Approximation Ratio for Preference Aggregation Using Tree CP-Nets

Abu Mohammad Hammad Ali
University of Regina
Canada
hammad@uregina.ca

Daniel Ogundare
University of Regina
Canada
dogun10517@gmail.com

Boting Yang
University of Regina
Canada
boting.yang@uregina.ca

Sandra Zilles
University of Regina and Amii
Canada
sandra.zilles@uregina.ca

ABSTRACT

Aggregating preferences of multiple entities is a problem that has been studied in various models of preference representation, including Conditional Preference Networks (CP-nets). Since optimal aggregation of CP-nets (for a specific natural choice of objective function) is known to require exponential time, efficient approximation algorithms have been proposed in the literature, yet with very limited results on the corresponding approximation ratio. In this paper, we show that a very simple and efficient method yields a $\frac{4}{3}$-approximation for aggregating CP-nets from a proper superset of the set of all tree CP-nets—a well-studied class of CP-nets of relevance to many applications.

KEYWORDS

Conditional Preference Networks; Preference Aggregation

ACM Reference Format:
Abu Mohammad Hammad Ali, Daniel Ogundare, Boting Yang, and Sandra Zilles. 2025. Approximation Ratio for Preference Aggregation Using Tree CP-Nets. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Preference aggregation is the task of finding the best collective preference model from individual preference models, or, in some cases, simply determining the best collective outcome. The definition of "best" typically depends on the specific application. Preference aggregation is crucial in areas such as recommender systems, social choice theory, and multi-agent systems [19]. Individual preferences can be represented using various models, from complete/partial orderings over outcomes [11, 12, 19] to more concise hypercube-based models [14, 32]. In this paper, we study preference aggregation using Conditional Preference Networks (CP-nets) [10], a graphical representation model which uses conditional dependencies and the Ceteris Paribus interpretation to compactly encode preferences in combinatorial domains. With CP-net semantics, outcomes are represented as vectors of attribute-value pairs. Through conditional dependencies, a CP-net can represent preferences using compact statements that apply to a large number of outcome pairs. For example, consider a set of attributes $V = \{V_1, V_2, V_3, V_4\}$ and the preference statement "Given V_1 is assigned 0, it is preferred that V_4 be assigned 0 over being assigned 1". Given this information, for any pair of outcomes with V_1 assigned 0, one would prefer the outcome with V_4 also assigned 0, all else being equal.

The setting that we target is one in which the preferences of an entity (i.e., of an individual user or of a group of users) are represented in the form of a CP-net. Assuming multiple such CP-nets are given as inputs (representing preferences of multiple entities), the goal is to aggregate these CP-nets into a single output CP-net that best represents a form of consensus among the entities' preferences.

To measure the dissensus of the output CP-net with the given input CP-nets, we adopt an objective function from the literature [5]. Since this dissensus function cannot in general be minimized in polynomial time [5], efficient approximation algorithms for minimizing dissensus were introduced, and were shown to yield an approximation ratio of $\frac{4}{3}$ for very special classes of input instances [5, 6]. For general input instances, a standard argument shows that an approximation ratio of at most 2 can always be achieved by using as aggregate CP-net any of the *input CP-nets* whose dissensus value is smallest [6]. A core open question is whether the approximation ratio of $\frac{4}{3}$, which was obtained for special input instances, can also be obtained for *every* input instance, by an efficient algorithm.

We make notable progress towards answering this question. Our main result states that a very simple and efficient algorithm (proposed by Ali et al. [6]) obtains an approximation ratio at most $\frac{4}{3}$ for input instances consisting of CP-nets whose graphs have maximum in-degree 1. This includes the class of all tree CP-nets (CP-nets whose underlying graphical structure is a tree).

Tree CP-nets represent preference relations in which the preference for one attribute depends on the value of at most one other attribute. It has been argued that tree CP-nets are highly relevant for applications, since most user preferences are not conditioned on the value of a large number of attributes [22]. This makes tree CP-nets expressive enough for many practical purposes, while being easier to handle algorithmically in many contexts. Hence, prior research produced results focused specifically on tree CP-nets in various contexts, such as, for instance, learning of CP-nets [2–4, 7].

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

Our main result substantially generalizes the result by Ali et al. [6], which proved an approximation ratio of $\frac{4}{3}$ to be efficiently obtainable for a small and not very expressive subclass $\mathcal{I}$ of input instances. In addition, we show that the approximation ratio for this class $\mathcal{I}$ (using the same efficient method proposed by Ali et al.) converges to 1, as the number of attributes of the underlying combinatorial domain increases.

We further provide new results on the structural properties of input instances for which Ali et al.'s method obtains the worst approximation ratios. The insights obtained from these results and their corresponding proof techniques may be of value for further research into approximation algorithms for preference aggregation.

2 RELATED WORK

Early studies on preference aggregation focus on the case where individual preferences are given as a list of outcome pairs, and aim to find an ordering minimizing some objective function [8, 18, 19, 31]. This is not feasible in multi-attribute combinatorial domains, where compact representation models such as dependency graphs [1], CP-nets [10], LP-trees [9], utility-based models or logical representations [23] are used instead. Our focus is on CP-nets.

One of the first CP-net based preference aggregation approaches introduces mCP-nets [30] which do not construct a consensus model but rather work with a set of partial CP-nets, all of which must be preserved as given. Reasoning tasks are then carried out by running queries on each CP-net in the set and applying some voting rule to choose the collective response. The complexity of reasoning depends on the voting rule chosen [27–29]. However, each input CP-net must be stored and reasoned with for each query.

In contrast, Lang [24] also works with CP-nets as input, applying some voting rule over the attributes sequentially and in topological order. This ensures that votes on an attribute are taken only after the best consensus outcome has been chosen for all its parent attributes, and their goal is to find a consensus outcome given individual CP-nets. Xia et al. [32, 33] and Lang and Xia [25] showed that this approach can lead to paradoxes, which can be avoided when all input CP-nets have the same dependency graph. For more general CP-nets, hypercube-based approaches are proposed, in order to find the set of non-dominated consensus outcomes [13, 14, 26, 32].

Several past studies aimed at methods that output a preference representation model [15–17]. These aggregate a set of CP-nets, building a single model represented by a Probabilistic CP-net (PCP-net)—an extension of CP-nets allowing probabilistic reasoning.

In contrast, we focus on finding the best aggregate CP-net given a set of input CP-nets. As argued by Ali et al. [5, 6], this saves us from querying each input CP-net to answer a query, and also allows us to apply existing CP-net preference reasoning algorithms. Since CP-net aggregation with minimal dissensus cannot in general be done in polynomial time [5], a recent focus has been on approximation algorithms for the same problem [6].

For preference aggregation with partial orderings, as well as for judgment aggregation, optimization has been proven intractable [19, 21]. Even in the domain of approximate aggregation, it is not possible to guarantee better than a 2-approximation in general [20]. However, Ali et al. [6] showed that for CP-net based aggregation and some specific classes of input profiles, approximation ratios strictly better than 2 are possible. Our paper is the first to show that CP-net aggregation within a factor of $\frac{4}{3}$ can be done efficiently for tree CP-nets (even for a more general class of CP-nets).

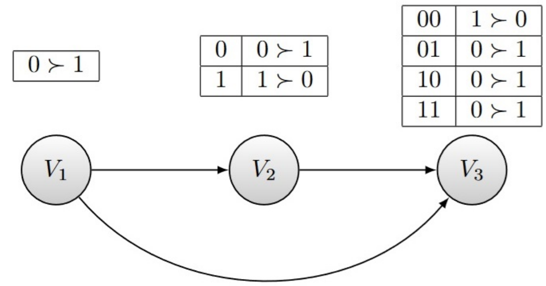

Figure 1: A CP-net N over $\{V_1, V_2, V_3\}$. Here $\mathrm{Pa}(N, V_1) = \{\}$, $\mathrm{Pa}(N, V_2) = \{V_1\}$, and $\mathrm{Pa}(N, V_3) = \{V_1, V_2\}$. Each node V_i is annotated with its CPT, which contains $2^{|\mathrm{Pa}(N,V_i)|}$ rules.

3 PRELIMINARIES AND NOTATION

A CP-net, as introduced in [10], is a directed graph with a set of vertices $V = \{V_1, \ldots, V_n\}$ representing n attributes, each with domain $\{0, 1\}$. A preference statement for V_i is a total ordering of $\{0, 1\}$ associated with V_i. A directed edge (V_i, V_j) indicates that the preferences for V_j are conditioned on the value assigned to V_i, and we say V_i is a parent of V_j. We denote the set of all parents of V_i in a CP-net N by $\mathrm{Pa}(N, V_i)$.

