

# An Argumentation Inspired Heuristic for Resolving Normative Conflict

Nir Oren, Michael Luck, Simon Miles, Timothy J. Norman

Department of Computer Science, King's College London, UK

**Abstract.** In multi-agent systems, norms provide a means for regulating agent behaviour at a system or society level rather than by constraining agent behaviour directly. In order to cope effectively with scenarios in which norms conflict, agents must be able to reason about norms and the consequences of compliance and violation. In particular, agents should follow the parsimony principle, that is, if an agent must violate a norm (because norms conflict, for example) then it should determine which norm to violate in such a way that enables it to otherwise maximise its compliance with the remaining set of applicable norms. This concept of maximising compliance and minimising violation or conflict is similar to the notion of preferred extensions from argument theory, which provides a potentially valuable way to analyse sets of norms to determine which norms to violate. In this paper, therefore, we map normative structures to argument theory, and show how some resulting heuristics may be applied to minimising normative conflict.

## 1 Introduction

Research on multi-agent systems is concerned with both the internal reasoning an agent must undertake when operating in some environment, and the behaviour that emerges as the agent interacts with other agents in the system. BDI architectures [1] are among the most popular ways of modelling an agent's reasoning. In such frameworks, an agent has a set of *beliefs* encoding its current knowledge about the world, and a set of desired future world states that it would like to see come about. Since an agent cannot commit resources to achieving all of these *desires*, it must select a subset to commit to, its *intentions*. In modelling a single agent, such BDI architectures generally perform well, but they are limited in failing to take into account the societal aspects of multi-agent systems.

For example, if we consider a scenario in which Bob promises to help Alice move her piano, the BDI model allows us to understand and represent beliefs and desires, possibly about others, and even provides a means to represent Bob's commitment to help move the piano. However, it tells us nothing about how such a promise might be enforced, nor how the relationships between agents are affected nor, importantly, how the breaking of a promise might impact on the encompassing society or on Bob himself. In fact, it may be possible, through a very complicated mix of beliefs, desires, and intentions, to encode some of these aspects (in terms of individual impact) within an

agent, but this is *internal* to an agent. The key point is that there is no way to encode the more general rule, at a societal level, that *an agent should attempt to keep its promises*<sup>1</sup>.

In general, such societal interactions in human societies are regulated by *norms*, which represent what *ought* to be done, and provide a means to constrain an agent's behaviour. In human society, such norms include the hard *laws* of a country, as determined by government, courts, and enforced by police, but apply equally to social niceties as well as other cultural and religiously imposed restrictions on, or guides for, behaviour. While such norms are indispensable for the effective operation of any society, one of their key properties, in the real world at least, is that they are not always complied with and are sometimes violated, even if this is undesirable, or not permitted. Thus, norms are societal rules that influence agent behaviour. Unlike beliefs, desires and intentions, norms are societal constructs, rather than individual (although of course, an agent must have the capacity to reason about norms affecting itself).

Computationally, in many multi-agent system frameworks, norms are only implicitly represented, in which case they find manifestation through being hard-wired into all agents, which are unable to deviate from their programmed behaviour. For example, by excluding the ability to cheat in an agent's reasoning procedure, a system containing only such agents implicitly imposes a *no cheating* norm on all its agents. However, such agents are not aware of their norms. Contrast this with a system in which agents are *norm-aware* in that not only are they aware of what they are expected to do, but they may decide not to comply with the norm. Systems containing such norm-aware agents have become the subject of much investigation recently, for several reasons, as follows.

- Norms are aimed at improving overall system behaviour by constraining the behaviour of individual agents within it. In general, this improves the conditions for the majority of agents within the system, but may restrict possibilities for some. However, an unintended beneficial consequence for individuals is that in this way, norms can limit the range of options that agents may need to reason about, potentially reducing computational load.
- Norms are generally declarative. By specifying what should, or should not, be done, rather than how to achieve it, and by allowing agents themselves to determine how and when to comply with (or indeed to violate) norms, autonomous behaviour is preserved for agents.
- Decisions on whether to comply with or violate norms are taken independently by norm-aware agents, which are better able to exploit their environments through violation if individual utility is higher than with compliance.

In this paper, we examine how an agent may reason about, and resolve, normative conflict. Such conflict arises when an agent, by complying with one norm, is unable to comply with one or more of its other norms. Our main contribution lies in the proposal of argumentation theory based heuristics allowing an agent to decide with which norms it should comply. Rather than describing a complex normative model, our work focuses on the introduction of these heuristics within a very simple environment.

