8

Introduction to Combinatorial Voting

We recall from Section 2.3 that one major direction in Computational Social Choice is to investigate the computational complexity of winner determination for some common voting rules, and then design heuristic, fixed-parameter tractable, or approximation algorithms for voting rules for which the winner is hard to compute. In those situations, the computational complexity mainly comes from the choice of voting rule.

However, in many real life group decision making problems, the computational complexity comes from the extremely large number of alternatives. In such cases it may take an unbearably long time to compute the winner even for simple voting methods such as Borda. Perhaps the most natural and prominent voting setting in real-life with an extremely large number of alternatives is *combinatorial voting*, a.k.a. *voting in multi-issue domains*. In combinatorial voting, the set of alternatives has a combinatorial (namely, multi-issue) structure. That is, there are multiple *issues* (or *attributes*, or *characteristics*) and each alternative can be uniquely characterized by a vector of the values these issues take. For example, consider a situation where the inhabitants of a county vote to determine a government plan. The plan is composed of multiple sub-plans for several interrelated issues, such as transportation, environment, and health (Brams et al., 1998). Another example is *voting by committees*, in which the voters select a subset of objects (Barbera et al., 1991), where each object can be seen as a binary issue. In such situations, a voters' preferences over one issue may well depend on the values of other issues. For example, a voter may prefer creating a natural reserve if a highway is built, but if the highway is not built, she may prefer not creating a reserve.

In the remainder of this dissertation (Chapter 8–12), we will focus on the design and analysis of voting rules when the set of alternatives has a multi-issue structure. In this chapter, we give the formal definitions and notation that will be used throughout these chapters.

Definition 8.0.1 (Combinatorial voting)**.** *Let $\mathcal{I} = \{X_1, \ldots, X_p\}$ denote a set of $p \geqslant 2$ issues, where for each $i \leqslant p$, X_i takes a value in a* local domain *D_i, where $|D_i| \geqslant 2$.*[1] Combinatorial voting *refers to the voting setting where the set of alternatives is $\mathcal{X} = D_1 \times \cdots \times D_p$. $\mathcal{X}$ is called a* multi-issue domain *or* combinatorial domain*.*

Example 8.0.2. *A group of people must make a joint decision on the menu for dinner (the caterer can only serve a single menu to everyone). The menu is composed of two issues: the main course (**M**) and the wine (**W**). There are three choices for the main course: beef (b), fish (f), or salad (s). The wine can be either red wine (r), white wine (w), or pink wine (p). The set of alternatives is a multi-issue domain: $\mathcal{X} = \{b, f, s\} \times \{r, w, p\}$.*

We call that the set of alternatives $\mathcal{C}$ studied in previous chapters constitutes an *unstructured domain*, because it does not need to have a multi-issue structure. In the above definition, we use $\mathcal{X}$ (instead of $\mathcal{C}$) to emphasize that the set of alternatives

[1] This is the standard assumption for studying voting in multi-issue domains, because otherwise either the domain can be simplified (by removing issues that can only take one value), or it has no multi-issue structure (when there is only one issue).

has a multi-issue structure. Following convention, for any $i \leqslant p$ we let $D_{-i} = D_1 \times \cdots \times D_{i-1} \times D_{i+1} \times \cdots \times D_p$.

A special case of multi-issue domains consists of the domains where all variables are binary, that is, for all $i \leqslant p$, $D_i = \{0_i, 1_i\}$. Such multi-issue domains are called *multi-binary-issue domains*. Even in multi-binary-issue domains, the number of alternatives is 2^p, which is already exponentially large. Moreover, we recall that the voting setting we defined in Chapter 2 requires a voter to submit a linear order over the set of alternatives. This requirement causes the major problem in combinatorial voting, which is that it is infeasible for a voter to give a full ranking over an exponentially large number of alternatives. Therefore, in combinatorial voting, the voters need to use another *voting language* to represent their preferences, and then we can design novel voting rules to aggregate voters' preferences represented by such a voting language.

