A Soft Constraint-Based Framework for Ethical Reasoning

Hiroshi Hosobe[1][a] and Ken Satoh[2]
[1]*Faculty of Computer and Information Sciences, Hosei University, Tokyo, Japan*
[2]*National Institute of Informatics, Tokyo, Japan*
hosobe@acm.org, ksatoh@nii.ac.jp

Keywords: Ethical norm, Constraint programming, Soft constraint.

Abstract: Artificial intelligence is becoming more widely used for making decisions in many application areas, where it often needs to consider legal rules and ethical norms. However, ethical norms are more difficult to treat than legal rules that have logical nature. Taheri et al. proposed a framework that formalized several important aspects of ethical decision making. However, their framework is still not powerful enough for more general problem solving. In this paper, we propose a soft constraint-based framework for ethical reasoning. Especially by devising the notion of norm constraints, we integrate Taheri et al.'s framework into Borning et al.'s constraint hierarchy framework for treating soft constraints. We also present a case study on the application of our framework to ethical reasoning.

1 INTRODUCTION

Artificial intelligence is becoming more widely used for making decisions in many application areas, where it often needs to consider legal rules and ethical norms. To treat legal rules, legal reasoning has long been studied. An effective and promising approach is logic programming. For example, a Prolog-based system was developed for the purpose of reasoning about the Japanese civil code (Satoh et al., 2010).

Ethical reasoning also has been studied. However, ethical norms are more difficult to treat than legal rules that have logical nature. One problem is that ethical norms are not explicitly specified unlike legal rules that are specified in laws. Another more essential problem is that how to represent and evaluate ethical norms is unclear. Another is that there are many kinds of ethical norms that should be simultaneously considered but that might be conflicting.

To tackle these problems, Taheri et al. proposed a framework for ethical decision making (Taheri et al., 2023). Their framework formalized several important aspects of ethical decision making including how to organize ethical norms, how to evaluate choices according to individual norms, how to aggregate such evaluations for different classes of norms, and how to determine the best choices according to the entire norms.

However, Taheri et al.'s framework is still not powerful enough for more general problem solving. Basically, what the framework does is to compare choices according to given norms. Therefore, it needs to be combined with another method to handle rules and knowledge other than ethical norms. In fact, in addition to the ethical decision making framework, they needed to use a planner and a legal checker to develop a multi-agent real-time compliance mechanism (Hayashi et al., 2023). The ethical decision making framework worked as an ethical checker to select the ethically best plans from the legal plans generated by the planner and the legal checker.

In this paper, we propose a soft constraint-based framework for ethical reasoning. Our goal is to construct a unifying framework that enables expressive ethical decision making as well as powerful constraint programming. Especially by devising the notion of norm constraints, we integrate Taheri et al.'s framework into Borning et al.'s constraint hierarchy framework (Borning et al., 1992) for treating soft constraints. The resulting framework has twofold characters. On one hand, it can be regarded as an extension of Taheri et al.'s ethical decision making framework to constraint-based reasoning. On the other hand, it can be regarded as a special case of Borning et al.'s constraint hierarchy framework with concrete application to ethical reasoning.

The rest of this paper is organized as follows. Section 2 describes previous research that is related to our

[a] https://orcid.org/0000-0002-7975-052X

ICAART2024—https://icaart.scitevents.org/?y=2024
This is the author's version. The final authenticated version is available online at https://doi.org/10.5220/0012474100003636.

work. Section 3 briefly explains Taheri et al.'s framework and Borning et al.'s framework. Section 4 proposes our soft constraint-based framework for ethical reasoning. Section 5 presents a case study on the application of our framework to ethical reasoning. Section 6 discusses our framework. Finally, Section 7 gives conclusions and future work.

2 RELATED WORK

Satoh et al. proposed the use of constraints for legal and ethical reasoning (Satoh et al., 2021). More specifically, they proposed to use hard constraints for legal rules and to use soft constraints for ethical norms. They showed their concept by developing a system based on abductive planning with event calculus (Eshghi, 1988). However, their treatment of soft constraints was still limited as compared to other soft constraint frameworks such as constraint hierarchies (Borning et al., 1992).

Fungwacharakorn et al. proposed the use of constraint hierarchies to treat ethical norms (Fungwacharakorn et al., 2022a; Fungwacharakorn et al., 2022b). They represented ethical norms as soft constraints and studied the property of fundamental revisions and the debugging of represented norms. However, they used logical constraints that were evaluated as true or false, and did not consider more general soft constraints that could be evaluated in a more detailed manner as shown in this paper.

3 PRELIMINARIES

In this section, we explain two frameworks as preliminaries to our framework.

3.1 Ethical Decision Making

First, we briefly explain Taheri et al.'s ethical decision making framework, which was originally proposed in (Taheri et al., 2023) and also was briefly explained in (Hayashi et al., 2023). The framework treats various ethical norms such as data minimality, data sensitivity, gender fairness, racial fairness, and even system performance. In addition, it provides a mechanism for organizing such norms into multiple norm classes (that can be hierarchically structured). For example, data minimality and data sensitivity can be classified in a norm class called privacy. There may be other norm classes such as fairness and performance.