For each V_i and for each possible value assignment to $\mathrm{Pa}(N, V_i)$, the preference ordering of $\{0, 1\}$ for V_i is listed in a Conditional Preference Table (CPT), denoted $\mathrm{CPT}(N, V_i)$, with each conditional preference ordering referred to as a CPT rule. For example, if $\mathrm{Pa}(N, V_i) = \{V_1\}$, the CPT for V_i may have two rules, one indicating the preference ordering of $\{0, 1\}$ for V_i when $V_1 = 0$, and one indicating the preference when $V_1 = 1$. See Figure 1 for an example.

A CPT that contains preference orderings for all possible value assignments to the parent set is called a complete CPT. In this paper, we limit ourselves to complete CPTs. In particular, since attributes are binary, if $|\mathrm{Pa}(N, V_i)| = k$, then $\mathrm{CPT}(N, V_i)$ has 2^k rules. We use the phrase *1-bounded CPT* to refer to a CPT whose parent set has size at most 1. A 1-bounded CP-net is a CP-net N with only 1-bounded CPTs, i.e., in which $|\mathrm{Pa}(N, V_i)| \leq 1$ for all $V_i \in V$. Note that the class of all 1-bounded CP-nets contains not only all tree-structured CP-nets, but also some cyclic CP-nets.

An instantiation of some $V' \subseteq V$ is an assignment of values to each attribute in V' and $\mathrm{Inst}(V')$ denotes the set of all instantiations of V'. Assuming a fixed order over V, each element $\gamma \in \mathrm{Inst}(V')$ is simply a boolean vector with $|V'|$ components. Elements of $\mathrm{Inst}(V)$ are called outcomes, and an outcome pair (o, o') is called a *swap* over V_i if o, o' differ only in their value for V_i. Any swap of some attribute V_i can be ordered by reference to a single CPT rule in $\mathrm{CPT}(N, V_i)$ [10], corresponding to the appropriate instantiation of $\mathrm{Pa}(N, V_i)$. A complete CP-net N can use these CPT rules to order all swaps, resulting in a preference order over swaps which we denote by $\succ_N$. For the CP-net N in Figure 1, the swap $(100, 101)$ over V_3 is ordered via the CPT for V_3; the instantiation 10 of the parent pair (V_1, V_2) corresponds to the third row of this CPT, resulting in $100 \succ_N 101$. Taking the transitive closure $\succ_{[N]}$ of $\succ_N$, one can then order outcome pairs that are not swaps; for example, for the CP-net in Figure 1, we obtain $000 \succ_{[N]} 011$, since $000 \succ_N 010 \succ_N 011$.

In general, a CP-net N with a cyclic graph may not induce a consistent preference ordering, i.e., there may be outcome pairs (o, o') for which both $o \succ_{[N]} o'$ and $o' \succ_{[N]} o$ hold [10]. Moreover, for both cyclic and acyclic (complete) CP-nets, there may be outcome pairs (o, o') for which neither $o \succ_{[N]} o'$ nor $o' \succ_{[N]} o$ holds [10]. However, if (o, o') is a swap, then exactly one of $o \succ_N o'$ and $o' \succ_N o$ is satisfied.

Ali et al. [5, 6] propose a distance function $f_{\text{swap}}(N, N')$, called swap disagreement, defined as the number of swaps (o, o') on which $\succ_N$ and $\succ_{N'}$ disagree, i.e., for which $o \succ_N o'$, but $o' \succ_{N'} o$.

Given a multiset $\{N_1, \ldots, N_z\}$ of CP-nets and an aggregate CP-net N, the term $f_{\text{swap}}(N, \{N_1, \ldots, N_z\})$ denotes the cumulative swap disagreement for N with all given N_i, that is,

$$f_{\text{swap}}(N, \{N_1, \ldots, N_z\}) = \sum_{1 \le i \le z} f_{\text{swap}}(N, N_i) .$$

If N and N' are defined over attributes $V_1, \ldots, V_n$, then we can separately count the swaps over each attribute V_j for which $\succ_N$ and $\succ_{N'}$ disagree: $f_{\text{swap},j}(N, N') = |\{(o, o') \mid (o, o') \text{ is a swap over } V_j, \text{ and } o \succ_N o',\ o' \succ_{N'} o\}|$. Note that

$$f_{\text{swap}}(N, N') = \sum_{1 \le j \le n} f_{\text{swap},j}(N, N') .$$

We obtain a similar equation for $f_{\text{swap},j}$ as for f_{swap}:

$$f_{\text{swap},j}(N, \{N_1, \ldots, N_z\}) = \sum_{1 \le i \le z} f_{\text{swap},j}(N, N_i) .$$

The algorithmic problem in the focus of our study is the following: given a multiset $\{N_1, \ldots, N_z\}$ of CP-nets over binary attributes $V_1, \ldots V_n$, compute a CP-net N^* over $V_1, \ldots V_n$, such that N^* minimizes the cumulative swap disagreement $f_{\text{swap}}(N^*, \{N_1, \ldots, N_z\})$. Since aggregating CP-nets can be done separately for each CPT, this problem reduces to the problem of, given *CPTs* $\{N_1, \ldots, N_z\}$ for attribute V_n (with parents in $\{V_1, \ldots, V_{n-1}\}$), computing a *CPT* N^* for V_n that minimizes $f_{\text{swap},n}(N^*, \{N_1, \ldots, N_z\})$. The cost of such a minimizer, i.e., of an optimal aggregate CPT, is denoted by

$$f^{\text{opt}}_{\text{swap},n}(\{N_1, \ldots, N_z\}) = \min_N f_{\text{swap},n}(N, \{N_1, \ldots, N_z\}) .$$

It was shown by Ali et al. [5] that, for some multisets of given CPTs, the size of any optimal solution to the above problem is exponential in the sum of the sizes of the given CPTs. In particular, no polynomial-time algorithm solving this problem exists, which motivates the study of approximation algorithms. Ali et al. [6] studied two approximation algorithms. The simpler one computes $f_{\text{swap},n}(N_i, \{N_1, \ldots, N_z\})$ for all $i \in \{1, \ldots, z\}$ and outputs an N_i minimizing this value. In other words, it outputs one of its inputs whose value in the objective functions is smallest; we refer to this method as *picking the best input CPT*, and we call the corresponding output N_i the *best input CPT*. The second algorithm computes, for each $i \in \{1, \ldots, z\}$, a CPT N'_i that has the smallest value of $f_{\text{swap},n}(N'_i, \{N_1, \ldots, N_z\})$ among all CPTs with the same parent set as N_i. Both algorithms are guaranteed to provide 2-approximations, but it was shown that the approximation ratio of the best input CPT in general exceeds $2 - \epsilon$ for every $\epsilon > 0$.

One of the main results by Ali et al. [6] states that, when all input CPTs have pairwise disjoint parent sets and satisfy a certain symmetry condition, then the best input CPT achieves an approximation ratio of $\frac{4}{3}$. Below we will relax the premises of this result by (i) allowing for the empty parent set, which does not satisfy Ali et al.'s symmetry condition, and by (ii) allowing pairs of parent sets to be either identical or pairwise disjoint. Our main result (stated in Theorem 5.1 below) is that for 1-bounded CP-nets, the disjointness requirement on the parent sets can be dropped. That means, for *any* set $\{N_1, \ldots, N_z\}$ of CPTs of 1-bounded CP-nets, we get

$$\min_{1 \le i \le z} f_{\text{swap},n}(N_i, \{N_1, \ldots, N_z\}) \le \frac{4}{3} f^{\text{opt}}_{\text{swap},n}(\{N_1, \ldots, N_z\}) .$$

In other words, picking the best input CPT is an efficient method achieving a $\frac{4}{3}$-approximation for any set of 1-bounded CP-nets.

4 COMBINATORIAL APPROACH

This section gives details of the notation and approach we use to state most of our results. We begin by proving a lemma that allows us to make some simplifying assumptions about the problem instances with which we have to deal.

This lemma says that two 1-bounded CPTs with identical parent sets but exactly opposite preference ordering can be removed from a set of input tree CPTs without decreasing the approximation ratio obtained by picking the best input CPT.

