Discrete Applied Mathematics 96–97 (1999) 363–373

DISCRETE APPLIED MATHEMATICS

On generating the irredundant conjunctive and disjunctive normal forms of monotone Boolean functions

V. Gurvich[a, 1], L. Khachiyan[b, *]

[a] RUTCOR, Rutgers University, New Brunswick, NJ 08903, USA
[b] Department of Computer Science, Rutgers University, Hill Center, Busch Campus, New Brunswick, NJ 08903, USA

Received 28 August 1995; revised 30 July 1996; accepted 9 September 1996

Abstract

Let $f:\{0,1\}^n \to \{0,1\}$ be a monotone Boolean function whose value at any point $x\in\{0,1\}^n$ can be determined in time t. Denote by $c=\bigwedge_{I\in C}\bigvee_{i\in I} x_i$ the irredundant CNF of f, where C is the set of the prime implicates of f. Similarly, let $d=\bigvee_{J\in D}\bigwedge_{j\in J} x_j$ be the irredundant DNF of the same function, where D is the set of the prime implicants of f. We show that given subsets $C'\subseteq C$ and $D'\subseteq D$ such that $(C',D')\neq(C,D)$, a new term in $(C\backslash C')\cup(D\backslash D')$ can be found in time $\mathrm{O}(n(t+n))+m^{\mathrm{o}(\log m)}$, where $m=|C'|+|D'|$. In particular, if $f(x)$ can be evaluated for every $x\in\{0,1\}^n$ in polynomial time, then the forms c and d can be jointly generated in incremental quasi-polynomial time. On the other hand, even for the class of $\wedge,\vee$-formulae f of depth 2, i.e., for CNFs or DNFs, it is unlikely that uniform sampling from within the set of the prime implicates and implicants of f can be carried out in time bounded by a quasi-polynomial $2^{\mathrm{polylog}(\cdot)}$ in the input size of f. We also show that for some classes of polynomial-time computable monotone Boolean functions it is NP-hard to test either of the conditions $D'=D$ or $C'=C$. This provides evidence that for each of these classes neither conjunctive nor disjunctive irredundant normal forms can be generated in total (or incremental) quasi-polynomial time. Such classes of monotone Boolean functions naturally arise in game theory, networks and relay contact circuits, convex programming, and include a subset of $\wedge,\vee$-formulae of depth 3. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Incremental polynomial time; Quasi-polynomial time; Dualization; NP-hardness; Monotone Boolean function; Monotone Boolean formula; Conjunctive normal form; Disjunctive normal form; Prime implicate; Prime implicant; Relay contact circuit; Positional game; Convex programming

* Corresponding author.
E-mail address: leonid@cs.rutgers.edu (L. Khachiyan)

[1] On leave from the International Institute of Earthquake Prediction Theory and Mathematical Geophysics, Moscow.

0166-218X/99/$ - see front matter
PII: S0166-218X(99)00099-2

1. Introduction

Let $f:\{0,1\}^n \to \{0,1\}$ be a monotone Boolean function of n variables:

$$x \geqslant x' \Rightarrow f(x) \geqslant f(x') \quad \text{for any } x, x' \in \{0,1\}^n.$$

Denote by

$$c = \bigwedge_{I \in C} \bigvee_{i \in I} x_i, \tag{1.1}$$

the irredundant conjunctive normal form (CNF) of f, where C is the set of the prime implicates $I \subseteq \{1,\dots,n\}$ of f. Note that the anti-characteristic vector of any prime implicate $I \in C$ is a maximal false vector of f, and vice versa. Thus, there is a natural one-to-one correspondence $C \rightleftharpoons \mathrm{MAX}\{x \mid f(x)=0\}$.