Norms are represented as a tuple of norm classes $N = \langle N_1, N_2, \ldots, N_l \rangle$, where l is some positive integer, and each N_i indicates a norm class consisting of norms. Each norm is used to rank *alternatives* (or choices to be compared in decision making) in a given set A. A norm class aggregates the results of the rankings of the norms in the class by using a voting mechanism. The framework especially chooses Copeland's rule (Saari and Merlin, 1996) for this purpose. Intuitively, Copeland's rule computes the Copeland score $s(a, N_i)$ of an alternative $a \in A$ according to a norm class N_i. A higher Copeland score of a indicates that a is better.

The framework selects the best alternatives by using a relation called *superiority* for comparing alternatives according to norm classes with an order, which is defined as follows.[1]

Definition 1 (superiority)**.** Given alternatives $a, a' \in A$, a tuple $N = \langle N_1, N_2, \ldots, N_l \rangle$ of norm classes, and a partial order $<_N$ on $\{1, \ldots, l\}$, the superiority of a over a' according to N with $<_N$ is defined as

$$\begin{aligned}&\mathsf{superior}(a, a', N, <_N) \equiv \\ &\quad \exists k \in \{1, \ldots, l\} (\forall i \in \{1, \ldots, l\} \\ &\qquad (i <_N k \rightarrow s(a, N_i) = s(a', N_i)) \wedge \\ &\qquad s(a, N_k) > s(a', N_k)).\end{aligned}$$

3.2 Constraint Hierarchies

Next, we briefly explain Borning et al.'s constraint hierarchy framework (Borning et al., 1992). Let X be a set of variables. How to assign values to variables is expressed as a *valuation* that is a function from variables to their values. Let Θ be the set of all the valuations. Then a valuation $\theta \in \Theta$ obtains the value of a variable $x \in X$ as $\theta(x)$. Let C be a set of constraints. How much a constraint is satisfied by a valuation is given by an *error function* e. Specifically, $e(c, \theta)$ returns a non-negative real number: returning 0 means that c is exactly satisfied by θ; returning a larger number means that c is less satisfied.

A constraint hierarchy is typically represented as $H = \langle H_0, H_1, \ldots, H_l \rangle$, where l is some positive integer, and each H_i, called a *level*, is a set of constraints. Level H_0 consists of *required* (or hard) constraints that must be exactly satisfied while each H_i with $i \geq 1$ consists of *preferential* (or soft) constraints that can be relaxed if necessary. A typical constraint hierar-

[1] We simplified the formulation of the superiority relation by using a partial order and a standard definition of lexicographic ordering. The original framework (Taheri et al., 2023) gives a more complex formulation of this relation by using a total preorder.

ICAART2024—https://icaart.scitevents.org/?y=2024
This is the author's version. The final authenticated version is available online at https://doi.org/10.5220/0012474100003636.

chy is totally ordered, which means that a preferential level H_i with smaller i consists of more important constraints.

In this paper, we treat *partially ordered hierarchies*. A partially ordered hierarchy is represented as $\langle H, <_H \rangle$, where $H = \langle H_0, H_1, \ldots, H_l \rangle$, and $<_H$ is a partial order on $\{0, 1, \ldots, l\}$ with 0 as the only smallest element. The intuitive meaning of the required level H_0 is the same as in the case of totally ordered hierarchies. By contrast, the importance of the preferential levels is specified by partial order $<_H$; that is, if $i <_H j$, H_i has more important constraints than H_j.

To define solutions to a partially ordered hierarchy, the notion of *consistent totally ordered hierarchies* is used. Given a partially ordered hierarchy $\langle H, <_H \rangle$, $\langle H, <'_H \rangle$ is a consistent totally ordered hierarchy if $<'_H$ is a total order on $\{0, 1, \ldots, l\}$ with 0 as the smallest element and there is a bijective mapping m of type $\{0, 1, \ldots, l\} \to \{0, 1, \ldots, l\}$ such that $i <_H j \to m(i) <'_H m(j)$.[2] Intuitively, a consistent totally ordered hierarchy is a modification of a partially ordered hierarchy that rearranges preferential levels in a total order in such a way that any pair of ordered levels in the original partially ordered hierarchy will keep the same order in the resulting totally ordered hierarchy.

The importance of constraints in a constraint hierarchy is evaluated by a *comparator* better. Intuitively, $\mathsf{better}(\theta, \theta', S_0, \langle H, <'_H \rangle)$ judges whether a valuation θ is better than another θ' with respect to alternative valuations S_0 according to a totally ordered hierarchy $\langle H, <'_H \rangle$ that is consistent with a partially ordered hierarchy $\langle H, <_H \rangle$. It should be noted that the better comparator internally uses the error function e to evaluate individual constraints.

The solution set of a partially ordered hierarchy is defined as the union of the solution sets of all the totally ordered constraint hierarchies consistent with the original partially ordered hierarchy.