An obvious solution is the following: we can simply ask voters to report only a (small) part of their preference relation and apply a voting rule that needs this information only. For example, we can ask the voters to report their most-preferred alternatives, and then apply the plurality rule. The voting language used in this case is the set of all alternatives instead of the set of all linear orders over the alternatives. One problem with this type of solution is the following: as soon as the number of alternative is large ($2^p \gg n$), the voters are likely to be unhappy about only expressing a small portion of their preferences. Moreover, the result of voting is likely to be completely insignificant or even catastrophic. For instance, with 5 voters and 6 binary issues, it is very likely that all 5 voters vote for different alternatives (since there are $2^6 = 64$ alternatives), and the winner under the plurality rule might be disliked by all but one voter. In fact, this phenomenon is a type of *multiple-election paradox*, which we will discuss in more detail in the next section.

Even though the above (plurality) solution itself does not work very well, it reveals the following two important high-level criteria for voting rules in multi-issue domains.

The first criterion: The quality of the voting language, which includes *compactness* and *expressiveness*.

The second criterion: The quality of the voting rule after the votes have been collected. Here the quality is measured by computational efficiency, satisfiability of axiomatic properties (see Section 2.2), resistance to multiple-election paradoxes, etc.

The compactness of a voting language can be measured by the number of bits that is used to represent a voter's preferences. For example, $\Theta(p \cdot 2^p)$ bits are necessary and sufficient to represent the voting language that consists of all linear orders over $\mathcal{X}$, because $\log((2^p)!)$ is $\Theta(p \cdot 2^p)$. Measuring the expressiveness of a voting language is more complicated. We consider the following two dimensions.

The first dimension of expressiveness: the *general usability* of the language. That is, the percentage of voters who are comfortable using this language to express their preferences. For example, if we only ask the voters to report their top-ranked alternative, no voter will feel ill at ease to do so. However, as we will see in the next section, voters are not always comfortable expressing their preferences in issue-by-issue voting and sequential voting.

The second dimension of expressiveness: the *informativeness* of the language. That is, how much of the voters' preferences are expressed by the language. For example, the top-ranked alternative only represents a tiny portion of the voter's preferences. The languages used in issue-by-issue voting rules and sequential voting rules both allow voters to express much more of their preferences.

8.1 Multiple-Election Paradoxes

Combinatorial voting has been extensively studied by economists. Most of previous work has focused on letting voters vote on the issues separately, in the following way. For each issue (simultaneously, not sequentially), each voter reports her preferences for that issue, and then, a *local rule* is used to select the winning value that the issue will take. This voting process is called *issue-by-issue* or *seat-by-seat* voting.[2] Recently, Ahn and Oliveros (2011) studied a Bayesian game of combinatorial voting, and showed the existence equilibrium under some conditions. We will not discuss the Bayesian setting in this dissertation.

Issue-by-issue voting has some drawbacks. First, a voter may feel uncomfortable expressing her preferences over one issue independently of the values that the other issues take. This means that, even though the voting language used in issue-by-issue voting can express more of a voter's preferences than the voting language that is used in plurality, it lack usability. That is, only voters whose preferences are *separable* (that is, for any issue i, regardless of the values for the other issues, the voter's preferences over issue i are always the same) are comfortable expressing their preferences in issue-by-issue voting (Kadane, 1972; Schwartz, 1977). Second, *multiple-election paradoxes* arise in issue-by-issue voting (Brams et al., 1998; Scarsini, 1998; Lacy and Niou, 2000), which we will discuss below in more detail.

Brams et al. (1998) showed that for multi-binary-issue domains, there exists a profile where the winner under issue-by-issue voting (where all local voting rules are majority rules) receives zero votes (that is, it is never ranked in the top position by any voter). Scarsini (1998) showed an even stronger paradox: there exists a profile

[2] The names "issue-by-issue" and "seat-by-seat" are a little bit misleading. It may sound like there is an ordering over issues, according to which the voters vote over issues sequentially. Even though the election can be organized in this sequential way, effectively these issue-wise elections are conducted in parallel in issue-by-issue voting, because the voters do not learn the outcomes of other issues before deciding on an issue.

where any alternative that is "close" to the winner in terms of Hamming distance under issue-by-issue voting receives zero votes. These paradoxes exist even when the voters' preferences are separable.

