for each vertex in the partition. We obtained DP-completeness results for a number of such problems. These results are summarized in Table 2 in Section 4.1.

In particular, the minimization problems $\texttt{Exact-}(5, \{0,1\}, \mathbb{N})\texttt{-Partition}$ and $\texttt{Exact-}(5, \{1\}, \mathbb{N})\texttt{-Partition}$ both are DP-complete, and so are the maximization problems $\texttt{Exact-}(3, \mathbb{N}^+, \mathbb{N}^+)\texttt{-Partition}$ and $\texttt{Exact-}(5, \mathbb{N}, \mathbb{N}^+)\texttt{-Partition}$. Since $\texttt{Exact-}(k, \mathbb{N}, \mathbb{N}^+)\texttt{-Partition}$ equals $\texttt{Exact-}k\texttt{-DNP}$, the latter result says that, for each given integer $i \geq 5$, it is DP-complete to determine whether or not $\delta(G) = i$ for a given graph G. In contrast, $\texttt{Exact-2-DNP}$ is coNP-complete, and thus this problem cannot be DP-complete unless the boolean hierarchy collapses. For $i \in \{3, 4\}$, the question of whether or not the problems $\texttt{Exact-}i\texttt{-DNP}$ are DP-complete remains an interesting open problem.

The same question arises for the other problems studied: It is open whether or not the value of $k = 3$ for $\sigma = \rho = \mathbb{N}^+$ and the value of $k = 5$ in the other cases is optimal in the results stated above. We were only able to show these problems NP-complete or coNP-complete for the value of $k = 1$ if $\sigma = \rho = \mathbb{N}^+$, and for the value of $k = 2$ in the other cases, thus leaving a gap between DP-completeness and membership in NP or coNP.

Another interesting open question is whether one can obtain similar results for the minimization problems $\texttt{Exact-}(k, \sigma, \{0,1\})\texttt{-Partition}$ for $\sigma \in \{\{0\}, \{0,1\}, \{1\}\}$. It appears that the constructions that we used in proving Theorems 13, 16, 21, and 24 do not work here.

As mentioned in the introduction and in Section 4, the corresponding gap for the exact chromatic number problem was recently closed [Rot03]. The reduction in [Rot03] uses both the standard reduction from $\texttt{3-SAT}$ to $\texttt{3-Colorability}$ (cf. [GJ79]) and a very clever reduction found by Guruswami and Khanna [GK00]. The decisive property of the Guruswami–Khanna reduction is that it maps each satisfiable formula φ to a graph G with $\chi(G) = 3$, and it maps each unsatisfiable formula φ to a graph G with $\chi(G) = 5$. That is, the graphs they construct are never 4-colorable. To close the above-mentioned gap for the exact domatic number problem, one would have to find a reduction from some NP-complete problem to $\texttt{DNP}$ with a similarly strong property: the reduction would have to yield graphs that never have a domatic number of three.

In Sections 4.4 and 4.5, the DP-completeness results of Sections 4.2 and 4.3 are lifted to complexity classes widely believed to be more powerful than DP. In Section 4.4, Theorem 26 generalizes Theorem 13, which states that Exact-5-DNP is DP-complete, by showing that certain exact domatic number problems are complete in the higher levels of the boolean hierarchy over NP. The open questions raised above for, e.g, Exact-i-DNP with $i \in \{3, 4\}$ apply to Theorem 26 as well, which is not optimal either. Section 4.5 proves the variants DNP-Odd, DNP-Equ, and DNP-Geq of the domatic number problem $\mathrm{P}_{||}^{\mathrm{NP}}$-complete.

In Section 5, we studied the exact conveyor flow shop problem using similar techniques. We proved that Exact-1-CFSP is DP-complete and Exact-k-CFSP is $\mathrm{BH}_{2k}(\mathrm{NP})$-complete. Note that in defining these problems, we do not specify a fixed set S_k with k fixed values as problem parameters; see Definition 31. Rather, only the cardinality k of such sets is given as a parameter, and S_k is part of the problem instance of Exact-k-CFSP. The reason is that the actual values of S_k depend on the input of the reduction f defined in the proof of Theorem 33. In particular, the number z_φ from Lemma 30, which is used to define the number $z = \sum_{i=1}^{2k} z_i$ in the proof of Theorem 33, has the following form (see [Esp01, Rie02]):

$$z_\varphi = 28n_K + 27n_{\overline{K}} + 8n_U + 90mt + 99m,$$

where t is the number of variables and m is the number of clauses of the given boolean formula φ, and n_K, $n_{\overline{K}}$, and n_U denote respectively the number of “coupling, inverting coupling, and interrupting elements” of the “minimum valid block cover” constructed in the Espelage reduction [Esp01] from 3-SAT to CFSP. It would be interesting to know whether one can obtain $\mathrm{BH}_{2k}(\mathrm{NP})$-completeness of Exact-$k$-CFSP even if a set S_k of k fixed values is specified a priori.