Definition 2 (solution)**.** Given a partially ordered hierarchy $\langle H, <_H \rangle$, the set $S_{\mathsf{po}}(\langle H, <_H \rangle)$ of all the solutions to $\langle H, <_H \rangle$ is defined as

$$S_{\mathsf{po}}(\langle H, <_H \rangle) = \bigcup_{\langle H, <'_H \rangle \in \mathcal{H}} S_{\mathsf{to}}(\langle H, <'_H \rangle)$$

where $\mathcal{H}$ is the set of all the totally ordered hierarchies consistent with $\langle H, <_H \rangle$ and

$$S_{\mathsf{to}}(\langle H, <'_H \rangle) = \{\theta \in S_0 \mid \forall \theta' \in S_0 \; (\neg \mathsf{better}(\theta', \theta, S_0, \langle H, <'_H \rangle))\}$$

where

$$S_0 = \{\theta \in \Theta \mid \forall c \in H_0 (e(c, \theta) = 0)\}.$$

In this definition, S_0 indicates the set of all the valuations that satisfy the required constraints. Also, $S_{\mathsf{to}}(\langle H, <'_H \rangle)$ indicates the set of all the solutions to a totally ordered hierarchy $\langle H, <'_H \rangle$, which is determined by collecting the "best" valuations from S_0.

4 PROPOSED FRAMEWORK

In this section, we propose a soft constraint-based framework for ethical reasoning. To begin with, we introduce a new kind of constraints called *norm constraints*. Intuitively, a norm constraint evaluates a given valuation according to the associated norm and assigns a rank to the valuation. Let C_{norm} be a set of norm constraints. Then the meaning of norm constraints is defined with a *ranking function* r. Intuitively, $r(c, \theta)$ obtains the rank of a valuation θ according to a norm constraint c. Ranks are expressed with positive integers, and a smaller positive integer indicates a better rank, typically with 1 as the best. This is formally defined as follows.

Definition 3 (ranking function)**.** Let $\mathbb{Z}^+$ be the set of all the positive integers. A ranking function r is a function of type $C_{\text{norm}} \times \Theta \to \mathbb{Z}^+$.

We use the notion of partially ordered hierarchies, which we explained in Subsection 3.2. Therefore, we define a constraint hierarchy as a pair $\langle H, <_H \rangle$, where $H = \langle H_0, H_1, \ldots, H_l \rangle$ and $<_H$ is a partial order on $\{0, 1, \ldots, l\}$ with 0 as the only smallest element. We assume that the required level H_0 consists of only ordinary constraints and that the other levels $H_1, \ldots, H_l$, called the *norm levels*, consist of only norm constraints

To evaluate how much a valuation satisfies a norm level, we use a *combining function* g that computes the Copeland score of the valuation according to the norm level. Intuitively, $g(\theta, S_0, H_i)$ computes the Copeland score of a valuation θ with respective to alternative valuations S_0 according to a norm level H_i. This is defined as follows.

Definition 4 (combining function)**.** Given a valuation θ, a set S_0 of valuations with θ (i.e., $\theta \in S_0$), and a set H_i of norm constraints, the combining function g is defined as

$$g(\theta, S_0, H_i) = \sum_{\theta' \in S_0 \setminus \{\theta\}} p(\theta, \theta', H_i)$$

[2] We simplified the definition of consistent totally ordered hierarchies by removing a possible operation of merging levels. This operation may impose a problem, which is discussed in (Borning et al., 1992). Chiu and Lee also discuss the problem and present a more complex definition of solutions (Chiu and Lee, 1998).

ICAART2024—https://icaart.scitevents.org/?y=2024
This is the author's version. The final authenticated version is available online at https://doi.org/10.5220/0012474100003636.

where

$$p(\theta,\theta',H_i) = \begin{cases} 1 & \text{if } w(\theta,\theta',H_i) > w(\theta',\theta,H_i) \\ 1/2 & \text{if } w(\theta,\theta',H_i) = w(\theta',\theta,H_i) \\ 0 & \text{otherwise} \end{cases}$$

where

$$w(\theta,\theta',H_i) = \left|\{c \in H_i \,|\, r(c,\theta) < r(c,\theta')\}\right|.$$

This definition is based on an adaptation of Copeland scores to our framework. Function $w(\theta,\theta',H_i)$ uses the ranking function r to compute the number of the wins of θ against θ' according to the norm constraints in H_i. Function $p(\theta,\theta',H_i)$ computes the elemental score of θ against θ' according to H_i. Finally, function $g(\theta,S_0,H_i)$ accumulates the elemental scores to compute the Copeland score of θ according to H_i

Next, we define a comparator better for our framework. Although we treat a partially ordered hierarchy, we define it for a totally ordered hierarchy by following the original framework.

Definition 5 (comparator)**.** Given a constraint hierarchy $\langle H, <_H\rangle$, valuations θ and θ', a set S_0 of valuations with θ and θ' (i.e., $\theta,\theta' \in S_0$), and a total order $<'_H$ consistent with $<_H$, the comparator better is defined as

$$\begin{aligned} &\mathsf{better}(\theta,\theta',S_0,\langle H,<'_H\rangle) \equiv \\ &\quad \exists k \in \{1,\ldots,l\}(\forall i \in \{1,\ldots,l\} \\ &\qquad (i <'_H k \rightarrow g(\theta,S_0,H_i) = g(\theta',S_0,H_i)) \wedge \\ &\qquad g(\theta,S_0,H_k) > g(\theta',S_0,H_k)). \end{aligned}$$

This comparator is defined as the lexicographic ordering using the total order and the Copeland scores of norm levels. The nature of the lexicographic ordering more highly evaluates a valuation if it obtains a higher Copeland score at an upper level.