Similarly, let

$$d = \bigvee_{J \in D} \bigwedge_{j \in J} x_j \tag{1.2}$$

be the irredundant disjunctive normal form (DNF) of the function f, where D is the set of the prime implicants $J \subseteq \{1,\dots,n\}$ of f. The characteristic vector of any prime implicant $J \subseteq \{1,\dots,n\}$ is a minimal true vector of f, which gives a bijection $D \rightleftharpoons \mathrm{MIN}\{x \mid f(x)=1\}$. By definition,

$$f(x) = c(x) = d(x) \quad \text{for all } x \in \{0,1\}^n. \tag{1.3}$$

In this paper, we investigate the complexity of generating the irredundant normal forms c and/or d for various input representations of f. Let $\{\cdot\}$ denote either C, or D, or the set $C \sqcup D$ of all $|C|+|D|$ prime implicates and implicants of f. We consider the following problems:

Gen$\{\cdot\}$:*Given a subset* $S \subseteq \{\cdot\}$, *either prove that* $S = \{\cdot\}$, *or find a new element in* $\{\cdot\} \backslash S$.

Section 2 deals with problem *Gen*$\{C \sqcup D\}$. In Theorem 1 we show that this problem can be solved in incremental quasi-polynomial time provided that $f(x)$ can be evaluated for any $x \in \{0,1\}^n$ in polynomial time. Specifically, given two subsets $C' \subseteq C$ and $D' \subseteq D$ of total size $m = |C'|+|D'| < |C|+|D|$, a new element in $(C \backslash C') \cup (D \backslash D')$ can be generated in time $\mathrm{O}(n(t+n)) + m^{\mathrm{o}(\log m)}$, where t is the complexity of evaluating $f(x)$ at a binary point x. Note that this result implies that the condition $(C', D') = (C, D)$ can also be checked in $\mathrm{O}(n(t+n)) + m^{\mathrm{o}(\log m)}$ time.

An important special case of Theorem 1 is for $D' = D$. In such a case, f is already represented by its irredundant DNF and consequently $f(x)$ can be evaluated in polynomial time. Next, computing the irredundant CNF for f is equivalent to computing the irredundant DNF for the dual function $f^{\mathrm{d}}(x) \doteq \neg f(\neg x)$. This problem is known as *Dualization* or *Transversal Hypergraph* – see e.g. [1–5,10,13]. Theorem 1 thus implies that the dualization problem for monotone DNFs can be solved in incremental quasi-polynomial time ([5] – see Theorem 2 below). In fact, Theorem 1 rests upon this result, and the polynomial-time solvability of the dualization problem would imply the solvability of problem $Gen\{C \sqcup D\}$ in incremental polynomial time [1].

Another straightforward consequence of Theorem 1 is as follows. Suppose that $f(x)$ can be evaluated for each $x \in \{0,1\}^n$ in quasi-polynomial time $2^{\mathrm{polylog}(\cdot)}$, where $(\cdot)$ is the size of the input encoding of f and x. Then the set $C \sqcup D$ can be constructed in time bounded by a quasi-polynomial in the *total* input and output size. Theorem 3 in Section 2 shows that, even for the class of $\wedge, \vee$-formulae f of depth 2, it is unlikely that uniform sampling from within $C \sqcup D$ can be carried out in time bounded by a quasi-polynomial $2^{\mathrm{polylog}(\cdot)}$ in the input size of f. Specifically, the existence of such a randomized algorithm would imply that any NP-complete problem can be solved in quasi-polynomial time by a randomized algorithm with arbitrarily small failure probability. Our arguments are similar to those used by Jerrum et al. [9] for the problem of uniformly generating cycles in a digraph.

Finally, in Section 3 we consider problems $Gen\{C\}$ and $Gen\{D\}$. In Theorems 4–7 we show that for some natural classes of polynomial-time computable monotone Boolean functions it is NP-hard to test either of the conditions $C' = C$ or $D = D'$. Modulo the standard bijections $C \rightleftharpoons \mathrm{MAX}\{x \mid f(x) = 0\}$ and $D \rightleftharpoons \mathrm{MIN}\{x \mid f(x) = 1\}$, our examples of such sets C (or D) are as follows:

- all prime implicates (or implicants) of a $\wedge, \vee$-formula of depth 3;
- all minimal subsets of relays connecting (or disconnecting) two terminals in a monotone relay circuit;
- all minimal winning sets of Player 1 (or 2) for a positional game form with perfect information;
- all maximal feasible (or minimal infeasible) subsystems of a system of convex inequalities.

For each of the above examples, problems $Gen\{C\}$ and $Gen\{D\}$ cannot be solved in total (and hence incremental) quasi-polynomial time, unless any problem in NP is solvable in quasi-polynomial time. But for all these examples, Theorem 1 guarantees that problem $Gen\{C \sqcup D\}$ can be solved in incremental quasi-polynomial time.

