Theoretical Computer Science 51 (1987) 53-80
North-Holland

MORE COMPLICATED QUESTIONS ABOUT MAXIMA AND MINIMA, AND SOME CLOSURES OF NP

Klaus W. WAGNER
Institut für Mathematik, Universität Augsburg, D-8900 Augsburg, Fed. Rep. Germany

Communicated by G. Mirkowska
Received June 1986
Revised September 1986

Abstract. Starting from NP-complete problems defined by questions of the kind 'max ... $\geqslant m$?' and 'min ... $\leqslant m$?' we consider problems defined by more complicated questions about these maxima and minima, as for example 'max ... $= m$?'. 'min ... $\in M$?' and 'is max ... odd?'. This continues a work started by Papadimitriou and Yannakakis (1982). It is shown that these and other problems are complete in certain subclasses of the Boolean closure of NP and other classes in the interesting area below the class Δ_2^P of the polynomial-time hierarchy. Special methods are developed to prove such completeness results. For this it is necessary to establish some properties of the classes in question which might be interesting in their own right.

1. Introduction

Many of the NP-complete problems investigated so far (see [3]) are defined using the maximum or minimum relation in the following way. Let Σ be a finite alphabet, let the *property* $P_A : \Sigma^* \times \Sigma^* \to \{0, 1\}$ and the *valuation function* $\beta_A : \Sigma^* \times \Sigma^* \to \mathbb{N}$ be polynomial-time computable functions, and let

$$I_A(w) = \{\beta_A(w, v) : v \in \Sigma^*, |v| \leqslant |w| \text{ and } P_A(w, v) = 1\}.$$

Then the *optimum problems*

$$A = \{(w, m) : w \in \Sigma^*, m \in \mathbb{N} \text{ and } \max I_A(w) \geqslant m\},$$

$$A = \{(w, m) : w \in \Sigma^*, m \in \mathbb{N} \text{ and } \min I_A(w) \leqslant m\}$$

are obviously in NP. And, in fact, many problems defined in this way are NP-complete. As examples take

CLIQUE $= \{(G, m) : G \text{ graph}, m \in \mathbb{N} \text{ and } \max I_{CL}(G) \geqslant m\}$,

MAX SAT ASG $= \{(H, m) : H \text{ Boolean formula}, m \in \mathbb{N} \text{ and } \max I_{MSA}(H) \geqslant m\}$,

TRAVELING SALESMAN $= \{(d, m) : d \text{ distance function}, m \in \mathbb{N} \text{ and } \min I_{TS}(d) \leqslant m\}$, where

$$I_{\mathrm{CL}}(G)=\{\text{card } C: C \text{ is a clique in } G\},$$

$$I_{\mathrm{MSA}}(H)=\left\{\sum_{i=0}^{n}\alpha_i\cdot 2^i:\alpha_0,\ldots,\alpha_n\in\{0,1\} \text{ and } H(\alpha_0,\ldots,\alpha_n)=1\right\},$$

$$I_{\mathrm{TS}}(d)=\left\{\sum_{i=1}^{n} d(\pi(i),\pi(i+1)):\pi \text{ permutation of } (1,2,\ldots,n)\right\}$$

$$(\text{let } \pi(n+1)=\pi(1)).$$

Indeed, the question 'max $I_A(w)\geq m$?' and 'min $I_A(w)\leq m$?' are the simplest questions which can be asked about max $I_A(w)$ and min $I_A(w)$. In the present paper we investigate the complexities of combinatorial problems based on more complicated questions about opt $I_A(w)$ (here and in what follows "opt" stands for "max" or "min") as for example 'opt $I_A(w)=m$?', 'opt $I_A(w)\in\{a_1,\ldots,a_k\}$?' and 'is opt $I_A(w)$ odd?'. For a given optimum problem A, we define

$$A_k=\{(w,a_1,\ldots,a_k): w\in\Sigma^*, a_1,\ldots,a_k\in\mathbb{N} \text{ and opt } I_A(w)\in\{a_1,\ldots,a_k\}\},\quad k\geq 1,$$

$$A_+=\{(w,a_1,\ldots,a_k): w\in\Sigma^*, k, a_1,\ldots,a_k\in\mathbb{N} \text{ and opt } I_A(w)\in\{a_1,\ldots,a_k\}\},$$

$$A'_k=\left\{(w,a_1,\ldots,a_k,b_1,\ldots,b_k): w\in\Sigma^*, a_1,\ldots,a_k,b_1,\ldots,b_k\in\mathbb{N} \text{ and opt } I_A(w)\in\bigcup_{i=1}^{k}[a_i,b_i]\right\},\quad k\geq 1,$$

$$A'_+=\left\{(w,a_1,\ldots,a_k,b_1,\ldots,b_k): w\in\Sigma^*, k, a_1,\ldots,a_k,b_1,\ldots,b_k\in\mathbb{N} \text{ and opt } I_A(w)\in\bigcup_{i=n}^{k}[a_i,b_i]\right\},$$

$$A_{\mathrm{odd}}=\{w: w\in\Sigma^* \text{ and max } I_A(w) \text{ is odd}\}.$$

In [14] it was proved that the problem CLIQUE$_1$ is complete in $\boldsymbol{D}^P=\mathbf{NP}\wedge\mathbf{coNP}$ (for classes $\boldsymbol{M}$ and $\boldsymbol{N}$ of languages we define: $\boldsymbol{M}\wedge\boldsymbol{N}=\{A\cap B: A\in\boldsymbol{M} \text{ and } B\in\boldsymbol{N}\}$, $\boldsymbol{M}\vee\boldsymbol{N}=\{A\cup B: A\in\boldsymbol{M} \text{ and } B\in\boldsymbol{N}\}$ and $\mathbf{co}\boldsymbol{M}=\{\bar{A}: A\in\boldsymbol{M}\}$). To generalize this result to CLIQUE$_k$ for arbitrary $k\geq 1$ and to establish the completeness results for the other problems we need

- the subclasses $\boldsymbol{C}_k^{\mathrm{NP}}$ and $\boldsymbol{D}_k^{\mathrm{NP}}$ ($k\geq 1$) of the Boolean closure of NP which are defined by $\boldsymbol{C}_1^{\mathrm{NP}}=\mathbf{coNP}$, $\boldsymbol{D}_1^{\mathrm{NP}}=\mathbf{NP}$, $\boldsymbol{C}_{k+1}^{\mathrm{NP}}=\boldsymbol{D}_k^{\mathrm{NP}}\wedge\mathbf{coNP}$ and $\boldsymbol{D}_{k+1}^{\mathrm{NP}}=\boldsymbol{C}_k^{\mathrm{NP}}\vee\mathbf{NP}$;
- the class $\boldsymbol{P}_{\mathrm{bf}}^{\mathrm{NP}}$ of all sets reducible to an NP set by polynomial-time Boolean formula reducibility (a restriction of the polynomial-time truth-table reducibility introduced in [9]; and
- the class $\boldsymbol{P}^{\mathrm{NP}}=\Delta_2^p$ of all sets reducible to an NP set by polynomial-time Turing reducibility.

As an important tool for proving our completeness results we need some results on the classes $\boldsymbol{C}_{2k}^{\mathrm{NP}}$ (taken from [17, 18]) and on the class $\boldsymbol{P}_{\mathrm{bf}}^{\mathrm{NP}}$ (developed in the present paper). These results might be interesting in their own right.

It turns out that there are three categories of NP-complete optimum problems A which differ in the complexities of the derivated problems A_k, A'_k, A_+, A'_+ and A_{odd}. The problems CLIQUE, MAX SAT ASG and TRAVELING SALESMAN are representants of these three categories (see Table 2 in Section 9). Such a different behavior of different NP-complete optimum problems can be explained by different properties of the valuation functions.

The completeness results summarized in Table 2 show that the present paper meets with a growing recent interest in the area between the complexity classes NP and $\boldsymbol{P}^{\text{NP}} = \Delta_2^p$. Many natural $\boldsymbol{D}^P$-complete problems have been exhibited in [2, 13, 14, 19]. The Boolean NP-hierarchy, which consists of the classes $\boldsymbol{C}_k^{\text{NP}}$ and $\boldsymbol{D}_k^{\text{NP}}$, for $k \geq 1$, has been introduced and investigated independently in [1, 7, 17]. Complete sets for the classes of the Boolean hierarchy have been exhibited in [1, 7]. However, the complete sets for these classes investigated in the present paper are much more natural. Finally, let us mention that our result on the Δ_2-completeness of the problem of whether the optimal tour length for a traveling-salesman instance is odd is close to a result in [12].

2. The Boolean closure of NP

Let $\boldsymbol{P}$ (NP) be the class of all sets which can be accepted by deterministic (nondeterministic) polynomial-time bounded Turing machines.

The Boolean closure of NP (for short BC(NP)) is the smallest class containing NP and being closed under union, intersection and complementation. Starting from NP one can construct new subclasses of BC(NP) using the operations $\wedge$, $\vee$ and **co**. For classes $\boldsymbol{A}$ and $\boldsymbol{B}$ of sets we define

$$\boldsymbol{A} \wedge \boldsymbol{B} = \{A \cap B : A \in \boldsymbol{A} \text{ and } B \in \boldsymbol{B}\},$$

$$\boldsymbol{A} \vee \boldsymbol{B} = \{A \cup B : A \in \boldsymbol{A} \text{ and } B \in \boldsymbol{B}\},$$

$$\mathbf{co}\boldsymbol{A} = \{\bar{A} : A \in \boldsymbol{A}\}.$$

The *Boolean* NP *-hierarchy* is the smallest family of classes containing NP as an element and being closed under the operations $\wedge$, $\vee$ and **co**. It is obvious that

$$\text{BC(NP)} = \{A : A \text{ belongs to some class of the Boolean NP-hierarchy}\}.$$

From the many classes of the Boolean NP-hierarchy we separate the classes $\boldsymbol{C}_k^{\text{NP}}$, and $\boldsymbol{D}_k^{\text{NP}}$ for $k \geq 1$ defined by

$$\boldsymbol{C}_{2k-1}^{\text{NP}} = \text{coNP} \vee \bigvee_{i=1}^{k-1} (\text{NP} \wedge \text{coNP}), \qquad \boldsymbol{D}_{2k-1}^{\text{NP}} = \text{NP} \vee \bigvee_{i=1}^{k-1} (\text{NP} \wedge \text{coNP}),$$

$$\boldsymbol{C}_{2k}^{\text{NP}} = \bigvee_{i=1}^{k} (\text{NP} \wedge \text{coNP}), \qquad \boldsymbol{D}_{2k}^{\text{NP}} = \text{NP} \vee \text{coNP} \vee \bigvee_{i=1}^{k-1} (\text{NP} \wedge \text{coNP}).$$

The class $\boldsymbol{C}_2^{\text{NP}}$ has been called $\boldsymbol{D}^P$ in [14].

Theorem 2.1 (Wagner [18]). *Let* $k \geq 1$.

(1) *Every class of the Boolean* NP-*hierarchy coincides with one of the classes* C_1^{NP}, D_1^{NP}, C_2^{NP}, D_2^{NP}, C_3^{NP}, $D_3^{NP}, \ldots$.

(2) $D_k^{NP} = \mathrm{co}C_k^{NP}$.

(3) $C_{k+1}^{NP} = D_k^{NP} \wedge \mathrm{coNP}$ and $D_{k+1}^{NP} = C_k^{NP} \vee \mathrm{NP}$.

(4) $C_k^{NP} \cup D_k^{NP} \subseteq C_{k+1}^{NP} \cap D_{k+1}^{NP}$.

Thus the Boolean NP-hierarchy has the same structure as the polynomial-time hierarchy (see [16]). However, the Boolean NP-hierarchy is included in the class Δ_2^p of the polynomial-time hierarchy (see Fig. 1 at the end of Section 3). As the classes of the polynomial-time hierarchy, the classes of the Boolean NP-hierarchy are closed under polynomial-time many-one reducibility.

Independently, in [7], two hierarchies are introduced and investigated which start with NP and which have BC(NP) as the union of all classes. The *difference hierarchy* coincides with the sequence $D_1^{NP}, C_2^{NP}, D_3^{NP}, C_4^{NP}, D_5^{NP}, \ldots$. The *truth-table* hierarchy is a sequence of classes whose kth level includes $C_k^{NP} \cup D_k^{NP}$ and is included in $C_{k+1}^{NP} \cap D_{k+1}^{NP}$. For an updated and improved version of this paper see [8]. Also independently, in [1], a hierarchy is introduced and investigated which coincides also with the sequence $D_1^{NP}, C_2^{NP}, D_3^{NP}, C_4^{NP}, D_5^{NP}, \ldots$.