We are more interested in the paradoxes demonstrated by Lacy and Niou (2000) for issue-by-issue voting when the voters' preferences are non-separable. Of course in such cases the voters may feel ill at ease reporting their preferences over a single issue without knowing the values of the other issues. In Lacy and Niou (2000), it is assumed that voters vote according to their top-ranked alternative. That is, when a voter is asked to report her preferences over issue X_i, she will report the value of the X_i component in her top-ranked alternative. This behavior in some sense corresponds to very optimistic voters, and Lacy and Niou argued that when a voter does not know the votes of the other voters, she is likely to vote in this way. They illustrated the paradoxes in the following example.

Example 8.1.1. *Suppose there are three voters and the multi-issue domain is composed of three binary issues. The top-ranked alternatives of the three voters are* 110*,* 101*, and* 011*, respectively; and all voters rank* 111 *in their bottom positions. Now, by voting over each issue separately in parallel using the majority rule, the winner is* 111*, which is the least-preferred alternative for all voters.*

The above example illustrates the following three types of multiple-election paradoxes for issue-by-issue voting:

First type of paradox: the winner is a Condorcet loser (who loses to all the other alternatives in their pairwise elections).

Second type of paradox: the winner is Pareto-dominated by another alternative (that is, that alternative is preferred to the winner by all voters).

Third type of paradox: the winner is ranked in a very low position in all voters' true preferences.

8.2 CP-nets

We have seen so far that none of the approaches mentioned above works well. One common deficiency of them is that the voting languages are not expressive enough. We have seen that the voting language used by plurality has a high usability (meaning that all voters are comfortable using it), but it lacks informativeness (meaning that it only represents a tiny portion of the voters' preferences). The language used by issue-by-issue voting is much less usable, because only voters whose preferences are separable are comfortable with reporting their preferences in issue-by-issue voting, and only a tiny fraction of the linear orders are separable (Hodge, 2006). But in general it is much more informative when the voters' preferences are separable. However, none of these languages model the preferential dependence among the issues.

Fortunately, a new language for preference representation in multi-issue domains, called *conditional preference networks, or CP-nets*, that captures the dependence of voters' preferences among individual issues, was recently proposed in Artificial Intelligence (Boutilier et al., 2004). Next, we first give the formal definition of CP-nets, then discuss how to use them as the voting language for sequential voting.

The definition of a CP-net is similar to that of a Bayesian network (Pearl, 1988). We first give the formal definitions, and the present an example.

Definition 8.2.1. *A CP-net $\mathcal{N}$ over $\mathcal{X}$ consists of two parts:*

(a) A directed graph $G = (\mathcal{I}, E)$.

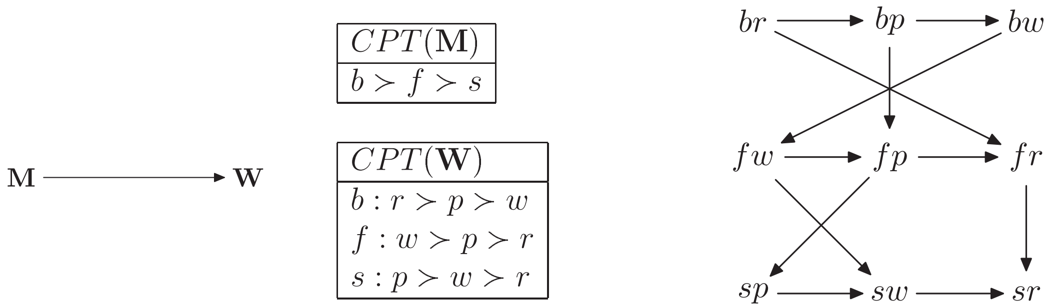
(b) A set of conditional linear preferences $>^i_{\vec{d}}$ over D_i, for each setting $\vec{d}$ of the parents of X_i in G. Let $CPT(X_i)$ be the set of the conditional preferences of a voter on D_i; this is called a conditional preference table (CPT).