2. Simultaneously generating *C* and *D*

In this section we show that problem $Gen\{C \sqcup D\}$ can be solved in incremental quasi-polynomial time.

Theorem 1. *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone Boolean function whose value at any point $x \in \{0,1\}^n$ can be determined in time t, and let C and D be the sets of the prime implicates and the prime implicants of f, respectively. Given two subsets $C' \subseteq C$ and $D' \subseteq D$ of total size $m = |C'| + |D'| < |C| + |D|$, a new element in $(C \backslash C') \cup (D \backslash D')$ can be found in time $\mathrm{O}(n(t+n)) + m^{\mathrm{o}(\log m)}$.*

As mentioned in the Introduction, Theorem 1 follows from its special case which deals with the dualization problem for monotone DNFs – cf. [1]. For this reason, we start with the dualization problem:

Problem ($\mathscr{D}\mathscr{D}^*$). *Given a pair of irredundant DNFs*

$$d[A] = \bigvee_{I \in A} \bigwedge_{i \in I} x_i, \qquad d[B] = \bigvee_{J \in B} \bigwedge_{j \in J} x_j,$$

test whether $d[A]$ *and* $d[B]$ *are mutually dual*:

$$d[A](x_1,\dots,x_n) = \neg d[B](\neg x_1,\dots,\neg x_n) \quad \text{for all } x = (x_1,\dots,x_n) \in \{0,1\}^n. \qquad (\mathscr{D})$$

If $d[A]$ *and* $d[B]$ *are not dual, find a Boolean vector* $x^* \in \{0,1\}^n$ *such that*

$$d[A](\neg x_1^*,\dots,\neg x_n^*) = d[B](x_1^*,\dots,x_n^*). \qquad (\mathscr{D}^*)$$

It is easy to see that any dual disjunctive normal forms $d[A]$ and $d[B]$ must satisfy the condition

$$I \cap J \neq \emptyset \quad \text{for all } I \in A \quad \text{and} \quad J \in B. \qquad (2.1)$$

Suppose to the contrary that there is a pair of disjoint sets $I \in A$ and $J \in B$. Then the characteristic vector of J satisfies ($\mathscr{D}^*$).

Lemma 1 below shows that for any pair of dual irredundant forms we also have

$$\max\{|I| : I \in A\} \leqslant |B|, \qquad \max\{|J| : J \in B\} \leqslant |A|. \qquad (2.2)$$

Lemma 1. *Suppose that irredundant DNFs* $d[A]$ *and* $d[B]$ *satisfy* (2.1). *If condition* (2.2) *is violated, Eq.* ($\mathscr{D}^*$) *can be solved in* $O(|A| + |B| + n^2)$ *time.*

Proof. First of all, (2.2) can be checked in $O(|A| + |B|)$ time. If $|J| > |A|$ for some $J \in B$, a solution x^* of Eq. ($\mathscr{D}^*$) can be found as follows:

Initialize $x^* \leftarrow 0$

For each $I \in A$, select an index $i \in I \cap J$ and set $x_i^* \leftarrow 1$.

This procedure takes $O(n|A|)$ time. Since $|A| < |J| \leqslant n$, we obtain the time bound as required. Similarly, if $|I| > |B|$ for some $I \in A$, Eq. ($\mathscr{D}^*$) can be solved in $O(n|B|)$ time. Again, $|B| < |I| \leqslant n$, which proves the lemma. □

Theorem 2 (Fredman and Khachiyan [10]). *Suppose that* $d[A]$ *and* $d[B]$ *satisfy* (2.1) *and* (2.2). *Then problem* ($\mathscr{D}\mathscr{D}^*$) *can be solved in time* $v^{\chi(v)+O(1)}$, *where* $v = |A||B|$ *and* $\chi^\chi = v$.

From

$$\chi(v) \sim \log v / \log\log v = o(\log v),$$

the trivial inequality

$$v = |A||B| \leqslant (|A| + |B|)^2,$$

and Lemma 1 we obtain the following complexity bound.