Lemma 4.1. *Let $\mathcal{N} = \{N_1, \ldots, N_z\}$ be any multiset of 1-bounded CPTs for attribute V_n. Suppose there exist $r, s \in \{1, \ldots, z\}$ with $r \neq s$ such that $\mathrm{Pa}(N_r, V_n) = \mathrm{Pa}(N_s, V_n)$ but, for all swaps (o, o') over V_n, we have $o \succ_{N_r} o'$ iff $o' \succ_{N_s} o$. Let $N_i \in \mathcal{N}$ such that $f_{\text{swap},n}(N_i, \mathcal{N}) = \min_{N \in \mathcal{N}} f_{\text{swap},n}(N, \mathcal{N})$, and let $\mathcal{N}' = \mathcal{N} \setminus \{N_r, N_s\}$. Suppose $\mathcal{N}' \neq \emptyset$. Then*

(1) *There is an $N'_i \in \mathcal{N}'$ with $f_{\text{swap},n}(N'_i, \mathcal{N}) = f_{\text{swap},n}(N_i, \mathcal{N})$.*
(2) *Such N'_i fulfills*

$$f_{\text{swap},n}(N_i, \mathcal{N}') = f_{\text{swap},n}(N'_i, \mathcal{N}') = \min_{N \in \mathcal{N}'} f_{\text{swap},n}(N, \mathcal{N}') .$$

(3) $\frac{f_{\text{swap},n}(N_i, \mathcal{N}')}{f^{\text{opt}}_{\text{swap},n}(\mathcal{N}')} \ge \frac{f_{\text{swap},n}(N_i, \mathcal{N})}{f^{\text{opt}}_{\text{swap},n}(\mathcal{N})}$.

Proof. Since N_r and N_s order each swap over V_n differently, we know that there exists some fixed $\delta \ge 0$ such that $f_{\text{swap},n}(N, \mathcal{N}') = f_{\text{swap},n}(N, \mathcal{N}) - \delta$ holds for *all* $N \in \mathcal{N}$. This means that

$$f_{\text{swap},n}(N_i, \mathcal{N}') = \min_{N \in \mathcal{N}} f_{\text{swap},n}(N, \mathcal{N}') ,$$

and each $N'_i \in \mathcal{N}'$ with $f_{\text{swap},n}(N_i, \mathcal{N}') = f_{\text{swap},n}(N'_i, \mathcal{N}')$ must also fulfill $f_{\text{swap},n}(N_i, \mathcal{N}) = f_{\text{swap},n}(N'_i, \mathcal{N})$. Thus, to prove (1) and (2), we only need to show that $\mathcal{N}'$ contains one of the best input CPTs for $\mathcal{N}$.

If this were false, then the chosen N_r or N_s—but no further N in the multiset $\mathcal{N}$—would be the best input CPT in $\mathcal{N}$. Let us assume, without loss of generality, that N_r is the best input CPT in $\mathcal{N}$. If $\mathcal{N}$ contains a second copy of N_r, we are done. So consider the case when $\mathcal{N}$ contains only one copy of N_r.

Then $\mathcal{N}$ also contains only one copy of N_s, since otherwise $f_{\text{swap},n}(N_s, \mathcal{N}) < f_{\text{swap},n}(N_r, \mathcal{N})$, contradicting the assumption that N_r is the best input CPT in $\mathcal{N}$. (To see why having more copies of N_s than copies of N_r implies $f_{\text{swap},n}(N_s, \mathcal{N}) < f_{\text{swap},n}(N_r, \mathcal{N})$, note that any two input CPTs with different parent sets agree on exactly half the swaps, since each parent set is of size at most 1.)

Since $\mathcal{N}' \neq \emptyset$, choose a CPT $N \in \mathcal{N}'$ over a parent set P, such that the number of copies of N in $\mathcal{N}$ is no smaller than the number of copies of $\overline{N}$ in $\mathcal{N}$, where $\overline{N}$ is the CPT with parent set P that entails the opposite ordering of $\succ_N$. Again exploiting the fact that any two input CPTs with different parent sets agree on exactly half the swaps, it is not hard to see that $f_{\text{swap},n}(N, \mathcal{N}) \leq f_{\text{swap},n}(N_r, \mathcal{N})$. This is a contradiction. Thus, there is a best input CPT for $\mathcal{N}$ that is still contained in $\mathcal{N}'$. Hence (1) and (2) are proven.

For every swap over V_n, the pair (N_r, N_s) contributes exactly one error to both $f_{\text{swap},n}(N_i, \mathcal{N})$ and $f^{\text{opt}}_{\text{swap},n}(\mathcal{N})$ since they entail opposite orderings. Therefore, $f_{\text{swap},n}(N_i, \mathcal{N}') = f_{\text{swap},n}(N_i, \mathcal{N}) - 2^{n-1}$ and $f^{\text{opt}}_{\text{swap},n}(\mathcal{N}') = f^{\text{opt}}_{\text{swap},n}(\mathcal{N}) - 2^{n-1}$. Now statement (3) follows from $f_{\text{swap},n}(N_i, \mathcal{N}) \geq f^{\text{opt}}_{\text{swap},n}(\mathcal{N})$. □

By Lemma 4.1, deleting N_r and N_s increases the approximation ratio of the best input CPT. Recall that our goal is to establish an upper bound on the worst-case approximation ratio. The latter cannot be maximized by instances containing pairs like (N_r, N_s). In what follows, we hence assume that, in our input instances, all 1-bounded CPTs for V_n with the same parent set are identical.

Note that a 1-bounded CPT for attribute V_n can have one of n possible parent sets, namely $\{V_1\}, \ldots, \{V_{n-1}\}$, or $\emptyset$. For each parent set, we only need to consider one of the two possible preference orderings, by Lemma 4.1. Thus, we can write any input instance as a vector $(q_1, \ldots, q_n)$ where the entry q_n is the number of occurrences of the CPT with empty parent set, and q_i, for $1 \leq i \leq n-1$, is the number of occurrences of the CPT with parent set $\{V_i\}$.

Now consider any two parent sets M_1 and M_2 for V_n, each of size at most 1. The number of swaps of V_n for which two CPTs with parent sets M_1 and M_2, resp., disagree equals 2^{n-2} (half the total number of swaps of V_n). Note that this holds regardless of whether one of M_1 and M_2 is the empty set or not (this argument is also used in the proof of Lemma 4.1). Therefore, the best input CPT is simply one that occurs most frequently, i.e., one for which the value q_i in the vector $(q_1, \ldots, q_n)$ is largest. We therefore consider the following representation of input instances, called input profiles.

Definition 4.2. An input profile (of 1-bounded CPTs for V_n) is a tuple $P = (t^P_1, \ldots, t^P_n)$ with $t^P_1 \geq t^P_2 \geq \ldots \geq t^P_n$, where $(t^P_1, \ldots, t^P_n)$ is a permutation of $(q_1, \ldots, q_n)$, which represents an input instance of 1-bounded CPTs as described above. For any such input profile, the best input CPT, denoted N^P, is an input CPT corresponding to the frequency value t^P_1.

If $t^P_1 = t^P_2$, the best input CPT is not unique. We then use N^P to refer to any CPT most frequently occurring in the input instance.

Given a tuple $P = (t^P_1, \ldots, t^P_n)$ with $t^P_1 \geq t^P_2 \geq \ldots \geq t^P_n$, let p be the largest index satisfying $t^P_p > 0$. Let T_i, $1 \leq i \leq p$, be the CPT corresponding to t^P_i. Let (Π_1, Π_2) be a partition of the set $\{T_1, \ldots, T_p\}$ on a swap w such that

(1) Π_1 contains T_1 and all T_i that have the same ordering as T_1 on w, and
(2) Π_2 contains all elements that are not in Π_1.

z	Sum_1	Sum_2	z	Sum_1	Sum_2
0	t^P_1	$t^P_2 + t^P_3 + t^P_4$	2	$t^P_1 + t^P_2 + t^P_3$	t^P_4
1	$t^P_1 + t^P_2$	$t^P_3 + t^P_4$	2	$t^P_1 + t^P_2 + t^P_4$	t^P_3
1	$t^P_1 + t^P_3$	$t^P_2 + t^P_4$	2	$t^P_1 + t^P_3 + t^P_4$	t^P_2
1	$t^P_1 + t^P_4$	$t^P_2 + t^P_3$	3	$t^P_1 + t^P_2 + t^P_3 + t^P_4$	0

Table 1: Possible (Sum_1, Sum_2) pairs corresponding to all partitions (Π_1, Π_2) when $n = 4$ and $t^P_4 > 0$.

Since each of the $p-1$ CPTs $T_2, \ldots, T_p$ has a unique parent set, it is not hard to see that for any given tuple P, there are 2^{p-1} different partitions (Π_1, Π_2). Note that each partition (Π_1, Π_2) corresponds to 2^{n-p} swaps.

For each such partition and each $b \in \{1, 2\}$, we define Sum_b to be the sum of the values t^P_i where $T_i \in \Pi_b$.