The following theorem gives a simplified representation of the sets in the several classes of the Boolean NP-hierarchy. This theorem can be understood as the quantitative version of a theorem by Hausdorff (see [5]) stating, for every class K which is closed under union and intersection, that every set from the Boolean closure of K can be presented in the form $\bigcup_{i=1}^{k} (A_{2i-1} \cap \overline{A_{2i}})$ such that $A_1, A_2, \ldots, A_{2k} \in K$ and $A_1 \supseteq A_2 \supseteq \cdots \supseteq A_{2k}$ ('Hausdorff set differences').

Theorem 2.2 (Wagner and Wechsung [17]). *Let* $k \geq 1$.

(1) *A set A is in* C_{2k-1}^{NP} *if and only if there exist sets* $A_1, A_2, \ldots, A_{2k-1} \in \mathrm{NP}$ *such that* $A_1 \supseteq A_2 \supseteq \cdots \supseteq A_{2k-1}$ *and* $A = \overline{A_1} \cup \bigcup_{i=1}^{k-1} (A_{2i} \cap \overline{A_{2i+1}})$.

(2) *A set A is in* D_{2k-1}^{NP} *if and only if there exist sets* $A_1, A_2, \ldots, A_{2k-1} \in \mathrm{NP}$ *such that* $A_1 \supseteq A_2 \supseteq \cdots \supseteq A_{2k-1}$ *and* $A = \bigcup_{i=1}^{k-1} (A_{2i-1} \cap \overline{A_{2i}}) \cap A_{2k-1}$.

(3) *A set A is in* C_{2k}^{NP} *if and only if there exist* $A_1, A_2, \ldots, A_{2k} \in \mathrm{NP}$ *such that* $A_1 \supseteq A_2 \supseteq \cdots \supseteq A_{2k}$ *and* $A = \bigcup_{i=1}^{k} (A_{2i-1} \cap \overline{A_{2i}})$.

(4) *A set A is in* D_{2k}^{NP} *if and only if there exist sets* $A_1, A_2, \ldots, A_{2k} \in \mathrm{NP}$ *such that* $A_1 \supseteq A_2 \supseteq \cdots \supseteq A_{2k}$ *and* $A = \overline{A_1} \cup \bigcup_{i=1}^{k-1} (A_{2i} \cap \overline{A_{2i+1}}) \cup A_{2k}$.

3. Reducibility closures of NP

In this section we consider the closure of NP with respect to some deterministic polynomial-time reducibilities.

The *polynomial-time many-one reducibility* is denoted by $\leq_m^p$. Obviously, NP is closed under $\leq_m^p$.

The closure of NP with respect to the *polynomial-time Turing reducibility* $\leq_T^p$ is denoted by P^{NP}, or equivalently, by Δ_2^p. In other words, P^{NP} is the class of all sets which can be accepted by a polynomial-time Turing machine with an oracle from NP.

The notion of *polynomial-time truth-table reducibility* was originally defined in [10]. From several equivalent definitions given there, we choose the following. Let c_A be the characteristic function of A. Define $A \leq_{tt}^p B$ iff there exists a polynomial-time computable function f such that $c_A(x) = h_N(c_B(y_1), \ldots, c_B(y_k))$, where $f(x) = (N, y_1, \ldots, y_k)$, N is a natural encoding of a Boolean circuit with $\wedge$, $\vee$ and $\neg$ gates, and h_N is the Boolean function realized by this circuit. By P_{tt}^{NP} we denote the closure of NP with respect to $\leq_{tt}^p$. It has been shown in [10] that $\leq_{tt}^p$ is properly stronger than $\leq_T^p$; however, we do not know whether P_{tt}^{NP} is properly included in P^{NP}.

Using in the above definition for $\leq_{tt}^p$ Boolean formulas in $\wedge$, $\vee$ and $\neg$ instead of Boolean circuits we obtain the *polynomial-time Boolean formula reducibility* $\leq_{bf}^p$. By P_{bf}^{NP} we denote the closure of NP with respect to $\leq_{bf}^p$. It is obvious that $P_{bf}^{NP} \subseteq P_{tt}^{NP}$, but it is not known whether this inclusion is strict. For a discussion of this problem see [9].

As a special case of the polynomial-time Boolean formula reducibility we define the *polynomial-time Hausdorff reducibility* by: $A \leq_{hd}^p B$ iff there exists a polynomial-time computable function f such that

$$c_A(x) = (c_B(x_1) \wedge \neg c_B(x_2)) \vee \cdots \vee (c_B(x_{2k-1}) \wedge \neg c_B(x_{2k})) \quad \text{and}$$

$$c_B(x_1) \geq c_B(x_2) \geq \cdots \geq c_B(x_{2k}),$$

where $f(x) = (x_1, x_2, \ldots, x_{2k})$. Note that k depends on x. By P_{hd}^{NP} we denote the closure of NP with respect to $\leq_{hd}^p$. We do not know whether $\leq_{bf}^p$ and $\leq_{hd}^p$ generally coincide. However, we are able to prove $P_{hd}^{NP} = P_{bf}^{NP}$.

To do so we need the following lemma. Let N be a Boolean circuit with input nodes $z_1, \ldots, z_r$; let b be a node of N, and let $a_1, \ldots, a_r \in \{0, 1\}$. By $N_b(a_1, \ldots, a_r)$ we denote the value of N at node b when $a_1, \ldots, a_r$ are given as input to the input nodes $z_1, \ldots, z_r$.

Lemma 3.1. *There exists a polynomial-time algorithm computing, for every Boolean formula H with operations* $\wedge$, $\vee$ *and* $\neg$, *a Boolean circuit N with* $\wedge$, $\vee$, 0 *and* 1 *nodes, an* $s \geq 1$ *and nodes* $b_1, \ldots, b_{2s}$ *of N such that, for all* $a_1, \ldots, a_r \in \{0, 1\}$,

$$\text{(1)} \qquad H(a_1, \ldots, a_r) = \bigvee_{i=1}^{s} (N_{b_{2i-1}}(a_1, \ldots, a_r) \wedge \neg N_{b_{2i}}(a_1, \ldots, a_r));$$

$$\text{(2)} \qquad N_{b_i}(a_1, \ldots, a_r) \geq N_{b_{i+1}}(a_1, \ldots, a_r) \qquad \textit{for all } i = 1, \ldots, 2s-1.$$

Proof. Let H be a Boolean formula with the variables $z_1, \ldots, z_r$. We inductively construct, for all subformulas of H, Boolean circuits with the properties of the lemma. Actually, we construct only one Boolean circuit which is enlarged in every step of the induction. This Boolean circuit has the input nodes $z_1, \ldots, z_r$ and two further nodes e_0 and e_1 of indegree zero which are labeled with 0 and 1 respectively.

For the atomic subformulas of H, the variables z_i, we have

$$a_i = N_{z_i}(a_1, \ldots, a_r) \wedge \neg N_{e_0}(a_1, \ldots, a_r) \quad \text{and}$$

$$N_{z_i}(a_1, \ldots, a_r) \geq 0 = N_{e_0}(a_1, \ldots, a_r).$$

Let $H(z_1, \ldots, z_r) \equiv \neg H'(z_1, \ldots, z_r)$, and let $b_1, \ldots, b_{2s}$ be nodes of N such that

$$H'(a_1, \ldots, a_r) = \bigvee_{i=1}^{s} (N_{b_{2i-1}}(a_1, \ldots, a_r) \wedge \neg N_{b_{2i}}(a_1, \ldots, a_r)),$$

$$N_{b_i}(a_1, \ldots, a_r) \geq N_{b_{i+1}}(a_1, \ldots, a_r) \quad \text{for all } i = 1, \ldots, 2s-1.$$

Consequently,

$$\begin{aligned} H(a_1, \ldots, a_r) = {} & (N_{e_1}(a_1, \ldots, a_r) \wedge \neg N_{b_1}(a_1, \ldots, a_r)) \\ & \vee \bigvee_{i=1}^{s-1} (N_{b_{2i}}(a_1, \ldots, a_r) \wedge \neg N_{b_{2i+1}}(a_1, \ldots, a_r)) \\ & \vee (N_{b_{2s}}(a_1, \ldots, a_r) \wedge \neg N_{e_0}(a_1, \ldots, a_r)) \end{aligned}$$

and

$$N_{e_1}(a_1, \ldots, a_r) = 1 \geq N_{b_1}(a_1, \ldots, a_r),$$

$$N_{b_{2s}}(a_1, \ldots, a_r) \geq 0 = N_{e_0}(a_1, \ldots, a_r).$$

Let $H(z_1, \ldots, z_r) \equiv (H_1(z_1, \ldots, z_r) \wedge H_2(z_1, \ldots, z_r))$ and let $b_1, \ldots, b_{2s}, c_1, \ldots, c_{2t}$ be nodes of N such that

$$H_1(a_1, \ldots, a_r) = \bigvee_{i=1}^{s} (N_{b_{2i-1}}(a_1, \ldots, a_r) \wedge \neg N_{b_{2i}}(a_1, \ldots, a_r))$$

$$\text{and } N_{b_i}(a_1, \ldots, a_r) \geq N_{b_{i+1}}(a_1, \ldots, a_r) \quad \text{for all } i = 1, \ldots, 2s-1,$$

$$H_2(a_1, \ldots, a_r) = \bigvee_{i=1}^{t} (N_{c_{2j-1}}(a_1, \ldots, a_r) \wedge \neg N_{c_{2j}}(a_1, \ldots, a_r))$$

$$\text{and } N_{c_j}(a_1, \ldots, a_r) \geq N_{c_{j+1}}(a_1, \ldots, a_r) \quad \text{for all } j = 1, \ldots, 2t-1.$$

Defining

$$A = \{(a_1, \ldots, a_r) : H(a_1, \ldots, a_r) = 1\},$$

$$B = \{(a_1, \ldots, a_r) : H_1(a_1, \ldots, a_r) = 1\},$$

$$C = \{(a_1, \ldots, a_r) : H_2(a_1, \ldots, a_r) = 1\},$$

$$B_i = \{(a_1, \ldots, a_r) : N_{b_i}(a_1, \ldots, a_r) = 1\} \quad \text{for } i = 1, \ldots, 2s,$$

$$C_j = \{(a_1, \ldots, a_r) : N_{c_j}(a_1, \ldots, a_r) = 1\} \quad \text{for } j = 1, \ldots, 2t$$

we obtain

$$B=\bigcup_{i=1}^{s}(B_{2i-1}\cap\overline{B_{2i}})\quad\text{and}\quad B_i\supseteq B_{i+1}\quad\text{for } i=1,\ldots,2s-1,$$

$$C=\bigcup_{j=1}^{t}(C_{2j-1}\cap\overline{C_{2j}})\quad\text{and}\quad C_j\supseteq C_{j+1}\quad\text{for } j=1,\ldots,2t-1.$$

