Rutcor
Research
Report

Generalized sandwich problem for Π- and Δ-free multigraphs and its applications to positional extensions of game forms. [a]

Endre Boros [b] Vladimir Gurvich [c]

RRR 11-2013, November 2013

RUTCOR
Rutgers Center for
Operations Research
Rutgers University
640 Bartholomew Road
Piscataway, New Jersey
08854-8003
Telephone: 732-445-3804
Telefax: 732-445-5472
Email: rrr@rutcor.rutgers.edu
http://rutcor.rutgers.edu/∼rrr

[a]This first author gratefully acknowledges the partial support by the NSF grants IIS-0803444 and CMMI-0856663.

[b]RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003; (boros@rutcor.rutgers.edu)

[c]RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003; (gurvich@rutcor.rutgers.edu)

Rutcor Research Report
RRR 11-2013, November 2013

Generalized sandwich problem for Π- and Δ-free multigraphs and its applications to positional extensions of game forms. [1]

Endre Boros Vladimir Gurvich

Abstract. An *n-multigraph* $\mathcal{G} = (V; E_i \mid i \in I)$ is a complete graph $G = (V, E)$ whose edges are covered by $n = |I|$ sets $E = \cup_{i \in I} E_i$, some of which might be empty. If this cover is a partition, then we call $\mathcal{G}$ an *n-graph*. We say that an n-graph $\mathcal{G}' = (V; E'_i \mid i \in I)$ is an edge-subgraph of an n-multigraph $\mathcal{G} = (V; E_i \mid i \in I)$ if $E'_i \subseteq E_i$ for all $i \in I$. We denote by Δ the n-graph on three vertices with three nonempty sets each containing a single edge, and by Π the four-vertex n-graph with two non-empty sets each of which contains the edges of a P_4. In this paper, we recognize in polynomial time whether a given n-multigraph $\mathcal{G}$ contains a Π- and Δ-free n-subgraph, or not, and if yes provide a polynomial delay algorithm to generate all such subgraphs. The above decision problem can be viewed as a direct generalization of the sandwich problem for P_4-free graphs introduced and solved by Golumbic, Kaplan, and Shamir in 1995.
As a motivation and application, we consider the n-person positional game forms, which are known to be in a one-to-one correspondence with Π- and Δ-free n-graphs. Given game form g, making use of the above result, we recognize in polynomial time whether g is a subform of a positional (that is, tight and rectangular) game form and, if yes, we generate with polynomial delay all such positional extensions of g.

Keywords: edge-colored n-graph, Gallai graph, sandwich problem, positional, tight, and rectangular game forms; polynomial delay.

1 Introduction

Let n be a positive integer, $I = \{1, 2, ..., n\}$ be the set of *colors*, V be a finite set of vertices, and let us denote by E the set of edges of the complete graph on vertex set V. By a *multicoloring* $C : E \to 2^I$ of the edges of the complete graph (V, E) by color set I, we mean an assignment of a nonempty subset $C(e) \subseteq I$ of the colors to each of the edges $e \in E$.

Given a multicoloring, let us denote by $E_i = \{i \in I \mid e \in C(e)\}$ the set of edges colored by color $i \in I$. Clearly, these sets cover the edge set E, that is, we have

$$E \;=\; \bigcup_{i \in I} E_i. \tag{1}$$

Conversely, given an edge cover (as in (1)), we can define a multicoloring by setting $C(e) = \{i \in I \mid e \in E_i\}$ for all edges $e \in E$.

Given sets $E_i \subseteq E$, $i \in I$, satisfying (1), we call the tuple $\mathcal{G} = (V; E_1, \ldots, E_n) = (V; E_i \mid i \in I)$ an *n-multigraph*. We call the graphs $G_i = (V, E_i)$, $i \in I$ the *chromatic components* of $\mathcal{G}$, and the set I the set of colors of $\mathcal{G}$. We call an n-multigraph $\mathcal{G}$ an *n-graph*, if the chromatic components are pairwise edge disjoint, or in other words, if the sets E_i, $i \in I$ form a partition of E.

We do not require to use all the colors of I in a multicoloring C of E. Consequently, some of the sets E_i, $i \in I$ in an n-(multi)graph $\mathcal{G}$ may be empty. This implies that any n-(multi)graph $\mathcal{G}$ is also an n'-(multi)graph for all $n' \geq n$.

Let us also note that a 2-graph $\mathcal{G}$ is simply a graph, that is, its two chromatic components are a graph and its complement; furthermore, any pair of complementary graphs constitute a 2-graph in this way. Thus, we can view n-(multi)graphs as a generalization of graphs.

We will also talk about subgraphs and sub(multi)graphs of an n-(multi)graph $\mathcal{G} = (V; E_i \mid i \in I)$. Given a subset $S \subseteq V$ of the vertices, we denote by $\mathcal{G}[S] = (S; F_i \mid i \in I)$ the *vertex sub(multi)graph* of $\mathcal{G}$ induced by vertex set S, that is, where $F_i = E_i[S] = \{(u, v) \in E_i \mid u, v \in S\}$ for $i \in I$. Given subsets $E'_i \subseteq E_i$ for $i \in I$, we call the n-(multi)graph $\mathcal{G}' = (V; E'_i \mid i \in I)$ and *edge sub(multi)graph* of $\mathcal{G}$. Whenever it will not cause ambiguity, we shall omit the words "vertex" or "edge" and simply talk about submultigraphs or subgraphs. In particular, if $S \subseteq V$ is a vertex subset and $\mathcal{G}' = (S; F_i \mid i \in I)$ is an n-graph, which is a vertex subgraph of an edge submultigraph of $\mathcal{G}$, we simply call $\mathcal{G}'$ an n-subgraph of $\mathcal{G}$.

We will be concerned with the two special n-graphs, defined as follows:

- $\Pi = (V; E_1, E_2)$ is a 2-graph on four vertices $V = \{v_1, v_2, v_3, v_4\}$ defined by

  $E_1 = \{(v_1, v_2), (v_2, v_3), (v_3, v_4)\}$ and $E_2 = \{(v_2, v_4), (v_4, v_1), (v_1, v_3)\}$.

  In other words, both chromatic components of Π is a P_4 on the four vertices of V.

- $\Delta = (V; E_1, E_2, E_3)$ is a 3-graph on three vertices $V = \{v_1, v_2, v_3\}$ defined by

  $E_1 = \{(v_1, v_2)\}$, $E_2 = \{(v_2, v_3)\}$ and $E_3 = \{(v_1, v_3)\}$.

  In other words, Δ is a 3-colored triangle.

As we noted above, both Π and Δ are also n-graphs for any $n \geq 3$; Figure 1 for an illustration of these n-graphs.

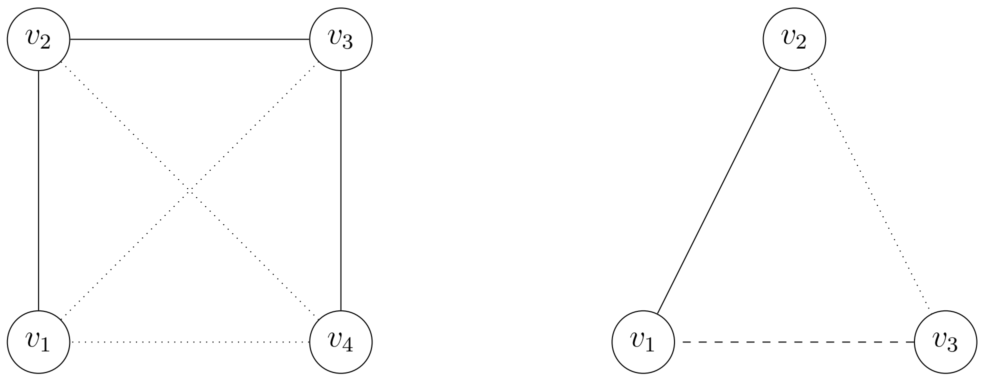


Figure 1: n-graphs Π and Δ.

Standardly, an n-graph $\mathcal{G}$ is called Π- *and* Δ-*free* if $\mathcal{G}$ does not contain Π or Δ as its (induced) vertex subgraph.

Let us remark that both Π and Δ, were introduced in 1967 by Gallai in his seminal paper [15]; Δ-free n-graphs are frequently called Gallai's graphs in the literature [4, 10, 30]. We will call them Gallai's n-graphs, which is, perhaps, more accurate.

In Section 3 we shall recall that Π- and Δ-free n-graphs are in a one-to-one correspondence with positional tree structures and hence with n-person positional game forms (see Section 5 for more details). Furthermore, to an n-person game form g one can naturally assign an n-multigraph $\mathcal{G}(g)$ such that g can be extended to a positional game form if and only if $\mathcal{G}(g)$ has a Π- and Δ-free edge n-subgraph. (We delay the precise definitions and the proofs of these facts till Section 5, due to their technical nature). These connections are the main motivation for the first part of our paper, in which we consider the following decision problem about the existence of a Π- and Δ-free edge subgraph:

$\Pi\Delta FREE(\mathcal{G})$: Given the n-multigraph $\mathcal{G} = (V; E_i \mid i \in I)$, does it have a Π- and Δ-free edge subgraph $\mathcal{G}' = (V; E'_i \mid i \in I)$?

Our first result asserts that the above problem can be answered efficiently.

Theorem 1. *Given an n-multigraph $\mathcal{G} = (V; E_i \mid i \in I)$, the problem $\Pi\Delta FREE(\mathcal{G})$ can be answered and if the answer is "YES", a Π- and Δ-free edge subgraph $\mathcal{G}' = (V; E'_i \mid i \in I)$ can be constructed, in $O(n|V|^3)$ time.*

Let us notice that the problem $\Pi\Delta FREE(\mathcal{G})$ and Theorem 1 are natural generalizations of the so called sandwich problem for the P_4-free graphs that was introduced and solved by Golumbic, Kaplan and Shamir in 1995 ([18]):

P_4-free sandwich problem (V, E'_1, E'_2): Given a complete graph $G = (V, E)$ and two disjoint subsets $E'_1 \subseteq E$ and $E'_2 \subseteq E$, whether there exists a subset $E' \subseteq E \setminus (E'_1 \cup E'_2)$ such that the graph $G' = (V, E'_1 \cup E')$ is P_4-free?

In [18] this problem was solved in $O(|V|^3)$ time.