---

<sup>1</sup> Attempts have been made to integrate social concepts into BDI frameworks, see [2, 3] for examples.

In the next section, we outline a simple model of norms. Starting from this simple model, we examine normative conflict in more detail, and show how we may transform an agent's norms, as specified by the model, into a form that may be operated upon by our heuristics. We then introduce our heuristics, and provide an empirical evaluation of their effectiveness. Finally, we suggest possible avenues of future work.

## 2 A Model of Norms

### 2.1 Properties of Norms

Many frameworks for norms have been proposed; rather than exhaustively examine them here (the interested reader is referred to [4]), we highlight some key features of the more widely used norm representation methods.

Moses and Tennenholtz's social laws [5] provide one means of including norms in a multi-agent system. Here, the system designer specifies the norms off-line, at design time; that is, the designer creates a set of norms that agents should comply with in advance of the operation of the system. While more flexible than a system without norms, the fixed nature of such social laws means that the system is still somewhat inflexible. If the environment changes in such a way that new norms are required, the system must be rebuilt.

More interesting is the situation in which agents may dynamically agree on a set of norms between them. Here, agents have maximum flexibility to adapt to new situations, at the cost of some complexity. Perhaps the most popular way of generating norms in this way at runtime is through contracts: an agent agrees to comply with the terms of an explicit contractual agreement, drawn up between itself and another agent. Of course, nothing precludes the agent also from being required to comply with norms imposed upon it by its designer, the environment, and other entities within the system.

Many different types of norms (such as obligations, permissions and prohibitions) have been identified. Generally, however, a norm either restricts or relaxes the constraints imposed on an agent's behaviour. While we do not consider them here, norms may be conditional; that is, their effects only come into force if certain preconditions are met. Some norms may also permanently constrain an agent's behaviour, while other norms may stop affecting the agent once they are satisfied. Often, norms are associated with sanctions and rewards that may be applied if a norm is satisfied or violated respectively. We take norms to be intentional; that is, if an agent adopts a norm, we assume it will attempt to comply with it.

Norms may vary in complexity. Compare, for example, the norm "Alice is obliged to pay Bob \$100", to the norm "Anyone supplying mission-critical systems is obliged to exhaustively test them, or will otherwise be liable for any injuries arising from their use". The former norm is unconditional, and may be discharged by a single action from Alice. The latter norm is conditional, mentions additional norms that come into effect when it is violated, and is temporally persistent. In what follows, we do not consider conditional or contrary to duty norms in this paper, and ignore the effects of time.

Finally, norms are typically imposed by some social entity upon an agent. The nature of this entity, and its relation with an agent may affect the way the agent deals with

the norm. We refer to this relation as the *social context* of a norm. For example, the social context of a norm created via a promise to a friend is very different to one created when an employer instructs an employee to do something. The idea of social context is fundamental in resolving normative conflict.

## 2.2 Example

Consider the following situation in which a set of norms is imposed on an agent, Alice, as represented in Table 1.

- a* Alice promised Bob she would go to the theatre with him. She is therefore obliged (with social context “Bob”) to go to the theatre.
- b* Alice promised her sick mother she would go visit her in the hospital. Here, Alice is obliged (with social context “mother”) to go to the hospital.
- c* It’s Alice’s turn to cook dinner for her friends. Alice is thus obliged to achieve the state of affairs “dinner”, with the social context being her friends.
- d* Alice is obliged to write a paper for her boss.

(a) $O_{bob}(theatre)$	(b) $O_{sickMother}(hospital)$
(c) $O_{friends}(cooking)$	(d) $O_{boss}(paper)$

**Table 1.** A sample set of norms imposed on an agent.

Now, suppose that Alice is unable to go both to the hospital and to the theatre, and that, since she has no food at home (and since cooking dinner requires food at home), she should go shopping. However, if Alice goes shopping, she will not have enough time for paper writing. It is clear that Alice is unable to comply with all of her norms, as some of their normative goals are mutually exclusive. Her norms are thus in conflict.

We will refer to this example through the paper to illustrate the various concepts and techniques introduced.