Corollary 1. *If* $d[A]$ *and* $d[B]$ *satisfy* (2.1), *Problem* ($\mathscr{D}\mathscr{D}^*$) *can be solved in time* $T_{dual} = \mathrm{O}(n^2) + (|A| + |B|)^{\mathrm{o}(\log(|A|+|B|))}$.

Proof of Theorem 1. Suppose that $C' \subseteq C$ and $D' \subseteq D$, where C and D are defined by (1.1) and (1.2). For $A \subseteq C$, let $c[A] = \bigwedge_{I \in A} \bigvee_{i \in I} x_i$. With this notation, (1.3) implies $c[C'](x) \geqslant c[C](x) \equiv c(x) \equiv f(x) \equiv d(x) \equiv d[D](x) \geqslant d[D'](x)$. Hence $(C', D') = (C, D)$ if and only if $c[C'](x) \equiv d[D'](x)$, which is equivalent to the duality of $d[C']$ and $d[D']$. In particular, we have $I \cap J \neq \emptyset$ for all $I \in C$ and $J \in D$. By Corollary 1, the duality of $d[C']$ and $d[D']$ can be tested in time $T_{dual} = \mathrm{O}(n^2) + m^{\mathrm{o}(\log m)}$, where $m = |C'| + |D'|$. If $(C', D') = (C, D)$, we are done. Otherwise we obtain a solution x^* of Eq. ($\mathscr{D}^*$). It is easy to see that $c[C'](x^*) = 1$ and $d[D'](x^*) = 0$. Now we compute $f(x^*)$ and split into two cases.

Case 1: $f(x^*) = 0$. By evaluating $f(\cdot)$ at $\mathrm{O}(n)$ binary points, we can find a vector $y^* \in \mathrm{MAX}\{x \mid f(x) = 0\}$ such that $x^* \leqslant y^*$. Since f is monotone, $0 = f(y^*) < 1 = c[C'](x^*) \leqslant c[C'](y^*)$. This means that $I = \{i \mid y_i^* = 0,\ i = 1, \ldots, n\} \in C \backslash C'$, i.e., we obtain a new prime implicate of f.

Case 2: $f(x^*) = 1$. Find a vector $y^* \in \mathrm{MIN}\{x \mid f(x) = 1\}$ such that $y^* \leqslant x^*$. The set $J = \{j \mid y_j^* = 1,\ j = 1, \ldots, n\} \in D \backslash D'$ is a new prime implicant of f. □

In the remainder of this section we discuss the complexity of uniformly sampling from $C \sqcup D$. A randomized algorithm $\mathscr{R}$ is an ε-*uniform generator* for a finite set Ω if

(i) $\mathscr{R}$ outputs only elements $\omega \in \Omega$, unless it stops with no output;

(ii) $\sum \{p(\omega) \mid \omega \in \Omega\} \geqslant 1/2$, where $p(\omega)$ is the probability that $\mathscr{R}$ outputs $\omega \in \Omega$;

(iii) $\max\{p(\omega)/p(\omega') \mid \omega, \omega'\} \leqslant 1 + \varepsilon$.

Theorem 3 below shows that a fast uniform generator for $C \sqcup D$ is unlikely to exist, even if we restrict the input to the class $\mathscr{DNF}_2$ of quadratic monotone DNFs. Note that the input size of any formula $f(x_1, \ldots, x_n) \in \mathscr{DNF}_2$ is polynomial in n.

Theorem 3. *Let* $\rho < 1$ *be a fixed constant, and let* $\varepsilon = 2^{n^\rho}$. *Suppose there exists a* (*quasi*) *polynomial-time randomized algorithm that, given a formula* $f(x_1, \ldots, x_n) \in \mathscr{DNF}_2$, *acts as an* ε-*uniform generator for the set* $C \sqcup D$ *of the prime implicates and implicants of f. Then any* NP-*complete problem can be solved in* (*quasi*) *polynomial time by a randomized algorithm with arbitrarily small one-sided failure probability.*

Proof. Since for any formula $f \in \mathscr{DNF}_2$ the set D is given explicitly and $|D| \leqslant n^2$, any ε-uniform generator $\mathscr{R}$ for $C \sqcup D$ can be used as an ε-uniform generator for C. This entails at most $\mathrm{O}(n^2)$ slowdown in the running time of $\mathscr{R}$. We can thus assume that there exists a (quasi) polynomial-time 2^{n^ρ}-uniform generator for C or equivalently, $\mathrm{MAX}\{x \mid f(x) = 0\}$.