Lemma 1. *The P_4-free sandwich problem (V, E'_1, E'_2) is a special case of $\Pi\Delta FREE(\mathcal{G})$.*

Proof. Let $\mathcal{G} = (V; E_1, E_2)$ be the 2-multigraph defined by $E_i = E \setminus E'_i$ for $i = 1, 2$. It is straightforward that all edge subgraphs are Δ-free; furthermore, an edge subgraph $\mathcal{G}' = (V; E''_1, E''_2)$ is also Π-free, if and only if the graph $G' = (V, E''_2)$ is an affirmative answer to the P_4-free sandwich problem for (V, E'_1, E'_2). This is due to the following obvious implications:

$$E''_2 = E \setminus E''_1 \supseteq E'_1 \;\Leftarrow\; E''_1 \subseteq E \setminus E'_1 \text{ and } E''_2 \subseteq E \setminus E'_2 \;\Rightarrow\; E''_2 \cap E'_2 = \emptyset. \qquad \square$$

Next, let us note that if the n-multigraph $\mathcal{G}$ has a Π- and Δ-free edge subgraph $\mathcal{G}' = (V; E'_i \mid i \in I)$ then it may have many of them and their number may even be exponential in the size of $\mathcal{G}$. Although it is natural to ask in such cases to generate all the Π- and Δ-free edge subgraphs of $\mathcal{G}$, this problem is clearly not polynomial, because of the potentially large number of the output graphs. For such situations it is customary to measure the efficiency of a generating procedure in both the input and output sizes. In particular, we say that this generation problem is solvable with *polynomial delay* if there is an algorithm outputting all Π- and Δ-free edge subgraphs $\mathcal{G}_1$, $\mathcal{G}_2$, ..., $\mathcal{G}_N$ of $\mathcal{G}$ in this order such that the computational time spent between the ith and $(i+1)$st outputs for any i is bounded by a polynomial of the size of the input graph $\mathcal{G}$. Our next result claims that all solutions to problem $\Pi\Delta FREE(\mathcal{G})$ can be generated efficiently, with polynomial delay:

Theorem 2. *Given an n-multigraph $\mathcal{G} = (V; E_i \mid i \in I)$, all Π- and Δ-free edge subgraphs of $\mathcal{G}$ can be generated with the polynomial delay of $O(n|V|^3)$.*

By Lemma 1, this Theorem implies that given a P_4-free sandwich problem (V, E'_1, E'_2), where $E'_1 \cap E'_2 = \emptyset$, all P_4-free graphs $G = (V, F)$ such that $E'_1 \subseteq F$ and $E'_2 \cap F = \emptyset$ can be generated with polynomial delay, further generalizing the result of [18].

The paper is organized as follows. In Section 2 we recall more properties of Π- and Δ-free graphs and prove Theorem 1. Then, in Section 3 we recall relations of the corresponding n-graphs and positional tree structures. in Section 4, we make use of this connection to provide an algorithmic proof for Theorem 2. In Section 5 we develop game theoretic applications of the obtained results.

2 Complementary connected multigraphs

2.1 Definition and hereditary property

An n-multigraph $\mathcal{G} = (V; E_i \mid i \in I)$ is called *complementary connected (CC)* [22, 24, 28] if the complementary graphs $\overline{G}_i = (V, \overline{E}_i)$ of its chromatic components $G_i = (V, E_i)$ are connected on V for all $i \in I$. (Since E is the edge set of the complete graph on vertex set V, we denote by $\overline{E'} = E \setminus E'$ the complementary edge set for an arbitrary edge subset $E' \subseteq E$.

It is very easy to see that the CC property is hereditary.

Lemma 2. *Each edge submultigraph (and in particular, every edge subgraph) of a CC n-multigraph is CC, too.*

Proof. It follows since adding edges to a graph connected on V keeps it connected. □

2.2 Minimal non CC n-graphs

Clearly, all chromatic components of an n-multigraph $\mathcal{G} = (V; E_1, \ldots, E_n)$ are empty if and only if $|V| = 1$. In this case G is called *trivial* and by convention we assume that it is *not* CC. Furthermore, up to an isomorphism there is only one n-graph with two vertices, $|V| = 2$, and it is not CC either. Yet, there are CC n-graphs Δ and Π with 3 and 4 vertices, respectively. Furthermore, there is no CC n-graph with $|V| \leq 3$ distinct from Δ. Hence, Π and Δ have no proper CC vertex subgraphs, that is, they are minimal CC n-graphs. The following theorem claims that there are no others.

Theorem 3. *Every CC n-graph contains Π or Δ as a subgraph.*

This statement can be strengthened as follows:

Theorem 4. *Every CC n-graph $\mathcal{G}$, except for Π and Δ, has a vertex $v \in V$ such that the induced subgraph $\mathcal{G}[V \setminus \{v\}]$ is still CC.*

In other words, not only each CC n-graph $\mathcal{G}$ contains a subgraph Π or Δ, but $\mathcal{G}$ can also be reduced to one of them by successive elimination of vertices, so that all intermediate n-graphs are CC, too. By Theorems 3, Π and Δ are the only minimal and, by Theorem 4, they are the only "locally minimal" CC n-graphs.

These two theorems have been announced in [22, 24] and complete proofs appeared in [6, 28]. Several other remarkable properties of n-graphs Π and Δ are discussed in [3, 29].

Remark 1. *The case of $n = 2$ is simpler than the general one, since Δ cannot exist when $n \leq 2$. For this reason, this case was studied earlier in [21, 39, 40, 41]. Theorem 3 for $n = 2$ was suggested as a problem for the 34th Moscow Mathematical Olympiad in 1971 (see Problem 72 in [16]) and successfully solved by seven high school students.*

2.3 Recoloring Π- and Δ-free n-graphs

The following simple statement plays an important role in our proof of Theorem 1. The statement claims that by recoloring to the same color all edges between the components of an arbitrary vertex partition we cannot create a new Π or Δ in an n-graph. Given an n-multigraph $\mathcal{G} = (V; E_i \mid i \in I)$, standardly a set-family $\mathcal{V} = \{V_j \subseteq V \mid j \in J\}$ is called a *vertex partition* of V if its subsets V_j are pairwise disjoint and their union is V.

Given a vertex partition $\mathcal{V}$, its *cut set* $F(\mathcal{V})$ is defined by all edges between the parts of $\mathcal{V}$, that is, $F(\mathcal{V}) = \{(v', v'') \mid v' \in V_{j'},\ v'' \in V_{j''},\ j' \neq j'',\ j', j'' \in J\}$.

Lemma 3. *Given an n-graph $\mathcal{G} = (V; E_i \mid i \in I)$, a vertex partition $\mathcal{V} = \{V_j \mid j \in J\}$, and a color $i_0 \in I$, let us set $E'_i = E_i \setminus F(\mathcal{V})$ for all $i \in I \setminus \{i_0\}$, and set $E'_{i_0} = E_{i_0} \cup F(\mathcal{V})$. Then the obtained n-graph $\mathcal{G}' = (V; E'_i \mid i \in I)$ contains a vertex subgraph Π or Δ if and only if the induced subgraph $\mathcal{G}[V_j] = \mathcal{G}'[V_j]$ does for some $j \in J$.*

Proof. We can prove this claim by showing that if $W \subseteq V$ induces a Π or Δ in $\mathcal{G}'$ then $W \subseteq V_j$ for some $j \in J$, and hence, it induces the same subgraph in $\mathcal{G}$ as well. Indeed, if the partition $\mathcal{V}$ induces a nontrivial partition of W, then it would be a partition of the induced Π or Δ in which all edges between the different parts have the *same* color i_0. Yet, it is easy to see that neither Π nor Δ admit such a partition. □

We will say that the n-graph $\mathcal{G}'$, of the above lemma, is obtained from $\mathcal{G}$ by *recoloring its cut set* $F(\mathcal{V})$ by color i_0.

2.4 Verification of existence of Π- and Δ-free edge subgraphs

From Lemma 3 we derive a polynomial time algorithm for the problem $\Pi\Delta FREE(\mathcal{G})$.

Proof of Theorem 1: Let us first test if a given n-multigraph $\mathcal{G}$ is CC, or not. If it is, then by Lemma 2, every its edge subgraphs $\mathcal{G}'$ is CC too. Then, by Theorem 3, every such edge subgraphs $\mathcal{G}'$ contains a vertex subgraph Π or Δ. Consequently, we answer problem $\Pi\Delta FREE(\mathcal{G})$ in the negative.

Otherwise, if $\mathcal{G}$ is not CC, then there is a color $i_0 \in I$ such that graph $\overline{G}_{i_0} = (V, \overline{E}_{i_0})$ is not connected. Let $\mathcal{V} = \{V_j \mid j \in J\}$ be the vertex partition of V induced by the connected components of $\mathcal{G}$. Note that we must have $F(\mathcal{V}) \subseteq E_{i_0}$. Then, let us delete all other colors assigned to the edges of $F(\mathcal{V})$ in $\mathcal{G}$, that is, define $E'_i = E_i \setminus F(\mathcal{V})$ for all $i \in I \setminus \{i_0\}$ and set $E'_{i_0} = E_{i_0}$. By this construction, the obtained n-multigraph $\mathcal{G}' = (V; E'_i \mid i \in I)$ is an edge submultigraph of $\mathcal{G}$. Furthermore, by recoloring the cut set $F(\mathcal{V})$ by color i_0 we obtain an edge subgraph of $\mathcal{G}'$ from any edge subgraph of $\mathcal{G}$, and all edge subgraphs of $\mathcal{G}'$ arise in this way. Consequently, Lemma 3 implies that $\mathcal{G}$ has a Π- and Δ-free edge subgraph if and only if $\mathcal{G}'$ does, which on its own turn is equivalent (again by Lemma 3) with the fact that all induced submultigraphs $\mathcal{G}[V_j] = \mathcal{G}'[V_j]$, $j \in J$, have Π- and Δ-free edge subgraphs.

In other words, problem $\Pi\Delta FREE(\mathcal{G})$ has the affirmative answer if and only if problem $\Pi\Delta FREE(\mathcal{G}')$ does, which is if and only if problems $\Pi\Delta FREE(\mathcal{G}'[V_j]) = \Pi\Delta FREE(\mathcal{G}[V_j])$ have affirmative answers for all $j \in J$. It is also clear that if we have obtained Π- and Δ-free edge subgraphs for all induced submultigraphs $\mathcal{G}[V_j] = \mathcal{G}'[V_j]$, $j \in J$, then taking their union and coloring the edges of $F(\mathcal{V})$ by color i_0 we obtain a Π- and Δ-free edge subgraph of $\mathcal{G}$.

Thus, we can proceed with problems $\Pi\Delta FREE(\mathcal{G}[V_j])$, $j \in J$ recursively as follows. Since $G_i = (V, E_{i_0})$ is not CC, we have $|V_j| < |V|$ for all $j \in J$, and consequently the recursion has depth at most $|V|$. Let the running time of this algorithm for $\mathcal{G}$ be $T(n, |V|)$. Since testing connectivity of a graph can be done in linear time in the number of its edges, the above recursion proves that

$$T(n, |V|) \leq O(n|V|^2) + \sum_{j \in J} T(n, |V_j|),$$

from which $T(n, |V|) = O(n|V|^3)$ follows. □

In Section 4 we will strengthen this result and show that in fact *all* such Π- and Δ-free edge subgraphs can be generated with polynomial delay.

3 One-to-one correspondence between Π- and Δ-free n-graphs and n-person positional tree structures

3.1 Decomposing Π- and Δ-free n-graphs

By Theorem 3, for any Π- and Δ-free n-graph $\mathcal{G} = (V; E_i \mid i \in I)$ there exists a color $i \in I$ such that the complementary graph $\overline{G}_i = (V, \overline{E_i}) = (V, \bigcup_{j \in I \setminus \{i\}} E_j)$ is not connected. The following lemma implies that there is exactly one such $i \in I$.