We define solutions to a constraint hierarchy in the same way as solutions to a partially ordered hierarchy in the original framework. Specifically, the set of all the solutions to a given constraint hierarchy $\langle H, <_H\rangle$ is $S_{\mathsf{po}}(\langle H, <_H\rangle)$, which is formulated by Definition 2.

Finally, we present a theorem that relates our framework with Taheri et al.'s ethical decision making framework. It allows us to regard our framework as an extension of Taheri et al.'s framework.

Theorem 1. *Let A be an arbitrary set of alternatives, $N = \langle N_1, N_2, \ldots, N_l\rangle$ be an arbitrary tuple of norm classes, and $<_N$ be an arbitrary partial order on $\{1,\ldots,l\}$. Let $X = \{x\}$ be the set of a variable x, and Θ be the set of all the valuations from X to A. Regard the norms in N as norm constraints and the ranking of alternatives in A by norms in N as a ranking function r. Let $H = \langle \emptyset, N_1, N_2, \ldots, N_l\rangle$ be a constraint hierarchy. Then the following holds:*

$$\begin{aligned} S_{\mathsf{po}}(\langle H, <_H\rangle) = \{\theta \in \Theta \,|\, &\theta(x) = a \wedge \\ &\forall a' \in A(\neg \mathsf{superior}(a', a, N, <_N))\}. \end{aligned}$$

Intuitively, this theorem says that the constraint hierarchy constructed from an ethical decision making problem based on Taheri et al.'s framework obtains the same set of solutions as the original problem. Note that the resulting constraint hierarchy has only one variable and no required constraints.

5 CASE STUDY

In this section, we conduct a case study on the application of the proposed framework to ethical reasoning.

5.1 Modeling a Problem

This case study uses a problem adapted from Hayashi et al.'s use case of their multi-agent real-time compliance mechanism (Hayashi et al., 2023). It supposes a job recommendation service operated by an employment platform company and performs legal and ethical reasoning. They implement this service as a distributed system consisting of multiple nodes in different regions, some of which are inside the European Union (EU) and the others of which are outside. They want to process their customers' data by using a remote processing node. Although both their user node and the remote processing node are inside the EU, some of the intermediate nodes between these nodes are outside the EU. Since the data include the customers' privacy information, they need to treat the data by following legal rules such as the EU's General Data Protection Regulation (GDPR). In addition, they want to treat the data by respecting ethical norms such as privacy and fairness.

This use case is represented as a simplified problem illustrated in Figure 1. The company has two datasets, data1 and data2, at the user node. While they can use data1 for two purposes, recommendation and analysis, they are legally allowed to use data2 only for analysis. The remote processing nodes can execute two kinds of processes, process1 and process2, which have different characteristics. The user node and the processing node are connected along two routes, one with node1 and the other with node2 as an intermediate node. While node1 is inside the EU, node2 is outside the EU. When they process a dataset, it is transferred from the user node to the processing node via either node1

ICAART2024—https://icaart.scitevents.org/?y=2024
This is the author's version. The final authenticated version is available online at https://doi.org/10.5220/0012474100003636.

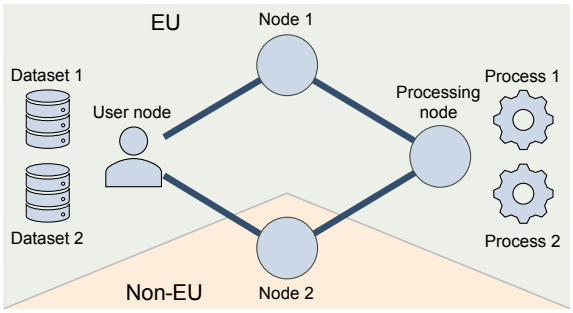

Figure 1: Problem of legal and ethical reasoning for a job recommendation service adapted from Hayashi et al.'s use case of their compliance mechanism (Hayashi et al., 2023).

or node2, then it is processed by either process1 or process2 at the processing node, and the result is returned to the user node via either node1 or node2.

Now we model this problem as a constraint hierarchy by using our framework. First, we introduce five variables with finite domains, $d \in \{\mathsf{data1}, \mathsf{data2}\}$, $u \in \{\mathsf{recommendation}, \mathsf{analysis}\}$, $n, m \in \{\mathsf{node1}, \mathsf{node2}\}$, and $p \in \{\mathsf{process1}, \mathsf{process2}\}$. Variable d indicates which of datasets data1 and data2 is used. Variable u indicates for which of purposes recommendation and analysis the dataset should be used. Variables n and m indicate which of intermediate nodes node1 and node2 should be used when the dataset is transferred to the processing node and when it is returned to the user node respectively. Variable p indicates which of processes process1 and process2 should be used when the dataset is processed at the processing node.

Next, to express the legal rule, we introduce the following required constraint:

$$d = \mathsf{data2} \rightarrow u = \mathsf{analysis}. \tag{1}$$

This constraint means that data2 is legally allowed to be used only for analysis.