## 2.3 Environments and Agents

We view norms as the states of affairs that an agent is obliged to bring about, permitted to do so, or prohibited from doing so. Syntactically, we may represent a norm that affects a single agent as follows:

$$N_c(g)$$

This may be read as a norm of type  $N$ , (where  $N \in \{O, P, F\}$  is, respectively, either an obligation, permission, or prohibition), is imposed on the agent, where the social context (that is, the social entity to which the agent is responsible for the discharge of the norm) is  $c$ . Such a social context is, most often, an agent, but we do not exclude other possibilities such as roles, groups of agents, or other social structures. The normative

goal associated with the norm is  $g$ . In the case of an obligation or prohibition,  $g$  indicates the state of affairs that the agent should ensure occurs, while in the case of a permission  $g$  indicates an allowable (rather than necessary) state.

In order to build up our model of norms and agents, we must first define our environment, and then our initial normative agent, as follows.

**Definition 1. (Environment)** *An environment is a tuple  $\langle S, C \rangle$  where  $S$  is a set of states of affairs, and  $C$  is a set of social contexts. We assume that the elements of  $S$  (and of  $C$ ) are independent of each other.*

**Definition 2. (Normative Agent)**

*A normative agent,  $A$ , is a pair  $(Norms, \leq)$  where  $Norms$  is a set of norms of the form  $N_c(s)$  with  $N \in \{O, P, F\}$  representing obligations, permissions and prohibitions respectively,  $c \in C$  identifies the social context imposing the norm on the agent, and  $s \in S$  is the normative goal of the norm; and  $\leq$  is a partial ordering over  $C$ , identifying the importance of complying with norms imposed by different social contexts.*

## 2.4 Normative Conflict

While some philosophers argue that normative conflicts do not in fact exist [6], most of their discussions revolve around some sort of ideal ethical system. Since such ideal systems are not common in practise, agents may often find themselves in a situation in which normative conflicts arise. Sources for these conflicting norms may include: interactions between elements of an agent's environment (for example, one societal group may require full disclosure of an agent's internal state, while another may simultaneously oblige the agent to keep its state private); an agent's position in some existing social structure (for example, if an agent is owned by one company and leased to another, certain behaviours may be prohibited by the owning company, but required by the leasing company); and the agent itself (an agent may accept conflicting norms in the hope that the situation in which they are actually instantiated may never arise). Thus, even excluding the final case, an agent may still find that it has had conflicting norms imposed on it.

Since it is clear that an agent may not avoid entering situations in which normative conflict occurs, it must be able to determine how to resolve such conflict. In an abstract sense, dealing with normative conflict involves deciding which norms to set aside when acting. More concretely, an agent may either drop a conflicting norm (in which case the norm will no longer affect its behaviour), or temporarily ignore it (in which case the agent will still intend to comply with the norm at a later stage). For the purposes of this paper, we assume that an agent in this situation has no choice but to drop a norm permanently, and must determine which norms to drop. Due to the intentional nature of norms, we assume that the agent attempts to minimise the number of norms it drops.

In the example of Table 1, inference had to take place to determine that the obligation to cook dinner conflicts with Alice's paper writing obligation. Integrating this into a normative model is outside the scope of this paper, and we thus assume that it takes place in the background.

Given this background, we can assume that an agent must sometimes reason about norms dealing with states of affairs  $S$  that may not occur at the same time or, more

specifically, are mutually exclusive. In order to deal with conflicting norms seeking to bring about mutually exclusive states, we introduce into our model a *mutuallyExclusive* relation, identifying pairs of mutually exclusive states of affairs. Given a pair  $(s, t) \in \text{mutuallyExclusive}$ , an agent is unable to meet both obligations to achieve  $s$  and  $t$ , and such obligations are in normative conflict with each other. In our example,  $(\text{theatre}, \text{hospital})$  is in Alice’s *mutuallyExclusive* relation.

**Definition 3. (Mutual Exclusiveness)** Given an environment  $\langle S, C \rangle$ ,

$$\text{mutuallyExclusive} : S \times S$$

is an irreflexive symmetric binary relation capturing the notion of norms that may not be complied with simultaneously.

## 2.5 Normative Conflict Graph

Now, if we have an agent with conflicting norms (because of states specified that are mutually exclusive), we can represent the set of norms and conflicts as a *normative conflict graph* in which norms are nodes, and edges are conflicts between norms. If such a graph contains no edges, then no normative conflict exists.