Acknowledgments. We are grateful to Gerd Wechsung for his interest in this paper, for many helpful conversations and, in particular, for pointing out the coNP-completeness of Exact-2-DNP. We also thank two anonymous referees whose comments and suggestions much helped to improve the presentation of this paper, and we thank Mitsunori Ogihara for his guidance during the editorial process.

References

[Adl78] L. Adleman. Two theorems on random polynomial time. In *Proceedings of the 19th IEEE Symposium on Foundations of Computer Science*, pages 75–83, 1978.

[AH92] E. Allender and L. Hemachandra. Lower bounds for the low hierarchy. *Journal of the ACM*, 39(1):234–251, 1992.

[BBJ+89] A. Bertoni, D. Bruschi, D. Joseph, M. Sitharam, and P. Young. Generalized boolean hierarchies and boolean hierarchies over RP. In *Proceedings of the 7th Conference on Fundamentals of Computation Theory*, pages 35–46. Springer-Verlag *Lecture Notes in Computer Science #380*, August 1989.

[BCO93] R. Beigel, R. Chang, and M. Ogiwara. A relationship between difference hierarchies and relativized polynomial hierarchies. *Mathematical Systems Theory*, 26(3):293–310, 1993.

[Bei91] R. Beigel. Bounded queries to SAT and the boolean hierarchy. *Theoretical Computer Science*, 84(2):199–223, 1991.

[Bei94] R. Beigel. Perceptrons, PP, and the polynomial hierarchy. *Computational Complexity*, 4(4):339–349, 1994.

[BH91] S. Buss and L. Hay. On truth-table reducibility to SAT. *Information and Computation*, 91(1):86–102, March 1991.

[BHW91] R. Beigel, L. Hemachandra, and G. Wechsung. Probabilistic polynomial time is closed under parity reductions. *Information Processing Letters*, 37(2):91–94, 1991.

[BKT94] H. Buhrman, J. Kadin, and T. Thierauf. On functions computable with nonadaptive queries to NP. In *Proceedings of the 9th Structure in Complexity Theory Conference*, pages 43–52. IEEE Computer Society Press, 1994.

[BLS99] A. Brandstädt, V. Le, and J. Spinrad. *Graph Classes: A Survey*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.

[Bon85] M. Bonuccelli. Dominating sets and dominating number of circular arc graphs. *Discrete Applied Mathematics*, 12:203–213, 1985.

[CGH$^+$88] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The boolean hierarchy I: Structural properties. *SIAM Journal on Computing*, 17(6):1232–1252, 1988.

[CGH$^+$89] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The boolean hierarchy II: Applications. *SIAM Journal on Computing*, 18(1):95–111, 1989.

[CK96] R. Chang and J. Kadin. The boolean hierarchy and the polynomial hierarchy: A closer connection. *SIAM Journal on Computing*, 25(2):340–354, April 1996.

[CM87] J. Cai and G. Meyer. Graph minimal uncolorability is D^{P}-complete. *SIAM Journal on Computing*, 16(2):259–277, April 1987.

[Esp01] W. Espelage. *Bewegungsminimierung in der Förderband-Flow-Shop-Verarbeitung*. PhD thesis, Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany, 2001. In German.

[EW00] W. Espelage and E. Wanke. Movement optimization in flow shop processing with buffers. *Mathematical Methods of Operations Research*, 51(3):495–513, 2000.

[EW01] W. Espelage and E. Wanke. A 3-approximation algorithmus for movement minimization in conveyor flow shop processing. In *Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science*, pages 363–374. Springer-Verlag *Lecture Notes in Computer Science #2136*, 2001.

[EW03] W. Espelage and E. Wanke. Movement minimization for unit distances in conveyor flow shop processing. *Mathematical Methods of Operations Research*, 57(2), 2003. To appear.

[Far84] M. Farber. Domination, independent domination, and duality in strongly chordal graphs. *Discrete Applied Mathematics*, 7:115–130, 1984.

[FHK00] U. Feige, M. Halldórsson, and G. Kortsarz. Approximating the domatic number. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 134–143. ACM Press, May 2000.

[Gil77] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6(4):675–695, 1977.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[GK00] V. Guruswami and S. Khanna. On the hardness of 4-coloring a 3-colorable graph. In *Proceedings of the 15th Annual IEEE Conference on Computational Complexity*, pages 188–197. IEEE Computer Society Press, May 2000.

[GNW90] T. Gundermann, N. Nasser, and G. Wechsung. A survey on counting classes. In *Proceedings of the 5th Structure in Complexity Theory Conference*, pages 140–153. IEEE Computer Society Press, July 1990.

[Gol80] M. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, 1980.

[GRW01] A. Große, J. Rothe, and G. Wechsung. Relating partial and complete solutions and the complexity of computing smallest solutions. In *Proceedings of the Seventh Italian Conference on Theoretical Computer Science*, pages 339–356. Springer-Verlag *Lecture Notes in Computer Science #2202*, October 2001.