A CP-net $\mathcal{N}$ captures dependencies across issues in the following sense. $\mathcal{N}$ induces a partial preorder $\succ_{\mathcal{N}}$ over the alternatives $\mathcal{X}$, representing the voter's preferences, as follows: for any $a_i, b_i \in D_i$, any setting $\vec{d}$ of the set of parents of X_i (denoted by $Par_G(X_i)$), and any setting $\vec{z}$ of $\mathcal{I} \backslash (Par_G(X_i) \cup \{X_i\})$, $(a_i, \vec{d}, \vec{z}) \succ_{\mathcal{N}} (b_i, \vec{d}, \vec{z})$ if and only if $a_i \succ^i_{\vec{d}} b_i$. In words, the preferences over issue X_i only depend on the setting of the parents of X_i (but not on any other issues). For any $1 \leqslant i \leqslant p$, CPT$(X_i)$ specifies conditional preferences over X_i. Now, if we obtain an alternative $\vec{d'}$ from $\vec{d}$ by only changing the value of the ith issue of $\vec{d}$, we can look up CPT(X_i) to conclude whether the voter prefers $\vec{d'}$ to $\vec{d}$, or vice versa. In general, however, from the CP-net, we will not always be able to conclude which of two alternatives a voter prefers, if the alternatives differ on two or more issues. This is why $\mathcal{N}$ usually induces a partial preorder rather than a linear order.

When the graph of $\mathcal{N}$ is acyclic, $\succ_{\mathcal{N}}$ is transitive and asymmetric, that is, a strict partial order (Boutilier et al., 2004). Let $\mathcal{O} = X_1 > \cdots > X_p$. We say that a CP-net $\mathcal{N}$ is *compatible* with (or, *follows*) $\mathcal{O}$, if the following is true: if X_i is a parent of X_j in the graph, then this implies that $i < j$. That is, preferences over any issue only depend on the values of earlier issues in $\mathcal{O}$. A CP-net is *separable* if there are no edges in its graph, which means that there are no preferential dependencies among issues.

Example 8.2.2. *Let $\mathcal{X}$ be the multi-issue domain defined in Example 8.0.2. We define a CP-net $\mathcal{N}$ as follows:* **M** *(the main course) is the parent of* **W** *(the wine), and the CPTs consist of the following conditional preferences: $CPT(\mathbf{M}) = \{b \succ f \succ s\}, CPT(\mathbf{W}) = \{b : r \succ p \succ w, f : w \succ p \succ r, s : p \succ w \succ r\}$, where $b : r \succ p \succ w$ is interpreted as follows: "when* **M** *is b, then, r is the most preferred value for* **W***, p is the second most preferred value, and w is the least preferred value." $\mathcal{N}$ and its induced partial order $\succ_{\mathcal{N}}$ are illustrated in Figure 8.1. $\mathcal{N}$ is compatible with $\mathbf{M} > \mathbf{W}$.*

$\mathcal{N}$ is not separable.



(a) A CP-net $\mathcal{N}$. (b) The partial order induced by $\mathcal{N}$.

FIGURE 8.1: A CP-net $\mathcal{N}$ and its induced partial order.

When all issues are binary, a CP-net $\mathcal{N}$ can be visualized as a hypercube with directed edges in p-dimensional space (Domshlak and Brafman, 2002), in the following way. Each vertex is an alternative, each dimension corresponds to an issue, and any two adjacent vertices differ in only one component (issue). That is, for any $i \leqslant p$ and any $\vec{d}_{-i} \in D_{-i}$, there is a directed edge connecting $(0_i, \vec{d}_{-i})$ and $(1_i, \vec{d}_{-i})$, and the direction of the edge is from $(0_i, \vec{d}_{-i})$ to $(1_i, \vec{d}_{-i})$ if and only if $(0_i, \vec{d}_{-i}) \succ_{\mathcal{N}} (1_i, \vec{d}_{-i})$.

Example 8.2.3. *Let $p = 3$ and let $\mathcal{N}$ be a CP-net defined as follows: the directed graph of $\mathcal{N}$ has an edge from X_1 to X_2 and an edge from X_2 to X_3; the CPTs are $CPT(X_1) = \{0_1 \succ 1_1\}$, $CPT(X_2) = \{0_1 : 0_2 \succ 1_2, 1_1 : 1_2 \succ 0_2\}$, $CPT(X_3) = \{0_2 : 0_3 \succ 1_3, 1_2 : 1_3 \succ 0_3\}$. $\mathcal{N}$ is illustrated as a hypercube in Figure 8.2 (for simplicity, in the figure, a vertex abc represents the alternative $a_1 b_2 c_3$, for example, the vertex 000 represents the alternative $0_1 0_2 0_3$).*