Lemma 4. *([22], see also [28]). Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be two graphs on the common vertex-set V such that both complementary graphs $\overline{G}_1 = (V, \overline{E}_1)$ and $\overline{G}_2 = (V, \overline{E}_2)$ are not connected. Then $E_1 \cap E_2 \neq \emptyset$.*

Proof. Let us consider the connected components of both $\overline{G}_1$ and $\overline{G}_2$, and denote by Q the smallest such component. We can assume w.l.o.g. that Q is a component of $\overline{G}_1$. Choose an arbitrary vertex $v \in Q$, and let R be the connected component of $\overline{G}_2$ containing v. Then $V \neq Q \cup R$, since otherwise $Q \setminus R$ would be the union of some of the other connected components of $\overline{G}_2$, contradicting that Q was chosen as a smallest component among all connected components of $\overline{G}_1$ and $\overline{G}_2$. Thus, we have a vertex $w \in V \setminus (Q \cup R)$. Then, we must have $(v, w) \in E_1 \cap E_2$. □

Thus, given a Π- and Δ-free n-graph $\mathcal{G} = (V; E_i \mid i \in I)$, by Theorem 3 and Lemma 4 there exists a unique color $i = i(\mathcal{G}) \in I$ such that the graph $\overline{G}_i = (V, \overline{E}_i)$ is not connected. Let us consider the connected components V_j, $j \in J$ of $\overline{G}_i$, and consider the corresponding induced n-graphs $\mathcal{G}_j = \mathcal{G}[V_j]$ for $j \in J$. Obviously, there are at least two of them and all of them are still Π- and Δ-free. Hence, by Theorem 3 and Lemma 4 there exists a unique colors $i(j) = i(\mathcal{G}[V_j]) \in I$ for each $j \in J$ such that the graphs $\overline{G[V_j]}_{i(j)} = (V_j, \overline{E_{i(j)}[V_j]})$ are not connected for all $j \in J$. We can repeat the above decomposition recursively, until all components contain only a single vertex. Let us notice that $i(j) \neq i(\mathcal{G})$, since V_j was defined as the vertex-set of a connected component of $\overline{G}_i = (V, \overline{E}_i)$.

Thus, we get a decomposition tree $T = T(\mathcal{G})$ whose leaves $L(T)$ are in a one-to-one correspondence with the vertices $V = \{v_1, \ldots, v_m\}$ of the n-graph $\mathcal{G}$, while the "interior vertices" $U(T) = V(T) \setminus L(T)$ of T correspond to the induced subgraphs of $\mathcal{G}$ and are labeled by colors of $I = \{1, \ldots, n\}$, in accordance with the above decomposition procedure.

Let us denote the obtained mappings by $\phi : L(T) \to V$ and $\psi : U(T) \to I$, respectively; by definition, the first one is a bijection. Furthermore, the vertex $u_0 \in U(T)$ that corresponds to the whole n-graph $\mathcal{G}$ will be called the *root* of T.

We will refer to this decomposition structure as $\mathcal{T}_\mathcal{G} = (T, u_0, \phi, \psi)$. Note that if $\mathcal{G}$ is a Π- and Δ-free n-graph then the mapping $\mathcal{G} \mapsto \mathcal{T}_\mathcal{G}$ is uniquely defined by the above procedure.

Remark 2. *This decomposition was suggested in [22, 24]. The special case* $n = 2$ *is simpler, since* Δ *cannot exist, and it was considered earlier in [15, 21, 23, 32, 31, 40, 41, 39]. For this case a more general (so called substitution or modular) decomposition was also introduced by Gallai [15] and then studied in many papers; see, for example, [8, 9, 36, 37, 38].*

3.2 Positional tree structures

The quadruple $\mathcal{T} = (T, u_0, \phi, \psi)$ will be called a *positional tree structure*. It can be naturally interpreted as a positional game (with perfect information and without moves of chance) in which $I = \{1, \ldots, n\}$ is the set of players and $V = \{v_1, \ldots, v_m\}$ is the set of outcomes.

Let $T = (U \cup L, E)$ be a tree. Its vertices $U \cup L$ are *positions*; they correspond to the subgraphs of $\mathcal{G}$ obtained by the above decomposition. The leaves L are called the *final or terminal positions*; they are in one-to-one correspondence ϕ with the vertices of $\mathcal{G}$ and form the set of *outcomes* of the game. Furthermore, to each non-terminal position $u \in U$ we assign a *player* $i = i(u) \in I$ who makes a decision, that is, a *move*, in the position u by choosing any immediate successor u' of u. This means that in the n-graph $\mathcal{G}(u)$ the complement to the chromatic component i is disconnected and one of its connected components forms the n-subgraph $\mathcal{G}(u')$. The game begins in the initial position u_0, corresponding to the original n-graph $\mathcal{G}$, and ends in a final position corresponding to a vertex v of $\mathcal{G}$. The unique path from u_0 to v in T is called a *play*.

According to section 2.4, we make the following two assumptions:

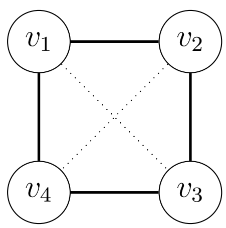
- (A1) there are at least two possible moves in every non-terminal position;
- (A2) no player makes two moves in a row.

Let us note that these two assumptions do not violate the generality, since they can always be enforced by trivial modifications of an arbitrary tree structure $\mathcal{T} = (T, u_0, \phi, \psi)$.

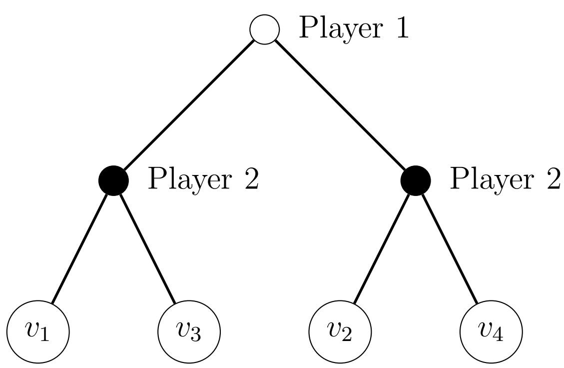
Four examples are given in Figures 2-5. On the right hand side of these figures some other objects are also shown that we will explain later.

3.3 Inverse mapping

Let $\mathcal{T} = (T, u_0, \phi, \psi)$ be a positional tree structure in which T is a tree with the root u_0, interior nodes $U = U(T)$, and leaves $L = L(T)$. Furthermore, let $\phi : L \to V$ be the *terminal bijection*, while $\psi : U \to I$ be the *player assignment*, $V = \{v_1, \ldots, v_m\}$ and $I = \{1, \ldots, n\}$.



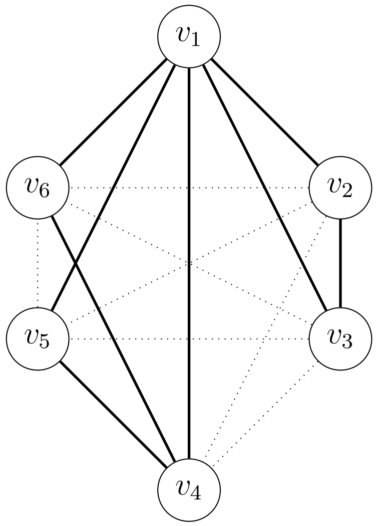
$$\mathcal{S}_1 = \{\{1,3\},\{2,4\}\}$$
$$\mathcal{S}_2 = \{\{1,2\},\{2,3\},\{3,4\},\{4,1\}\}$$



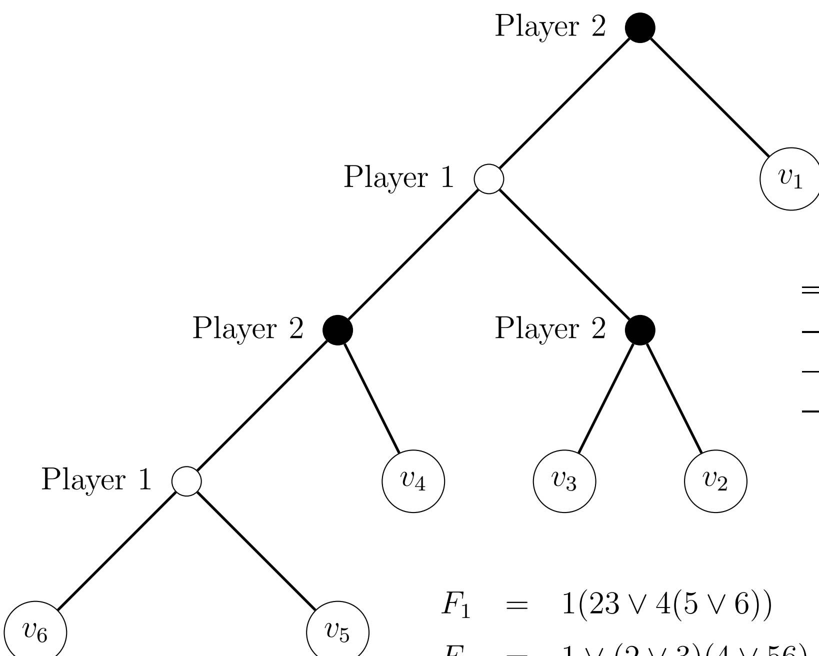
	12	23	34	41
13	1	3	3	1
24	2	2	4	4

$$F_1 = 13 \vee 24$$
$$F_2 = (1 \vee 3)(2 \vee 4) = 12 \vee 13 \vee 34 \vee 41$$

Figure 2: A Π-free 2-graph and the corresponding positional and normal game forms. Families $\mathcal{S}_1$ and $\mathcal{S}_2$ are the sets of maximal stable sets in the two chromatic components. For simplicity, on the right side we use index i instead of vertex v_i.