Now we can conclude

$$\begin{aligned}
A&=B\cap C\\
&=\left(\bigcup_{i=1}^{s}(B_{2i-1}\cap\overline{B_{2i}})\right)\cap\left(\bigcup_{i=1}^{t}(C_{2j-1}\cap\overline{C_{2j}})\right)\\
&=\bigcup_{\substack{i=1,\ldots,s\\ j=1,\ldots,t}}(B_{2i-1}\cap\overline{B_{2i}}\cap C_{2j-1}\cap\overline{C_{2j}})\\
&=\bigcup_{k=2}^{s+t}\bigcup_{i+j=k}(B_{2i-1}\cap C_{2j-1}\cap\overline{B_{2i}}\cap\overline{C_{2j}})\cup\bigcap_{\substack{m+n=k\\ m>i}}(B_{2i-1}\cap C_{2j-1}\cap\overline{B_{2i}})\\
&\quad\cap\bigcap_{\substack{m+n=k\\ m<i}}(B_{2i-1}\cap C_{2j-1}\cap\overline{C_{2j}})\\
&=\bigcup_{k=2}^{s+t}\bigcup_{i+j=k}(B_{2i-1}\cap C_{2j-1}\cap\overline{B_{2i}}\cap\overline{C_{2j}})\cap\bigcap_{\substack{m+n=k\\ m>i}}(B_{2i-1}\cap C_{2j-1}\cap\overline{B_{2m-1}})\\
&\quad\cap\bigcap_{\substack{m+n=k\\ m<i}}(B_{2i-1}\cap C_{2j-1}\cap\overline{C_{2n-1}})\\
&=\bigcup_{k=2}^{s+t}\bigcup_{i+j=k}(B_{2i-1}\cap C_{2i-1}\cap(\overline{B_{2i}}\cup\overline{C_{2j-1}})\cap(\overline{C_{2j}}\cup\overline{B_{2i-1}}))\\
&\quad\cap\bigcap_{\substack{m+n=k\\ m<i}}(B_{2i-1}\cap C_{2j-1}\cap(\overline{B_{2m}}\cup\overline{C_{2n-1}})\cap(\overline{C_{2n}}\cup\overline{B_{2m-1}}))\\
&\quad\cap\bigcap_{\substack{m+n=k\\ m>i}}(B_{2i-1}\cap C_{2j-1}\cap(\overline{B_{2m}}\cup\overline{C_{2n-1}})\cap(\overline{C_{2n}}\cup\overline{B_{2m-1}}))\\
&=\bigcup_{k=2}^{s+t}\bigcup_{i+j=k}\left(B_{2i-1}\cap C_{2j-1}\cap\bigcap_{m+n=k}((\overline{B_{2m}}\cup\overline{C_{2n-1}})\cap(\overline{C_{2n}}\cup\overline{B_{2m-1}}))\right)\\
&=\bigcup_{k=2}^{s+t}\left(\bigcup_{i+j=k}(B_{2i-1}\cap C_{2j-1})\cap\bigcap_{m+n=k}((\overline{B_{2m}}\cup\overline{C_{2n-1}})\cap(\overline{C_{2n}}\cup\overline{B_{2m-1}}))\right)\\
&=\bigcup_{k=2}^{s+t}\left(\bigcup_{i+j=k}(B_{2i-1}\cap C_{2j-1})\cap\overline{\bigcup_{m+n=k}((B_{2m}\cap C_{2n-1})\cup(C_{2n}\cap B_{2m-1}))}\right)\\
&=\bigcup_{k=1}^{s+t-1}(A_{2k-1}\cap\overline{A_{2k}}),
\end{aligned}$$

where we define, for $k=1,\ldots,s+t-1$,

$$A_{2k-1}=\bigcup_{i+j=k+1}(B_{2i-1}\cap C_{2j-1})\quad\text{and}$$

$$A_{2k}=\bigcup_{i+j=k+1}((B_{2i}\cap C_{2j-1})\cup(C_{2j}\cap B_{2i-1})).$$

Obviously, we have $A_k\supseteq A_{k+1}$ for $k=1,\ldots,2s+2t-3$.

Now we can easily add some nodes to N including the nodes $d_1,\ldots,d_{2s+2t-2}$ such that, for $k=1,\ldots,s+t-1$,

$$N_{d_{2k-1}}(a_1,\ldots,a_r)=\bigvee_{i+j=k+1}(N_{b_{2i-1}}(a_1,\ldots,a_r)\wedge N_{c_{2j-1}}(a_1,\ldots,a_r)),$$

$$N_{d_{2k}}(a_1,\ldots,a_r)=\bigvee_{i+j=k+1}((N_{b_{2i}}(a_1,\ldots,a_r)\wedge N_{c_{2j-1}}(a_1,\ldots,a_r)) \vee(N_{c_{2j}}(a_1,\ldots,a_r)\wedge N_{b_{2i-1}}(a_1,\ldots,a_r))).$$

Evidently, $A_k=\{(a_1,\ldots,a_r):N_{d_k}(a_1,\ldots,a_r)=1\}$ and therefore,

$$\begin{aligned}H(a_1,\ldots,a_r)=1 &\text{ iff } (a_1,\ldots,a_r)\in A\\ &\text{ iff } (a_1,\ldots,a_r)\in\bigcup_{k=1}^{s+t-1}(A_{2k-1}\cap\overline{A_{2k}})\\ &\text{ iff } \bigvee_{k=1}^{s+t-1}N_{d_{2k-1}}(a_1,\ldots,a_r)\wedge\neg N_{d_{2k}}(a_1,\ldots,a_r)=1.\end{aligned}$$

Furthermore, $A_k\supseteq A_{k+1}$ implies $N_{d_k}(a_1,\ldots,a_r)\geqslant N_{d_{k+1}}(a_1,\ldots,a_r)$.

Finally, let $H(x_1,\ldots,x_r)\equiv(H_1(x_1,\ldots,x_r)\wedge H_2(x_1,\ldots,x_r))$. By De Morgan's law, this case can be reduced to the cases already treated.

To see that the size of N is polynomially bounded in the length of H, observe that, in each step of the induction, the number s in the lemma is bounded by $|H|$ □

Theorem 3.2. $P^{\mathrm{NP}}_{\mathrm{hd}}=P^{\mathrm{NP}}_{\mathrm{bf}}$.

Proof. We have to show that, for given sets A and B such that $A\leqslant^p_{\mathrm{bf}}B$ and $B\in\mathrm{NP}$, there exist a $B'\in\mathrm{NP}$ such that $A\leqslant^p_{\mathrm{hd}}B'$. Let $A\leqslant^p_{\mathrm{bf}}B$ and $B\in\mathrm{NP}$. There exists a polynomial-time computable function f such that $c_A(x)=h_H(c_B(x_1),\ldots,c_B(x_r))$ for all $x\in\Sigma^*$ where $f(x)=(H,x_1,\ldots,x_r)$ and H is a Boolean formula with operations $\wedge$, $\vee$ and $\neg$. By Lemma 3.1, there exists a polynomial-time computable function g which yields $g(H)=(N,s,b_1,\ldots,b_{2s})$ such that N is a Boolean circuit with $\wedge$, $\vee$, 0 and 1 nodes: $b_1,\ldots,b_{2s}$ are nodes of N and, for all $a_1,\ldots,a_r\in\{0,1\}$,

$$H(a_1,\ldots,a_r)=\bigvee_{i=1}^{s}(N_{b_{2i-1}}(a_1,\ldots,a_r)\wedge\neg N_{b_{2i}}(a_1,\ldots,a_r)),$$

$$\text{and } N_{b_i}(a_1,\ldots,a_r)\geqslant N_{b_{i+1}}(a_1,\ldots,a_r)\quad\text{for all } i=1,\ldots,2s-1.$$

Now define

$$B'=\{(N,b,x_1,\ldots,x_r): N \text{ is a Boolean circuit with } \wedge,\vee, 0 \text{ and } 1 \text{ nodes, } b \text{ is a node of } N \text{ and } N_b(c_B(x_1),\ldots,c_B(x_r))=1\}.$$

Since the Boolean circuits in B' have only monotonic Boolean operations in the nodes, the set B' is in NP. We conclude

$$\begin{aligned} x \in A \text{ iff } & H(c_B(x_1), \ldots, c_B(x_r)) = 1 \\ \text{iff } & \bigvee_{i=1}^{s} (N_{b_{2i-1}}(c_B(x_1), \ldots, c_B(x_r)) \wedge \neg N_{b_{2i}}(c_B(x_1), \ldots, c_B(x_r))) = 1 \\ \text{iff } & \bigvee_{i=1}^{s} (c_{B'}(N, b_{2i-1}, x_1, \ldots, x_r) \wedge \neg c_{B'}(N, b_{2i}, x_1, \ldots, x_r)) = 1 \end{aligned}$$

and

$$\begin{aligned} c_{B'}(N, b_{2i-1}, x_1, \ldots, x_r) &= N_{b_{2i-1}}(c_B(x_1), \ldots, c_B(x_r)) \geq N_{b_{2i}}(c_B(x_1), \ldots, c_B(x_r)) \\ &= c_{B'}(N, b_{2i}, x_1, \ldots, x_r). \qquad \square \end{aligned}$$

We conclude this section with Fig. 1, showing the inclusional relationships between the classes mentioned above. Neither of these inclusions is known to be proper.

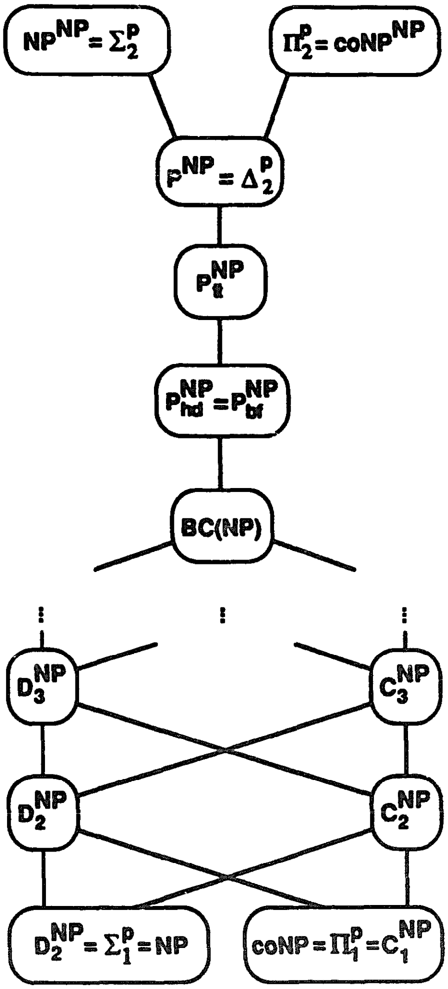

Fig. 1.

4. Upper bounds

To estimate the complexity of problems of the form A_k, A'_k, A_+, A'_+ and A_{odd} in terms of completeness in certain classes, we first have to answer the question to which classes these problems can belong. This will be done in this section.

Let A be an optimum problem defined by the property P_A and the valuation function β_A (which are assumed to be polynomial-time computable). The function β_A is said to be *polynomially bounded* iff there exists a polynomial p such that $\beta_A(w, v) \leq p(|w|)$ for all $w, v \in \Sigma^*$. The function β_A is said to be *polynomial-time invertible* iff the mapping $(w, k) \to \{v : v \in \Sigma^* \text{ and } \beta_A(w, v) = k\}$ is polynomial-time computable. Note that β_A cannot be both polynomially bounded and polynomial-time invertible.

Theorem 4.1. *Let A be an optimum problem (with polynomial-time computable P_A and β_A) and let $k \geq 1$.*

(1) $A \in \mathbf{NP}$.
(2) $A_k \in C_{2k}^{\mathbf{NP}}$. *If β_A is polynomial-time invertible, then* $A_k \in \mathbf{coNP}$.
(3) $A'_k \in C_{2k}^{\mathbf{NP}}$.
(4) $A_+ \in P_{\mathrm{bf}}^{\mathbf{NP}}$. *If β_A is polynomial-time invertible, then* $A_+ \in \mathbf{coNP}$.
(5) $A'_+ \in P_{\mathrm{bf}}^{\mathbf{NP}}$.
(6) $A_{\mathrm{odd}} \in P^{\mathbf{NP}}$. *If β_A is polynomially bounded, then* $A_{\mathrm{odd}} \in P_{\mathrm{bf}}^{\mathbf{NP}}$.

Proof. (1) is obvious.

(2): For polynomial-time invertible β_A, this is a special case of (4). For other β_A, this is a special case of (3).

(3): We restrict ourselves to the case that A is a maximum problem. The minimum case is proved analogously. Because of the representation

$$
\begin{aligned}
(w, a_1, \ldots, a_k, b_1, \ldots, b_k) \in A'_k &\leftrightarrow \bigvee_{i=1}^{k} (\max I_A(w) \in [a_i, b_i]) \\
&\leftrightarrow \bigvee_{i=1}^{k} (\max I_A(w) \geq a_i \wedge \max I_A(w) < b_i - 1) \\
&\leftrightarrow \bigvee_{i=1}^{k} ((w, a_i) \in A \wedge (w, b_i - 1) \notin A)
\end{aligned}
$$

and the fact that $A \in \mathrm{NP}$, we can conclude $A'_k \in C_{2k}^{\mathbf{NP}}$.

(4): Let β_A be polynomial-time invertible. Define

$$g(w, a) := \{v : |v| \leq |w|, \beta_A(w, v) = a \text{ and } P_A(w, v) = 1\};$$

$$f(w, a_1, \ldots, a_k) = \begin{cases} a_{i_0} & \text{if } i_0 = \max\{i : g(w, a_i) \neq \emptyset\}, \\ -1 & \text{if } g(w, a_i) = \emptyset \text{ for all } i = 1, \ldots, k. \end{cases}$$

Since β_A is polynomial-time invertible, the functions g and f are polynomial-time computable. Because of the representation

$$(w, a_1, \ldots, a_k) \in A_+ \leftrightarrow \bigwedge_{|v| \leq |w|} (P_A(w, v) = 1 \rightarrow \beta_A(w, v) \leq f(w, a_1, \ldots, a_k)),$$

we have $A_+ \in \mathbf{coNP}$.