In the following, we associate norm constraints with symbolic norms to distinguish how important they are. Specifically, we have six symbolic norms, data_minimality, data_sensitivity, transfer_safety, node_safety, algo_unbiasedness, and transfer_efficiency. More generally, norms data_minimality, data_sensitivity, transfer_safety, and node_safety are related to the ethical norm of privacy, algo_unbiasedness (for algorithmic unbiasedness) is related to fairness, and transfer_efficiency is related to performance. (In the next subsection, we consider such a classification of the norms.)

We represent each norm constraint as a sequence of more primitive constraints separated with either relation of $\triangleright$ or $\circ$. Relation $\triangleright$ indicates that satisfying its left-hand side is better than satisfying its right-hand side, and relation $\circ$ indicates that satisfying its left-hand side is as good as satisfying its right-hand side in the sense of the associated norm.

To represent ethical norms in this problem, we introduce the following norm constraints:

$$\mathsf{data\_minimality}\ (d = \mathsf{data1}) \triangleright (d = \mathsf{data2}) \tag{2}$$

$$\mathsf{data\_sensitivity}\ (d = \mathsf{data1}) \triangleright (d = \mathsf{data2}) \tag{3}$$

$$\begin{aligned} \mathsf{transfer\_safety}\ & (n = \mathsf{node1} \wedge m = \mathsf{node1}) \triangleright \\ & (n = \mathsf{node1} \wedge m = \mathsf{node2}) \circ \\ & (n = \mathsf{node2} \wedge m = \mathsf{node1}) \triangleright \\ & (n = \mathsf{node2} \wedge m = \mathsf{node2}) \end{aligned} \tag{4}$$

$$\begin{aligned} \mathsf{node\_safety}\ & (n = \mathsf{node1} \wedge m = \mathsf{node1}) \triangleright \\ & (n = \mathsf{node1} \wedge m = \mathsf{node2}) \circ \\ & (n = \mathsf{node2} \wedge m = \mathsf{node1}) \triangleright \\ & (n = \mathsf{node2} \wedge m = \mathsf{node2}) \end{aligned} \tag{5}$$

$$\begin{aligned} \mathsf{algo\_unbiasedness}\ & (p = \mathsf{process1}) \triangleright \\ & (p = \mathsf{process2}) \end{aligned} \tag{6}$$

$$\begin{aligned} \mathsf{transfer\_efficiency}\ & (n = \mathsf{node2} \wedge m = \mathsf{node2}) \triangleright \\ & (n = \mathsf{node1} \wedge m = \mathsf{node2}) \circ \\ & (n = \mathsf{node2} \wedge m = \mathsf{node1}) \triangleright \\ & (n = \mathsf{node1} \wedge m = \mathsf{node1}). \end{aligned} \tag{7}$$

Intuitively, constraints (2) and (3) mean that, in the senses of data_minimality and data_sensitivity respectively, using data1 is better than using data2. Constraints (4) and (5) mean that, in the senses of transfer_safety and node_safety respectively, using node1 twice is the best, using node2 only once is the second best, and using node2 twice is the worst. Constraint (6) means that, in the sense of algo_unbiasedness, using process1 is better than using process2. Constraint (7) means that, in the sense of transfer_efficiency, using node2 twice is the best, using node1 only once is the second best, and using node1 twice is the worst.

This problem especially considers the case where a dataset is used for the purpose of recommendation, which is expressed by further introducing the following required constraint:

$$u = \mathsf{recommendation}. \tag{8}$$

Now the entire problem has nearly been modeled as a constraint hierarchy consisting of two required constraints (1) and (8) and six norm constraints (2) to (7) although norm levels are yet to be introduced.

5.2 Solving the Problem

We solve the problem modeled in the previous subsection.[3] First, we satisfy the required constraints

[3]In this paper, we manually solve the problem by following the formulation of our framework, without introducing a concrete algorithm.

ICAART2024—https://icaart.scitevents.org/?y=2024
This is the author's version. The final authenticated version is available online at https://doi.org/10.5220/0012474100003636.

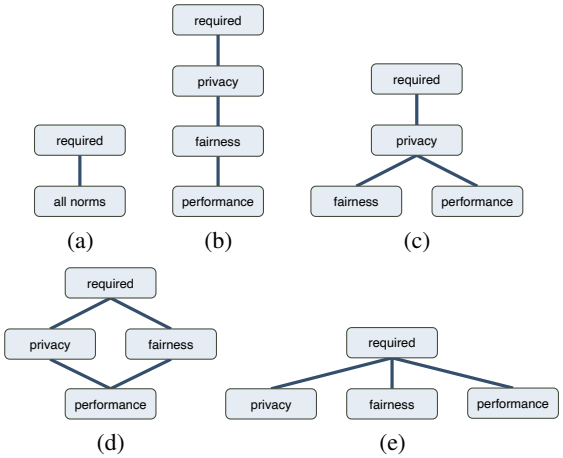

Figure 2: Structures of constraint hierarchies.

(1) and (8). Constraint (8) uniquely determines u as recommendation, which further determines d as data1 because of constraint (1) and the domain of d. There are still eight potential solutions that can assign different combinations of values to variables n, m, and p. Their values need to be determined by further considering the norm constraints.