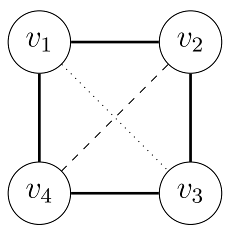
$$\mathcal{S}_1 = \{\{1\},\{2,4\},\{3,4\},\{2,5,6\},\{3,5,6\}\}$$
$$\mathcal{S}_2 = \{\{1,2,3\},\{1,4,5\},\{1,4,6\}\}$$



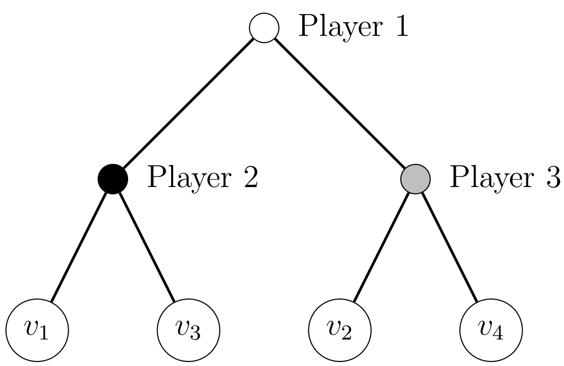
	1	24	34	256	356
123	1	2	3	2	3
145	1	4	4	5	5
146	1	4	4	6	6

$$F_1 = 1(23 \vee 4(5 \vee 6)) = 123 \vee 145 \vee 146$$
$$F_2 = 1 \vee (2 \vee 3)(4 \vee 56) = 1 \vee 24 \vee 34 \vee 256 \vee 356$$

Figure 3: Another Π-free 2-graph and the corresponding positional and normal game forms.



$$\mathcal{S}_1 = \{\{1,3\},\{2,4\}\}$$
$$\mathcal{S}_2 = \{\{1,2,4\},\{2,3,4\}\}$$
$$\mathcal{S}_3 = \{\{1,2,3\},\{1,3,4\}\}$$



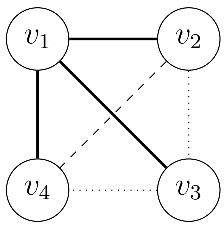
13	123	134
124	1	1
234	3	3

24	123	134
124	2	4
234	2	4

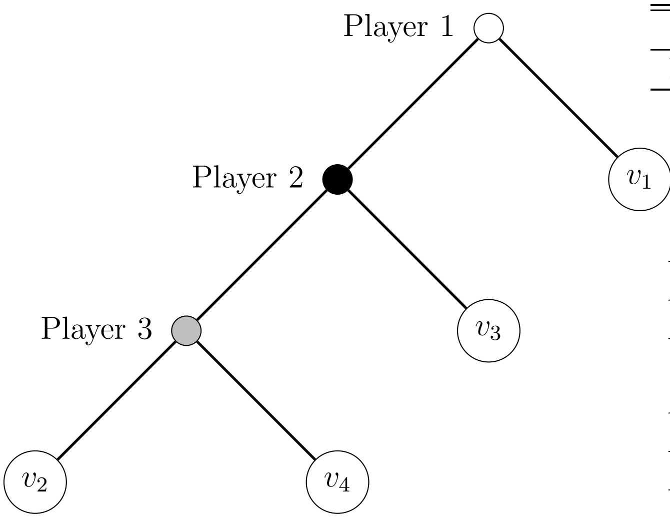
$$F_1 = 13 \vee 24 = 13 \vee 24$$
$$F_2 = (1 \vee 3)24 = 124 \vee 234$$
$$F_3 = 13(2 \vee 4) = 123 \vee 134$$

$$F_{23} = (1 \vee 3)(2 \vee 4) = 12 \vee 23 \vee 34 \vee 41$$
$$F_{13} = 13 \vee 2 \vee 4 = 13 \vee 2 \vee 4$$
$$F_{12} = 1 \vee 3 \vee 24 = 1 \vee 3 \vee 24$$

Figure 4: A Π- and Δ-free 3-graph and corresponding positional and normal game forms. Note that this normal form is a $2 \times 2 \times 2$ array. We show above its two-dimensional slices, corresponding to player 1's two possible strategies, 13 and 24, as indicated in the upper left corner of the tables.



$$
\begin{aligned}
\mathcal{S}_1 &= \{\{1\},\{2,3,4\}\}\\
\mathcal{S}_2 &= \{\{1,3\},\{1,2,4\}\}\\
\mathcal{S}_3 &= \{\{1,2,3\},\{1,3,4\}\}
\end{aligned}
$$



1	123	134
13	1	1
124	1	1

234	123	134
13	3	3
124	2	4

$$
\begin{aligned}
F_1 &= 1 \vee 324 &&= 1 \vee 234\\
F_2 &= 1(3 \vee 24) &&= 13 \vee 124\\
F_3 &= 13(2 \vee 4) &&= 123 \vee 134
\end{aligned}
$$

$$
\begin{aligned}
F_{23} &= 1(3 \vee 2 \vee 4) &&= 12 \vee 13 \vee 14\\
F_{13} &= 1 \vee 3(2 \vee 4) &&= 1 \vee 23 \vee 34\\
F_{12} &= 1 \vee 3 \vee 24 &&= 1 \vee 3 \vee 24
\end{aligned}
$$

Figure 5: Another Π- and Δ-free 3-graph and the corresponding positional and normal forms.

We can assign to $\mathcal{T}$ an n-graph $\mathcal{G} = (V; E_i \mid i \in I\}$ as follows. For any two distinct leaves $v', v'' \in L(T)$ let $u = u(v', v'') \in U(T)$ denote their last common ancestor. In other words, let us consider the plays p', p'' from u_0 to v' and to v'', respectively and define $u = u(v', v'') \in U(T)$ as the last common position of these two plays. Furthermore, we color the edge $(\phi(v'), \phi(v''))$ by color i whenever $\psi(u) = i \in I$. By the above construction, all edges of the complete graph on vertex set V receive colors form I.

Thus, we obtain an n-graph $\mathcal{G} = (V; E_i \mid i \in I)$. It is Π- and Δ-free. To show this it is enough to consider all tree structures with three and four leaves and verify that none of them generate Π or Δ. We leave the details for the reader.

Let us finally remark that if the positional tree structure $\mathcal{T} = (T, u_0, \phi, \psi)$ satisfies axioms (A1) and (A2), then the above definitions yield a unique Π- and Δ-free n-graph $\mathcal{G} = \mathcal{G}_{\mathcal{T}}$.

Furthermore, the following stronger claim can be recalled from the literature.

Theorem 5. *([22], see also [24, 28]). The mappings, $\mathcal{G} \mapsto \mathcal{T}_{\mathcal{G}}$ and $\mathcal{T} \mapsto \mathcal{G}_{\mathcal{T}}$ defined above are mutually inverse one-to-one correspondences between positional tree structures satisfying assumptions* (A1, A2) *and* Π- *and* Δ-*free n-graphs.* □

4 Generating all Π- and Δ-free edge-subgraphs of a n-multigraph with polynomial delay

4.1 General plan

Given an n-multigraph $\mathcal{G} = (V; E_i \mid i \in I)$, we will generate all its Π- and Δ-free edge-subgraphs with polynomial delay. In the previous section, it was shown that existence of such a subgraph can be verified in polynomial time as follows.

If $|V| = 1$ then there are no edges, $\mathcal{G}$ is trivial, and it is Π- and Δ-free itself. Otherwise, let us verify whether $\mathcal{G}$ is CC. If "yes" then it has no Π- and Δ-free edge-subgraphs. If "no", let us find an $i \in I$ such that the graph $\overline{G}_i = (V, \overline{E}_i)$ is not connected and partition it into *connected* vertex-components $P_i : (V = \cup V^j \mid j \in J_i)$. (Obviously, $|J_i| > 1$, since $|V| > 1$.)

Then, for each $j \in J_i$, let us repeat the similar procedure for the induced vertex-submultigraph $\mathcal{G}[V^j]$. If it is CC then $\mathcal{G}$ has no Π- and Δ-free edge-subgraphs.

Otherwise, there is a color $i' \in I$ such that the vertex-subgraph $\overline{G}_{i'}[V^j]$ is not connected on V^j. (Obviously, $i' \neq i$, since V^j is a connected component of $\overline{G}_i$, by construction.)

Repeating this procedure recursively, for all appearing CC components, we will either prove that the original n-multigraph $\mathcal{G}$ has no Π- and Δ-free edge-subgraphs, or obtain such a subgraph; more precisely, we will get the corresponding positional tree structure.

Remark 3. *Conditions (A1) and (A2) hold, by construction. Indeed, $|J_i| > 1$ whenever $|V| > 1$ and $i' \neq i$ for any two successive colors implying (A1) and (A2), respectively.*

Let us note, however, that the above algorithm produces not all but only some special tree structures, because we restrict ourselves by the partitions $P_i : (V = \cup V^j \mid j \in J_i)$ of V into the *connected* vertex-components of the graph $\overline{G}_i = (V, \overline{E}_i)$. This is enough for verifying the existence of a Π- and Δ-free edge-subgraph (or equivalently, a positional tree structure) in $\mathcal{G}$ but this requirement must be relaxed if we want *all* such structures. Then, we must consider *all* partitions $Q : (V = \cup W^\ell \mid \ell \in L)$ satisfying the following weaker conditions:

- (B1) P_i is a subpartition of Q, that is, each W^ℓ is the union of V^j, or more precisely, there is a partition $R : (J_i = \cup J_i^\ell \mid \ell \in L)$ such that $W^\ell = \cup_{j \in J_i^\ell} V^j$ for all $\ell \in L$.

- (B2) For every $\ell \in L$ there is an $i' \in I$ such that $i' \neq i$ and the vertex-subgraph $\overline{G}_{i'}[W^\ell]$ is not connected on W^ℓ.

In particular, partition P_i itself satisfies (B1) and (B2) but there may be many others.

In the next two subsections, we will generate all such partitions with polynomial delay. (See [**?**] for the precise definitions related to efficiency of generating algorithms).

This will allow us to output all positional tree structures corresponding to $\mathcal{G}$ with polynomial delay, too, by the standard depth first search procedure.

4.2 Legal subsets and partitions

Given a partition $P_i : (V = \cup V^j \mid j \in J_i)$, a partition Q satisfying (B1) and (B2), as well as each part of Q, and the corresponding subsets of V and of J_i will be called *legal*. More precisely, an index-subset $J' \subseteq J_i$ and the corresponding vertex-subset $V_{J'} = \cup_{j \in J'} V_j$ are legal if there is a color i' distinct from i such that the induced vertex-subgraph $\overline{G}_{i'}[V_{J'}]$ is not connected on $V_{J'}$. Otherwise, the sets J' and $V_{J'}$ are called *illegal*.

If a singleton $\{j\} \subset J$ is illegal then the original n-multigraph $\mathcal{G}$ has no Π- and Δ-free edge-subgraphs. Thus, without loss of generality, we will assume that $|J| > 1$ and that all singletons of J are legal. By convention, the empty set is legal too.

The following statement will be instrumental for the generating algorithms.

Lemma 5. *Any legal packing (that is, a collection of pairwise disjoint legal subsets of J) can be extended to a legal partition of J.*

Proof. It would suffice to add all missing singletons. □

Lemma 6. *If J' and J'' are illegal and $J' \cap J'' \neq \emptyset$ then $J' \cup J''$ is illegal too.*

Proof. Indeed, if two vertex-sets of a graph intersect and induce connected subgraphs then the subgraph induced by their union is connected too. □

We will also need the following three corollaries.

Lemma 7. *If $J' \cup \{j\}$ and $J' \cup \{j'\}$ are illegal then $J' \cup \{j, j'\}$ is illegal too.*

Proof. Obviously, $J' \neq \emptyset$, since all singletons are legal. The rest follows from Lemma 6. □

Lemma 8. *If $J'' \subset J' \subseteq J$ and $J'' \cup \{j\}$ is illegal for every $j \in J' \setminus J''$ then J' is illegal too.*

Proof. It can be immediately derived from the previous lemma, by induction. □

In other words, it is a polynomial problem to verify whether a legal subset $J'' \subset J$ can be extended to a larger legal subset J' (such that $J'' \subset J' \subseteq J$). It is sufficient (and necessary) to verify whether $J'' \cup \{j\}$ is legal for a $j \in J' \setminus J''$. A subset $J'' \subset J$ satisfying the conditions of the last lemma will be called *forbidden*. Let us remark that it may be legal or illegal.