A linear order V over $\mathcal{X}$ *extends* a CP-net $\mathcal{N}$, denoted by $V \sim \mathcal{N}$, if it extends the partial order that $\mathcal{N}$ induces. (This is merely saying that V is consistent with the preferences implied by the CP-net $\mathcal{N}$.) V is *separable* if it extends a separable

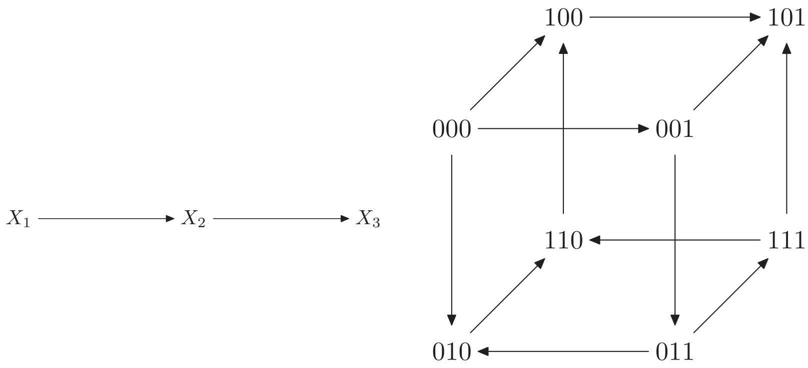


FIGURE 8.2: The hypercube representation of the CP-net in Example 8.2.3.

CP-net. Given an ordering $\mathcal{O}$ over issues, V is $\mathcal{O}$*-legal* if it extends a CP-net that is compatible with $\mathcal{O}$. The set of all $\mathcal{O}$-legal linear orders is denoted by $Legal(\mathcal{O})$.

To present our results, we will frequently use notations that represent the projection of a vote/CP-net/profile to an issue X_i (that is, the voter's local preferences over X_i), given the setting of all parents of X_i. These notations are defined as follows. For any issue X_i, any setting $\vec{d}$ of $Par_G(X_i)$, and any linear order V that extends $\mathcal{N}$, we let $V|_{X_i:\vec{d}}$ and $\mathcal{N}|_{X_i:\vec{d}}$ denote the the projection of V (or, equivalently $\mathcal{N}$) to X_i, given $\vec{d}$. That is, each of these notations evaluates to the linear order $\succ^i_{\vec{d}}$ in the CPT associated with X_i. For example, let $\mathcal{N}$ be the CP-net defined in Example 8.2.2. $\mathcal{N}|_{\mathbf{W}:b} = r \succ p \succ w$. For any $\mathcal{O}$-legal profile P, $P|_{X_i:\vec{d}}$ is the profile over D_i that is composed of the projections of each vote in P on X_i, given $\vec{d}$. That is, $P|_{X_i:\vec{d}} = (V_1|_{X_i:\vec{d}}, \ldots, V_n|_{X_i:\vec{d}}) = (\mathcal{N}_1|_{X_i:\vec{d}}, \ldots, \mathcal{N}_n|_{X_i:\vec{d}})$, where $P = (V_1, \ldots, V_n)$, and for any $1 \leqslant i \leqslant p$, V_i extends $\mathcal{N}_i$.

Let $\mathcal{O} = X_1 > \cdots > X_p$. The *lexicographic extension* of an $\mathcal{O}$-compatible CP-net $\mathcal{N}$ w.r.t. $\mathcal{O}$, denoted by $Lex_{\mathcal{O}}(\mathcal{N})$, is an $\mathcal{O}$-legal linear order V over $\mathcal{X}$ such that for any $1 \leqslant i \leqslant p$, any $\vec{d_i} \in D_1 \times \cdots \times D_{i-1}$, any $a_i, b_i \in D_i$, and any $\vec{y}, \vec{z} \in D_{i+1} \times \cdots \times D_p$, if $a_i \succ_{\mathcal{N}|_{X_i:\vec{d_i}}} b_i$, then $(\vec{d_i}, a_i, \vec{y}) \succ_V (\vec{d_i}, b_i, \vec{z})$. Intuitively, in the lexicographic extension of $\mathcal{N}$, X_1 is the most important issue, X_2 is the next-most important issue, and so on; a desirable change to an earlier issue always outweighs any changes to later issues. We note that the lexicographic extension of any CP-net is unique w.r.t. the order $\mathcal{O}$. Again, the subscript "$\mathcal{O}$" is sometimes omitted when there is no risk of confusion. We say that $V \in L(\mathcal{X})$ is *lexicographic* if it is the lexicographic extension of a CP-net $\mathcal{N}$. For example, let $\mathcal{N}$ be the CP-net defined in Example 8.2.2. We have $Lex(\mathcal{N}) = br \succ bp \succ bw \succ fw \succ fp \succ fr \succ sp \succ sw \succ sr$. A profile P is $\mathcal{O}$-legal/separable/lexicographic, if each of its votes is in $Legal(\mathcal{O})$/ is separable/ is lexicographic.