To show the effects of our framework, we consider different structures of constraint hierarchies. First, we suppose the simplest case where all the six norms belong to one norm level as shown in Figure 2(a). In this case, all the norms are aggregated to compare the potential solutions that satisfy the required constraints, which determines the solutions to the constraint hierarchy. Table 1 shows this process. Initially, each potential solution is associated with the six ranks that are indicated as norm-wise ranks in this table. For example, potential solution $\langle d,u,n,m,p\rangle = \langle$data1, recommendation, node1, node1, process1$\rangle$ has the first rank for norm transfer_safety since assigning node1 to both n and m is specified as the best by the norm constraint (4) associated with transfer_safety. Next, for each potential solution, its Copeland score is computed from their norm-wise ranks. Finally, $\langle d,u,n,m,p\rangle = \langle$data1, recommendation, node1, node1, process1$\rangle$ is determined as only the solution to the constraint hierarchy since it was given the highest Copeland score of 7.0. Intuitively, this solution means that data1 should be used for recommendation, should be transferred to the processing node via node1, should be processed by process1, and should be returned to the user node via node1. Note that this solution is given the first norm-wise ranks by the five norms other than transfer_efficiency.

Next, we consider a case where there are three norm levels, privacy, fairness, and performance. In this case, level privacy consists of four norms

Table 1: Case where all the six norms belong to one norm level. Norms data_minimality, data_sensitivity, transfer_safety, node_safety, algo_unbiasedness, and transfer_efficiency and variable values data1, recommendation, node1, node2, process1, and process2 are abbreviated as dm, ds, ts, ns, au, te, d1, rec, n1, n2, p1, and p2 respectively.

Potential solution	Norm-wise ranks						Copeland score	Level-wise rank
$\langle d,\ u,\ n,\ m,\ p\rangle$	dm	ds	ts	ns	au	te		
⟨d1, rec, n1, n1, p1⟩	1	1	1	1	1	3	7.0	**1**
⟨d1, rec, n1, n2, p1⟩	1	1	2	2	1	2	5.0	2
⟨d1, rec, n2, n1, p1⟩	1	1	2	2	1	2	5.0	2
⟨d1, rec, n2, n2, p1⟩	1	1	3	3	1	1	2.5	4
⟨d1, rec, n1, n1, p2⟩	1	1	1	1	2	3	4.5	3
⟨d1, rec, n1, n2, p2⟩	1	1	2	2	2	2	2.0	5
⟨d1, rec, n2, n1, p2⟩	1	1	2	2	2	2	2.0	5
⟨d1, rec, n2, n2, p2⟩	1	1	3	3	2	1	0.0	6

Table 2: Case where four out of the six norms belong to norm level privacy. The norms and the variable values are abbreviated in the same way as in Table 1.

Potential solution	Norm-wise ranks				Copeland score	Level-wise rank
$\langle d,\ u,\ n,\ m,\ p\rangle$	dm	ds	ts	ns		
⟨d1, rec, n1, n1, p1⟩	1	1	1	1	6.5	**1**
⟨d1, rec, n1, n2, p1⟩	1	1	2	2	3.5	2
⟨d1, rec, n2, n1, p1⟩	1	1	2	2	3.5	2
⟨d1, rec, n2, n2, p1⟩	1	1	3	3	0.5	3
⟨d1, rec, n1, n1, p2⟩	1	1	1	1	6.5	**1**
⟨d1, rec, n1, n2, p2⟩	1	1	2	2	3.5	2
⟨d1, rec, n2, n1, p2⟩	1	1	2	2	3.5	2
⟨d1, rec, n2, n2, p2⟩	1	1	3	3	0.5	3

data_minimality, data_sensitivity, transfer_safety, and node_safety, level fairness consists of one norm algo_unbiasedness, and level performance consists of the other one norm transfer_efficiency. Since levels privacy and fairness consist of only one norm, their level-wise rankings of the potential solutions are the same as the underlying norm-wise rankings due to the nature of Copeland's rule. For level privacy, we need to compute its level-wise ranking, which is shown in Table 2. In the same way as the previous case, the four norm-wise ranks are aggregated to calculate their Copeland scores, which separates the potential solutions into three ranks. Especially, note that two potential solutions $\langle d,u,n,m,p\rangle = \langle$data1, recommendation, node1, node1, process1$\rangle$, $\langle$data1, recommendation, node1, node1, process2$\rangle$ are given the first level-wise rank in this case.

To handle the three norm levels, we further consider their partial orders. It should be noted that, in the typical use of a constraint hierarchy, such a partial order is determined at its modeling stage. However, in the following, we show all the possible partial orders to illustrate our framework. Since there are three norm levels, we can consider the

ICAART2024—https://icaart.scitevents.org/?y=2024
This is the author's version. The final authenticated version is available online at https://doi.org/10.5220/0012474100003636.

following three subcases: (A) they are totally ordered, e.g., privacy $<_H$ fairness $<_H$ performance as shown in Figure 2(b); (B) two of them are incomparable, e.g., privacy $<_H$ fairness $\wedge$ privacy $<_H$ performance with incomparable norms fairness and performance as shown in Figure 2(c), which is denoted as privacy $<_H \{\text{fairness}, \text{performance}\}$ below, and also, e.g., privacy $<_H$ performance $\wedge$ fairness $<_H$ performance with incomparable norms privacy and fairness as shown in Figure 2(d), which is denoted as $\{\text{privacy}, \text{fairness}\} <_H$ performance below; (C) all of them are incomparable with each other as shown in Figure 2(e), which is denoted as $\{\text{privacy}, \text{fairness}, \text{performance}\}$ below.