Lemma 9. *Every legal subset $J' \subseteq J$ contains an element $j \in J'$ such that $J' \setminus \{j\}$ is legal.*

Proof. If $|J'| \leq 2$, the claim holds, since all singletons and the empty set are legal.
For $|J'| > 2$, the claim follows immediately from the previous lemma. □

4.3 Accessible and strongly accessible set-families

Given a ground set V, a family F of its subsets is called:

- *accessible* if for any non-empty set $W \in F$ there is an element $w \in W$ such that $W \setminus \{w\} \in F$; in particular, $\emptyset \in F$;
- *strongly accessible* if for any two sets $W, W' \in F$ such that $W' \subset W$ there is an element $w \in W \setminus W'$ such that $W \setminus \{w\} \in F$.

For example, given a connected graph $G = (V, E)$, it is easily seen that all subsets $W \subseteq V$ such that the induced subgraphs $G[W]$ are connected form a strongly accessible set-family.

Furthermore, let $\mathcal{G} = (V; E_i \mid i \in I)$ be a CC n-graph and F be the family of all subsets $W \subseteq V$ such that the vertex-subgraphs $\mathcal{G}[W]$ are CC. Obviously, F is not even accessible. Indeed, by Theorem 3, F contains a subset $W \subseteq V$ such that $\mathcal{G}[W]$ is a Π or Δ and, obviously, the above definition does not hold for any such subset W.

Nevertheless, it is very easy to extend F to an accessible family F'. To do so, let us just add to F all subsets of each set W inducing a Π or Δ. Still, it is important to notice that although F' is accessible, yet, it may be not strongly accessible. To obtain such an example, it is sufficient to substitute a Π for a middle vertex of another Π.

According to Lemma 9, the legal subsets of J form an accessible family.

4.4 Three depth first search poly-delay generating algorithms

4.4.1 Generating accessible set-families

By this lemma, all legal subsets of J can be obtained by extensions of the smaller ones. Thus, the following recursive generating procedure looks natural. Let us begin with the singletons (or even with the empty set), in each step extend every new legal set by one element, ignoring all illegal sets that will appear. It is easily seen that the above procedure is only incremental polynomial. Indeed, having M legal subsets of cardinality m, we should verify $M(k - m)$ subsets of cardinality $m + 1$ from $J = \{1, \ldots, k\}$.

Yet, just a very minor modification is needed to obtain a polynomial delay procedure. Namely, let us keep the generating tree but only replace the depth by the breadth first search.

Remark 4. *Applying the breadth first search is natural, because we immediately output all obtained sets. Yet, we may output exponentially many of them and then wait an exponential (in the size of the ground set $|J|$) time for the next one.*

In contrast, the depth first search procedure will hold potentially ready outputs to make the delay "more uniform", which guarantees that it becomes polynomial.

Remark 5. *Let us also notice that the above polynomial delay generating algorithm was further developed and improved in [7] for the strongly accessible set-families.*

4.4.2 Generating all legal subsets and partitions

By lemma 9, the legal subsets of J form an accessible family and, hence, they can be generated with polynomial delay, according to the previous section. Yet, we will have to generate not only all legal subsets but all legal partitions of J, as well.

Before doing so, let us recall that a packing or a partition of J is legal whenever every its set is legal and furthermore, by Lemma 5, any legal packing can be extended to a legal partition by just adding all missing singletons.

To avoid repetitions, we will order the sets in partitions. To do so, first, let us order the elements of J, say, $J = \{j_1, \ldots, j_k\}$, and for any considered partition $J = J_1 \cup \ldots, J_\ell$ assume that the minimum element of $J \setminus \cup_{i=1}^{s-1} J_i$ belongs to J_s for all $s \in \{1, \ldots, \ell\}$. In particular, $j_1 \in J_1$, the minimum element of $J \setminus J_1$ is in J_2, etc.

Then, the algorithm of the previous section can easily be adapted to output all such legal partitions as follows: First, output all legal subsets J_1 containing j_1, then, all legal subsets $J_2 \subseteq J \setminus J_1$ containing its minimum element, etc. Again, applying the depth (rather than the breadth) first search we will output all legal partitions of J with polynomial delay and without repetitions.

4.4.3 Generating all Π and Δ-free edge-subgraphs of a multigraph

Given an n-multigraph $\mathcal{G} = (V; E_i \mid i \in I)$, we generate all its Π- and Δ-free edge-subgraph (in fact, the corresponding positional structures), in accordance with Sections 4.1 and 4.4.2.

If $|V| = 1$ then there are no edges, $\mathcal{G}$ is trivial, and it is Π- and Δ-free itself.

Otherwise, let us verify whether $\mathcal{G}$ is CC. If "yes" then it has no Π- and Δ-free edge-subgraphs. If "no", let us choose the minimum $i \in I = \{1, \ldots, n\}$ such that $\overline{G}_i = (V, \overline{E}_i)$ is not connected and partition it into connected vertex-components $P : (V = \cup V^j \mid j \in J)$. (Obviously, $|J| > 1$, since $|V| > 1$.)

Recall that a subset $J' \subseteq J$ is called legal if there is an $i' \in I$ such that $i' \neq i$ and the corresponding component of the vertex-subgraph $\overline{G}_{i'}[V^{J'}] = (V^{J'}, \overline{E}_{i'})$ is not connected on $V^{J'} = \cup_{j \in J'} V^j$. In particular, any singleton ($|J'| = 1$) is legal, even when $|V^{J'}| = 1$. Hence, every legal packing in J can be extended to a legal partition of J, by just adding the singletons. (A packing or partition are called legal if all its sets are legal.) Both legal sets and partitions can be output with polynomial delay.

Let us enumerate $J = \{j_1, \dots j_\ell\}$ and output the legal $J' \subseteq J$ containing j_1. minimizing the corresponding $i' \in I \setminus \{i\}$. Then we apply the same procedure to J', etc., applying the depth first search. This procedure will output the positional trees, corresponding to Π- and Δ-free edge-subgraph of $\mathcal{G}$, with polynomial delay and without repetitions.

5 Game forms

5.1 Basic definitions

Let $V = \{v_1, \dots, v_m\}$ be a set of outcomes (or candidates), $I = \{1, \dots n\}$ be a set players (or voters), and X_i be a finite set of strategies of the player $i \in I$.

A mapping $g : X \to V$ is called a *game form*, where $X = \prod_{i \in I} X_i$ is the direct product of the sets of strategies. The elements $x = (x_1, \dots, x_n) \in \prod_{i \in I} X_i = X$ are called *strategy profiles* or *situations*. Mapping g assigns an outcome to every such profile.

Let us notice that g may be (and typically is) not injective; in other words, the same outcome can be chosen in several different situations. A game form can be viewed as an n-person game in normal form in which payoffs are not defined yet.

A subset $X' \subseteq X$ is called a *box* if it is the product $X' = \prod_{i \in I} X'_i$ of n subsets $X'_i \subseteq X_i$.

A restriction of $g : X \to V$ to a box $X' \subseteq X$ is called a *subform* and denoted $g' \leq g$.

Subsets $K \subseteq I$ and $B \subseteq V$ are called *coalitions* and *blocks*, respectively.

To each outcome $v \in V$ let us assign a Boolean variable, which will be denoted by the same symbol v, for simplicity.

Given an n-person game form g and a coalition $K \subseteq I$, let us introduce a Boolean function $F_K = F_K(g)$ defined by the following disjunctive normal form (DNF)

$$F_K(g) = \bigvee_{x_K} \bigwedge_{x_{\overline{K}}} g(x_K, x_{\overline{K}}) = \bigvee_{x_K} \bigwedge_{v \in g(x_K)} v = \bigvee_{x_K} y_K, \tag{2}$$

where $x_K = (x_i \mid i \in K)$ and $x_{\overline{K}} = (x_i \mid i \notin K)$ are strategies of coalitions K and $\overline{K} = I \setminus K$, respectively; clearly, each pair $(x_K, x_{\overline{K}})$ forms a strategy profile.

By (2), for every coalition $K \subseteq I$, the implicants y_K of the DNF F_K are in one-to-one correspondence with the strategies $x_K \in X_K$. Then, the set of variables of y_K is $[y_K] = g(X_K) = \{g(x_K, x_{\overline{K}}) \mid \forall x_{\overline{K}} \in X_{\overline{K}}\}$, where $\overline{K} = I \setminus K$ is the complementary to K coalition.

For simplicity, we will write $K = i$, rather than $K = \{i\}$, when coalition K consists of only one player $i \in I$, for example, x_i, $x_{\bar{i}}$, F_i, $F_{\bar{i}}$, y_i, $y_{\bar{i}}$, $g(x_i)$, $g(x_{\bar{i}})$.

By convention, we assume that $F_I = v_1 \vee \dots \vee v_m$ and $F_\emptyset = v_1 \wedge \dots \wedge v_m = v_1 \dots v_m$.

Figures 2-5 provide four examples that illustrate all above definitions.

5.2 Equivalent strategies

Two strategies x'_K and x''_K of a coalition $K \in I$ in a game form g are called *equivalent* if $g(x') = g(x'')$ for any two strategy profiles $x' = (x'_K, x_{\overline{K}})$ and $x'' = (x''_K, x_{\overline{K}})$ in which the strategies of all players $i \notin K$ are the same.

From now on, we will assume that in every considered game form $g : X \to V$ *no player* $i \in I$ has a pair of equivalent strategies $x'_i, x''_i \in X_i$. (Although larger coalitions may have them. For example, the total coalition I has no equivalent strategies if and only if the mapping $g : X \to V$ is an injection, which is not typically the case.)

The above assumption can be easily enforced as follows. If there are several equivalent strategies in X_i then we just eliminate all but one. Doing so for each class of equivalent strategies and for each player $i \in I$, we obtain a subform $g' : X' \to V$ in which no player has equivalent strategies. "Essentially", g and g' are equivalent. Let us remark, however, that there is no way to get rid of equivalent strategies of a larger coalition.

5.3 Rectangularity

Given three strategy profiles, $x' = (x'_i \mid i \in I)$, $x'' = (x''_i \mid i \in I)$, and $x = (x_i \mid i \in I)$ in $X = \prod_{i \in I} X_i$, we will call x a *mixture* of x' and x'' if $x_i = x'_i$ or $x_i = x''_i$ for each $i \in I$.

A game form g is called *rectangular* if $g(x) = g(x') = g(x'')$ whenever $g(x') = g(x'')$ and x is a mixture of x' and x''.

It is easy to verify that all four game forms in the considered examples are rectangular.

The next two properties of rectangular game forms from [22] will be important.