For instance, assume the CPTs in Π_1 have ordering $0 \succ 1$, and those in Π_2 have ordering $1 \succ 0$. Then the number of CPTs with vote $0 \succ 1$ is Sum_1 and the number of CPTs with vote $1 \succ 0$ is Sum_2; one can simply swap the numbers for the opposite ordering.

On each of the 2^{n-1} swaps of V_n, an optimal solution N^* has a preference ordering not in the minority, and the number of errors made by N^* is the frequency of the ordering that is in the minority. For all swaps whose ordering entailed by N^P is not in the minority, N^P and N^* thus make the same number of errors. For swaps where the ordering entailed by N^P is in the minority, N^P makes more errors than N^*. From our notion of partitions, N^P and N^* have the same ordering whenever $Sum_1 \geq Sum_2$, and opposite orderings when $Sum_1 < Sum_2$. For the former, N^P and N^* both make error Sum_2. For the latter, N^P makes error Sum_2 and N^* makes error Sum_1. Given any input profile P as in Definition 4.2, we introduce some notation to calculate the errors made by N^P and N^*.

A profile P lists the multiplicities of n distinct input CPTs. Suppose, for some swap, there are z ($0 \leq z \leq n-1$) distinct input CPTs other than N^P with the same ordering as N^P. These input CPTs then agree on 2^{n-z-1} swaps. Depending on the values of t^P_i and z, the ordering of N^P may be the same as the majority ordering for these swaps. This is equivalent to saying that N^P and the z CPTs in Π_1 have the same ordering. We can exhaustively list all possible partitions where z frequency values t^P_i have the same ordering as N^P for any specific swap. Summing up these t^P_i values gives us the number Sum_1 of input CPTs that agree with N^P on that swap, while the sum Sum_2 of the remaining t^P_i values is the number of input CPTs that disagree with N^P on that same swap. An example for $n = 4$ is shown in Table 1.

Each partition corresponds to some number α of swaps. For each row, the error made by N^P is $\alpha \cdot Sum_2$, while the error made by N^* is $\alpha \cdot \min(Sum_1, Sum_2)$. When $Sum_1 = Sum_2$, we use $\alpha \times Sum_2$ as the error made by N^* for convenience with some of our proofs.

We conclude this section with a technical lemma needed for proving our main result. In what follows, we say that a given partition *aligns* with a swap iff the CPTs corresponding to each t^P_i have orderings consistent with the two parts of the given partition.

Lemma 4.3. *For an input profile $P = (t_1^P, \ldots, t_n^P)$ with $t_1^P \geq t_2^P \geq \ldots \geq t_n^P \geq 1$, each of the 2^{n-1} possible partitions (Π_1, Π_2) aligns with exactly one swap of V_n.*

Proof. Let (Π_1, Π_2) be an arbitrary partition of the CPTs corresponding to all t_i^P. We prove our claim by considering two cases.

Case 1. Π_1 contains a CPT $N_\emptyset$ with empty parent set. WLOG, assume $N_\emptyset$ entails $0 \succ 1$ for all 2^{n-1} swaps of V_n. Then all CPTs in Π_1 entail $0 \succ 1$ and all those in Π_2 entail $1 \succ 0$. A swap with this ordering exists, as all input CPTs other than $N_\emptyset$ can take on either of the two possible orderings. Clearly, there is exactly one swap with this configuration of orderings.

Case 2. Π_2 contains a CPT $N_\emptyset$ with empty parent set. WLOG, assume $N_\emptyset$ entails $0 \succ 1$ for all 2^{n-1} swaps of V_n. Then all CPTs in Π_1 entail $1 \succ 0$ and all those in Π_2 entail $0 \succ 1$. As in Case 1, exactly one swap with this ordering exists. □

Remark. There might be some $n' < n$ with $t_{n'}^P > t_{n'+1}^P = t_{n'+2}^P = \ldots = t_n^P = 0$. Then each Sum_1/Sum_2 combination corresponds to $2^{n-n'}$ swaps. While this changes the actual error values, the ratio will remain unchanged. We thus present all results as if $n' = n$.

5 BOUNDED APPROXIMATION RATIO FOR 1-BOUNDED CP-NETS

This section establishes our main result:

Theorem 5.1. *For any input profile of tree CPTs over $n \geq 2$ attributes, the best input CPT has an approximation ratio of at most $\frac{4}{3}$.*

The proof of this result makes use of a sequence of lemmas, the first of which proves the main result for the case when $n = 2$.

Lemma 5.2. *For any input profile of tree CPTs over $n = 2$ attributes, the best input CPT is an optimal solution.*

Proof. Let $P = (t_1^P, t_2^P)$ be an input profile of tree CPTs. Then the best input CPT N^P gives $f^{\mathrm{a}}_{\mathrm{swap},n}(N^P, P) = f^{\mathrm{opt}}_{\mathrm{swap},n}(P) = t_2^P$. □

Thus, when $n = 2$, the approximation ratio of the best input CPT is 1. We now focus on the case $n \geq 3$. The next two lemmas, which are needed for the proof of Lemma 5.9, apply to input profiles P of a specific structure.

Lemma 5.3. *Let P be an input profile of tree CPTs of the form $P = (k, \ldots, k, k') \in \mathbb{N}^n$ where $1 \leq k' \leq k$. Let N^P be the best input CPT, and $t = (n-1)k + k'$. Then*

$$f_{\mathrm{swap},n}(N^P, P) = 2^{n-2} \sum_{i=2}^{n} t_i^P = (t-k)2^{n-2}.$$

Proof. For each Sum_1/Sum_2 combination, the error made by N^P is the Sum_2 term, and the total error is the sum of all these Sum_2 terms. Each t_i^P, $2 \leq i \leq n$, appears on the Sum_2 in exactly half of the 2^{n-1} Sum_1/Sum_2 combinations. This proves the lemma. □

Lemma 5.4. *Let $n = 2c - 1$ for some $c > 1$, and let P be an input profile of tree CPTs of the form $P = (k, \ldots, k, k') \in \mathbb{N}^n$ where $1 \leq k' \leq k$. Let $t = (n-1)k + k'$. Then*

$$f^{\mathrm{opt}}_{\mathrm{swap},n}(P) = (2^{2c-3} - \binom{2c-3}{c-2}) \cdot t\,.$$

Proof. $f^{\mathrm{opt}}_{\mathrm{swap},n}(P)$ is the sum of all Sum_1 terms that are no larger than their corresponding Sum_2 terms, plus the sum of all Sum_2 terms that are smaller than their Sum_1 terms (multiplied by the number of swaps each combination represents). Since $(c-1)k < \frac{1}{2}(n-1)k + k'$, we know that no Sum_2 term summing up at most $c-1$ entries of P can form a majority over Sum_1. Hence we only consider Sum_1/Sum_2 combinations in which Sum_2 is a sum of at least c entries of P, so that Sum_1 contains t_1^P $(= k)$ plus at most $c-2$ further entries of P. We derive an expression for $f^{\mathrm{opt}}_{\mathrm{swap},n}$ by summing up the errors made by an optimal N^* for each of the 2^{n-1} combinations. Since t_1^P is in Sum_1, the number of times t_1^P appears in $f^{\mathrm{opt}}_{\mathrm{swap},n}$ is $\sum_{i=0}^{c-2}\binom{2c-2}{i}$ (the number of combinations for which $Sum_1 < Sum_2$). For $2 \leq i \leq n$, t_i^P may appear in the expression for $f^{\mathrm{opt}}_{\mathrm{swap},n}$, in one of two possible ways:

- Case 1. t_i^P is in Sum_1 and $Sum_1 < Sum_2$
- Case 2. t_i^P is in Sum_2 and $Sum_1 \geq Sum_2$

For Case 1, since $Sum_1 < Sum_2$, the Sum_1 has at most $c-3$ terms other than t_1^P and t_i^P, which corresponds to $\sum_{i=0}^{c-3}\binom{2c-3}{i}$ combinations. For Case 2, since $Sum_1 \geq Sum_2$, the Sum_2 has at most $c-2$ terms other than t_1^P and t_i^P, yielding $\sum_{i=0}^{c-2}\binom{2c-3}{i}$ combinations. Combining these we get $2^{2c-3} - \binom{2c-3}{c-2}$ (the frequency with which each t_i^P occurs in the error made by an optimal solution). Thus

$$\begin{aligned}
f^{\mathrm{opt}}_{\mathrm{swap},n}(P) &= t_1^P \cdot \sum_{\kappa=0}^{c-2}\binom{2c-2}{\kappa} + \sum_{i=2}^{n} t_i^P \cdot (2^{2c-3} - \binom{2c-3}{c-2}) \\
&= (2^{2c-3} - \frac{1}{2}\cdot\binom{2c-2}{c-1}) \cdot t_1^P + ((2^{2c-3} - \binom{2c-3}{c-2})(t - t_1^P) \\
&= (2^{2c-3} - \binom{2c-3}{c-2}) \cdot t_1^P + ((2^{2c-3} - \binom{2c-3}{c-2})(t - t_1^P) \\
&= (2^{2c-3} - \binom{2c-3}{c-2}) \cdot t\,,
\end{aligned}$$

which completes our proof. □

The following lemma addresses the situation when each input CPT occurs equally often. A similar result was obtained by Ali et al. [6], but in their case the empty parent set was not included.