We first consider subcase (A). In this subcase, we can determine solutions in the same way as normal, totally ordered constraint hierarchies, which is achieved by using lexicographic ordering. Table 3 shows this process. In this table, each potential solution is associated with three level-wise ranks. For example, potential solution $\langle d,u,n,m,p\rangle = \langle$data1, recommendation, node1, node1, process1$\rangle$ has the first rank for privacy, the first rank for fairness, and the third rank for performance. The final ranks of the potential solutions are determined as the "lexicographic ranks" according to the used order of the norm levels. For example, for order privacy $<_H$ fairness $<_H$ performance, potential solution $\langle d,u,n,m,p\rangle = \langle$data1, recommendation, node1, node1, process1$\rangle$ is given the first lexicographic rank since its level-wise ranks 1, 1, and 3 (in the order of the norm levels) is lexicographically better than those of the other potential solutions. Such potential solutions given the first ranks are finally determined as the solutions to the constraint hierarchy according to the associated order of the norm levels. Note that different lexicographic ranks can be obtained for different orders of norm levels.

Finally, we consider subcases (B) and (C). In these subcases, we determine solutions by using the notion of partially ordered hierarchies. Table 4 shows this process. In this table, each partially ordered hierarchy is associated with consistent totally ordered hierarchies. For example, partially ordered hierarchy privacy $<_H \{\text{fairness}, \text{performance}\}$ have two consistent totally ordered hierarchies, privacy $<_H$ fairness $<_H$ performance and privacy $<_H$ performance $<_H$ fairness. Solutions to a partially ordered hierarchies are obtained by collecting the solutions to all the consistent totally ordered hierarchies. For example, partially ordered hierarchy privacy $<_H \{\text{fairness}, \text{performance}\}$ has only one solution $\langle d,u,n,m,p\rangle = \langle$data1, recommendation, node1, node1, process1$\rangle$ since it is only the solution to both of its consistent totally ordered hierarchies. Note that different solutions can be obtained for different partially ordered hierarchies. Also, note that subcase (C) (shown in Figure 2(e)) obtained the different solution set (with two solutions) from that of the simplest case (shown in Figure 2(a)) of having all the six norms in one level (with only one solution).

6 DISCUSSION

In our framework, we treated only norm constraints at preferential levels of constraint hierarchies. However, we can modify our framework to allow other kinds of constraints at preferential levels. This can be realized by separating different kinds of constraints into different levels, as already suggested for the purpose of user interface applications (Hosobe et al., 1994). For example, we can use ordinary constraints at stronger levels and norm constraints at weaker levels.

We did not present a concrete algorithm for solving constraint hierarchies based on our framework. For this purpose, we think that an SMT- or SAT-based approach will be promising. In this approach, a constraint hierarchy is first encoded as an SMT or SAT problem and then is solved by an external SMT or SAT solver. Especially, the hill climbing method (Hosobe and Satoh, 2022) is plausible because it is simple and applicable to various constraint hierarchies. Also, it might be possible to use binary search-based methods (Hosobe and Satoh, 2023) to construct a more efficient algorithm although it will be more difficult and challenging.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a soft constraint-based framework for ethical reasoning. By devising the notion of norm constraints, we integrated Taheri et al.'s ethical decision making framework into Borning et al.'s constraint hierarchy framework. We also presented a case study on the application of our framework to ethical reasoning.

Our future work includes the development of a constraint solver based on our framework. Another direction is to integrate our framework into hierarchical constraint logic programming (Wilson and Borning, 1993) to enable more expressive modeling of ethical reasoning problems. It is also necessary to evaluate the usefulness of our framework by applying it to more practical problems.

ICAART2024—https://icaart.scitevents.org/?y=2024
This is the author's version. The final authenticated version is available online at https://doi.org/10.5220/0012474100003636.

Table 3: Subcase (A) where the three norm levels are totally ordered. Levels privacy, fairness, and performance are abbreviated as pv, fr, and pf respectively, and the variable values are abbreviated in the same way as in Table 1.

Potential solution					Level-wise ranks			Lexicographic ranks					
$\langle d,$	$u,$	$n,$	$m,$	$p\rangle$	pv	fr	pf	pv $<_H$ fr $<_H$ pf	pv $<_H$ pf $<_H$ fr	fr $<_H$ pv $<_H$ pf	fr $<_H$ pf $<_H$ pv	pf $<_H$ pv $<_H$ fr	pf $<_H$ fr $<_H$ pv
⟨d1,	rec,	n1,	n1,	p1⟩	1	1	3	**1**	**1**	**1**	3	5	5
⟨d1,	rec,	n1,	n2,	p1⟩	2	1	2	3	3	2	2	3	3
⟨d1,	rec,	n2,	n1,	p1⟩	2	1	2	3	3	2	2	3	3
⟨d1,	rec,	n2,	n2,	p1⟩	3	1	1	5	5	3	**1**	**1**	**1**
⟨d1,	rec,	n1,	n1,	p2⟩	1	2	3	2	2	4	6	6	6
⟨d1,	rec,	n1,	n2,	p2⟩	2	2	2	4	4	5	5	4	4
⟨d1,	rec,	n2,	n1,	p2⟩	2	2	2	4	4	5	5	4	4
⟨d1,	rec,	n2,	n2,	p2⟩	3	2	1	6	6	6	4	2	2

Table 4: Subcase (B) where two of the three norm levels are incomparable, and subcase (C) where the three levels are incomparable with each other. The norms and the variable values are abbreviated in the same way as in Table 3.