Lemma 10. *If g is rectangular and $g(x'_K) \subseteq g(x''_K)$ then strategies x'_K and x''_K are equivalent.*

In other words, for all $K \subseteq I$ the DNFs F_K contain only prime implicants and have the same set of variables. Moreover, by convention of Section 5.2 , containment $g(x'_i) \subseteq g(x''_i)$ cannot hold for a player $i \in I$.

Proof of the lemma. Containment $g(x'_K) \subseteq g(x''_K)$ and rectangularity of g imply equality $g(x'_K, x_{\overline{K}}) = g(x''_K, x_{\overline{K}})$ for each strategy $x_{\overline{K}}$ of the complementary coalition $\overline{K}$. □

Given a game form g' and its subform $g \le g'$, that is, $g : X \to V$, $g' : X' \to V$, $X \subseteq X'$ is a box in X', and g is a restriction of g' to X. Obviously, g is rectangular whenever g' is rectangular. Moreover, the following property holds too.

Lemma 11. *If game form g is rectangular and its subform g' has the same set of outcomes, that is, $V = g(X) = g'(X') = V'$, then $g(x_K) = g'(x_K)$ for every $K \subseteq I$ and $x_K \in X_K$.*

In other words, $F_K(g')$ may contain more implicants than $F_K(g)$, yet all implicants y_K of $F_K(g)$ remain unchanged in $F_K(g')$; in particular, $F_K(g') \ge F_K(g)$, for all $K \subseteq I$.

Proof of the lemma. Containment $[y_K] \subseteq [y'_K]$ obviously holds, since g' is an extension of g. Let us show that it can be strengthen to equality $[y_K] = [y'_K]$ if g and g' are rectangular and have the same set of outcomes. Assume indirectly that $v \in [y'_K] \setminus [y_K]$. Then, $v \in g(X)$, since $g(X) = g'(X')$. Hence, by rectangularity, $v \in g(x_K) = [y_K]$ and we get a contradiction. □

In the rest of this section we will give several equivalent characterizations of rectangularity obtained in [22] and based on the following definitions.

Given a game form $g : X \to V$ and a pair of outcomes $v', v'' \in V$, let $I(v', v'') \subseteq I$ denote the set of players that do *not* have a strategy allowing to reach both outcomes v' or v''; in other words, there is no implicant y_i of F_i that contains both variables v' and v''

$$I(v', v'') = \{i \mid v', v'' \in g(x_i) \text{ for no strategy } x_i \in X_i\} \subseteq I. \quad (3)$$

Let $X^v = g^{-1}(v) \subseteq X$ denote the set of all strategy profiles $x \in X$ such that $g(x) = v$.

Lemma 12. *The following five properties of an n-person game form g are equivalent:*

- *(i)* g *is rectangular;*
- *(ii)* X^v *is a box for each* $v \in V$*;*
- *(iii)* $| \cap_{i \in I} g(x_i)| = 1$ *for every strategy profile* $x = (x_i \mid i \in I)$*;*
- *(iv)* g *is uniquely defined by the n-tuple of DNFs* $F(g) = (F_i(g) \mid i \in I)$*;*
- *(v)* $I(v', v'') \neq \emptyset$ *for any pair* $v', v'' \in V$.

Proof. All these statements are just simple reformulations of the definition. □

More details on the above two lemmas are given in [22].

By claim (ii), rectangular game forms can be viewed as discrete box partitions, since the box $X = \prod_{i \in I} X_i$ is partitioned into m boxes X^v for $v \in V = \{v_1, \ldots, v_m\}$.

Claim (iii) shows that for each strategy profile $x = (x_i \; ; | \; i \in I) \in X$ the corresponding n sets $(g(x_i) \; ; | \; i \in I)$ of outcomes have exactly one outcome in common, which immediately implies (iv). In particular, DNF $F_K(g)$ is uniquely defined by the vector of n DNFs $F = (F_i \mid i \in I)$ according to the formula $F_K(g) = \cap_{i \in K} F_i(g)$ meaning that

$$g(x_K) = \bigcap_{i \in K} g(x_i), \quad \text{where} \quad x_K = (x_i \mid i \in K). \quad (4)$$

All 2^n DNFs F_K have equal sets of variables, $[F_K(g)] = \cup_{i \in K} g(x_K) = [F(g)] \; \forall \; K \subseteq I$.

Claim (v) means that pre-images $g^{-1}(v')$ and $g^{-1}(v'')$ in X intersect for *no* pair $v', v'' \in V$. Indeed, by (3), $g^{-1}(v') \cap g^{-1}(v'') = \emptyset$ if and only if $I(v', v'') \neq \emptyset$.

Thus, to every n-person rectangular game form g we can assign an n-multigraph $\mathcal{G}(g) = (V; E_i \mid i \in I)$ in which $I(e) = I(v', v'') \neq$ for every edge $e = (v', v'') \in E$. This multigraph will be instrumental for our main results; see Sections 5.6.

5.4 On DNF profiles $F = (F_1, \ldots, F_n)$

By (iv), a rectangular game form g is uniquely defined by the n-tuple of DNFs $F(g) = (F_i(g) \mid i \in I)$, where $F_i(g) = \bigvee_{j_i \in J_i} y_i^{j_i}$; the implicants $y_i^{j_i}$ of DNF F_i are in one-to-one correspondence with the strategies $x_i^{j_i} \in X_i$ and $[y_i^{j_i}] = g(x_i^{j_i}) \subseteq V$ denote the corresponding sets of outcomes/variables. By (iii), there is exactly one common variable for every selection of n implicants $(y_i^{j_i} \mid i \in I)$, that is, $| \cap_{i \in I} [y_i^{j_i}]| = 1$ for any $j_i \in J_i$.

Furthermore, by Lemma 10 and assumptions A1 and A2 of Section 3.2, containments $[y_i'] \subseteq [y_i'']$ hold for no pair of implicants y_i', y_i'' of F_i and $i \in I$; in other words, all n DNFs F_i of F are irredundant, that is, have only prime implicants each of which appears only once.

These properties, but last, hold also for 2^n DNFs $F_K, K \subseteq I$: their implicants are also prime, yet, they may be redundant, since the same implicant may appear in F_K several times if $|K| > 1$. All 2^n DNFs ave the same set of variables, $[F_K] = \cup_{j_K \in J_K}[y_K^{j_K}] = [F] \; \forall \; K \subseteq I$.

Let us notice that $|X| = \prod_{i \in I}|X_i|$, while the size of $F(g)$ is bounded by $m\sum_{i \in I}|X_i|$ is typically much smaller. Hence, as an input, $F = F(g)$ would be more convenient than g. Yet, not every DNF-vector $F = (F_1, \ldots, F_n)$ is associated with a rectangular game form g.

Lemma 13. *Given a DNF profile $F = (F_1, \ldots, F_n)$, there is a rectangular game form g such that $F = F(g)$ if and only if*

$$\left|\bigcap_{i \in I}[y_i]\right| = 1 \textit{ for any implicant profile } y = (y_i \mid i \in I). \tag{5}$$

Proof. The "only if" part follows from (iii) of Lemma 12. To prove the "if" part, let us consider a DNF-vector F satisfying (5) and define a game form $g = g(F)$ as follows. For every $i \in I$, let us assign a strategy x_i to each implicant y_i of F_i and and for each strategy profile $x = (x_1, \ldots, x_n)$ set $g(x) = \cap_{i \in I}[y_i]$. It is easily seen that the obtained game form g is rectangular and that $F = F(g(F))$. □

A DNF profile $F = (F_i \mid i \in I)$ satisfying (5) will be called *rectangular*. We will show that condition (5) can be verified in polynomial time. To do so, let us split it into two parts:

$$(a) \; \left|\bigcap_{i \in I}[y_i]\right| \neq \emptyset \quad \text{and} \quad (b) \; \left|\bigcap_{i \in I}[y_i]\right| \leq 1 \; \text{ for any implicant profile } \; y = (y_i \mid i \in I). \tag{6}$$

Clearly, part (b) means exactly that $I(v', v'') \neq \emptyset$ for any $v', v'' \in V$, or in other words, that for any two variables $v', v'' \in V$ there is an $i \in I$ such that no implicant y_i of F_i contains both v' and v''. The latter condition can be easily verified. Furthermore, if it holds then one can also verify (a) by means of the following simple counting arguments.

Let k_i denote the total number of implicants y_i of F_i, while k_i^v be the number of implicants that contain variable v; standardly $i \in I = \{1, \ldots, n\}$ and $v \in V = \{v_1, \ldots, v_m\}$.

Lemma 14. *Condition (6 a) holds if and only if*

$$\prod_{i \in I} k_i = \sum_{v \in V}\prod_{i \in I} k_i^v. \tag{7}$$

Proof. Equality (7) means that the "n-dimensional volume" of the box X and the sum of volumes of the boxes X^v for all $v \in V$ are equal. This happens if and only if X is partitioned by X^v, $v \in V$, since by (5), boxes X^v are pairwise disjoint. □

Thus, condition (5) was reduced to equality (5) and the latter can be easily verified.

Remark 6. *Let us notice, yet, that without assumption (6 b), it becomes NP-complete to verify the intersection property (6 a), already when $F_i = y_i' \vee y_i''$ for every $i \in I$.*

A polynomial reduction from SAT is simple. Given a CNF C of n variables $z_1, \ldots, z_n$ and m clauses $v_1, \ldots, v_m$, let us assign to C a (monotone) DNF profile $F(C) = F = (F_1, \ldots, F_n)$ as follows: for each $i \in I$ the DNF F_i consists of two implicants, $F_i = y'_i \vee y''_i$, whose variables $[y'_i]$ and $[y'_i]$ correspond to the sets of clauses that do not *contain z_i and $\bar{z}_i$, respectively. It is easy to verify that C is satisfiable intersection property (6 a) holds for F.*

Somewhat similarly, "tautology of a DNF" is an NP-complete problem but in case of the so-called orthogonal DNFs it becomes polynomial and can be solved by means of simple counting arguments; see, for example, [11]. In general, typically enough, verifying whether "objects cover a space" is NP-complete, in contrast to checking whether they form a partition, which is polynomial whenever it is polynomial (i) to c that any two objects are disjoint and (ii) to compute the volume of an object and the whole space then the "partition problem" is polynomial, while its "covering analog" might remain NP-hard.

5.5 Normal forms of positional tree structures

Let $\mathcal{T} = (T, u_0, \phi, \psi)$ be a tree structure, as defined in Section 3. A strategy x_i of a player $i \in I$ is a mapping that to each position $u \in U_i$ controlled by i assigns a move in this position. Then, each strategy profile $x = (x_i \mid i \in I)$ naturally defines a play that begins in u_0 and ends in a final position $v(x) \in V$. A game form $g = g(\mathcal{T}) : X \to V$ is defined by formula $g(x) = v(x)$ and called the *normal form* of the tree structure $\mathcal{T}$. Let us recall that all equivalent strategies of a player $i \in I$ are identified, in accordance with Section 5.2.