Lemma 5.5. *Let $n = 2c - 1$ for some $c > 1$, and let P be an input profile of tree CPTs of the form $P = (k, \ldots, k) \in \mathbb{N}^n$. Let N^P be the best input CPT. Then*

$$\begin{aligned}
f_{\mathrm{swap},n}(N^P, P) &= k(n-1)2^{n-2} = k(n-1)2^{2c-3} \text{ and} \\
f^{\mathrm{opt}}_{\mathrm{swap},n}(P) &= (2^{2c-3} - \binom{2c-3}{c-2}) \cdot kn\,.
\end{aligned}$$

In particular, the best input CPT has an approximation ratio of at most $\frac{4}{3}$.

Proof. Lemmas 5.3 and 5.4 yield the formulas for $f_{\mathrm{swap},n}(N^P, P)$ and $f^{\mathrm{opt}}_{\mathrm{swap},n}(P)$. It suffices to show that $f_{\mathrm{swap},n}(N^P, P)/f^{\mathrm{opt}}_{\mathrm{swap},n}(P) \leq \frac{4}{3}$. Since the factor of k cancels out, we only need to show that

$$\frac{(n-1)2^{n-2}}{n(2^{2c-3} - \binom{2c-3}{c-2})} \leq \frac{4}{3}\,.$$

To this end, Ali et al. [6] showed that, when $n = 2c - 1$,

$$\frac{(t-1)2^{n-2}}{t2^{n-2} - 2^{n-t-1}c\binom{2c-1}{c}} \leq \frac{4}{3},$$

for any value of t. For $t = n$, this shows that

$$\frac{(n-1)2^{n-2}}{n2^{n-2} - \frac{c}{2}\binom{2c-1}{c}} \leq \frac{4}{3}.$$

Note that $n(2^{2c-3} - \binom{2c-3}{c-2}) = n2^{n-2} - n\binom{n-2}{c-2}$. So, it remains to show that $\frac{c}{2}\binom{n}{c} = n\binom{n-2}{c-2}$. The latter holds true since $\frac{c}{2}\binom{n}{c} = \frac{n}{c} \cdot \frac{n-1}{c-1} \cdot \frac{c}{2}\binom{n-2}{c-2} = \frac{n(n-1)}{2c-2}\binom{n-2}{c-2} = n\binom{n-2}{c-2}$. □

Subsequently, we will often analyze how the approximation ratio of the best input CPT changes when changing the input profile. For ease of presentation, we therefore introduce the following notation.

Definition 5.6. Let P be an input profile of CPTs. Then r_P denotes the approximation ratio of the best input CPT in P.

The following corollary is an additional strengthening of one of the main results by Ali et al. [6]. They showed that, for input profiles in which each CPT occurs equally often, the best input CPT gives a $\frac{4}{3}$-approximation. We demonstrate (for odd n) that this approximation ratio actually converges to 1 as n increases.

Corollary 5.7. *Let $(P_{2c-1})_{c\in\mathbb{N}}$ be a sequence of input profiles of 1-bounded CPTs of the form $P_n = (k_n, \ldots, k_n) \in \mathbb{N}^n$, where $n = 2c-1$. Then $r_{P_{2c-1}}$ is a decreasing function of c with $\lim_{2c-1\to\infty} r_{P_{2c-1}} = 1$.*

Proof. The approximation ratio of the best input CPT for P_n is

$$\begin{aligned}\frac{(n-1)2^{2c-3}}{(2^{2c-3} - \binom{2c-3}{c-2}) \cdot n} &= \frac{(2c-2)2^{2c-3}}{(2^{2c-3} - \binom{2c-3}{c-2}) \cdot (2c-1)} \\ &= \frac{(2c-2)2^{2c-3}}{(2^{2c-3} - \frac{2^{2c-3}}{\theta(\sqrt{c})}) \cdot (2c-1)} \\ &= \frac{(2c-2)}{(1 - \frac{1}{\theta(\sqrt{c})}) \cdot (2c-1)},\end{aligned}$$

which decreases and converges to 1 as c increases. □

This corollary gives suggests that *near-optimal* aggregation of interesting sub-classes of CP-nets may be possible: for a small number n of attributes, one could obtain optimal solutions directly, while for larger n (when optimal aggregation is infeasible), one could invoke an efficient approximation algorithm with a low approximation ratio. Pursuing this idea is beyond the scope of this paper.

We now formulate two crucial lemmas that will allow us to compare the approximation ratios of specific kinds of input profiles.

Lemma 5.8. *Let $n = 2c$ for some $c > 1$. Let P and Q be input profiles of 1-bounded CPTs of the form $P = (k, \ldots, k, k') \in \mathbb{N}^n$ and $Q = (k, \ldots, k, k'-1) \in \mathbb{N}^n$ where $1 \leq k' \leq k$. Then $r_Q > r_P$.*

Proof. Since $t_n^P = t_n^Q + 1$, both $f_{\text{swap},n}(N^S, S)$ and $f^{\text{opt}}_{\text{swap},n}(S)$ are lower for $S = Q$ than for $S = P$. Let us denote the difference in these two errors by $\delta^{\text{app}}_{\text{swap}}$ and $\delta^{\text{opt}}_{\text{swap}}$, resp. From Lemma 5.3, it follows that $\delta^{\text{app}}_{\text{swap}} = 2^{n-2} = 2^{2c-2}$.

Note that Q results from P by decrementing t_n^P, which decreases $f^{\text{opt}}_{\text{swap},n}$ for all Sum_1/Sum_2 combinations where t_n^P is in the minority:

- Case 1. t_n^P is in Sum_1 and $Sum_1 < Sum_2$
- Case 2. t_n^P is in Sum_2 and $Sum_1 \geq Sum_2$

For Case 1, since $Sum_1 < Sum_2$, the Sum_1 has at most $c-2$ terms other than t_1^P and t_n^P. The number of combinations satisfying these constraints is $\sum_{\kappa=0}^{c-2}\binom{2c-2}{\kappa}$. For Case 2, since $Sum_1 \geq Sum_2$, the Sum_1 has at least $c-1$ terms other than t_1^P and t_n^P. The number of combinations satisfying these constraints is $\sum_{\kappa=c-1}^{2c-2}\binom{2c-2}{\kappa}$. In total, from both cases, we get $\sum_{\kappa=0}^{2c-2}\binom{2c-2}{\kappa} = 2^{2c-2}$ combinations. Thus, $\delta^{\text{app}}_{\text{swap}} = \delta^{\text{opt}}_{\text{swap}} = 2^{2c-2}$. Since $r_P > 1$ and $\delta^{\text{app}}_{\text{swap}} = \delta^{\text{opt}}_{\text{swap}}$, we obtain $r_Q > r_P$. □

Interestingly, for odd values of n, one observes the opposite effect to that described in Lemma 5.8.

Lemma 5.9. *Let $n = 2c-1$ for some $c > 1$. Let P and Q be input profiles of 1-bounded CPTs of the form $P = (k, \ldots, k, k') \in \mathbb{N}^n$ and $Q = (k, \ldots, k, k'+1) \in \mathbb{N}^n$ where $1 \leq k' < k$. Then $r_Q > r_P$.*

Proof. The proof is in part similar to that of Lemma 5.8. Since $t_n^P = t_n^Q - 1$, both $f_{\text{swap},n}(N^S, S)$ and $f^{\text{opt}}_{\text{swap},n}(S)$ are higher for $S = Q$ than for $S = P$. Let us denote the difference in these two errors by $\delta^{\text{app}}_{\text{swap}}$ and $\delta^{\text{opt}}_{\text{swap}}$, resp. By Lemma 5.3, $\delta^{\text{app}}_{\text{swap}} = 2^{n-2} = 2^{2c-3}$.