8.3 Sequential Voting

One natural approach to combinatorial voting is sequential voting. Let $\mathcal{O}$ denote be an ordering over the issues. W.l.o.g. $\mathcal{O} = X_1 > X_2 > \cdots > X_p$. Sequential voting selects the winner in p rounds. In round i, the voters report their preferences over the ith issue in $\mathcal{O}$, based on which the winning value is selected by applying a local voting rule, and this winning value is then announced to all the voters. The idea of sequential voting is not new. For example, Lacy and Niou (2000) suggested to use sequential voting to circumvent multiple-election paradoxes. But, again, in the sequential voting process they proposed, voters are sometimes ill at ease reporting their preferences over issues, and the voters are still assumed to behave optimistically.[3] Moreover, Lacy and Niou argued that the sequential voting process "*takes too long*," because the voters must wait for the results of previous issues to be announced before moving to the subsequent issues. They argued that "*the cost to voters of going to the polls and the cost to governments of keeping polls open for several days will likely prevent the use of sequential voting schemes*" (Lacy and Niou, 2000).

In fact, the voters do not need to go to voting booths multiple times. It suffices

[3] Lacy and Niou (2000) also suggested to let voters vote strategically and sequentially, and showed that the outcome will always be the Condorcet winner whenever one exists. This is the strategic sequential voting procedure that will be discussed in Chapter 11.

for them to report in one shot all their local preferences over single issues, given all (relevant) valuations of the previous issues. That is, to apply sequential voting w.r.t. the order $\mathcal{O}$ over issues, it suffices for the voters to use an $\mathcal{O}$-compatible CP-net to represent their preferences. Of course the voters need to report more of their preferences, and some of them are not used for the voting rule to decide the winner. This is not a big problem as long as the language is compact (as we will see later in this section). Similarly to the situation in issue-by-issue voting (where only the voters with separable preferences are comfortable with reporting their preferences), in sequential voting we have a similar criterion: if a voter's preferences are $\mathcal{O}$-legal, then she is comfortable with reporting their preferences; otherwise she is not comfortable with reporting her preferences.

The ground-breaking systematic method for analyzing sequential voting was proposed by Lang (2007), who focused on the profiles where voters are comfortable reporting their preferences (that is, $\mathcal{O}$-legal profiles), and defined sequential voting rules on top of these profiles.

Definition 8.3.1. *(Lang, 2007) Given a vector of* local rules $(r_1, \ldots, r_p)$*, where for each* $1 \leqslant i \leqslant p$*,* r_i *is a voting rule on* D_i*, the* sequential composition *of* $r_1, \ldots, r_p$ *w.r.t.* $\mathcal{O}$*, denoted by* $Seq_{\mathcal{O}}(r_1, \ldots, r_p)$*, is defined for all* $\mathcal{O}$*-legal profiles as follows:* $Seq_{\mathcal{O}}(r_1, \ldots, r_p)(P) = (d_1, \ldots, d_p) \in \mathcal{X}$*, so that for any* $1 \leqslant i \leqslant p$*,* $d_i = r_i(P|_{X_i:d_1 \cdots d_{i-1}})$*.*

The *sequential composition* of *local correspondences* $r_1^c, \ldots, r_p^c$, denoted by $Seq_{\mathcal{O}}(r_1^c, \ldots, r_p^c)$, is defined in a similar way: for any $\mathcal{O}$-legal profile P, $\vec{d} \in Seq_{\mathcal{O}}(r_1^c, \ldots, r_p^c)(P)$ if and only if for each $i \leqslant p$, we have that $d_i \in r_i^c(P|_{X_i:d_1 \ldots d_{i-1}})$.