For a given graph $G = (V, E)$ with n vertices, define $f_G(x_1, \ldots, x_n) = \vee\{x_i x_j \mid (ij) \notin E\}$. Then $\mathrm{MAX}\{x \mid f_G(x) = 0\}$ is the set of (the characteristic vectors of) all maximal cliques in G. In other words, $\mathscr{R}$ can be used to 2^{n^ρ}-uniformly generate maximal cliques in G. To show that this implies the theorem, we need only slightly modify the proof suggested by Jerrum et al. [9] for the problem of generating cycles in digraphs.

Let $H_k = \mathscr{K}_{2,2,\dots,2}$ be the complete k-partite graph, each "part" of which consists of two isolated vertices. Thus, H_k has $2k$ vertices and 2^k maximal cliques of size k each. Let $G(H_k)$ be the $2nk$-vertex graph obtained by substituting H_k for each vertex of G. Then $N(G(H_k), kl) = 2^{kl} N(G, l)$, where $N(\cdot, t)$ is the number of maximal cliques of size t in $(\cdot)$. Furthermore, $N(G(H_k), t) = 0$ if $t \neq 0 \bmod k$. Since the total number of cliques in G is bounded by 2^n, we conclude that for $k \geqslant 1 + n + (2nk)^\rho$, any $2^{(2nk)^\rho}$-uniform generator $\mathscr{R}$ of maximal cliques in $G(H_k)$ produces a clique of *maximum size* with probability $\geqslant \frac{1}{4}$. Letting $k = \Theta(n^{1/(1-\rho)})$, we can satisfy the inequality $k \geqslant 1 + n + (2nk)^\rho$ and find a maximum clique in $G(H_k)$ with high probability in (quasi) polynomial time. But this is equivalent to solving the NP-complete clique problem for any imput graph G. $\square$

The proof of Theorem 3 also shows that there is little hope that false vectors of a monotone quadratic DNF can be uniformly generated in polynomial time. It should be pointed out that Karp and Luby [12] gave a simple polynomial-time algorithm for uniformly generating true vectors of an arbitrary, not necessarily monotone or quadratic, DNF.

We also mention in passing that problem $Gen\{x \mid x$ a maximal clique in $G\}$ can be solved in incremental polynomial time. In fact, all maximal cliques in a graph can be generated with polynomial delay – see [10].

3. Generating C or D

In this section we describe some classes of monotone Boolean functions for which it is NP-hard to separately check either of the conditions $C' = C$ or $D' = D$. This provides evidence that for each of these classes, problems $Gen\{C\}$ and $Gen\{D\}$ cannot be solved in total (or incremental) quasi-polynomial time. Our first example is as follows.

3.1. Monotone Boolean formulae of depth 3

Theorem 4. *Let $\mathscr{F}_3$ be the class of $\wedge, \vee$-formulae of depth* 3. *For a formula $f \in \mathscr{F}_3$, let C and D denote the sets of the prime implicates and the prime implicants of f, respectively.*

(i) *Given a formula $f \in \mathscr{F}_3$ and a subset C' of C, it is co*NP-*complete to decide whether $C' = C$.*

(ii) *Similarly, for a formula $f \in \mathscr{F}_3$ and a subset D' of D, it is co*NP-*complete to determine whether $D' = D$.*

Proof. Since the class $\mathscr{F}_3$ is self-dual, parts (i) and (ii) of the theorem are equivalent. To show part (ii), it is convenient to state (ii) in the following equivalent form:

$\mathscr{E}$: *Given a formula* $f(x) \in \mathscr{F}_3$ *and a monotone DNF* $d(x)$ *such that* $f(x) \geqslant d(x)$ *for all* $x \in \{0,1\}^n$, *it is co*NP-*complete to check whether* $f(x) \equiv d(x)$.