Note that Q results from P by incrementing t_n^P, which increases $f^{\text{opt}}_{\text{swap},n}$ for all Sum_1/Sum_2 combinations where t_n^P is in the minority:

- Case 1. t_n^P is in Sum_1 and $Sum_1 < Sum_2$
- Case 2. t_n^P is in Sum_2 and $Sum_1 \geq Sum_2$

For Case 1, since $Sum_1 < Sum_2$, the Sum_1 has at most $c-3$ terms other than t_1^P and t_n^P. The number of combinations satisfying these constraints is $\sum_{\kappa=0}^{c-3}\binom{2c-3}{\kappa}$. For Case 2, since $Sum_1 \geq Sum_2$, the Sum_1 has at least $c-1$ terms other than t_1^P and t_n^P. The number of combinations satisfying these constraints is $\sum_{\kappa=c-1}^{2c-3}\binom{2c-3}{\kappa}$. For the total number of combinations from both cases, after simplifying with the help of combinatorial identities, we get $2^{2c-3} - \binom{2c-3}{c-2}$.

Let $t = (n-1)k + k'$. From Lemmas 5.3 and 5.4, we have

$$\begin{aligned} r_P &= \frac{(t - t_1^P)2^{2c-3}}{(2^{2c-3} - \binom{2c-3}{c-2}) \cdot t} < \frac{t \cdot 2^{2c-3}}{(2^{2c-3} - \binom{2c-3}{c-2}) \cdot t} \\ &= \frac{2^{2c-3}}{(2^{2c-3} - \binom{2c-3}{c-2})} = \frac{\delta^{\text{app}}_{\text{swap}}}{\delta^{\text{opt}}_{\text{swap}}}.\end{aligned}$$

The above shows that the ratio of the changes to $f_{\text{swap},n}(N^S, S)$ and $f^{\text{opt}}_{\text{swap},n}(S)$, resp., is strictly greater than r_P. From this it follows that $r_Q > r_P$, which completes our proof. □

We are now able to prove a special case of our main result:

Theorem 5.10. *Let P be an input profile of 1-bounded CPTs of the form $P = (k, \ldots, k, k') \in \mathbb{N}^n$ where $1 \leq k' \leq k$. Then the best input CPT has an approximation ratio of at most $\frac{4}{3}$, i.e., $r_P \leq \frac{4}{3}$.*

Proof. Depending on whether n is odd or even, we recursively apply Lemma 5.8 or Lemma 5.9, resp. For $n = 2c$, we stop when we arrive at some Q with $t_i^Q = k$ for all $i \neq n$, and $t_n^Q = 0$. Since this gives us $2c-1$ non-zero t_i^Q values, all of which equal k, Lemma 5.5 proves our claim. For $n = 2c-1$, we stop when we arrive at some Q with $t_i^Q = k$ for all n. Once again our claim holds by Lemma 5.5. □

To extend this special case result to a full proof of Theorem 5.1, we need just one more lemma.

Lemma 5.11. *Let $P = (t_1^P, \ldots, t_n^P)$ and $Q = (t_1^Q, \ldots, t_n^Q)$ be input profiles of 1-bounded CPTs over n attributes, $n \geq 3$. Suppose there are i and j, $1 < i < j \leq n$, such that $t_i^Q = t_i^P + 1$, $t_j^Q = t_j^P - 1$, and $t_m^Q = t_m^P$ for $m \notin \{i, j\}$. Then $r_Q \geq r_P$.*

Proof. Since $i > 1$, we have $t_1^P = t_1^Q$. Further, since P and Q have the same total number of CPTs, we have $f_{\mathrm{swap},n}(N^P, P) = f_{\mathrm{swap},n}(N^Q, Q)$. It suffices to show that $f^{\mathrm{opt}}_{\mathrm{swap},n}(P) \geq f^{\mathrm{opt}}_{\mathrm{swap},n}(Q)$. Note that $t_i^P \geq t_j^P$. Let $M = \{t_m^P \mid m \notin \{i, j\}\}$ be the multiset of entries in P except t_i^P and t_j^P. Note that M has $n-2$ members. Let M_1 and M_2 denote some partition of M, and S_1 and S_2 the sum of the integers in M_1 and M_2, respectively. For convenience, we assume M_1 always contains t_1^P. Note that $M_2 = \emptyset$ is possible.

Since $t_j^Q = t_j^P - 1$ and $t_i^Q = t_i^P + 1$, we can abstract all possible combinations of Sum_1 and Sum_2 to the following four.

(a) $t_i^P + S_1$ and $t_j^P + S_2$ (b) $t_j^P + S_1$ and $t_i^P + S_2$
(c) $t_i^P + t_j^P + S_1$ and S_2 (d) S_1 and $t_i^P + t_j^P + S_2$

For (c) and (d), the CPTs corresponding to indices i and j have identical orderings and there is no net difference between $f^{\mathrm{opt}}_{\mathrm{swap},n}(P)$ and $f^{\mathrm{opt}}_{\mathrm{swap},n}(Q)$ contributed by the swaps associated with such Sum_1 / Sum_2 combinations. It suffices to show that in cases (a) and (b) the corresponding swaps do not contribute a net positive difference $\delta := f^{\mathrm{opt}}_{\mathrm{swap},n}(Q) - f^{\mathrm{opt}}_{\mathrm{swap},n}(P)$.

The δ value for a given partition (M_1, M_2) in case (a) (or (b)) depends on which of the two sums in (a) (or (b)) is larger. This leads to the following four cases.

(a.1) $t_i^P + S_1 < t_j^P + S_2$, $t_j^P + S_1 < t_i^P + S_2$
(a.2) $t_i^P + S_1 < t_j^P + S_2$, $t_j^P + S_1 \geq t_i^P + S_2$
(b.1) $t_i^P + S_1 \geq t_j^P + S_2$, $t_j^P + S_1 < t_i^P + S_2$
(b.2) $t_i^P + S_1 \geq t_j^P + S_2$, $t_j^P + S_1 \geq t_i^P + S_2$

By Lemma 4.3, each combination corresponds to exactly one swap. Thus, each of the abstractions above corresponds to 2^{n-3} swaps. For cases (a) and (b), we consider the contribution of the corresponding pair of swaps to the value δ, in each of these four cases. In a case where t_j^P is counted in the minority, the corresponding swap contributes -1 to δ, and when t_j^P is counted in the majority, it contributes +1 to δ. We thus obtain the following net contributions for each of the above cases, for a given M_1, M_2 pair:

(a.1) Net contribution 0.
(a.2) Since $t_i^P \geq t_j^P$, this combination is not possible.
(b.1) Net contribution -2.
(b.2) Net contribution 0.

Hence, for any given partition (M_1, M_2), the contribution to the value $\delta := f^{\mathrm{opt}}_{\mathrm{swap},n}(Q) - f^{\mathrm{opt}}_{\mathrm{swap},n}(P)$ is non-positive. So, $f^{\mathrm{opt}}_{\mathrm{swap},n}(P) \geq f^{\mathrm{opt}}_{\mathrm{swap},n}(Q)$, as desired, which yields $r_Q > r_P$. □

Finally, we are in a position to prove our main result:

Theorem 5.1. *For any input profile of tree CPTs over $n \geq 2$ attributes, the best input CPT has an approximation ratio of at most $\frac{4}{3}$.*

Proof. For $n = 2$, this follows from Lemma 5.2. So, let P be an input profile of 1-bounded CPTs over $n \geq 3$ attributes. We build a profile P' from P by repeatedly applying Lemma 5.11 as follows.

Initialize $P' = P$, $i = 2$, and $j = n$. As long as $i < j$, repeat the following steps:

(i) If $t_i^{P'} = t_1^{P'}$ then increment i; if $t_j^{P'} = 0$ then decrement j.
(ii) Consider an input profile Q such that $t_i^Q = t_i^{P'} + 1$ and $t_j^Q = t_j^{P'} - 1$, while $t_m^Q = t_m^{P'}$ for $m \notin \{i, j\}$. By Lemma 5.11, $r_Q \geq r_{P'}$. Set $P' := Q$.

When the above process halts, the profile P' is of the form $P' = (k, \ldots, k, k') \in \mathbb{N}^{n'}$, where $n' \leq n$ and $k' \leq k$. By construction, $r_{P'} \geq r_P$. Now Theorem 5.10 yields $\frac{4}{3} \geq r_{P'} \geq r_P$. □

Note that our upper bound in Theorem 5.1 is tight, since Ali et al. proved that the best input CPT for the profile (m, m, m) has an approximation ratio of $\frac{4}{3}$ [6].