Lemma 15. *For every tree structure $\mathcal{T}$ the game form $g(\mathcal{T})$ is rectangular.*

Proof. For each final position $v \in V = L(T)$ there is a unique play $p(v)$ from u_0 to v, since T is a tree. It is easy to see that $g(x) = v$ for a strategy profile $x = (x_i \mid i \in I)$ if and only if for every $i \in I$ the strategy x_i recommends to follow p in each position that belongs to p and is controlled by i. It is also clear that $g(x) = v$ whenever $g(x') = g(x'') = v$ and x is a mixture of x' and x''. □

5.6 Positional and subpositional game forms and DNF profiles; main concepts and results

A game form g will be called *positional* if it is the normal form of a positional tree structure, that is, $g = g(\mathcal{T})$ for some $\mathcal{T}$. Respectively, g will be called *subpositional* if it is a subform of a positional game form g', that is, if $g \le g' = g'(\mathcal{T})$ for some $\mathcal{T}$. In this case g' is called a *positional extension* of g.

Every positional or subpositional game form g is rectangular, by Lemma 16.

Remark 7. *In [22], subforms $g \le g' = g'(\mathcal{T})$ were used to model the concept of* information. *"To apply a strategy x_i in $\mathcal{T}$, player i may need a resource. In particular, information may be viewed as such a resource." Due to a lack of information, X_i may be reduced to a subset $X'_i \subseteq X_i$ for some (or for all) $i \in I$; then, X is reduced to a box $X' \subseteq X$. The obtained* *subform* $g'(\mathcal{T}) : X' \to V$ *of* $g(\mathcal{T})$ *defines a tree structure* $\mathcal{T}$ *with imperfect information. This approach generalizes the classic one with* information sets *introduced by Kuhn in [34, 35].*

A DNF profile $F = (F_i \mid i \in I)$ is *rectangular* if it satisfies (5) or, equivalently, (6 a, b). By Section 5.3, F is rectangular if and only if there is a rectangular game form g such that $F = F(g)$. By Section 5.4, (b) is easy to verify and if it holds then (a) is easy to verify too.

A DNF profile F will be called *positional* or *subpositional* whenever the corresponding rectangular game form $g(F)$ is.

To every n-multigraph $\mathcal{G} = (V; E_i \mid i \in I)$ let us assign a DNF profile $F = F(\mathcal{G})$ as follows. Let $\mathcal{S}_i = \{S_i^j \mid j \in J_i\}$ denote the family of all maximal independent sets (MIS) in the graph $G = (V, E_i)$; see four examples in Figures 2-5. For every $i \in I$ and $j \in J_i$, let us treat the vertex set $S_i^j \subseteq V$ as the set of variables, $S_i^j = [y_i^j]$, of an implicant y_i^j. The obtained n DNFs $F_i = \vee_{j \in J_i} y_i^j$ determine the desired profile $F = F(\mathcal{G}) = (F_i(\mathcal{G}) \mid i \in I)$.

Positional game forms were characterized in several equivalent ways in [22], see also [23, 28, 29]. One of these characterizations can be reformulated in terms of n-graphs and DNF profiles as follows. Let us recall that a game form g is positional if and only if the corresponding DNF profile $F = F(g)$ is positional.

Theorem 6. *A DNF profile* $F = (F_i \mid i \in I)$ *is positional if and only if* $F = F(\mathcal{G})$ *for a* Π- *and* Δ*-free* n*-graph* $\mathcal{G}$*.* □

Remark 8. *Conversely, it seems that if an* n*-graph* $\mathcal{G} = (V; E_i \mid i \in I)$ *contains a* Δ *then there exists a collection of* n *vertex-sets* $S_i \subseteq V$ *such that* S_i *is a maximal stable sets of the graph* $G_i = (V, E_i)$ *for all* $i \in I$ *and* $\cap_{i \in I} S_i = \emptyset$*. This* Δ*-conjecture was firstly suggested in [22] (see page 71, right after Claim 17); see also [2, 3] for some partial results.*

The goal of this paper is to characterize the *subpositional* game forms and DNF profiles, and also to obtain algorithms enumerating all their positional rectangular extensions.

Let g' be a positional rectangular extension of a (subpositional) game form g, that is, $g \le g' = g(\mathcal{T})$. Both g and g' are rectangular, by Lemma 16. Let I, I' and V, V' denote the sets of players and outcomes in g and g', respectively. Obviously, $I \subseteq I'$ and $V \subseteq V'$.

An extension g' of g will be called *limited* if equalities hold: $I = I'$ and $V = V'$.

Lemma 16. *Each subpositional game form* g *has a limited positional extension* $g' = g(\mathcal{T})$*.*

Proof. Let $g'' = g(\mathcal{T}'')$ be a positional extension of g. Then, $I \subseteq I''$ and $V \subseteq V''$. If $I'' = I$ and $V'' = V$ then the extension is limited and there is nothing to prove. Otherwise, we will reduce $\mathcal{T}''$ to $\mathcal{T}$, so that $g \le g' = g(\mathcal{T})$ and $I' = I, V' = V$.

Indeed, coalition $K = I'' \setminus I$ has a strategy $x_K \in X''_K$ that allows to the complementary coalition $\overline{K} = I$ to realize g. Let us fix the moves corresponding to x_K in the tree structure $\mathcal{T}''$ and delete all other moves of the players from K. Then, they become "dummies" and can be eliminated, by assumption A1.

Then, in the obtained reduced tree structure T', let us delete all leaves $v \in V'' \setminus V$ and all edges incident to them to obtain the desired $\mathcal{T}$. It is easily seen that $g \le g' = g(\mathcal{T})$. □

Let us notice that, by the above procedure, we reduce in a unique way a positional extension $g'' \geq g$ to a limited one, $g' \geq g$. Vice versa, given a limited positional extension $g' = g(\mathcal{T}) \geq g$, we can extend it further by adding extra players and outcomes to $\mathcal{T}$. From now on we will restrict ourselves by the limited extensions only.

In Section 5.3, to every rectangular n-person game form g we assigned an n-multigraph $\mathcal{G}(g) = (V; E_i \mid i \in I)$ in which $I(e) = I(v', v'') \neq \emptyset$ for every edge $e = (v', v'') \in E$.

Theorem 7. *A game form g is subpositional if and only if it is rectangular and its multigraph $\mathcal{G}(g)$ contains a Π- and Δ-free edge-subgraph $\mathcal{G}'$. Moreover, these subgraphs are in one-to-one correspondence with the limited positional rectangular extensions $g' \geq g$. Their DNF profiles are given by the equality $F(g') = F(\mathcal{G}')$.*

Proof. Let $g = g(\mathcal{G})$ and $g' = g(\mathcal{G}')$. By Theorem 6, DNF profile $F(\mathcal{G}')$ is positional, since $\mathcal{G}'$ is a Π- and Δ-free n-graph. Hence, $F(\mathcal{G}') = F(g')$ for a rectangular game form g'. Thus, $F = F(g) = F(\mathcal{G})$ and $F' = F(g') = F(\mathcal{G}')$. Furthermore, n-graphs $\mathcal{G}$ and $\mathcal{G}'$ have the same set of vertices $V = V'$. Then, by Lemma 11, for each $i \in I$ the DNF $F_i(g')$ contains all implicants of $F_i(g)$ and, perhaps, some others, that is, $F_i(g') = F_i(g) \vee F_i'$. Both DNFs $F(g)$ and $F'(g')$ are irredundant, since both game forms g and g' are rectangular. Thus, g is a subform of g'. Conversely, let g' be a limited rectangular extension of g, that is, $g \leq g'$ and g' is rectangular. Then, the n-multigraphs $\mathcal{G}' = \mathcal{G}(g')$ is an edge-subgraph of $\mathcal{G} = \mathcal{G}(g)$. Furthermore, by Theorem 5, $\mathcal{G}'$ is a Π- and Δ-free n-graph. ☐

Remark 9. *Let us noticce that the size of $F_i(g')$ may be (and frequently is) exponential in the size of $F_i(g)$ and the whole multigraph $\mathcal{G}(g)$. Thus, to verify whether a game form g is subpositional is much easier than to get its positional extension when it exists.*

In Sections 2.4 it was shown that existence of Π- and Δ-free subgraphs in an n-multigraph can be verified in polynomial time. Moreover, in Section 4, we demonstrated that all these subgraphs can be generated with polynomial delay.

If $\mathcal{G} = \mathcal{G}(g)$ itself is a Π- and Δ-free n-graph then the limited rectangular extension $g' \geq g$ is unique, $g' = g(\mathcal{G})$, but still it might be non-trivial, $g' \neq g$. Indeed, for some $i \in I$ DNFs $F_i(g)$ may miss some maximal independent sets of the graphs $G_i = (V, E_i)$.

5.7 Tightness

A game form g is called *tight* if the Boolean functions defined by DNFs $F_K(g)$ and $F_{\overline{K}}(g)$ are dual, $F_K^d = F_{\overline{K}}$, for all $K \subseteq I$ and g is called *weakly tight* if $F_i^d(g) = F_{\overline{i}}(g)$, for all $i \in I$.

Theorem 8. *A rectangular game form g is tight if (and only if, of course) it is weakly tight. Furthermore, g is positional if and only if it (weakly) tight and rectangular.*

Remark 10. *This is the main result of [22] Chapter 5. In case $n = 2$ it can be viewed as a necessary and sufficient conditions for a pair of dual DNFs to be read-once [21, 22]; see also [12, 13, 17, 26, 28, 33, 42]. Let us also note that for $n = 2$ tightness is necessary and sufficient for Nash-solvability of a two-person game form [20]; see also [22] chapter 2 and*

[25, 5]. Yet, already for $n = 3$ both if and only if parts of the above statement fail. In [20] Remark 3, it was mistakenly claimed that tightness is still necessary for any n. Yet, in [25] this mistake was corrected, due to remark of V.I. Danilov. However, by Theorem 8, tight and rectangular game forms are Nash-solvable for an arbitrary n. This statement first appeared in [20] Remark 3; see also [22, 23, 28]. Then it was generalized in [1].

Also, it was mistakenly announced in [22] chapter 5 (and then repeated in [28]) that a rectangular game form g is tight whenever $F_i^d(g) = F_{\bar{i}}(g)$ for all but maybe one $i \in I$. Yet, this is an overstatement, as the following simple example shows.

Example 1. *Let $\mathcal{T} = (T, u_0, \phi, psi)$ be a positional tree structure in which T be a binary tree of depth n and n players move in order $1, \ldots, n$. Hence, each player $i \in I$ has 2^{2^i} strategies. In the normal form $g = g(\mathcal{T})$, let us eliminate one (arbitrary) strategy of the last player n and denote the obtained game form g'. Obviously, both g and g' are rectangular and g is tight, while g' is not. Moreover, g' is not weakly tight. Indeed, it is easily seen that $F_K(g) = F_K(g')$ for all coalitions $K \subseteq I$, except for the singleton $K = \{n\}$; DNF $F_n(g)$ consists of 2^{2^i} implicants one of which is missing in $F_n(g')$. Thus, duality $F_K(g')$ and $F_{\overline{K}}(g')$ holds for all pairs $(K, \overline{K})$ but one, $(i, \bar{i})$.*

Let us also remark that the unique exceptional pair must be $(i, \bar{i})$ in case of a rectangular game form g. Indeed, by Theorem 8, g is tight whenever it is weakly tight. The next example shows that an arbitrary given pair $(K_0, \overline{K_0})$ (of course, except for $(\emptyset, I)$), may be the unique exception from duality when a game form g is not rectangular.