An agent may create a *normative conflict graph* from a set of norms and the *mutuallyExclusive* relation. This graph identifies conflicts between the agent’s norms, and is used as the basis for our heuristics. We represent this graph as  $\langle \text{Nodes}, \text{Edges} \rangle$ , where *Edges* is a set of binary relations with signature  $\text{Nodes} \times \text{Nodes}$ . Figure 1 shows how an agent may compute its normative conflict graph. In this algorithm, the agent connects all those norms that are in normative conflict with edges. Lines 6 and 7, for example, cause conflicting obligations to be linked within the graph. As expected of prohibitions, lines 8 and 10 state that a prohibition only causes a normative conflict if there is an obligation or permissions to achieve the same state. By iterating through all of the agent’s norms, the algorithm is guaranteed to generate all possible edges.

Agents may prune some edges from the normative conflict graph. Such pruning may be based on their social context preferences. This is captured by the *social context pruning* algorithm described in Figure 2. This algorithm removes the edge going from a lower priority norm towards a higher priority norm.

Certain agents may perform additional edge-pruning based on preferences over norm types. For example, an agent may decide that when they are obliged and prohibited from doing something, they will do it anyway, meaning that the obligation is, in a sense, preferred by the agent over the prohibition. To capture this concept, we define an extended normative agent as follows:

**Definition 4. (Extended Normative Agent)** An extended normative agent is a normative agent as defined in Definition 2, with an additional strict partial ordering,  $\prec$ , over norm types  $(O, P, F)$ .

$$EA = (\text{Norms}, \leq, \prec)$$

```

Normative Graph Creation
Given a normative agent  $A = (Norms, contexts)$ 
an environment  $E = \langle S, C \rangle$ 
a mutuallyExclusive relation
Define a normative conflict graph  $NCG = \langle Nodes, Edges \rangle$ 

01  $\forall n = X_c g \in Norms$  such that  $X \in \{O, P, F\}$ ,  $c \in C$  is
02 a social context and  $g \in S$  a state of affairs
03   add  $n$  as a node to  $NCG$ 
04    $\forall m \in Norms$  where  $m = Y_d h$ ,  $Y \in \{O, P, F\}$ ,  $d \in C$  is
05   a social context, and  $h \in S$  a state of affairs.
06     if  $X = O$  and  $Y = O$ ,  $(g, h) \in mutuallyExclusive$ 
07       add an edge  $(n, m)$  to  $NCG$ 
08     if  $(X = O$  or  $X = P)$  and  $Y = F$ ,  $g \equiv h$ 
09       add an edge  $(n, m)$  to  $NCG$ 
10     if  $X = F$  and  $(Y = O$  or  $Y = P)$ ,  $g \equiv h$ 
11       add an edge  $(n, m)$  to  $NCG$ 
12     if  $X = P$  and  $Y = O$ ,  $(g, h) \in M$ 
13       add an edge  $(n, m)$  to  $NCG$ 
14     if  $X = O$  and  $Y = P$ ,  $(g, h) \in M$ 
15       add an edge  $(n, m)$  to  $NCG$ 
16 return  $NCG$ 

```

**Fig. 1.** The algorithm used to create the normative conflict graph for an agent.

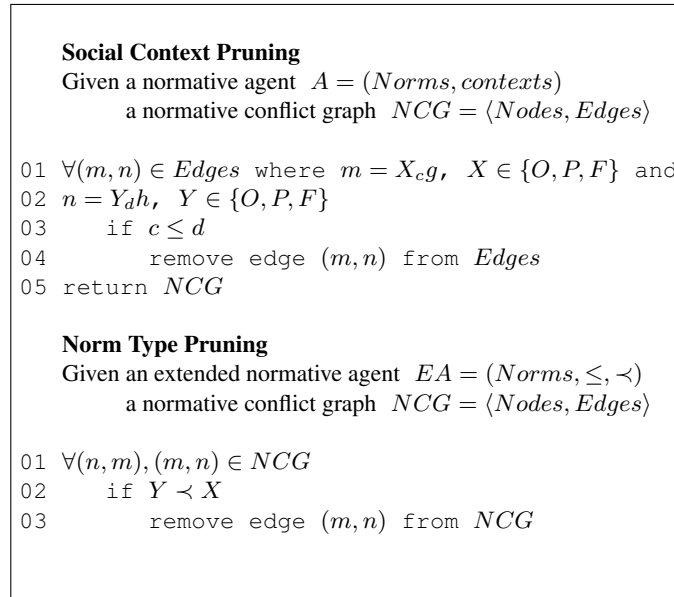
Given this  $\prec$  relation, an agent may further prune the normative conflict graph according to the *norm type pruning* algorithm defined in Figure 2.