The subscript "$\mathcal{O}$" in $Seq_{\mathcal{O}}$ is sometimes omitted when there is no risk of confusion. We note that when a voter's preferences are $\mathcal{O}$-legal, she only needs to submit an $\mathcal{O}$-compatible CP-net instead of reporting the entire $\mathcal{O}$-legal linear order. Hence, the voting language used by a sequential rule is essentially the set of all $\mathcal{O}$-compatible CP-nets. Similarly, the voting language used in issue-by-issue voting is essentially the set of all separable CP-nets (where there are no edges in the graph). We note that if the voters' profile is separable, then sequential voting rules become issue-by-issue voting rules. In that sense, sequential voting rules are extensions of issue-by-issue voting rules.

To examine the compactness of the set of all $\mathcal{O}$-compatible CP-nets as a voting language, let us calculate the size of an $\mathcal{O}$-compatible CP-net (which is the sum of the sizes of all CPTs). It is easy to see that the size of a CP-net largely depends on how many parents each issue has in the graph. In fact, the size of a CP-net is

$$\sum_{i=1}^{p} \prod_{X_j \in Par_G(X_i)} |D_j| \log(|D_i|!)$$

Therefore, if both the number of members in each local domain and the number of parents for each issue are small, then the size of the CP-net is polynomial in the number of issues (for comparison, we recall that in multi-issue domains we need $\Theta(p \cdot 2^p)$ bits to represent a linear order, which is exponential in the number of issues); on the other hand, in the worst case the size of an $\mathcal{O}$-compatible CP-net is exponentially large in the number of issues. However, in practice it is reasonable to expect that all local domains are small, and the voters' preferences over each issue only depends on a few other issues. Hence, we can expect in practice that $\mathcal{O}$-compatible CP-nets are a compact language. Obviously $\mathcal{O}$-compatible CP-nets, as a voting language, are more expressive than the language used by issue-by-issue voting (that is, separable CP-nets), simply because separable CP-nets are special cases of $\mathcal{O}$-compatible CP-nets. In fact, it has been shown that the ratio between the number of $\mathcal{O}$-legal linear orders and the number of separable linear orders is

$\Omega(\frac{2^p}{\sqrt{\pi p}})$ (Lang and Xia, 2009), which in some sense shows quantitatively how much more usable $\mathcal{O}$-compatible CP-nets are, compared to separable CP-nets. Table 8.1 provides a comparison of plurality, common voting rules that require voters to report linear orders (e.g., Borda), issue-by-issue voting rules, sequential voting rules, the framework introduced in Chapter 9, and the MLE approach taken in Chapter 10, in terms of the following three aspects: (1) computational efficiency of computing the winner, (2) compactness of the voting language, and (3) expressiveness of the voting language, which includes general usability and informativeness.

Table 8.1: Comparing voting rules and languages for combinatorial voting.

Voting method	Computational efficiency	Compactness	Expressiveness	
			General usability	Informativeness
Plurality	High	High	High	Low
Borda, etc.	Low	Low	High	High
Issue-by-issue	High	High	Low	Medium
Sequential voting	High	Usually high	Medium	Medium
H-composition in Chapter 9	Low–High (depends on the voters' common preference structure)	Usually high	High	Medium
MLE approach in Chapter 10	Low–High (depends on the probabilistic model)	Usually high	High	Medium

For (truthful) sequential voting, multiple-election paradoxes are alleviated (Lacy and Niou, 2000; Lang and Xia, 2009), though they return when voters vote strategically, as we will see in Chapter 11. One natural question to ask is whether sequential voting rules satisfy some other desired axiomatic properties for voting rules (see Section 2.2). Not surprisingly, the answer depends on whether the local voting rules satisfy these axiomatic properties. Lang and Xia (2009) asked the following two questions for any axiomatic property Y.

1. If the sequential voting rule satisfies Y, is it true that all its local voting rules also satisfy Y? This corresponds to the "Global to local" column in Table 8.2.

2. If all local voting rules satisfy Y, is it true that their sequential composition also satisfies Y? This corresponds to the "Local to global" column in Table 8.2.

The answers for some of the axiomatic properties described in Section 2.2 are summarized in Table 8.2.[4]

Table 8.2: Local vs. global for sequential rules (Lang and Xia, 2009).