For arbitrary β_A, statement (4) is a special case of (5).

(5) is obvious by the following representation (see the proof of (3))

$$(w, a_1, \ldots, a_k, b_1, \ldots, b_k) \in A'_+ \leftrightarrow \bigvee_{i=1}^{k} ((w, a_i) \in A \wedge (w, b_i - 1) \notin A).$$

(6): It is evident that A_{odd} can be reduced to the problem of computing $\max I_A(w)$ (for maximum problems, and analogously $\min I_A(w)$ for minimum problems). Thus we describe a polynomial-time algorithm which uses A as an oracle and computes $\max I_A(w)$. Since β_A is polynomial-time computable, there exists a polynomial p such that $\max I_A(w) \leq 2^{p(|w|)}$. First, the oracle is asked for $\max I_A(w) \geq 2^{p(|w|)-1}$. If the answer is "no", the oracle is asked for $\max I_A(w) \geq 2^{p(|w|)-2}$, otherwise it is asked for $\max I_A(w) \geq 2^{p(|w|)-1} + 2^{p(|w|)-2}$. In such a manner the interval for $\max I_A(w)$ is halved in every step. Thus, after $p(|w|)$ steps, the exact value of $\max I_A(w)$ is estimated.

If β_A is polynomially bounded, then there exists a polynomial p such that $\beta_A(w, v) \leq p(|w|)$. Because of the representation

$$\max I_A(w) \text{ is odd} \leftrightarrow \bigvee_{i=1}^{p(|w|)} ((w, 2i-1) \in A \wedge (w, 2i) \notin A),$$

we have $A_{\text{odd}} \in P_{\text{bf}}^{\text{NP}}$. $\square$

Now we know in which classes the problems A_k, A'_k, A_+, A'_+ and A_{odd} can expected to be complete.

Beside the problems CLIQUE, TRAVELING SALESMAN and MAX SAT ASG already defined in Section 1, we consider seven further problems which are defined as follows.

$$\text{INDEPENDENT SET} = \{(G, m) : G \text{ graph}, m \in \mathbb{N} \text{ and } \max I_{\text{IS}}(G) \geq m\},$$

where $I_{\text{IS}}(G) = \{\text{card } I : I \text{ is an independent set in } G\}$.

$$\text{VERTEX COVER} = \{(G, m) : G \text{ graph}, m \in \mathbb{N} \text{ and } \min I_{\text{VC}}(G) \leq m\},$$

where $I_{\text{VC}}(G) = \{\text{card } V : V \text{ is a vertex cover on } G\}$.

$$\text{COLOR} = \{(G, m) : G \text{ graph}, m \in \mathbb{N} \text{ and } \min I_{\text{CO}}(G) \leq m\},$$

where $I_{\text{CO}}(G) = \{\text{card } R_f : f \text{ is an admissible coloring of } G\}$.

MAX 3-SAT $=\{(H, m)$: H Boolean formula in conjuctive normal form with 3 literals per clause, $m \in \mathbb{N}$ and $\max I_{\mathrm{SAT}}(H) \geqslant m\}$,

where $I_{\mathrm{SAT}}(H)=\{$number of clauses of H satisfied by $\varphi : \varphi$ assignment to the variables of $H\}$. The well-known problem 3-SAT is a sub-problem of MAX 3-SAT:

3-SAT $=\{(H, m)$: H Boolean formula in conjunctive normal form with three literals per clause, $m \in \mathbb{N}$ and $\max I_{\mathrm{SAT}}(H) \geqslant$ number of clauses of $H\}$.

SIMPLE DI CIRCUIT $=\{(G, m)$: G digraph, $m \in \mathbb{N}$ and $\max I_{\mathrm{SC}}(G) \geqslant m\}$,

where $I_{\mathrm{SC}}(G)=\{$length of $C : C$ is a simple circuit in $G\}$.

SIMPLE CIRCUIT $=\{(G, m)$: G graph, $m \in \mathbb{N}$ and $\max I_{\mathrm{SC}}(G) \geqslant m\}$.

The well-known problem HAMILTONIAN CIRCUIT is a subproblem of SIMPLE CIRCUIT:

HAMILTONIAN CIRCUIT $=\{G$: G graph and $\max I_{\mathrm{SC}}(G) \geqslant$ number of vertices of $G\}$.

MAX SOS $=\{(a_1, \ldots, a_n, b, c) : n, a_1, \ldots, a_n, b, c \in \mathbb{N}$ and $\max I_{\mathrm{SOS}}(a_1, \ldots, a_n, b) \geqslant c\}$,

where $I_{\mathrm{SOS}}(a_1, \ldots, a_n, b)=\{\sum_{i=1}^{n} \alpha_i \cdot a_i : \alpha_1, \ldots, \alpha_n \in\{0,1\}$ and $\sum_{i=1}^{n} \alpha_i \cdot a_i \leqslant b\}$. The well-known problem SOS is a sub-problem of MAX SOS:

SOS $=\{(a_1, \ldots, a_n, b) : n, a_1, \ldots, a_n, b \in \mathbb{N}$ and

$\max I_{\mathrm{SOS}}(a_1, \ldots, a_n, b) \geqslant b\}$.

Now, for the ten specific problems considered in this paper, let us observe which properties their valuation functions have.

Proposition 4.2. (1) *The valuation function is polynomially bounded for the problems* CLIQUE, INDEPENDENT SET, VERTEX COVER, COLOR, MAX 3-SAT, SIMPLE DI CIRCUIT *and* SIMPLE CIRCUIT.

(2) *The valuation function is polynomial-time invertible for the problem* MAX SAT ASG.

(3) *The valuation function is neither polynomially bounded nor polynomial-time invertible for the problems* TRAVELING SALESMAN *and* MAX SOS.

Rember that the valuation function cannot be both polynomially bounded and polynomial-time invertible.

Combining Theorem 4.1 and Proposition 4.2 we obtain the memberships shown in Table 1. In Section 6 and 7 we shall see that every membership shown in Table 1 is actually a completeness (see Table 2 at the end of Section 9).

Table 1.

	1st category	2nd category	3rd category
$A_k \in$	C_{2k}^{NP}	coNP	C_{2k}^{NP}
$A'_k \in$	C_{2k}^{NP}	C_{2k}^{NP}	C_{2k}^{NP}
$A_+ \in$	$P_{\mathrm{bf}}^{\mathrm{NP}}$	coNP	$P_{\mathrm{bf}}^{\mathrm{NP}}$
$A'_+ \in$	$P_{\mathrm{bf}}^{\mathrm{NP}}$	$P_{\mathrm{bf}}^{\mathrm{NP}}$	$P_{\mathrm{bf}}^{\mathrm{NP}}$
$A_{\mathrm{odd}} \in$	$P_{\mathrm{bf}}^{\mathrm{NP}}$	P^{NP}	P^{NP}
Examples of A	CLIQUE INDEPENDENT SET VERTEX COVER COLOR MAX 3-SAT SIMPLE DI CIRCUIT SIMPLE CIRCUIT	MAX SAT ASG	TRAVELING SALESMAN MAX SOS
Properties of β_A	polynomially bounded	polynomial-time invertible	neither of them

5. How to prove hardness for C_k^{NP}, D_k^{NP} and $P_{\mathrm{bf}}^{\mathrm{NP}}$?

A set A is said to be hard for the class C of sets w.r.t. the reducibility $\propto$ if $B \propto A$ for every $B \in C$. A set is said to be complete in C if, in addition, $A \in C$. If the reducibility is not specified explicitly, we mean hardness and completeness w.r.t. $\leq_m^p$.

Since the classes C_k^{NP}, D_k^{NP} and $P_{\mathrm{bf}}^{\mathrm{NP}}$ are not defined as complexity classes in the usual sense (i.e., by a restricted machine class), it is not immediately clear how one can prove hardness results for these classes. In this section we shall give sufficient conditions for a set to be hard for the classes C_k^{NP}, D_k^{NP} and $P_{\mathrm{bf}}^{\mathrm{NP}}$. Note that a hardness proof by this method corresponds to a master reduction, i.e., a reduction of all problems of the class in question to the problem in question (without knowing already any complete problem for this class).

Theorem 5.1. *Let D be an* NP*-complete set, let A be an arbitrary set, and let $k \geq 1$.*
(1) *If there exists a polynomial-time computable function f such that*

$$|\{i : x_i \in D\}| \text{ is odd } \leftrightarrow f(x_1, \ldots, x_{2k-1}) \in A$$

for all $x_1, \ldots, x_{2k-1} \in \Sigma^*$ with $c_D(x_1) \geq c_D(x_2) \geq \cdots \geq c_D(x_{2k-1})$, then A is D_{2k-1}^{NP}-hard.
(2) *If there exists a polynomial-time computable function f such that*

$$|\{i : x_i \in D\}| \text{ is even } \leftrightarrow f(x_1, \ldots, x_{2k-1}) \in A$$

for all $x_1, \ldots, x_{2k-1} \in \Sigma^*$ with $c_D(x_1) \geq c_D(x_2) \geq \cdots \geq c_D(x_{2k-1})$, then A is C_{2k-1}^{NP}-hard.
(3) *If there exists a polynomial-time computable function f such that*

$$|\{i : x_i \in D\}| \text{ is odd } \leftrightarrow f(x_i, \ldots, x_{2k}) \in A$$

for all $x_1, \ldots, x_{2k} \in \Sigma^*$ with $c_D(x_1) \geq c_D(x_2) \geq \cdots \geq c_D(x_{2k})$, then A is C_{2k}^{NP}-hard.

(4) *If there exists a polynomial-time computable function f such that*

$$|\{i : x_i \in D\}| \text{ is even} \leftrightarrow f(x_1, \ldots, x_{2k}) \in A$$

for all $x_1, \ldots, x_{2k} \in \Sigma^*$ *with* $c_D(x_1) \geq c_D(x_2) \geq \cdots \geq c_D(x_{2k})$, *then A is* D_{2k}^{NP}*-hard.*

Proof. Because of $\mathrm{co}C_k^{\mathrm{NP}} = D_k^{\mathrm{NP}}$ for all $k \geq 1$, it is sufficient to prove (1) and (3). We restrict ourselves to the proof of (3), the proof of (1) is analogous.

Let B be an arbitrary set from C_{2k}^{NP}. By Theorem 2.2(3), there exist sets $B_1, B_2, \ldots, B_{2k} \in \mathrm{NP}$ such that $B_1 \supseteq B_2 \supseteq \cdots \supseteq B_{2k}$ and $B = \bigcup_{i=1}^{k} (B_{2i-1} \cap \overline{B_{2i}})$. Hence, there exist polynomial-time computable functions $f_1, f_2, \ldots, f_{2k}$ such that

$$x \in B_i \leftrightarrow f_i(x) \in D$$

for $i = 1, 2, \ldots, 2k$ and $x \in \Sigma^*$. From $B_i \supseteq B_{i+1}$ we can conclude that $f_{i+1}(x) \in D$ implies $f_i(x) \in D$. Consequently,

$$x \in B \leftrightarrow |\{i : f_i(x) \in D\}| \text{ is odd} \leftrightarrow f(f_1(x), f_2(x), \ldots, f_{2k}(x)) \in A.$$

Thus, A is C_{2k}^{NP}-hard. □

Theorem 5.2. *Let D be an* NP*-complete set and let A be an arbitrary set. If there exists a polynomial-time computable function f such that*

$$|\{i : x_i \in D\}| \text{ is odd} \leftrightarrow f(x_1, \ldots, x_{2k}) \in A$$

for all $k \geq 1$, $x_1, \ldots, x_{2k} \in \Sigma^*$ *with* $c_D(x_1) \geq c_D(x_2) \geq \cdots \geq c_D(x_{2k})$, *then A is* $P_{\mathrm{bf}}^{\mathrm{NP}}$*-hard.*

Proof. Let B be an arbitrary set from $P_{\mathrm{bf}}^{\mathrm{NP}}$. Because of Theorem 3.2, we have $B \in P_{\mathrm{hd}}^{\mathrm{NP}}$. Since D is NP-hard w.r.t. $\leq_m^p$, it is $P_{\mathrm{hd}}^{\mathrm{NP}}$-hard w.r.t. $\leq_{\mathrm{hd}}^p$. Consequently, $B \leq_{\mathrm{hd}}^p D$, and there exist a polynomial-time computable function g such that

$$x \in B \leftrightarrow \bigvee_{i=1}^{k} c_D(x_{2i-1}) \wedge \neg c_D(x_{2i}) = 1$$

and $c_D(x_1) \geq c_D(x_2) \geq \cdots \geq c_D(x_{2k})$ where $g(x) = (x_1, x_2, \ldots, x_{2k})$. Note that k depends on x. Now we can conclude

$$x \in B \leftrightarrow |\{i : x_i \in D\}| \text{ is odd} \leftrightarrow f(g(x)) \in A.$$

Thus, A is $P_{\mathrm{bf}}^{\mathrm{NP}}$-hard. □

Following [7, 8] we define

$$D^{k-\Delta} = \{(x_1, \ldots, x_k) : |\{i : x_i \in D\}| \text{ is odd}\}, \quad \text{for } k \geq 1;$$

$$D^{\omega-\Delta} = \{(x_1, \ldots, x_k) : k \geq 1 \text{ and } |\{i : x_i \in D\}| \text{ is odd}\}.$$

Corollary 5.3. (1) (Köbler [7]; Köbler, Schöning, Wagner [8]). *Let* $k \geq 1$. *The set* $D_k^{k-\Delta}$ *is complete in* D_k^{NP} *for odd k, and it is complete in* C_k^{NP} *for even k.*

(2) (Köbler, Schöning, Wagner [8]). *The set* $D^{\omega-\Delta}$ *is complete in* $P_{\mathrm{bf}}^{\mathrm{NP}}$.