Example 2. *Let us define a voting scheme g with n voters I and two candidates $V = \{v_1, v_2\}$. Each voter has three strategies specified by three voting cards: green, blue, and red. Let us fix an arbitrary coalition $K_0 \subseteq I$ (distinct from $\emptyset$ and I) and introduce the election rule as follows: v_1 is elected if and only if K_0, but not the whole I, vote unanimously by green cards or by blue cards. Clearly, $F_{K_0} = v_1 v_2$, since K_0 can not veto neither v_2 nor even v_1, while $F_{\overline{K}_0} = v_1$, since $\overline{K}_0$ can veto v_2 but not v_1. These two DNFs are not dual and it is also not difficult to verify that $F_K^d(g) = F_{\overline{K}}(g)$ for all other pairs $(K, \overline{K})$.*

An open question: if a similar "Boolean" example, with $|X_i| = 2$ for all $i \in I$, exists.

Remark 11. *Given a game form g and partition $P : I = K_1 \cup \ldots \cup K_\ell$, let g_P denote the corresponding ℓ-person game form. It is easy to verify that, in Example 1, game form g_P is* not *tight if and only if $K_j = \{n\}$ for some $j \in \{1, \ldots, \ell\}$.*

Let us write that $P \supseteq P'$ if P' is a subpartition of P. Obviously, $\supseteq$ is a partial order and tightness of g_P is a monotone property with respect to it. Example 1 shows that even for a rectangular game form g there may be several minimal partitions P such that g_P is tight. Generating all such minimal partitions is an interesting open problem.

5.8 Read-once DNF profiles

Given a positional tree structure $\mathcal{T} = (T, u_0, \phi, \psi)$, let us consider the corresponding normal game form $g = g(\mathcal{T})$, and 2^n DNFs $F_K, K \subseteq I$. It is not difficult to show (see [22] and also

[28]) that each of these DNFs is read-once, that is, it can be expressed via their common set of variables $V = \{v_1, \ldots, v_m\}$ by a formula in which every variable appears exactly once. Moreover, all these 2^n formulas have the same construction of clauses determined by the tree T and the operations $\vee$ and $\wedge$ appear according to the following simple rule. Given a position u in T and all its immediate successors $u_j, j \in J$, disjunction $\vee$ takes place for the corresponding clauses of F_K if and only if $i \in K$, where $i = \psi(u)$, that is, i is the player who controls position u in T. Four examples are provided by Figures 2-5.

In particular, $\vee$ in F_K is always associated with $\wedge$ in $F_{\overline{K}}$. This observation immediately implies tightness of the corresponding game form $g = g(\mathcal{T})$, since duality $F_K^d = F_{\overline{K}}$ holds for all $K \subseteq I$, by the de Morgan rules: $(F_1 \vee F_2)^d = F_1^d \wedge F_2^d$ and $(F_1 \wedge F_2)^d = F_1^d \vee F_2^d$.

For the same reason, in the DNF profile $F(\mathcal{T}) = F = (F_1, \ldots, F_n)$, for every position u, disjunction takes place for the unique $i_0 = \psi(u) \in I$, while conjunctions correspond to the $n-1$ remaining $i \in I \setminus \{i_0\}$; see examples in Figures 2-5 again. Such DNF profiles are called read-once. It is shown in [22] (and easy to see) that a DNF profile F is positional if and only if it is read-once. Thus, by previous results, the following properties of F are equivalent:

- (i) F is read-once;
- (ii) F is rectangular and the game form $g = g(F)$ is (weakly) tight;
- (iii) n-multigraph $\mathcal{G} = \mathcal{G}(F) = (V; E_i \mid i \in I)$ is a Π- and Δ-free n-graph and to each maximal independent set S_i of $G_i = (V, E_i)$ an implicant y_i of the DNF F_i is assigned.

Each of these properties can be verified in polynomial time.

References

[1] J. Abdou, Tight and Effectively Rectangular Game Forms: A Nash Solvable Class, Games and Economic Behavior 23 (1998) 111.

[2] D. Andrade, E. Boros, and V. Gurvich, On Graphs Whose Maximal Cliques and Stable Sets Intersect, Rutcor Research Report, RRR-17-2006.

[3] D. Andrade, E. Boros, and V. Gurvich, Not complementary connected and not CIS d-graphs form weakly monotone families, Discrete Math. 310 (5) (2010) 1089 – 1096.

[4] R.N. Ball, A. Pultr, and P. Vojtěchovský, Colored graphs without colorful cycles, Combinatorica 27 (4) (2007) 407–427.

[5] E. Boros, K. Elbassioni, V. Gurvich, and K. Makino, On effectivity functions of game forms, Games and Economic Behavior 68 (2) (2010) 512–531.

[6] E. Boros, and V. Gurvich, Vertex- and edge-minimal and locally minimal graphs, RUTCOR Research Report RRR-25-2007, Rutgers University, Discrete Mathematics 309 (12) (2009) 3853–3865.

[7] M. Boley, T. Horvath, A. Poigne, and S. Wrobel, Listing closed sets of strongly accessible set systems with applications to data mining, Theoretical Comp. Sci., 411:3 (2010) 691-700.

[8] A Brandstädt, V.B. Le, and J.P. Spinrad, Graph classes: A survey, SIAM, Phila., 1999.

[9] H. Buer and R. Möring, A fast algorithm for decomposition of graphs and posets. Math. Oper. Res. 3 (1983) 170-184.

[10] F.R.K. Chung and R.L. Graham, Edge-colored complete graphs with precisely colored subgraphs, Combinatorica 3 (1983) 315-324.

[11] Y. Crama and P.L. Hammer, Boolean Functions: Theory, Algorithms, and Applications, Cambridge University Press, 2011.

[12] T. Eiter, Exact transversal hypergraphs and application to Boolean μ-functions, J. Symbolic Comp. 17 (1994) 215-225.

[13] T. Eiter, Generating Boolean μ-expressions, Acta Informatica, 32 (1995) 171-187.

[14] M. Fredman and L. Khachiyan, On the complexity of dualization of monotone disjunctive normal forms. *J. Algorithms* 21 (3) (1996) 618–628.

[15] T. Gallai, Transitiv orientierbare graphen, Acta Math. Acad. Sci. Hungar., 18 (1-2) (1967), 25-66. English translation by F. Maffray and M. Preissmann, in: Perfect Graphs (Chapter 3) eds. J.L.R Alfonsin and B.A. Reed, John Wiley and Sons, 2002.

[16] G. Gal'perin and A.Tolpygo, Moscow Mathematical Olympiads, (ed. A.N. Kolmogorov), Prosvetstschenie (Enlightenment), Moscow, USSR, 1986 (in Russian).

[17] M.C. Golumbic and V. Gurvich, Read once Boolean functions, Chapter 10, 448-486, in: Boolean functions: Theory, algorithms, and applications, Y. Crama and P.L. Hammer eds., Cambridge University Press, 2011.

[18] M. C. Golumbic, H. Kaplan, R. Shamir, Graph Sandwich Problems, J. Algorithms 19(3) (1995) 449-473.

[19] V.Gurvich, To theory of multi-step games, USSR Computer Math. and Math. Physics 13 (6) (1973), 143–161.

[20] V.Gurvich, Nash-solvability of positional games in pure strategies, USSR Computer Math. and Math. Physics 15 (2) (1975) 74–87.

[21] V.Gurvich, On repetition-free Boolean functions, Uspechi mat. nauk (Russian Math. Surveys) 32 (1) (1977) 183–184 (in Russian).

[22] V. Gurvich, Applications of Boolean functions and networks in game theory (Chapter 2, Nash solvability in pure strategies; Chapter 5, Repetition-free Boolean functions and normal forms of positional games), Ph.D. thesis, Moscow Inst. of Physics and Technology, Moscow, USSR, 1978 (in Russian).

[23] V. Gurvich, On the normal form of positional games, Soviet math. dokl. 25 (3) (1982) 572–575.

[24] V. Gurvich, Some properties and applications of complete edge-chromatic graphs and hypergraphs, Soviet math. dokl. 30 (3) (1984) 803–807.

[25] V. Gurvich, Equilibrium in pure strategies, Soviet Math. Dokl. 38 (3) (1988) 597–602.

[26] V. Gurvich, Criteria for repetition-freeness of functions in the Algebra of Logic, Russian Acad. of Sci. Dokl. Math. 43 (3) (1991) 721–726.

[27] V. Gurvich, Positional game forms and edge-chromatic graphs, Russian Acad. Sci. Dokl. Math. 45 (1) (1992) 168–172.

[28] V. Gurvich, Decomposing complete edge-chromatic graphs and hypergraphs, Revisited; RUTCOR Research Report RRR 29-2006, Rutgers University; Discrete Applied Math. 157 (2009) 3069–3085.

[29] V. Gurvich On exact blockers and anti-blockers, Δ-conjecture, and related problems, Discrete Applied Math. 159 (5) (2011) 311-321.

[30] A. Gyárfás and G. Simonyi, Edge coloring complete graphs without tricolored triangles, J. Graph Theory 46 (2004), 211-216.

[31] M. Habib, Substitution de structures combinatoires, theorie et algorithmes, Thesis, Universite Pierre et Marie Curie, Paris VI, 1981.

[32] M. Habib and M. Maurer, On the X-join decomposition of undirected graphs, Discrete Applied Math. 3 (1979), 198-207.

[33] M. Karchmer, N. Linial, L. Newman, M. Saks, and A. Wigderson, A combinatorial characterization of read-once formulae, Discrete Math. 114 (1993) 275-282.

[34] H. Kuhn, Extensive games, Proc. Natl. Acad. Sci. 36 (1950) 286–295.

[35] H. Kuhn, Extensive games and the problem of information, in Contributions to the theory of games, Volume 2, Princeton (1953) 193–216.

[36] R.M. McCollel and J.P. Spinrad, Modular decomposition and transitive orientation, Discrete Math. 201 (1999) 189-241.

[37] R. Möring, Algorithmic aspects of the substitution decomposition in optimization over relations, set systems, and Boolean functions, Ann. Oper. Res. 4 (1985/6) 195-225.

[38] J. Muller and J. Spinrad, Incremental modular decomposition, J. ACM 36 (1) (1989) 1–19.

[39] D. Seinsche, On a property of the class of n-colorable graphs, *J. Combin. Theory* B 16 (1974), 191–193.

[40] D.P. Sumner, Indecomposable graphs, Thesis, Univ. of Mass. at Amherst, 1971.

[41] D.P. Sumner, Graphs indecomposable with respect to the X-join, Discrete Math. 6 (1973), 281-298.

[42] I. Newman, On read-once boolean functions, LMS Vol. Symp. on Boolean functions, Durham, 1989.