It is well known that it is *co*NP-complete to test whether a given (non-monotone) DNF $D(x_1,\dots,x_n)$ is a tautology. Substituting y_i for $\neg x_i$, $i=1,\dots,n$, we can transform $D(x_1,\dots,x_n)$ into a monotone form $d(x_1,y_1,\dots,x_n,y_n)$ such that

$$d(x,y) \equiv D(x) \text{ for } y = \neg x.$$

Let $\phi(x,y) = \bigwedge_{i=1}^{n}(x_i \vee y_i)$. It is easy to see that $D(x)$ is a tautology, i.e.,

$$D(x) = 1 \quad \text{for all } x \in \{0,1\}^n,$$

if and only if

$$d(x,y) \vee \phi(x,y) = d(x,y) \quad \text{for all } x,y \in \{0,1\}^n.$$

Since $f(x,y) \doteq \phi(x,y) \vee d(x,y)$ is a $\wedge,\vee$-formula of depth 3 such that $f(x,y) \geqslant d(x,y)$, claim $\mathscr{E}$ and the theorem follows. □

Note that since any Boolean formula can be evaluated at any binary point in polynomial time, from Theorem 1 it follows that problem $Gen\{C \sqcup D\}$ can be solved in incremental quasi-polynomial time for any $\wedge,\vee$-formula. Observe also that any $\wedge,\vee$-formula of depth 2 is in conjunctive or disjunctive normal form. Theorem 2 thus implies that for $\wedge,\vee$-formulae of depth 2, problems $Gen\{C\}$ and $Gen\{D\}$ can be solved in incremental quasi-polynomial time. In addition, it is not hard to show that problems $Gen\{C\}$ and $Gen\{D\}$ can be solved with polynomial delay for any *read-once* $\wedge,\vee$-formula. (A formula is read-once if each variable appears in it exactly once – see e.g., [8,11].

3.2. Monotone relay circuits

Let $G=(V,E)$ be a graph with two distinguished vertices $s,t \in V$. A *monotone relay circuit* is a mapping $R : E \to \{1,\dots,n\}$, which assigns a relay $R(e) \in \{1,\dots,n\}$ to each edge $e \in E$ – cf. [14]. For a relay set $X \subseteq \{1,\dots,n\}$, let $ON(X) = \{e \in E \mid R(e) \in X\}$ and $OFF(X) = E \setminus ON(X)$.We say that X *connects* the terminals s and t if the graph $(V, ON(X))$ contains an s,t-path. Similarly, X *disconnects* the terminals if s and t are not connected in $(V, OFF(X))$. We shall call a minimal X connecting (disconnecting) the terminals s and t a *relay path* (*cut*), respectively.

Theorem 5. *Let* Π *be the class of series–parallel monotone relay circuits. Given a circuit in* Π *and a collection of relay* s,t*-cuts* (*or relay* s,t*-paths*), *it is co*NP-*complete to determine whether the given collection is complete.*

Proof. For a relay circuit R, let $f_R : \{0,1\}^n \to \{0,1\}$ be the monotone Boolean function realized by the circuit:

$$\begin{aligned} f_R(x_1,\dots,x_n) &= 1 \quad \text{if the set } \{i \mid x_i = 1,\ i = 1,\dots,n\} \text{ connects } s \text{ and } t, \\ f_R(x_1,\dots,x_n) &= 0 \quad \text{otherwise.} \end{aligned} \tag{3.1}$$

Clearly, each relay s,t-cut (path) is a prime implicate (implicant) of f_R, and vice versa. Since any $\wedge,\vee$-formula can be easily realized by a series–parallel relay circuit, Theorem 5 follows from Theorem 4. □

As before, $f_R(x)$ can be evaluated for each binary vector x in polynomial time. Hence all relay cuts and paths in an arbitrary monotone relay circuit can be jointly generated in incremental quasi-polynomial time.

If the relay mapping $R : E \to \{1,\dots,n\}$ is bijective, the relay cuts and paths turn into the usual cuts and paths, which can be (separately) generated with polynomial delay for any graph G.

3.3. *Positional two-person games with perfect information*

Let $G = \langle V, E\rangle$ be a directed acyclic graph with a distinguished vertex s such that all vertices $v \in V$ are reachable form s. A *two-person positional game on* G is a partitioning

$$V = V_1 \cup V_2, \qquad V_1 \cap V_2 = \emptyset, \tag{3.2}$$

where V_1 and V_2 are the sets of positions controlled by Players 1 and 2, respectively.