6 INPUT INSTANCES WITH WORST APPROXIMATION RATIO

While we have proven that a very simple and efficient method obtains a $\frac{4}{3}$-approximation ratio when aggregating 1-bounded CP-nets, it remains open which input profiles P of a given number z of 1-bounded CP-nets yield the maximum value of r_P, amongst all such profiles. In other words, it is not known, given a number z, which combination of z 1-bounded CP-nets gives rise to the highest approximation ratio when using the best input as an aggregate. We are interested in finding such problem instances since results along these lines may ultimately help us characterize the problem instances that have highest ratio, and to bound the average approximation ratio taken over all problem instances. Such a bound might be substantially below $\frac{4}{3}$. We conjecture that the ratio of $\frac{4}{3}$ is attained only for certain problem instances, and that, for large values of n, the approximation ratio may even approach 1, which would be a significant result. This section provides first steps in this direction, by identifying a particular such combination for the special case when, in addition, the number n of attributes is fixed at $n = 3$. In particular, we will show the following:

> Among all input profiles of $z \geq 3$ 1-bounded CPTs over $n = 3$ attributes, a profile P maximizing r_P is
> - $P = (m, m, m)$ in case $z = 3m$;
> - $P = (m+1, m, m)$ in case $z = 3m+1$; (here it should be noted that, for $P' = (m, m, m, 1)$, we get $r_{P'} > r_P$, but P' uses more than 3 attributes;)
> - $P = (m+1, m+1, m)$ in case $z = 3m+2$.

Lemma 6.1. *Let $P = (t_1^P, t_2^P, t_3^P) \in (\mathbb{N} \setminus \{0\})^3$ be an input profile of 1-bounded CP-nets over $n = 3$ attributes, and define Q_k as follows.*

(1) If $t_2^P > t_3^P + 1$, then $Q_1 := (t_1^P, t_2^P - 1, t_3^P + 1)$.
(2) If $t_1^P > t_2^P > t_3^P$, then $Q_2 := (t_1^P - 1, t_2^P, t_3^P + 1)$.
(3) If $t_1^P > t_2^P + 1$, then $Q_3 := (t_1^P - 1, t_2^P + 1, t_3^P)$.

In each case, Q_k is an input profile with $r_{Q_k} \geq r_P$.

Proof. In each case, with $k \in \{1, 2, 3\}$ and $Q = Q_k$, we have $Q = (t_1^Q, t_2^Q, t_3^Q)$ for values t_1^Q, t_2^Q, t_3^Q satisfying $t_1^Q \leq t_2^Q \leq t_3^Q$.

Below, we exhaustively enumerate all four Sum_1 / Sum_2 combinations, with each combination corresponding to $2^{n-3} = 1$ swap.

We assume $t_1^P < t_2^P + t_3^P$ because otherwise the best input CPT is an optimal solution and $r_P = 1$, so that obviously $r_Q \geq r_P$. The remaining three inequalities follow from $t_1^P \geq t_2^P \geq t_3^P \geq 1$:

(i) $t_1^P < t_2^P + t_3^P$ (ii) $t_1^P + t_2^P > t_3^P$
(iii) $t_1^P + t_3^P > t_2^P$ (iv) $t_1^P + t_2^P + t_3^P > 0$

Case (1). $P = (t_1^P, t_2^P, t_3^P)$ and $Q = (t_1^P, t_2^P - 1, t_3^P + 1)$. Since $t_1^P = t_1^Q$ and P and Q have the same total number of CPTs, $f_{\text{swap},n}(N^Q, Q) = f_{\text{swap},n}(N^P, P)$. From P to Q, the number of errors made by an optimal solution increases for (ii) and decreases for (iii), cancelling each other out. For combinations (i) and (iv), there is no change in the number of errors made by an optimal solution, when moving from P to Q. Thus $f^{\text{opt}}_{\text{swap},n}(Q) = f^{\text{opt}}_{\text{swap},n}(P)$ and $r_Q = r_P$.

Case (2). $P = (t_1^P, t_2^P, t_3^P)$ and $Q = (t_1^P - 1, t_2^P, t_3^P + 1)$. Since $t_1^P > t_1^Q$ and P and Q have the same total number of CPTs, $f_{\text{swap},n}(N^Q, Q) = f_{\text{swap},n}(N^P, P) + 2^{n-2}$. Next let us assess the relationship between $f^{\text{opt}}_{\text{swap},n}(P)$ and $f^{\text{opt}}_{\text{swap},n}(Q)$. In what follows, the term *margin* refers to the absolute difference between Sum_1 and Sum_2.

For combinations (iii) and (iv), there is no change in the number of errors made by an optimal solution, when moving from P to Q. For (i), the number of errors made by an optimal solution decreases by at most 1 (and does not decrease at all if the margin is 1). Summing up over all swaps under this combination, this gives a decrease of at most 2^{n-3} in the optimal error, when moving from P to Q. For (ii), the margin is strictly greater than 1 since $t_1^P > t_2^P$ and $t_3^P > 0$, so the number of errors made by an optimal solution increases by 1. Summing over all swaps under (iii), the error increases by 2^{n-3}. Combined with swaps under (i), this gives a net increase $\leq -2^{n-3}$. Thus $f^{\text{opt}}_{\text{swap},n}(Q) \leq f^{\text{opt}}_{\text{swap},n}(P) + 2^{n-3}$ and $r_Q > r_P$.

Case (3). $P = (t_1^P, t_2^P, t_3^P)$ and $Q = (t_1^P - 1, t_2^P + 1, t_3^P)$. The proof for this case is symmetric to that for Case 2, with respect to combinations (i) and (iii), and again we have $r_Q > r_P$. □

We obtain a formal statement about the "worst" input profiles in terms of approximation ratio of the best input CPT, for $n = 3$.

Corollary 6.2. *Let $n = 3$, $z \in \mathbb{N}$, and $P = (t_1^P, t_2^P, t_3^P)$ any input profile of 1-bounded CPTs, with $t_3^P \geq 1$.*

(1) *If $z = 3m$, then $r_P \leq r_{P_0^*}$ where $P_0^* = (m, m, m)$.*
(2) *If $z = 3m + 1$, then $r_P \leq r_{P_1^*}$ where $P_1^* = (m + 1, m, m)$.*
(3) *If $z = 3m + 2$, then $r_P \leq r_{P_2^*}$ where $P_2^* = (m + 1, m + 1, m)$.*

Proof. Statement (1) is immediate from Theorem 5.1, since $r_{P_0^*} = \frac{4}{3}$ by a result in [6]. For (2) and (3), note that $t_1^P \geq m + 1$, since $t_1^P \geq t_2^P \geq t_3^P$. Now Lemma 6.1 implies that P can be converted into P_i^* by a sequence of moves none of which decreases the approximation ratio of the best input CPT. This proves the corollary. □

Remark. From Theorem 5.1, $r_{P_0^*} \geq r_P$ for *all* input profiles P of $3m$ 1-bounded CPTs, regardless of the number n of attributes. We conjecture similarly, if P has $3m + 2$ CPTs, that $r_{P_2^*} \geq r_P$, regardless of the number of attributes in P. However, we can prove that, among profiles of $3m+1$ CPTs, the profile $r_{P_1^*}$ does *not* maximize r_P. (It only does so when limiting the number n of attributes to 3.) In particular, consider two profiles $P = (m, m, m, 1)$ and $P_1^* = (m+1, m, m)$. Using our combinatorial approach, we can see $f_{\text{swap},n}(N^P, P) = 8m + 4$ and $f^{\text{opt}}_{\text{swap},n}(P) = 6m + 4$. Similarly, we obtain $f_{\text{swap},n}(N^{P_1^*}, P_1^*) = 4m$ and $f^{\text{opt}}_{\text{swap},n}(P_1^*) = 3m + 1$. For all $m > 0$, $r_P > r_{P_1^*}$ because $\frac{4m}{3m+1} = \frac{8m}{6m+2} < \frac{8m+4}{6m+4}$. This leads to the following conjecture.

> *Conjecture.* Among all input profiles of z 1-bounded CPTs (with a variable number n of attributes), a profile P with highest ratio r_P is
> - $P = (m, m, m, 1)$ in case $z = 3m + 1$;
> - $P = (m + 1, m + 1, m)$ in case $z = 3m + 2$.

Proving this conjecture would complement our result from above, which states that, among all input profiles of $z = 3m$ 1-bounded CPTs (with a variable number n of attributes), a profile P with highest ratio r_P is $P = (m, m, m)$.

7 CONCLUSIONS

We have shown that a very simple and efficient method for aggregating CP-nets guarantees an approximation ratio of at most $\frac{4}{3}$ when all input CP-nets are 1-bounded CP-nets. Notably, this method also produces 1-bounded CP-nets as outputs. Our results are significant in various ways:

Firstly, 1-bounded CP-nets (including tree-structured CP-nets) are of interest in multiple studies on CP-nets, since they are expressive enough to be useful for many applications, while simple enough to allow for efficient learning and reasoning in many settings. For general voting aggregation (with explicit representation of preferences), the simple aggregation algorithm that we use cannot have a better approximation ratio than 2 in the worst case [20]. The fact that for CP-nets (i.e., for compact representation of preferences), an approximation ratio of $\frac{4}{3}$ is attainable, at least for a non-trivial sub-class, is noteworthy.