Criteria	Global to local	Local to global
Anonymity	Y	Y
Neutrality	Y	N
Consistency	Y	Y
Participation	Y	N
Pareto efficiency	Y	N
(Strong) monotonicity	Y	Y

For neutrality and Pareto efficiency, Xia and Lang (2009) showed that the existence of voting correspondences that satisfy neutrality (respectively, Pareto efficiency) can be characterized by the structure of the multi-issue domain: if the multi-issue domain is composed of two binary variables, then there exists a voting correspondence that satisfies neutrality (respectively, Pareto efficiency); otherwise no voting correspondence satisfies neutrality (respectively, Pareto efficiency).[5]

Nevertheless, we may still argue that in order for voters to feel comfortable expressing their preferences, sequential voting is quite restrictive at two levels: first, at the individual voters' level, sequential voting requires that a voter's preferences must be represented by an acyclic CP-net. Second, at the profile level, it requires

[4] Since sequential voting rules are defined for $\mathcal{O}$-legal profiles, the definitions of neutrality, Pareto efficiency, and monotonicity are slightly different. See Lang and Xia (2009).

[5] Again, here the definitions of neutrality and efficiency are slightly different from the definitions in Section 2.2.

that all voters' preferences are compatible with the same ordering $\mathcal{O}$. To overcome these restrictions, we need to consider even more expressive voting languages. One option is the set of all (possibly cyclic) CP-nets. Obviously it is more expressive, because it is a superset of the set of all acyclic CP-nets. Chapters 9 and 10 aim at designing new voting rules for combinatorial voting where the voters use (possibly cyclic) CP-nets to represent their preferences. In Chapter 9, we will further show how much more general (possibly cyclic) CP-nets are, by showing the ratio between the number of $\mathcal{O}$-legal votes and the number of all linear orders over $\mathcal{X}$ (note that any voter should be comfortable with using a possibly cyclic CP-net to represent her preferences, in the sense that for any linear order, a possibly cyclic CP-net can be constructed such that the linear order extends this CP-net). Then, we propose an extension of sequential voting rules to aggregate (possibly cyclic) CP-nets, which we call *hypercubewise composition (H-composition)*. We will analyze its normative and computational aspects. This framework was further studied by Li et al. (2011) and Conitzer et al. (2011b). In Chapter 10, we extend Condorcet's MLE model to combinatorial voting.

Chapters 11 and 12 investigate game-theoretic aspects of combinatorial voting. In Chapter 11 we study the sequential voting game mentioned by Lacy and Niou (2000), that is, the game where voters cast votes strategically on one issue after another, following some ordering over the issues. We call this type of voting games the *strategic sequential voting procedure (SSP)*. Lacy and Niou (2000) proved that strategic sequential voting[6] always selects the Condorcet winner whenever one exists, but they did not examine whether there are any multiple-election paradoxes for SSP. In Chapter 11 we show that all three types of multiple-election paradoxes still arise in strategic sequential voting. Moreover, changing the ordering of the issues according to which the voters vote on them cannot avoid at least the first and the

[6] They called it *sophisticated sequential voting*, following the convention of Farquharson (1969).

third paradoxes. Then, in Chapter 12, we will see how to restrict voters' preferences over multi-issue domains to obtain strategy-proof voting rules.

At the end of this chapter, let me briefly mention some other work in preference aggregation over multi-issue domains. Rossi et al. (2004) studied aggregating voters' preferences represented by *partial CP-nets*, which allows voters to be "indifferent" with between the values of some issues. Gonzales et al. (2008) studied aggregating preferences represented by another compact language called *GAI-networks*. Xia et al. (2007a) slightly extended sequential voting rules by removing the constraint that the order $\mathcal{O}$ is fixed before the voting process. However, the above two levels of restrictions for sequential voting rules still exist. Recently, Conitzer et al. (2009a) studied the agenda control problem in sequential voting—that is, the chair gets to choose the over in which the issues are voted on, and investigated its computational complexity.

8.4 Summary

In this chapter, we introduced the notation used in this dissertation for combinatorial voting, multiple-election paradoxes, CP-nets, sequential voting rules, and important criteria for designing new voting rules in combinatorial domains. We also evaluated voting rules proposed in previous work by these criteria, and the result is summarized in Table 8.1. We observed that all previous approach either used voting languages that lack expressivity, or is computationally intractable.