Partially ordered hierarchy	Consistent totally ordered hierarchies						Solutions
	pv$<_H$fr $<_H$ pf	pv$<_H$pf $<_H$ fr	fr$<_H$pv $<_H$ pf	fr$<_H$pf $<_H$ pv	pf$<_H$pv $<_H$ fr	pf$<_H$fr $<_H$ pv	$\langle d,u,n,m,p\rangle$
pv $<_H$ {fr, pf}	✓	✓					⟨d1, rec, n1, n1, p1⟩
{pv, fr} $<_H$ pf	✓		✓				⟨d1, rec, n1, n1, p1⟩
{pv, pf} $<_H$ fr		✓			✓		⟨d1, rec, n1, n1, p1⟩, ⟨d1, rec, n2, n2, p1⟩
fr $<_H$ {pv, pf}			✓	✓			⟨d1, rec, n1, n1, p1⟩, ⟨d1, rec, n2, n2, p1⟩
{fr, pf} $<_H$ pv				✓		✓	⟨d1, rec, n2, n2, p1⟩
pf $<_H$ {pv, fr}					✓	✓	⟨d1, rec, n2, n2, p1⟩
{pv, fr, pf}	✓	✓	✓	✓	✓	✓	⟨d1, rec, n1, n1, p1⟩, ⟨d1, rec, n2, n2, p1⟩

ACKNOWLEDGEMENT

This work was supported by JST AIP Trilateral AI Research Grant Number JPMJCR20G4.

REFERENCES

Borning, A., Freeman-Benson, B., and Wilson, M. (1992). Constraint hierarchies. *Lisp Symbolic Comput.*, 5(3):223–270.

Chiu, C. K. and Lee, J. H. M. (1998). Extending HCLP with partially ordered hierarchies and composite constraints. *J. Expt. Theor. Artif. Intell.*, 10:5–24.

Eshghi, K. (1988). Abductive planning with event calculus. In *Proc. JICSLP*, pages 562–579.

Fungwacharakorn, W., Tsushima, K., and Satoh, K. (2022a). Debugging constraint hierarchies representing ethical norms with valuation preferences. In *Proc. Workshop on AI Compliance Mechanism (WAICOM)*, pages 114–124.

Fungwacharakorn, W., Tsushima, K., and Satoh, K. (2022b). Fundamental revisions on constraint hierarchies for ethical norms. In *Proc. JURIX*, volume 362 of *FAIA*, pages 182–187.

Hayashi, H., Mitsikas, T., Taheri, Y., Tsushima, K., Schäfermeier, R., Bourgne, G., Ganascia, J.-G., Paschke, A., and Satoh, K. (2023). Multi-agent online planning architecture for real-time compliance. In *RuleML+RR Companion*, volume 3485 of *CEUR WS*.

Hosobe, H., Miyashita, K., Takahashi, S., Matsuoka, S., and Yonezawa, A. (1994). Locally simultaneous constraint satisfaction. In *Proc. PPCP Workshop*, volume 874 of *LNCS*, pages 51–62.

Hosobe, H. and Satoh, K. (2022). Solving constraint hierarchies for hierarchical constraint logic programming. In *Proc. JSAI Conf.*, number 4F3-OS-8b-04, pages 1–4. In Japanese.

Hosobe, H. and Satoh, K. (2023). Binary search-based methods for solving constraint hierarchies over finite domains. In *Proc. IEEE ICTAI*, pages 186–193.

Saari, D. G. and Merlin, V. R. (1996). The Copeland method—I.: Relationships and the dictionary. *Econ. Theory*, 8:51–76.

Satoh, K., Asai, K., Kogawa, T., Kubota, M., Nakamura, M., Nishigai, Y., Shirakawa, K., and Takano, C. (2010). PROLEG: An implementation of the presupposed ultimate fact theory of Japanese civil code by PROLOG technology. In *Proc. JSAI-isAI*, volume 6797 of *LNCS*, pages 153–164.

Satoh, K., Ganascia, J.-G., Bourgne, G., and Paschke, A. (2021). Overview of RECOMP project. In *Proc. Workshop on Computational Machine Ethics (CME)*.

Taheri, Y., Bourgne, G., and Ganascia, J.-G. (2023). Modelling integration of responsible AI values for ethical decision making. In *Proc. Workshop on Computational Machine Ethics (CME)*. https://github.com/yousef-taheri/responsibleAi.

Wilson, M. and Borning, A. (1993). Hierarchical constraint logic programming. *J. Log. Program.*, 16(3–4):227–318.

ICAART2024—https://icaart.scitevents.org/?y=2024
This is the author's version. The final authenticated version is available online at https://doi.org/10.5220/0012474100003636.