At first glance, Theorem 5.2 could be interpreted only as the statement that $D^{\omega-\Delta} \leq_m^p A$ implies the $P_{\text{bf}}^{\text{NP}}$-hardness of A. However, Theorem 5.2 is stronger. It says that, for the proof of the $P_{\text{bf}}^{\text{NP}}$-hardness of A, it is sufficient to make the reduction $D^{\omega-\Delta} \leq_m^p A$ only for the tuples $(x_1, x_2, \ldots, x_{2k})$ with the property $c_D(x_1) \geq c_D(x_2) \geq \cdots \geq c_D(x_{2k})$. This can make the hardness proofs considerably easier. In fact, we were not able to prove the $P_{\text{bf}}^{\text{NP}}$-hardness results in Sections 6–8 without having this strong result.

As usual, if we have a problem A which is hard for one of the classes C_k^{NP}, D_k^{NP}, $P_{\text{bf}}^{\text{NP}}$ or P^{NP} and if we can show for another problem B that $A \leq_m^p B$, then B is also hard for this class. For the special case that A and B are problems of the kind A_k, A'_k, A_+, A'_+ and A_{odd}, we shall give sufficient conditions which imply $A \leq_m^p B$ in the following theorem. The proof of this theorem is obvious.

Theorem 5.4. *Let A and B be maximum or minimum problems, and let $f: \Sigma^* \to \Sigma^*$ and $g, h: \Sigma^* \times \mathbb{N} \to \mathbb{N}$ be polynomial-time computable functions.*

(1) *If* $[\text{opt } I_A(w) = m \leftrightarrow \text{opt } I_B(f(w)) = h(w, m)]$ *for all* $w \in \Sigma^*$ *and* $m \in \mathbb{N}$, *then* $A_k \leq_m^p B_k$ *for all* $k \geq 1$ *and* $A_+ \leq_m^p B_+$.

(2) *If* $[\text{opt } I_A(w) \in [r, m] \leftrightarrow \text{opt } I_B(f(w)) \in [g(w, r, m), h(w, r, m)]]$ *for all* $w \in \Sigma^*$ *and* $r, m \in \mathbb{N}$, *then* $A'_k \leq_m^p B'_k$ *for all* $k \geq 1$ *and* $A'_+ \leq_m^p B'_+$.

(3) *If* $[\text{opt } I_A(w)$ *is odd* $\leftrightarrow \text{opt } I_B(f(w))$ *is odd*$]$, *then* $A_{\text{odd}} \leq_m^p B_{\text{odd}}$.

6. Problems with a polynomially bounded valuation function

In this and the following two sections we shall establish the completeness results for the problems A_k, A'_k, A_+, A'_+ and A_{odd} where A is one of the ten problems defined in the preceding sections. In this section we shall deal with problems having a polynomially bounded valuation function. A characteristic for the proofs in this section is that for every A we need only one reduction to prove the hardness results for A_k, A'_k, A_+, A'_+ and A_{odd}. This can be a master reduction via Theorems 5.1 and 5.2 or a reduction from another problem for which the corresponding hardness results are already known (by Theorem 5.4).

Naturally, we have to start with a master reductions. Statement (1) of the following theorem is a generalization of the result in [14] that CLIQUE_1 is complete in $D^P = C_2^{\text{NP}}$.

Theorem 6.1. (1) *For* $k \geq 1$, CLIQUE_k *and* CLIQUE'_k *are* C_{2k}^{NP}*-complete.*

(2) CLIQUE_+, CLIQUE'_+ *and* $\text{CLIQUE}_{\text{odd}}$ *are* $P_{\text{bf}}^{\text{NP}}$*-complete.*

Proof. The memberships in the corresponding classes are already stated in Table 1.

To prove the hardness results let D be any NP-complete problem. Since CLIQUE is also NP-complete, there exists a polynomial-time computable function f reducing D to CLIQUE. Without loss of generality (see [14]), we can assume that, for every $x \in \Sigma^*$ and $f(x) = (G, a)$,

- a is odd;

- $x \in D$ implies $\max I_{CL}(G) = a$; and
- $x \notin D$ implies $\max I_{CL}(G) = a - 1$.

For two graphs $G = (V, E)$ and $G' = (V', E')$ such that $V \cap V' = \emptyset$ we define $G + G' = (V \cup V', E \cup E' \cup (V \times V))$. Note that + is associative. It is obvious that

$$\max I_{CL}(G + G') = \max I_{CL}(G) + \max I_{CL}(G').$$

Now, let $x_1, x_2, \ldots, x_{2k} \in \Sigma^*$ such that $c_D(x_1) \geqslant c_D(x_2) \geqslant \cdots \geqslant c_D(x_{2k})$. For $i = 1, 2, \ldots, 2k$, let $f(x_i) = (G_i, a_i)$, and let $a = (\sum_{i=1}^{2k} a_i) - 2k$. Since all a_i's are odd, the number a is even. Without loss of generality, assume that the vertex sets of $G_1, G_2, \ldots, G_{2k}$ are pairwise disjoint. We conclude

$$\begin{aligned}
|\{i : x_i \in D\}| \text{ is odd } &\leftrightarrow \bigvee_{i=1}^{k} (x_1, \ldots, x_{2i-1} \in D \wedge x_{2i}, \ldots, x_{2k} \notin D) \\
&\leftrightarrow \bigvee_{i=1}^{k} \left(\sum_{j=1}^{2k} \max I_{CL}(G_j) = \left(\sum_{j=1}^{2k} a_j \right) - 2k + 2i - 1 \right) \\
&\leftrightarrow \max I_{CL}(G_1 + G_2 + \cdots + G_{2k}) \\
&\qquad \in \{a+1, a+3, a+5, \ldots, a+2k-1\} \\
&\leftrightarrow \max I_{CL}(G_1 + G_2 + \cdots + G_{2k}) \text{ is odd.}
\end{aligned}$$

By Theorem 5.1(3), we obtain that CLIQUE$_k$ and CLIQUE$'_k$ are C_{2k}^{NP}-hard, and by Theorem 5.2, we obtain that CLIQUE$_+$, CLIQUE$'_+$ and CLIQUE$_{odd}$ are P_{bf}^{NP}-hard. □

The above proof (as well as the following two proofs) exhibits the power of Theorems 5.1 and 5.2. All constructions of a graph G from $(G_1, \ldots, G_{2k}, a_1, \ldots, a_{2k})$ that we have tried without having the property that $\max I_{CL}(G_{i+1}) \geqslant a_{i+1}$ implies $\max I_{CL}(G_i) \geqslant a_i$ have led to a G whose size increases exponentially in k. Thus, the proofs presented here differ essentially from that made in [14] to prove that CLIQUE$_1$ is complete in $D^P = C_2^{NP}$.

In the same spirit we now prove similar results for MAX 3-SAT and COLOR.

Theorem 6.2. (1) *For* $k \geqslant 1$, MAX 3-SAT$_k$ *and* MAX 3-SAT$'_k$ *are* C_{2k}^{NP}*-complete.*

(2) MAX 3-SAT$_+$, MAX 3-SAT$'_+$ *and* MAX 3-SAT$_{odd}$ *are* P_{bf}^{NP}*-complete.*

Proof. The memberships in the corresponding classes are already stated in Table 1.

To prove the hardness results let D be any NP-complete set. The master reduction made, for example, in [3] to prove the NP-completeness of 3-SAT (via the unrestricted satisfiability problem for Boolean formulas) can easily be modified to yield a polynomial-time computable function f such that for every $x \in \Sigma^*$

- the number of clauses in $f(x)$ is odd;
- $x \in D$ implies $\max I_{SAT}(f(x)) =$ number of clauses of $f(x)$; and
- $x \notin D$ implies $\max I_{SAT}(f(x)) =$ (number of clauses of $f(x)) - 1$.

Let $x_1, x_2, \ldots, x_{2k} \in \Sigma^*$ be such that $c_D(x_1) \geqslant c_D(x_2) \geqslant \cdots \geqslant c_D(x_{2k})$. For $k = 1, 2, \ldots, 2k$, define $H_i \equiv f(x_i)$; let a_i be the number of clauses of H_i; and set

$a=(\sum_{i=1}^{2k} a_i)-2k$. Since all a_i's are odd, the number a is even. W.l.o.g., assume that the Boolean formulas $H_1, H_2, \ldots, H_{2k}$ have pairwise disjoint sets of variables. We conclude

$$
\begin{aligned}
|\{i : x_i \in d\}| \text{ is odd } &\leftrightarrow \bigvee_{i=1}^{k} (H_1, \ldots, H_{2i-1} \in 3\text{-SAT} \wedge H_{2i}, \ldots, H_{2k} \notin 3\text{-SAT}) \\
&\leftrightarrow \bigvee_{i=1}^{k} \left(\sum_{j=1}^{2k} \max I_{\mathrm{SAT}}(H_j) = \left(\sum_{i=1}^{2k} a_j \right) - 2k + 2i - 1 \right) \\
&\leftrightarrow \max\ I_{\mathrm{SAT}}(H_1 \wedge H_2 \wedge \cdots \wedge H_{2k}) \\
&\qquad \in \{a+1, a+3, a+5, \ldots, a+2k-1\} \\
&\leftrightarrow \max\ I_{\mathrm{SAT}}(H_1 \wedge H_2 \wedge \cdots \wedge H_{2k}) \text{ is odd.}
\end{aligned}
$$

By Theorem 5.1(3), we obtain that MAX 3-SAT$_k$ and MAX 3-SAT$'_k$ are C_{2k}^{NP}-hard, and, by Theorem 5.2, we obtain that MAX 3-SAT$_+$, MAX 3-SAT$'_+$ and MAX 3-SAT$_{\mathrm{odd}}$ are $P_{\mathrm{bf}}^{\mathrm{NP}}$-hard. □

Theorem 6.3. (1) *For* $k \geqslant 1$, COLOR$_k$ *and* COLOR$'_k$ *are* C_{2k}^{NP}*-complete.*

(2) COLOR$_+$, COLOR$'_+$ *and* COLOR$_{\mathrm{odd}}$ *are* $P_{\mathrm{bf}}^{\mathrm{NP}}$*-complete.*

Proof. The memberships in the corresponding classes are already stated in Table 1.

To prove the hardness results let 3-COLOR be the set of all graphs which are colorable by at most three colors. The problem 3-SAT is NP-complete (see [3]) and in [4] it is shown that 3-SAT $\leqslant_m^p$ 3-COLOR. The latter is done by giving a polynomial-time computable function f such that

- $H \in$ 3-SAT implies $\min I_{\mathrm{CO}}(f(H))=3$, and
- $H \notin$ 3-SAT implies $\min I_{\mathrm{CO}}(f(H))=4$.