Let $E^+(v)$ be the set of arcs incident from a position $v \in V$. The game starts in the initial position s. If the current position v is in V_α, $\alpha = 1,2$, Player α selects a move from $E^+(v)$ until the game reaches a final position $u \in V_T = \{v \in V \mid E^+(v) = \emptyset\}$. The player who controls the final position wins – cf. [15].

A *game form* Γ specifies the partitioning (3.2) on $V \backslash V_T$, but does not indicate the winners on the set V_T of final positions. A subset $X \subseteq V_T$ is called a *winning set of Player* α if this player can force the game to finish in X, regardless of the adversary's moves.

Theorem 6. *Let* $\alpha = 1$ *or* 2. *Given a positional game form* Γ *and a list of minimal winning sets of Player* α, *it is co*NP-*complete to decide whether the given list is exhaustive.*

Proof. Assume $V_T = \{1,\dots,n\}$ and consider the following Boolean function:

$$\begin{aligned} f_\Gamma(x_1,\dots,x_n) &= 1 \quad \text{if the set } \{i \mid x_i = 1,\ i = 1,\dots,n\} \text{ is a winning set of Player 1}, \\ f_\Gamma(x_1,\dots,x_n) &= 0 \quad \text{otherwise} \end{aligned}$$

– see [6,7]. Clearly f_Γ is monotone, and each minimal winning set of Player 1 is a prime implicant of f_Γ and vice versa. Furthermore, the prime implicates of f_Γ are nothing but the minimal winning sets of Player 2. It is also easy to see that any $\wedge, \vee$-formula of size l and depth d can be realized by a game form of the same size and depth. For this reason, Theorem 6 follows from Theorem 4. □

Any positional game with perfect information can be solved in polynomial time by dynamic programming. Hence $f_\Gamma(x)$ can be evaluated for each x in polynomial time. From Theorem 1 we conclude that all minimal winning sets of Players 1 and 2 can be jointly generated in incremental quasi-polynomial time.

Let us remark that due to the obvious one-to-one correspondence between positional game forms and combinatorial $\wedge, \vee$-circuits, all minimal winning sets of each player can be generated with polynomial delay for positional games on trees.

3.4. Convex programming

Given a system $\mathscr{P} = (P_1, \dots, P_n)$ of polyhedra in $\mathscr{R}^d$ consider the monotone Boolean function

$$\begin{aligned} &f_{\mathscr{P}}(x_1, \dots, x_n) = 1 \quad \text{if} \bigcap_{\{i|x_i=1\}} P_i = \emptyset, \\ &f_{\mathscr{P}}(x_1, \dots, x_n) = 0 \quad \text{otherwise.} \end{aligned} \tag{3.3}$$

By definition, each maximal false vector of $f_{\mathscr{P}}$ corresponds to a maximal feasible subsystem of polyhedra in $\mathscr{P}$, whereas each minimal true vector of $f_{\mathscr{P}}$ can be viewed as a minimal infeasible subsystem of $\mathscr{P}$.

As an example, let $\mathscr{P}$ be the set of all facets of polytope $Q = \{y \in \mathscr{R}^d \mid a_i y \leqslant b_i, i = 1, \dots, n\}$. Then problem $Gen\{x \mid x$ a maximal feasible subsystem of $\mathscr{P}\}$ is equivalent to generating all vertices of Q. The complexity status of the latter problem is not known. In general, however, generating all maximal feasible subsystems of a system of polyhedra is hard. Analogously, generating all minimal infeasible systems of $\mathscr{P}$ can also be hard:

Theorem 7. *For a system $\mathscr{P}$ of nonempty polyhedra in $\mathscr{R}^d$ and a collection of maximal feasible (minimal infeasible) subsystems of $\mathscr{P}$, it is co*NP*-complete to tell whether the given collection is complete.*

Proof. Let $R : E \to \{1, \dots, n\}$ be an arbitrary relay circuit on a series–parallel graph $G = (V, E)$. It is easy to see that for any edge $e \in E$, all s–t-paths through e cross e in the same direction, which we refer- to as the s–t-orientation of e.