Returning to the example of Table 1, suppose we add a new norm, labelled (e), as  $P_{superior}(delayPaper)$ , representing permission from Alice's boss's superior to delay the paper. Now, assume that Alice can't decide whether visiting her mother is more important than going to the theatre with Bob, but believes that going to the theatre with him is more important than cooking dinner for her friends. In addition, her boss's paper is more important than any of these. However, since she doesn't want to get in trouble with either her boss, or his superior, she is unable to rank either social context. This leads to Alice having the following priorities over social contexts:

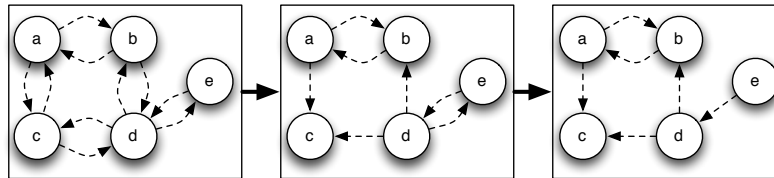
$$\begin{array}{ll}
 friends < bob & bob < boss \\
 sickMother < boss & friends < boss
 \end{array}$$

We further assume that many of the possible states of affairs are mutually exclusive, leading to the following pairs appearing within *mutuallyExclusive*:

$$\begin{array}{lll}
 (theatre, hospital) & (hospital, theatre) & (cooking, paper) \\
 (paper, delay) & (cooking, theatre) & (paper, cooking) \\
 (hospital, paper) & (theatre, cooking) & (delay, paper) \\
 (paper, hospital) & &
 \end{array}$$



**Fig. 2.** Pruning Algorithms for a normative conflict graph.

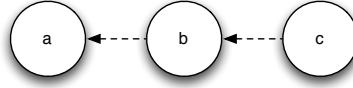


**Fig. 3.** Normative conflict graphs resulting from the norms in Table 1.

Finally, we assume that Alice prefers to comply with permissions rather than obligations where possible, i.e. that  $O \prec P$ . Figure 3 shows all stages of normative conflict graph creation. The leftmost graph contains all conflicts found in the system, while the centre graph is the result of pruning by social context preferences. The final, rightmost graph illustrates the effects of pruning by priorities over norm types.

### 3 Resolving Normative Conflict

Ideally, agents should attempt to maximise the number of norms they comply with, subject to their social context and preferences over norm types. As mentioned previously, a normative conflict graph containing no edges also contains no normative conflicts. By representing norms in such a graph, and then removing nodes (and their associated edges) until no normative conflicts exist, we can identify a set of non-conflicting



**Fig. 4.** A simple normative conflict graph illustrating the random drop heuristic.

norms. However, the important question identified in Section 2.4, of determining which norms to drop, requires further consideration. In this section, we consider three separate heuristics for node removal, and therefore for norm removal.

### 3.1 The Random Drop Heuristic

The simplest possible method of dropping norms (and nodes) is random. We introduce this to provide a baseline strategy against which to compare.

**Definition 5. (Random Drop Heuristic)** *Given a normative conflict graph, select an edge (i.e. a normative conflict) at random. If this edge is labelled  $(n, m)$ , node  $m$  (and all edges containing it) are removed from the graph. This process is repeated until no edges remain in the graph.*

By dropping nodes in this manner, we effectively always drop the lower priority norm from a conflicting pair, until no more normative conflicts remain. While guaranteeing an edge free normative conflict graph, this heuristic often drops more nodes than are strictly necessary. For example, consider the graph  $\langle \{a, b, c\}, \{(b, a), (c, b)\} \rangle$ , shown in Figure 4. The random drop heuristic might first choose to examine edge  $(b, a)$ , so that node  $a$  is dropped. After this, node  $b$  must still be dropped to create an edge free graph. If edge  $(c, b)$  was selected first, and node  $b$  was dropped, the resulting edge free graph contains two nodes, rather than one as in the former case.

### 3.2 Argumentation and the Maximal Conflict-free Set Heuristic

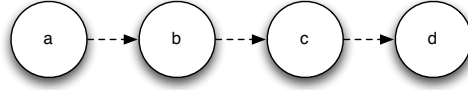
Many similarities exist between the normative conflict graph and the argument systems found in Dung’s abstract argument framework [7]. In the latter, an argument system may be represented as a graph, with attacks between arguments forming edges in the graph. Dung was interested in studying the interactions between arguments in this very abstract setting, and identified which sets of arguments a rational reasoner would find *consistent* in various situations. These consistent arguments are (at the simplest level) those that do not attack each other, much like the consistent norms in our normative conflict graph. Thus in this section we propose to apply argumentation theory to find consistent (or non-conflicting) norms.