Let $H_1, H_2, \ldots, H_{2k}$ be given such that $c_{3\text{-SAT}}(H_1) \geqslant \cdots \geqslant c_{3\text{-SAT}}(H_{2k})$. We set $G_i = f(H_i)$ for $i=1,\ldots,2k$, and we assume that the vertex sets of the graphs $G_1, G_2, \ldots, G_{2k}$ are pairwise disjoint. If we define the addition of graphs as in the proof of Theorem 6.1, we obtain

$$\min I_{\mathrm{CO}}(G+G') = \min I_{\mathrm{CO}}(G) + \min I_{\mathrm{CO}}(G').$$

We conclude

$$
\begin{aligned}
|\{i : H_i \in 3\text{-SAT}\}| \text{ is odd } &\leftrightarrow \bigvee_{i=1}^{k} (H_1, \ldots, H_{2i-1} \in 3\text{-SAT} \wedge H_{2i}, \cdots, H_{2k} \notin 3\text{-SAT}) \\
&\leftrightarrow \bigvee_{i=1}^{k} \left(\sum_{j=1}^{k} \min I_{\mathrm{CO}}(G_j) = 3(2i-1) + 4(2k-2i+1) \right) \\
&\qquad = 8k - 2i + 1 \\
&\leftrightarrow \min\ I_{\mathrm{CO}}(G_1 + G_2 + \cdots + G_{2k}) \\
&\qquad \in \{6k+1, 6k+3, \ldots, 8k-1\} \\
&\leftrightarrow \min\ I_{\mathrm{CO}}(G_1 + G_2 + \cdots + G_{2k}) \text{ is odd.}
\end{aligned}
$$

By Theorem 5.1(3) we obtain that COLOR_k and COLOR'_k are C_{2k}^{NP}-hard and, by Theorem 5.2, we obtain that COLOR_+, COLOR'_+ and $\text{COLOR}_{\text{odd}}$ are $P_{\text{bf}}^{\text{NP}}$-hard. □

The proof of Theorem 6.3(1) shows that even the problem

$$\text{COLOR}_{M_k} = \{G : \min I_{\text{CO}}(G) \in M_k\}$$

is C_{2k}^{NP}-complete, where $M_k = \{6k+1, 6k+3, \ldots, 8k-1\}$ is a fixed set of k elements. Thus, the question arises of how small the numbers in a k-element set M can be chosen such that COLOR_M is still C_{2k}^{NP}-complete. In particular, is COLOR_M still C_{2k}^{NP}-complete for $M = \{3, 5, 7, \ldots, 2k+1\}$?

So far the master reductions in this section. For the remaining four problems with polynomially bounded valuation functions, we make reductions of the form $A_k \leq_m^p B_k$, $A_+ \leq_m^p B_+$ and $A_{\text{odd}} \leq_m^p B_{\text{odd}}$. It is an interesting fact that, for these reductions between problems which are not in NP (unless NP = coNP), we can use slight modifications of the reductions made in the literature between the problems A and B (which are in NP). Our proofs are based on the reductions made in [3, 11]. We shall not repeat the entire proofs from there, but we shall only say how the constructions there are modified to obtain our results. Thus, the reader should have access to both books when reading the following proofs.

The obvious polynomial-time reductions from CLIQUE to the problems INDEPENDENT SET and VERTEX COVER (see [3]) immediately yield the following corollary.

Corollary 6.4. (1) *For* $k \geq 1$, INDEPENDENT SET_k, $\text{INDEPENDENT SET}'_k$, VERTEX COVER_k and $\text{VERTEX COVER}'_k$ are C_{2k}^{NP}*-complete.*

(2) INDEPENDENT SET_+, $\text{INDEPENDENT SET}'_+$, $\text{INDEPENDENT SET}_{\text{odd}}$, VERTEX COVER_+, $\text{VERTEX COVER}'_+$ *and* $\text{VERTEX COVER}_{\text{odd}}$ *are* $P_{\text{bf}}^{\text{NP}}$*-complete.*

Theorem 6.5. (1) *For* $k \geq 1$, $\text{SIMPLE DI CIRCUIT}_k$ *and* $\text{SIMPLE DI CIRCUIT}'_k$ *are* C_{2k}^{NP}*-complete.*

(2) $\text{SIMPLE DI CIRCUIT}_+$, $\text{SIMPLE DI CIRCUIT}'_+$ *and* $\text{SIMPLE DI CIRCUIT}_{\text{odd}}$ *are* $P_{\text{bf}}^{\text{NP}}$*-complete.*

Proof. The memberships in the corresponding classes are already stated in Table 1.

To prove the hardness results we modify the proof in [11] for 3-SAT $\leq_m^p$ HAMILTONIAN PATH by identifying the vertices v_{n+1} and v_1 in the digraph G constructed for the Boolean formula H (which has n variables and has conjunctive normal form with m clauses and three literals per clause). We denote the modified digraph by G'. It is easy to see that we have not only

$$H \in \text{3-SAT} \leftrightarrow G' \text{ has a Hamiltonian circuit,}$$

i.e.,

$$\max I_{\text{SAT}}(H) = m \leftrightarrow \max I_{\text{SC}}(G') = n + 18m,$$

but also

$$\max I_{\text{SAT}}(H) = r \leftrightarrow \max I_{\text{SC}}(G') = n + 18r \quad \text{for all } r \leq m.$$

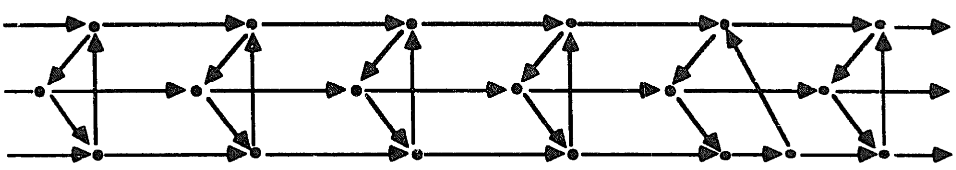

Fig. 2.

To achieve that the odd-number property for the maxima is also preserved, we modify the digraph G' by replacing all 'three-lane carriageways' occurring in G' by the subgraph presented in Fig. 2. The resulting digraph is denoted by G''. We obtain

$$\max I_{\mathrm{SAT}}(H)=r \leftrightarrow \max I_{\mathrm{SC}}(G')=n+19r.$$

Since we can, w.l.o.g., assume that the number n of variables in H is even, we can conclude

$$\max I_{\mathrm{SAT}}(H) \text{ is odd} \leftrightarrow \max I_{\mathrm{SC}}(G'') \text{ is odd}.$$

By Theorems 5.4(1) and 6.2, we obtain that SIMPLE DI CIRCUIT$_k$ and SIMPLE DI CIRCUIT$'_k$ are C_{2k}^{NP}-hard and that SIMPLE DI CIRCUIT$_+$ and SIMPLE DI CIRCUIT$'_+$ are $P_{\mathrm{bf}}^{\mathrm{NP}}$-hard. By Theorems 5.4(3) and 6.2, we obtain that SIMPLE DI CIRCUIT$_{\mathrm{odd}}$ is $P_{\mathrm{bf}}^{\mathrm{NP}}$-hard. □

Theorem 6.6. (1) *For* $k \geqslant 1$, SIMPLE CIRCUIT$_k$ *and* SIMPLE CIRCUIT$'_k$ *are* C_{2k}^{NP}*-complete.*

(2) SIMPLE CIRCUIT$_+$, SIMPLE CIRCUIT$'_+$ *and* SIMPLE CIRCUIT$_{\mathrm{odd}}$ *are* $P_{\mathrm{bf}}^{\mathrm{NP}}$*-complete.*

Proof. The memberships in the corresponding classes are already stated in Table 1.

To prove the hardness results, we modify the proof in [3] for VERTEX COVER $\leqslant_m^p$ HAMILTONIAN CIRCUIT. The graph constructed there for a given graph $G=(V,E)$ with $V=\{v_1, v_2, \ldots, v_n\}$ and $|E|=m$ is modified as follows

(1) Instead of the selector vertices we take the vertices $a_1, \ldots, a_n, b_1, \ldots, b_n$ which are joined with each other and with the cover-testing components as shown in Fig. 3 ($i=1,\ldots,n$ and $a_{n+1}=a_1$).

(2) In the cover-testing component for $e=\{u,v\}$, each of the vertices $(u,e,2)$, $(v,e,2)$, $(u,e,5)$ and $(v,e,5)$ is replaced by $n+1$ new vertices as shown in Fig. 4. Thus, this component has $4n+12$ vertices.

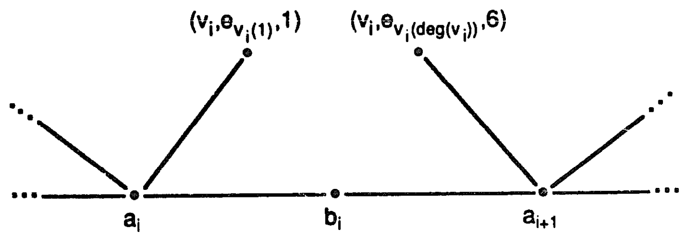

Fig. 3.

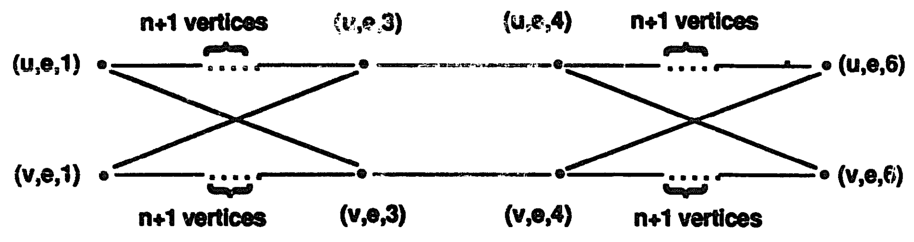

Fig. 4.

The resulting graph is denoted by G'. We observe: a simple circuit not meeting the vertices of a cover-testing component in one of the three configurations shown in [3, Fig. 3.5] does not meet at least one block of $n+1$ new vertices. Consequently, the length of such a circuit is at most $(4n+12)m+2n-(n+1)$. On the other hand, there exist a simple circuit of the shorter length $(4n+12)m+n$, namely that circuit which meets all a_i, no b_i, and which meets the cover-testing components as shown in [3, Fig. 3.5(b)]. Thus, a simple circuit of maximum length meets the vertices of the cover-testing components as shown in [3, Fig. 3.5]. We conclude

$$\min I_{\mathrm{VC}}(G)=r \leftrightarrow \max I_{\mathrm{SC}}(G')=(4n+12)m+2n-r.$$

By Theorem 5.4(1) and Corollary 6.4, we obtain that SIMPLE CIRCUIT$_k$ and SIMPLE CIRCUIT$'_k$ are C_{2k}^{NP}-hard and that SIMPLE CIRCUIT$_+$ and SIMPLE CIRCUIT$'_+$ are $P_{\mathrm{bf}}^{\mathrm{NP}}$-hard. By Theorem 5.4(3) and Corollary 6.4, we obtain that SIMPLE CIRCUIT$_{\mathrm{odd}}$ is $P_{\mathrm{bf}}^{\mathrm{NP}}$-hard. □

7. Problems with polynomial-time invertible valuation functions

Since, for problems with polynomial-time invertible valuation functions the complexities of the derived problems A_k, A_+ and A_{odd} differ from those for the problems in Section 6, we have to make master reductions for these cases.

Theorem 7.1. (1) *For* $k\geq 1$, MAX SAT ASG$_k$ *and* MAX SAT ASG$_+$ *are* coNP-*complete.*

(2) *For* $k\geq 1$, MAX SAT ASG$'_k$ *is* C_{2k}^{NP}-*complete.*

(3) MAX SAT ASG$'_+$ *is* $P_{\mathrm{bf}}^{\mathrm{NP}}$-*complete.*

(4) MAX SAT ASG$_{\mathrm{odd}}$ *is* P^{NP}-*complete.*

Proof. The memberships in the corresponding classes are already stated in Table 1.