Secondly, note that the method of the best input CPT, when applied to 1-bounded CPTs, will always simply output a most frequent CPT (after subtracting all CPTs with same parent set but opposite ordering). This means, the method of selecting the best input CPT can be applied efficiently in an online fashion, when input CPTs are observed in a stream rather than in a batch. The method simply has to keep count of how often each CPT has occurred, and can then output a most frequent one.

Thirdly, our results substantially extend those of Ali et al. [6]. While we do not know the best possible approximation ratio for inputs more complex than 1-bounded CP-nets, we hope that the combinatorial techniques we developed here can be used to obtain new results in this direction. In this context, our insights into input instances forcing the "worst" approximation ratios (see Section 6) might also be helpful.

Finally, simple methods like the one we study here are easily implemented and analyzed in practical settings, thus giving our result practical relevance.

ACKNOWLEDGMENTS

B. Yang was supported in part by an NSERC Discovery Research Grant, Application No.: RGPIN-2018-06800. S. Zilles was supported by an NSERC Discovery Research Grant, Application No.: RGPIN-2017-05336, an NSERC Canada Research Chair, and a Canada CIFAR AI Chair held at the Alberta Machine Intelligence Institute (Amii).

REFERENCES

[1] Stéphane Airiau, Ulle Endriss, Umberto Grandi, Daniele Porello, and Joel Uckelman. 2011. Aggregating dependency graphs into voting agendas in multi-issue elections. In *Proc. Intl. Joint Conf. on Artificial Intelligence (IJCAI)*. 18–23.

[2] Eisa Alanazi, Malek Mouhoub, and Sandra Zilles. 2016. The Complexity of Learning Acyclic CP-Nets. In *IJCAI*. 1361–1367.

[3] Eisa Alanazi, Malek Mouhoub, and Sandra Zilles. 2018. The Complexity of Learning Acyclic Conditional Preference Networks. *arXiv preprint arXiv:1801.03968* (2018).

[4] Eisa Alanazi, Malek Mouhoub, and Sandra Zilles. 2020. The complexity of exact learning of acyclic conditional preference networks from swap examples. *Artificial Intelligence* 278 (2020), 103182.

[5] Abu Mohammad Hammad Ali, Howard J Hamilton, Elizabeth Rayner, Boting Yang, and Sandra Zilles. 2021. Aggregating Preferences Represented by Conditional Preference Networks. In *Proc. Intl. Conf. on Algorithmic Decision Theory (ADT 2021)*. 3–18.

[6] Abu Mohammad Hammad Ali, Boting Yang, and Sandra Zilles. 2024. Approximation Algorithms for Preference Aggregation Using CP-Nets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 10433–10441.

[7] Thomas E Allen, Cory Siler, and Judy Goldsmith. 2017. Learning tree-structured CP-nets with local search. In *The Thirtieth International Flairs Conference*.

[8] Christian Bachmaier, Franz J Brandenburg, Andreas Gleißner, and Andreas Hofmeier. 2015. On the hardness of maximum rank aggregation problems. *Journal of Discrete Algorithms* 31 (2015), 2–13.

[9] Richard Booth, Yann Chevaleyre, Jérôme Lang, Jérôme Mengin, and Chattrakul Sombattheera. 2010. Learning Conditionally Lexicographic Preference Relations. In *Proc. European Conf. on Artificial Intelligence (ECAI)*. 269–274.

[10] Craig Boutilier, Ronen I Brafman, Carmel Domshlak, Holger H Hoos, and David Poole. 2004. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research* 21 (2004), 135–191.

[11] Franz J Brandenburg, Andreas Gleißner, and Andreas Hofmeier. 2012. Comparing and aggregating partial orders with Kendall tau distances. In *WALCOM*. Springer, 88–99.

[12] Franz J Brandenburg, Andreas Gleißner, and Andreas Hofmeier. 2013. Comparing and aggregating partial orders with kendall tau distances. *Discrete Mathematics, Algorithms and Applications* 5, 02 (2013), 1360003.

[13] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. 2016. *Handbook of Computational Social Choice*. Cambridge University Press.

[14] Vincent Conitzer, Jérôme Lang, and Lirong Xia. 2011. Hypercubewise preference aggregation in multi-issue domains. In *Proc. Intl. Joint Conf. on Artificial Intelligence (IJCAI)*.

[15] Cristina Cornelio, Judy Goldsmith, Umberto Grandi, Nicholas Mattei, Francesca Rossi, and K Brent Venable. 2021. Reasoning with PCP-Nets. *Journal of Artificial Intelligence Research* 72 (2021), 1103–1161.

[16] Cristina Cornelio, Judy Goldsmith, Nicholas Mattei, Francesca Rossi, and K Brent Venable. 2013. Updates and uncertainty in CP-nets. In *Proc. 26th Australasian Joint Conf. on Artificial Intelligence (AI)*. 301–312.

[17] Cristina Cornelio, Umberto Grandi, Judy Goldsmith, Nicholas Mattei, Francesca Rossi, and Kristen Brent Venable. 2015. Reasoning with PCP-nets in a Multi-Agent Context. In *Proc. Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*. 969–977.

[18] Liviu P Dinu and Florin Manea. 2006. An efficient approach for the rank aggregation problem. *Theoretical Computer Science* 359 (2006), 455–461.

[19] Cynthia Dwork, Ravi Kumar, Moni Naor, and Dandapani Sivakumar. 2001. Rank aggregation methods for the web. In *TheWebConf*. ACM, 613–622.

[20] Ulle Endriss and Umberto Grandi. 2014. Binary aggregation by selection of the most representative voters. In *Proc. AAAI Conf. on Artificial Intelligence (AAAI)*, Vol. 28.

[21] Ulle Endriss, Umberto Grandi, and Daniele Porello. 2012. Complexity of judgment aggregation. *Journal of Artificial Intelligence Research* 45 (2012), 481–514.

[22] Frédéric Koriche and Bruno Zanuttini. 2010. Learning Conditional Preference Networks. *Artificial Intelligence* 174, 11 (2010), 685–703.

[23] Jérôme Lang. 2004. Logical preference representation and combinatorial vote. *Annals of Mathematics and Artificial Intelligence* 42, 1-3 (2004), 37–71.

[24] Jérôme Lang. 2007. Vote and Aggregation in Combinatorial Domains with Structured Preferences.. In *Proc. Intl. Joint Conf. on Artificial Intelligence (IJCAI)*, Vol. 7. 1366–1371.

[25] Jérôme Lang and Lirong Xia. 2009. Sequential composition of voting rules in multi-issue domains. *Mathematical Social Sciences* 57, 3 (2009), 304–324.

[26] Minyi Li, Quoc Bao Vo, and Ryszard Kowalczyk. 2011. Majority-rule-based preference aggregation on multi-attribute domains with CP-nets. In *Proc. Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*. 659–666.

[27] Thomas Lukasiewicz and Enrico Malizia. 2016. On the complexity of mCP-nets. In *Proc. AAAI Conf. on Artificial Intelligence (AAAI)*, Vol. 30.

[28] Thomas Lukasiewicz and Enrico Malizia. 2019. Complexity results for preference aggregation over (m) CP-nets: Pareto and majority voting. *Artificial Intelligence* 272 (2019), 101–142.

[29] Thomas Lukasiewicz and Enrico Malizia. 2022. Complexity results for preference aggregation over (m)CP-nets: Max and rank voting. *Artificial Intelligence* 303 (2022), 103636.

[30] Francesca Rossi, Kristen Brent Venable, and Toby Walsh. 2004. mCP nets: representing and reasoning with preferences of multiple agents. In *Proc. AAAI Conf. on Artificial Intelligence (AAAI)*. 729–734.

[31] D Sculley. 2007. Rank aggregation for similar items. In *Proc. SIAM Intl. Conf. on Data Mining*. 587–592.

[32] Lirong Xia, Vincent Conitzer, and Jérôme Lang. 2008. Voting on Multiattribute Domains with Cyclic Preferential Dependencies.. In *Proc. AAAI Conf. on Artificial Intelligence (AAAI)*, Vol. 8. 202–207.

[33] Lirong Xia, Jérôme Lang, and Mingsheng Ying. 2007. Sequential voting rules and multiple elections paradoxes. In *Proc. Conf. on Theoretical Aspects of Rationality and Knowledge*. 279–288.