[GRW02] A. Große, J. Rothe, and G. Wechsung. Computing complete graph isomorphisms and hamiltonian cycles from partial ones. *Theory of Computing Systems*, 35(1):81–93, February 2002.

[Hau14] F. Hausdorff. *Grundzüge der Mengenlehre*. Walter de Gruyten and Co., 1914.

[Hem89] L. Hemachandra. The strong exponential hierarchy collapses. *Journal of Computer and System Sciences*, 39(3):299–322, 1989.

[HHR97a] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, 44(6):806–825, November 1997.

[HHR97b] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Raising NP lower bounds to parallel NP lower bounds. *SIGACT News*, 28(2):2–13, June 1997.

[HHW99] L. Hemaspaandra, H. Hempel, and G. Wechsung. Query order. *SIAM Journal on Computing*, 28(2):637–651, 1999.

[HR97] L. Hemaspaandra and J. Rothe. Unambiguous computation: Boolean hierarchies and sparse Turing-complete sets. *SIAM Journal on Computing*, 26(3):634–653, June 1997.

[HR98] E. Hemaspaandra and J. Rothe. Recognizing when greed can approximate maximum independent sets is complete for parallel access to NP. *Information Processing Letters*, 65(3):151–156, February 1998.

[HRS02] E. Hemaspaandra, J. Rothe, and H. Spakowski. Recognizing when heuristics can approximate minimum vertex covers is complete for parallel access to NP. In *Proceedings of the 28th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2002)*, pages 258–269. Springer-Verlag *Lecture Notes in Computer Science #2573*, June 2002.

[HT98] P. Heggernes and J. Telle. Partitioning graphs into generalized dominating sets. *Nordic Journal of Computing*, 5(2):128–142, 1998.

[HW91] L. Hemachandra and G. Wechsung. Kolmogorov characterizations of complexity classes. *Theoretical Computer Science*, 83:313–322, 1991.

[JT95] B. Jenner and J. Torán. Computing functions with parallel queries to NP. *Theoretical Computer Science*, 141:175–193, 1995.

[Kad88] J. Kadin. The polynomial time hierarchy collapses if the boolean hierarchy collapses. *SIAM Journal on Computing*, 17(6):1263–1282, 1988. Erratum appears in the same journal, 20(2):404, 1991.

[Kad89] J. Kadin. $\mathrm{P}^{\mathrm{NP}[\log n]}$ and sparse Turing-complete sets for NP. *Journal of Computer and System Sciences*, 39(3):282–298, 1989.

[Ko89] K. Ko. Relativized polynomial time hierarchies having exactly k levels. *SIAM Journal on Computing*, 18(2):392–408, 1989.

[Ko91] K. Ko. On adaptive versus nonadaptive bounded query machines. *Theoretical Computer Science*, 82:51–69, 1991.

[Kre88] M. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36:490–509, 1988.

[KS94] H. Kaplan and R. Shamir. The domatic number problem on some perfect graph families. *Information Processing Letters*, 49(1):51–56, January 1994.

[KSW87] J. Köbler, U. Schöning, and K. Wagner. The difference and truth-table hierarchies for NP. *R.A.I.R.O. Informatique théorique et Applications*, 21:419–435, 1987.

[LS95] T. Long and M. Sheu. A refinement of the low and high hierarchies. *Mathematical Systems Theory*, 28(4):299–327, July/August 1995.

[Ogi94] M. Ogiwara. Generalized theorems on relationships among reducibility notions to certain complexity classes. *Mathematical Systems Theory*, 27(3):189–200, 1994.

[Ogi96] M. Ogihara. Functions computable with limited access to NP. *Information Processing Letters*, 58(1):35–38, May 1996.

[Pap94] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[PY84] C. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). *Journal of Computer and System Sciences*, 28(2):244–259, 1984.

[PZ83] C. Papadimitriou and S. Zachos. Two remarks on the power of counting. In *Proceedings of the 6th GI Conference on Theoretical Computer Science*, pages 269–276. Springer-Verlag *Lecture Notes in Computer Science #145*, 1983.

[Rie02] T. Riege. Vollständige Probleme in der Booleschen Hierarchie über NP. *Diploma thesis*, Heinrich-Heine-Universität Düsseldorf, Institut für Informatik, Düsseldorf, Germany, August 2002. In German.

[Rot03] J. Rothe. Exact complexity of Exact-Four-Colorability. *Information Processing Letters*, 87(1):7–12, July 2003.

[RSV03] J. Rothe, H. Spakowski, and J. Vogel. Exact complexity of the winner problem for Young elections. *Theory of Computing Systems*, 36(4):375–386, June 2003.

[Sch78] T. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th ACM Symposium on Theory of Computing*, pages 216–226. ACM Press, May 1978.

[Val76] L. Valiant. The relative complexity of checking and evaluating. *Information Processing Letters*, 5(1):20–23, 1976.

[Wag87] K. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51:53–80, 1987.

[Wag90] K. Wagner. Bounded query classes. *SIAM Journal on Computing*, 19(5):833–846, 1990.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'