(1): Since we have $A_1 \leq_m^p A_2 \leq_m^p A_3 \leq_m^p \cdots \leq_m^p A_+$ for every optimum problem A, it is sufficient to prove the coNP-hardness of MAX SAT ASG$_1$. We reduce the complement of the NP-complete problem 3-SAT to MAX SAT ASG$_1$ as follows. For a Boolean formula $H(x_0,\ldots,x_n)$ in conjunctive normal form with three literals per clause, let $H'(x_0,\ldots,x_n,x_{n+1})$ be such a formula that is equivalent to

$H(x_0, \ldots, x_n) \vee (\neg x_0 \wedge \cdots \wedge \neg x_n \wedge \neg x_{n+1})$. Consequently,

$$H \notin 3\text{-SAT} \leftrightarrow \max I_{\text{MSA}}(H') = 0.$$

(2) and (3): We apply Theorems 5.1(3) and 5.2 with $D = 3\text{-SAT}$. Let $H_1, \ldots, H_{2k}$ be Boolean formulas in conjunctive normal form with three literals per clause such that $H_{i+1} \in 3\text{-SAT}$ implies $H_i \in 3\text{-SAT}$ for $i = 1, 2, \ldots, 2k-1$ W.l.o.g., let $H \equiv H_i(x_{r(i-1)}, \ldots, x_{ri-1})$ for $i = 1, 2, \ldots, 2k$, i.e., all H_i's have exactly r variables and no variable occurs in H_i and in H_j for $i \neq j$. Let $H(x_0, x_1, \ldots, x_{2rk+2k-1})$ be a Boolean formula in conjunctive normal form with three literals per clause which is equivalent to

$$\bigwedge_{i=1}^{2k} (x_{2rk-1+i} \leftrightarrow H_i(x_{r(i-1)}, \ldots, x_{ri-1})).$$

A maximum satisfying assignment to the variables of H is such a satisfying assignment which puts $x_{2rk-1+i} = 1$ for as many as possible $i \in \{1, \ldots, 2k\}$. But $x_{2rk-1+i} = 1$ is possible if and only if $H_i \in 3\text{-SAT}$. We conclude

$$\begin{aligned}
&|\{i : H_i \in 3\text{-SAT}\}| \text{ is odd} \\
&\leftrightarrow \bigvee_{i=1}^{k} (H_1, \ldots, H_{2i-1} \in 3\text{-SAT} \wedge H_{2i}, \ldots, H_{2k} \notin 3\text{-SAT}) \\
&\leftrightarrow \bigvee_{i=1}^{k} (\text{for the maximum satisfying assignment to } H \text{ it holds} \\
&\qquad \text{that } x_{2rk} = \cdots = x_{2rk+2i-2} = 1 \text{ and } x_{2rk+2i-1} = \cdots = x_{2rk+2k-1} = 0) \\
&\leftrightarrow \bigvee_{i=1}^{k} \max I_{\text{MSA}}(H) \in \left[\sum_{j=2rk}^{2rk+2i-2} 2^j, \sum_{j=0}^{2rk+2i-2} 2^j\right] \\
&\leftrightarrow \max I_{\text{MSA}}(H) \in \bigcup_{i=1}^{k} [2^{2rk+2i-1} - 2^{2rk}, 2^{2rk+2i-1} - 1].
\end{aligned}$$

(4) (*sketch*): Let $A \in P^{\text{NP}}$ and let M be a deterministic polynomial-time Turing machine accepting A with an oracle $B \in \text{NP}$, where the time bound is given by the polynomial p. Further, let M' be a nondeterministic polynomial-time Turing machine accepting B. For an input $x \in \Sigma^*$, let $H_x(\bar{u}, \bar{v}_1, \ldots, \bar{v}_{p(|x|)}, z_1, \ldots, z_{p(|x|)}, z)$ be a Boolean formula in conjunctive normal form with three literals per clause whose only satisfying assignment describes (as, for example, in the proof of the NP-completeness of 3-SAT given in [17]) the computation of M on x such that

- the set $\bar{u}$ of Boolean variables describes the tape contents and states of M;
- the set $\bar{v}_i$ of Boolean variables describes the query v_i made in the ith step of M to the oracle;
- the Boolean variable z_i describes the answer of the oracle to the query v_i made in the ith step of M, where $z_i = 1$ ($z_i = 0$) means "yes" ("no"); $i = 1, 2, \ldots, p(|x|)$;
- the Boolean variable z describes the result of the computation of M on x, where $z = 1$ ($z = 0$) means "accept" ("reject").

Further, let $G_x(\bar{v}, \bar{y}, z')$ be a Boolean formula in conjunctive normal form with three literals per clause whose satisfying assignments describe all (not only the accepting) computations of M' on inputs of maximum length $p(|x|)$ such that
- the set $\bar{v}$ of Boolean variables describes the tape contents and states of M;
- the set $\bar{y}$ of Boolean variables describes the tape contents and states of M';
- the Boolean variable z' describes the result of a computation of M' on v where $z=1$ $(z=0)$ means "accept" ("reject").

Since M' is a nondeterministic machine, there can be several computations of M' on a given input v. Thus, the result variable z' can have different values for different satisfying assignments to G. We define

$$F_x(z, u, y_1, \ldots, y_{p(|x|)}, v_1, \ldots, v_{p(|x|)}, z_1, \ldots, z_{p(|x|)})$$

$$\equiv H_x(u, v_1, \ldots, v_{p(|x|)}, z_1, \ldots, z_{p(|x|)}) \wedge \bigwedge_{i=1}^{p(|x|)} G_x(v_i, y_i, z_i).$$

A satisfying assignment to F_x describes a computation of M on x where, as the answer of the oracle to the query v_i, the result of any computation of M' on v_i is taken. Thus, it can happen that the result 0 of a rejecting computation is taken as an answer, though there also exists an accepting computation of M' on v_i (having result 1). Consequently, this satisfying assignment to F_x does not describe the correct computation of M on input x. However, the *maximum* satisfying assignment to F_x must make $z_i = 1$, if possible, for $i = 1, 2, \ldots, p(|x|)$. Hence, the maximum satisfying assignment describes an accepting computation of M' on v_i if such a computation exists. Consequently, the maximum satisfying assignment to F_x describes the correct computation of M on x. We conclude

$$x \in A \leftrightarrow \text{the maximum satisfying assignment to } F \text{ makes } z=1$$

$$\leftrightarrow \max I_{\mathrm{MSA}}(F_x) \text{ is odd.} \qquad \square$$

8. Problems whose valuation functions are not restricted

The problems MAX SOS and TRAVELING SALESMAN fall into this category. To obtain all completeness results, we make reductions from VERTEX COVER (which has a polynomially bounded valuation function) and MAX SAT ASG (which has a polynomial-time invertible valuation function).

Theorem 8.1. (1) *For $k \geq 1$,* MAX SOS$_k$ *and* MAX SOS$'_k$ *are C^{NP}_{2k}-complete.*
(2) MAX SOS$_+$ *and* MAX SOS$'_+$ *are $P^{\mathrm{NP}}_{\mathrm{bf}}$-complete.*
(3) MAX SOS$_{\mathrm{odd}}$ *is P^{NP}-complete.*

Proof. The membership in the corresponding classes are already stated in Table 1.
(1) and (2): We apply Theorem 5.4(1) with $A =$ VERTEX COVER and $B =$ MAX SOS. Let $G = (V, E)$ be a graph with $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$. We define

the natural numbers $a_1, \ldots, a_n, b_1, \ldots, b_n, c_1, \ldots, c_m$ and L by their 4-adic presentations which are of length $m+2n$ $(i=1, \ldots, n;\ j=1, \ldots, m)$:

$$\begin{array}{ll} a_i = a_{i1} \ldots a_{im} 00 \ldots 010 \ldots 00 & \\ b_i = 0 \ldots . .0\ \ 00 \ldots \underset{\substack{\uparrow \\ (m+i)\text{th digit}}}{0 1 0} \ldots 01 & \end{array} \quad \text{where} \quad a_{ij} = \begin{cases} 1 & \text{if } v_i \text{ in } e_j. \\ 0 & \text{otherwise;} \end{cases}$$

$$c_j = 00 \ldots \underset{\substack{\uparrow \\ j\text{th digit}}}{010} \ldots 00 \quad \textit{and} \quad L = 22 \ldots \underset{\substack{\uparrow \\ m\text{th digit}}}{21} 1 \ldots \underset{\substack{\uparrow \\ (m+n)\text{th digit}}}{1} 00 \ldots 0.$$

Let $\min I_{\mathrm{VC}}(G) = r$, and let $C \subseteq V$ be a vertex cover with $|C| = r$. From the numbers a_i, b_i and c_j we select a_i if $v_i \in C$ and b_i if $v_i \notin C$. Further, we take b_j if e_j has only one endpoint in C. The sum of all these numbers is $L+(n-r)$. Consequently,

$$\max I_{\mathrm{SOS}}(a_1, \ldots, a_n, b_1, \ldots, b_n, c_1, \ldots, c_m, L+n) \geq L+(n-r).$$

Let

$$\max I_{\mathrm{SOS}}(a_1, \ldots, a_n, b_1, \ldots, b_n, c_1, \ldots, c_m, L+n) = L+(n-r),$$

and let $S \subseteq \{a_1, \ldots, a_n, b_1, \ldots, b_n, c_1, \ldots, c_m\}$ such that $\Sigma_{s \in S}\, s = L+(n-r)$. Evidently, $\{v_i : a_i \in S\}$ is a vertex cover of G and $|\{v_i : a_i \in S\}| = n - |\{i : b_i \in S\}| = n-(n-r) = r$. Hence, $\min I_{\mathrm{VC}}(G) \leq r$. From this we conclude

$$\begin{aligned} \min I_{\mathrm{VC}}(G) = r \ \leftrightarrow\ & \max I_{\mathrm{SOS}}(a_1, \ldots, a_n, b_1, \ldots, b_n, c_1, \ldots, c_m, L+n) \\ & = L+n-r. \end{aligned}$$

(3): We apply Theorem 5.4(3) with $A =$ MAX SAT ASG and $B =$ MAX SOS. The proof is similar to that given in [11] for 3-SAT $\leq_m^p$ SOS. Let $H(x_0, \ldots, x_n)$ be a Boolean formula in conjunctive normal form with m clauses and three literals per clause. We define the natural numbers $a_1, \ldots, a_n, b_1, \ldots, b_n, c_1, \ldots, c_m, d_1, \ldots, d_m$ and L with the help of their 6-adic presentations which are of length $m+2n+2$ $(i=0, 1, \ldots, n;\ j=1, \ldots, m)$:

$$a_i = a_{i1} \ldots a_{im} 00 \ldots \underset{\substack{\uparrow \\ (m+i+1)\text{th digit}}}{010} \ldots 00 + 2^i \quad \text{where } a_{ij} = \begin{cases} 1 & \text{if } x_i \text{ is in clause } j, \\ 0 & \text{otherwise;} \end{cases}$$

$$b_i = b_{i1} \ldots b_{im} 00 \ldots \underset{\substack{\uparrow \\ (m+i+1)\text{th digit}}}{010} \ldots 00 \qquad \text{where } b_{ij} = \begin{cases} 1 & \text{if } \bar{x}_i \text{ is in clause } j, \\ 0 & \text{otherwise;} \end{cases}$$

$$c_j = d_j = 00 \ldots \underset{\substack{\uparrow \\ j\text{th digit}}}{010} \ldots 00 \quad \text{and} \quad L = 33 \ldots \underset{\substack{\uparrow \\ m\text{th digit}}}{31} 1 \ldots \underset{\substack{\uparrow \\ (m+n+1)\text{th digit}}}{1} 00 \ldots 0.$$

Now we can conclude

$$\begin{aligned} & \max I_{\mathrm{MSA}}(H) = r \\ & \quad \leftrightarrow \max I_{\mathrm{SOS}}(a_1, \ldots, a_n, b_1, \ldots, b_n, c_1, \ldots, c_m, d_1, \ldots, d_m, L+2^{n+1}-1) \\ & \qquad = L+r. \end{aligned}$$

Since L is even, we have

$$\max I_{\mathrm{MSA}}(H) \text{ is odd}$$

$$\leftrightarrow \max I_{\mathrm{SOS}}(a_1, \ldots, a_n, b_1, \ldots, b_n, c_1, \ldots, c_m, d_1, \ldots, d_m, L+2^{n+1}-1) \text{ is odd.} \quad \square$$

Theorem 8.2. (1) *For $k \geq 1$,* TRAVELING SALESMAN$_k$ *and* TRAVELING SALESMAN$'_k$ *are C_{2k}^{NP}-complete.*

(2) TRAVELING SALESMAN$_+$ *and* TRAVELING SALESMAN$'_+$ *are $P_{\mathrm{bf}}^{\mathrm{NP}}$-complete.*

(3) TRAVELING SALESMAN$_{\mathrm{odd}}$ *is P^{NP}-complete.*

Proof. The memberships in the corresponding classes are already stated in Table 1.