Denote by G_i the graph $(V, OFF(\{x_i\}))$ with the s–t-orientation on the set of its edges, and let P_i be the s–t-flow polyhedron for the diagraph $G_i, i = 1, \dots, n$. In other words, P_i consists of all vectors $y \in \mathscr{R}^E$ such that

$$y(e) = 0, \quad e \in ON(\{x_i\}),$$

$$y(e) \geqslant 0, \quad e \in OFF(\{x_i\}),$$

$$\sum\{y(e)\,|\,e\,\text{incident from } s\} = 1,$$

$$\sum\{y(e)\,|\,e\,\text{incident from } v\} - \sum\{y(e)\,|\,e\,\text{incident to } v\} = 0, \quad v \in V\backslash\{s,t\}.$$

For this polyhedral system we have $f_{\mathscr{P}}(x) \equiv \neg f_R(\neg x)$, i.e., Definitions (3.3) and (3.1) give mutually dual Boolean functions. This means that Theorem 7 is a corollary of Theorem 5. □

Since linear programming is polynomial-time solvable, from Definition (3.3) it follows that $f_{\mathscr{P}}(x)$ can be computed for each $x \in \{0,1\}^n$ in polynomial time. Again, we conclude that all maximal feasible and minimal infeasible subsystems of an arbitrary system of convex polyhedral sets can be jointly generated in incremental quasi-polynomial time.

Acknowledgements

The first author gratefully acknowledges the partial support of the Office of Naval Research under Grants N0014-92-J-1375 and N0014-92-J-4083. The second author's research was supported in part by ONR Grant N0014-92-J-1375 and NSF Grants CCR-9208371 and CCR-9208539.

References

[1] J.C. Bioch, T. Ibaraki, Complexity of identification and dualization of positive Boolean functions, Inform. and Comput. 123 (1995) 50–63.

[2] E. Boros, V. Gurvich, P.L. Hammer, Dual subimplicants of positive Boolean functions, RUTCOR Research Report 11-93, Rutgers Center for Operations Research, Rutgers University, New Brunswick, New Jersey 08903, 1993.

[3] Y. Crama, Dualization of regular Boolean functions, Discrete Appl. Math. 16 (1987) 79–85.

[4] T. Eiter, G. Gottlob, Identifying the minimal transversals of a hypergraph and related problems, SIAM J. Comput. 24 (6) (1995) 1278–1304.

[5] M. Fredman, L. Khachiyan, On the complexity of dualization of monotone disjunctive normal forms, J. Algorithms 21 (1996) 618–628.

[6] V. Gurvich, To theory of multistep games, USSR Comput. Math. Math. Phys. 13 (6) (1973) 1485–1500.

[7] V. Gurvich, Solvability of positional games in pure strategies, USSR Comput. Math. Math. Phys. 15 (2) (1975) 357–371.

[8] V. Gurvich, On repetition-free Boolean functions, Uspekhi Mat. Nauk 32 (1977) 183–184 (in Russian).

[9] M.R. Jerrum, L.G. Valiant, V.V. Vazirani, Random generation of combinatorial structures from a uniform distribution, Theoret. Comput. Sci. 43 (1986) 169–188.

[10] D.S. Johnson, M. Yannakakis, C.H. Papadimitriou, On generating all maximal independent sets, Inform. Process. Lett. 27 (1988) 119–123.

[11] M. Karchmer, N. Linial, I. Newman, M. Saks, A. Widgerson, Combinatorial characterization of read-once formulae, Discrete Math. 114 (1993) 275–282.

[12] R.M. Karp, M. Luby, Monte-Carlo algorithms for enumeration and reliability problems, Proceedings of the 24th IEEE Symposium on Foundations of Computer Science, 1983, pp. 56–64.

[13] U.N. Peled, B. Simeone, An $O(nm)$ – time algorithm for computing the dual of a regular Boolean function, Discrete Appl. Math. 49 (1994) 309–323.
[14] C.E. Shannon, A symbolic analysis of relay and switching circuits, Trans. AIEE 57 (1938) 713–723.
[15] E. Zermelo, Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels, Proceedings of the 5th Congress of Mathematicians, Cambridge 1912, Cambridge Univ. Press, Cambridge, 1913, pp. 501–504.