(1) and (2) (*sketch*): We apply Theorem 5.4(1) with $A =$ VERTEX COVER and $B =$ TRAVELING SALESMAN. The proof given in [3] for

VERTEX COVER $\leq_m^p$ HAMILTONIAN CIRCUIT

is modified as follows. Instead of the selector vertices in the graph G' constructed from the graph G we take the vertices $a_1, \ldots, a_n$ which are joined with each other and with the cover-testing components as shown in Fig. 5 ($i = 1, \ldots, n$; $a_{n+1} = a_1$).

Now we construct a traveling-salesman instance T as follows:

- the cities of T are vertices of G';
- the edges $(a_i, (v_i, e_{v_i[1]}, 1))$ have length 1; $i = 1, \ldots, n$;
- all other edges of G' have length 0;
- all connections between cities not connected by an edge of G' have length $n+1$.

We observe:

- G' has a Hamiltonian circuit;
- tours of T corresponding to a Hamiltonian circuit of G' have length at most n;
- tours of T not corresponding to a Hamiltonian circuit of G' have length at least $n+1$;
- $\min I_{\mathrm{VC}}(G) = m \leftrightarrow \min I_{\mathrm{TS}}(T) = m$.

(3) (*sketch*): We apply Theorem 5.4(3) with $A =$ MAX SAT ASG and $B =$ TRAVELING SALESMAN. The proof given in [11] for 3-SAT $\leq_m^p$ HAMILTONIAN PATH (for digraphs) is modified as follows. For a given Boolean formula $H(y_1, \ldots, y_n)$ in conjunctive

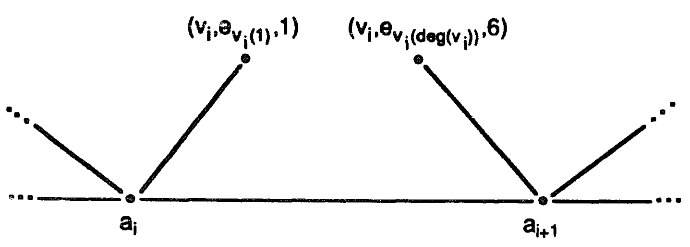

Fig. 5.

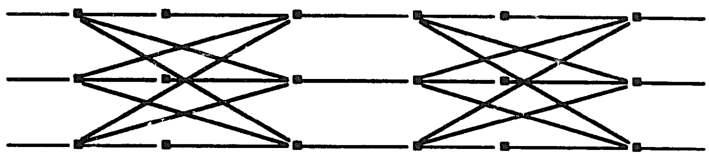

Fig. 6.

normal form with three literals per clause, we construct an undirected graph G rather than a digraph where

- v_{n+1} is identified with v_1;
- the "three-lane carriageways" made for digraphs are replaced by the subgraph shown in Fig. 6 which works in the same way for undirected graphs.

Consequently,

$$H \in 3\text{-SAT} \leftrightarrow G \text{ has a Hamiltonian circuit.}$$

From G we construct a traveling-salesman instance T as follows:

- the cities of T are the vertices of G;
- the edge from v_i to the first vertex of the 'y_i-path' has length $2^n - 2^{i-1}$, $i = 1, \ldots, n$;
- the edge from v_i to the first vertex of the '$\bar{y}_i$-path' has length 2^n, $i = 1, \ldots, n$;
- all other edges of G have length 0;
- all connections between cities not connected by an edge of G have length $n2^{n+1}$.

We observe:

- tours of T corresponding to a Hamiltonian circuit of G have length at most $n2^n$;
- tours of T not corresponding to a Hamiltonian circuit of G have length at least $n2^{n+1}$;
- $\max I_{\text{MSA}}(H) = m \leftrightarrow \min I_{\text{TS}}(T) = n2^n - m$.

W.l.o.g., we assume $n \geq 1$ and obtain

$$\max I_{\text{MSA}}(H) \text{ is odd} \leftrightarrow \min I_{\text{TS}}(T) \text{ is odd.} \qquad \square$$

Note that Theorem 8.2(3) is close to a result in [12] saying that the problem of whether a given traveling-salesman instance has only one optimal solution is complete in P^{NP}. In [6] it has been noted that a minor modification of the proof of this result would yield the following result: the problem of whether the optimal tour length of a given traveling-salesman instance is divisible by a given $k \in \mathbb{N}$ is complete in P^{NP}. Our result replaces k by the fixed number 2.

9. Conclusions

Our results on the complexity of the problems A_k, A_k, A_+, A_+ and A_{odd} for the ten specific problems A are summarized in Table 2.

Thus we have found natural problems which are complete in the classes C_{2k}^{NP} $(k \geq 1)$ of the Boolean NP-hierarchy. To find problems which are complete in the

Table 2.

	1st category	2nd category	3rd category
A_k $(k\geqslant 1)$	complete in C_{2k}^{NP}	complete in coNP	complete in C_{2k}^{NP}
A'_k $(k\geqslant 1)$	complete in C_{2k}^{NP}	complete in C_{2k}^{NP}	complete in C_{2k}^{NP}
A_+	complete in P_{bf}^{NP}	complete in coNP	complete in P_{bf}^{NP}
A'_+	complete in P_{bf}^{NP}	complete in P_{bf}^{NP}	complete in P_{bf}^{NP}
A_{odd}	complete in P_{bf}^{NP}	complete in P^{NP}	complete in P^{NP}
Examples of A	CLIQUE INDEPENDENT SET VERTEX COVER COLOR MAX 3-SAT SIMPLE DI CIRCUIT SIMPLE CIRCUIT	MAX SAT ASG	TRAVELING SALESMAN MAX SOS
Properties of β_A	polynomially bounded	polynomial-time invertible	neither of them

other classes of the Boolean NP-hierarchy, we can modify the problems A'_k as follows:

$$A_k^2=\Big\{(w,a_1,\ldots,a_k,b_1,\ldots,b_k): w\in\Sigma^*, a_1,\ldots,a_k,b_1,\ldots,b_k\in\mathbb{N} \text{ and } \operatorname{opt} I_A(w)\in\bigcup_{i=4}^{k-1}[a_i,b_i]\cup[0,b_k]\cup[a_k,\infty)\Big\},$$

$$A_k^3=\Big\{(w,a_1,\ldots,a_k,b_1,\ldots,b_{k-1}): w\in\Sigma^*, a_1,\ldots,a_k,b_1,\ldots,b_{k-1}\in\mathbb{N} \text{ and } \operatorname{opt} I_A(w)\in\bigcup_{i=1}^{k-1}[a_i,b_i]\cup[a_k,\infty)\Big\},$$

$$A_k^4=\Big\{(w,a_1,\ldots,a_{k-1},b_1,\ldots,b_k): w\in\Sigma^*, a_1,\ldots,a_{k-1},b_1,\ldots,b_k\in\mathbb{N} \text{ and } \operatorname{opt} I_A(w)\in\bigcup_{i=1}^{k-1}[a_i,b_i]\cup[0,b_k]\Big\}.$$

In the same way as in Sections 6, 7 and 8, one can prove for all our ten problems A that A_k^2 is D_{2k}^{NP}-complete, that A_k^3 is D_{2k-1}^{NP}-complete (C_{2k-1}^{NP}-complete) for maximum (minimum) problems A and that A_k^4 is C_{2k-1}^{NP}-complete (D_{2k-1}^{NP}-complete) for maximum (minimum) problems A.

The results in Table 2 show that the class P_{bf}^{NP} can be considered as a natural ω-jump of NP w.r.t. the Boolean NP-hierarchy. This is an analogue to the result in [16] that PSPACE is a natural ω-jump of NP w.r.t. the polynomial-time hierarchy. However, in [7, 8] it is shown that, in the case of the Boolean NP-hierarchy, the ω-jump is not uniquely determined. Using other complete sets for the classes C_k^{NP} and D_k^{NP}, $k\geqslant 1$, one obtains P_{tt}^{NP} as the ω-jump of NP w.r.t. the Boolean NP-hierarchy. Generally speaking, this ω-jump heavily depends on the succinctness of the presentation of the Boolean conditions in the complete sets for C_k^{NP} and D_k^{NP}, $k\geqslant 1$.

From a wide variety of 'more complicated' questions about maxima and minima we have chosen only five. It would be interesting to study other such questions which possibly give rise to other types of results than the ones obtained in this paper. We have additionally investigated the question "opt $I_A(w_1) = \text{opt } I_A(w_2)$?" where we obtained the same results as for "opt $I_A(w)$ is odd?".

Questions about opt $I_A(w)$ are special cases of questions about $I_A(w)$. The complexity of problems based on questions like "$x \in I_A(w)$?", "$I_A(w_1) \neq I_A(w_2)$?", "$I_A(w_1) \subseteq I_A(w_2)$?", "$I_A(w_1) \neq I_A(w_2)$?" and "card $I_A(w) \geq m$?" have been investigated for problems A with not polynomially-bounded valuation functions in [15, 19] where it has been shown that, in many cases, such questions lead to problems which are complete in suitable classes of the polynomial-time hierarchy or of the counting polynomial-time hierarchy (for the latter see [19]). It is obvious that, for problems A with polynomially bounded valuation function, the above-mentioned questions lead to problems in P^{NP}.

References

[1] J. Cai and L. Hemachandra, The Boolean hierarchy: hardware over NP, *Proc. Struct. in Compl. Theory Conf.*, Lecture Notes in Computer Science **223** (Springer, Berlin, 1986) 105-124.

[2] J. Cai and G.E. Meyer, Graph minimal uncolorability is D^P-complete, *SIAM J. Comput.* (1986), to appear.

[3] M.R. Garey and D.S. Johnson, *Computers and Intractibility: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979).

[4] M.R. Garey, D.S. Johnson and L. Stockmeyer, Some simplified NP-complete graph problems, *Theoret. Comput. Sci.* **1** (1976) 237-267.

[5] F. Hausdorff, *Grundzüge der Mengenlehre* (Leipzig, 1914).

[6] D.S. Johnson, The NP-completeness column: an ongoing guide 15th edition, *J. Algorithms* **6** (1985) 291-305.

[7] J. Köbler, Die Differenz- und die truth-table-Hierarchie für NP, Diploma thesis, University of Stuttgart, 1985.

[8] J. Köbler, U. Schöning and K. W. Wagner, The difference and truth-table hierarchies for NP, Bericht 7/86, Seminar für Informatik, EWH Koblenz, 1986; also: *RAIRO Inform. Théoriq.* to appear.

[9] R.E. Ladner and N.A. Lynch, Relativizations of questions about log-space computability, *Math. Systems Theory* **10** (1976) 19-32.

[10] R.E. Ladner, N.A. Lynch and A.L. Selman, A comparison of polynomial time reducibilities, *Theoret. Comput. Sci.* **1** (1975) 103-123.

[11] M. Machtey and P. Young, *An Introduction to General Theory of Algorithms* (North-Holland, New York, 1978).

[12] C.H. Papadimitriou, On the complexity of unique solutions, *J. ACM* **31** (1984) 392-400.

[13] C.H. Papadimitriou and D. Wolfe, The complexity of facets resolved, *Proc. 26th Ann. Symp. on Found. of Comp. Sci.* (1985) 74-78.

[14] C.H. Papadimitriou and M. Yannakakis, The complexity of facets (and some facets of complexity), *Proc. 14th STOC* (1982) 255-260; see also: *J. Comput. System Sci.* **28** (1984) 244-259.

[15] L.J. Stockmeyer and A.R. Meyer, Word problems requiring exponential time, *Proc. 5th STOC* (1973) 1-9.

[16] L.J. Stockmeyer, The polynomial-time hierarchy, *Theoret. Comput. Sci.* **3** (1977) 1-22.

[17] K. Wagner and G. Wechsung, *Computational Complexity* (Verlag der Wissenschaften, Berlin, 1986; and Reidel, Dordrecht, 1986).

[18] K. Wagner, On ω-regular sets, *Inform. and Control* **43** (1979) 123-177.

[19] K. Wagner, Compact descriptions and the counting polynomial-time hierarchy, *Proc. 2nd Frege Conf.* (1984) 383-392; see also: The complexity of combinatorial problems with succinct input representation, *Acta Inform.* **23** (1986) 325-356.

[20] K. Wagner, The complexity of graphs with regularities, *Proc. 11th MFCS Conf.*, Lecture Notes in Computer Science **176** (Springer, Berlin, 1984) 544-552.

[21] G. Wechsung and K.W. Wagner, On the Boolean closure of NP, Manuscript (extended abstract as: G. Wechsung, On the Boolean closure of NP, *Proc. FCT Conf. 1985*, Lecture Notes in Computer Science **199** (Springer, Berlin, 1985) 485-493.