Formally, Dung defines an argument system as follows:

**Definition 6. (Argument System)** *An argument system is a pair*

$$AS = \langle AR, attacks \rangle$$

*where  $AR$  is a set of arguments, and  $attacks$  is a binary relation on  $AR$ .*



**Fig. 5.** A simple normative conflict graph illustrating the maximal conflict-free set heuristic.

Note that for convenience, if  $(A, B) \in attacks$ , we may write  $attacks(A, B)$ . Then, by equating normative conflict with attack, we can represent the rightmost normative conflict graph of Figure 3 as the following argument system.

$$\langle \{a, b, c, d, e\}, \{(a, b), (b, a), (a, c), (d, c), (c, b), (e, d)\} \rangle$$

This is identical to the formal representation of the normative conflict graph.

Now, we can define the notion of attack for a set: a set  $S$  attacks an argument  $A$  if some argument in  $S$  attacks  $A$ . Clearly, any set containing arguments attacking each other is, in some sense, in conflict. In contrast, we can define *conflict-free* as follows.

**Definition 7. (Conflict Free)** A set of arguments  $S$  is conflict free iff  $\nexists A, B \in S$  such that  $attacks(A, B)$ .

Clearly, in the example, sets such as  $\{a\}$ ,  $\{b\}$ , and  $\{a, d\}$  are conflict free. By converting our normative conflict graph to an argument system (turning nodes into arguments, and conflicts into attacks), we are able to define our second heuristic.

**Definition 8. (Maximal Conflict-free Set Heuristic)** Select the norms found in the maximal (with respect to number of norms) conflict free set, as computed from the normative conflict graph. If multiple such sets exist, select one at random.

Choosing from among several equally sized maximal conflict free sets is problematic, as such sets do not consider priorities between norms, and are thus not suitable in many situations. Although we choose randomly, we may not have a suitable result. For example, in the normative conflict graph of Figure 5, both  $\{a, c\}$  and  $\{b, d\}$  are maximal conflict free sets, but it is clear that the former set is preferable to the latter, as no node attacks  $a$ , while attacks exist against all other nodes. Nevertheless, this heuristic does provide an upper limit to the number of norms that an agent may be able to satisfy.

### 3.3 Preferred Extension Based Norm Conflict Resolution Heuristic

Continuing with our use of heuristics inspired by argumentation approaches, we move to the third possibility. Given a conflict free set of arguments, we may determine which arguments in an argument system are, in some sense, *acceptable*, or not attacked in such a way that the attack against them stands:

**Definition 9. (Acceptability and Admissibility)** An argument  $A \in AR$  is acceptable with respect to a set of arguments  $S$  iff for every argument  $B \in AR$ , such that  $attacks(B, A)$ , there is a  $C \in S$  such that  $attacks(C, B)$ .

Then we may define a conflict free set of arguments  $S$  as admissible iff each argument in  $S$  is acceptable with respect to  $S$ .

Thus, an admissible set contains a self consistent set of arguments that defend each other from attack by any other arguments. In the example illustrated by the rightmost graph of Figure 3, the sets  $\{a\}$ ,  $\{e\}$ ,  $\{a, e\}$ , and  $\{b, e\}$  are all admissible sets.

A *credulous* agent is one that deems any defended argument acceptable, so that all of these are possible. However, it is better to accept a larger number of non-conflicting arguments. Indeed, a credulous agent would accept the *preferred extension* of a set of arguments; that is, the maximal (with regards to set inclusion) admissible set of arguments.

**Definition 10. (Preferred Extension)** *A preferred extension of an argument system AR is a maximal (with respect to set inclusion) admissible set of AR.*

This allows us to provide our third, preferred extension based heuristic.

**Definition 11. (Preferred Extension Based Norm Conflict Resolution Heuristic)** *Given a normative conflict graph, choose those norms present in the maximal (with respect to set size) preferred extension, and drop all other norms.*

While there may be multiple maximal preferred extensions, they all contain the most important norms (in terms of the priorities assigned by social context and the agent’s conflict resolution mechanism), assuming that these are not in conflict. If they are in conflict, each extension will contain one such norm. In Section 4, we compare this heuristic to the two described previously.

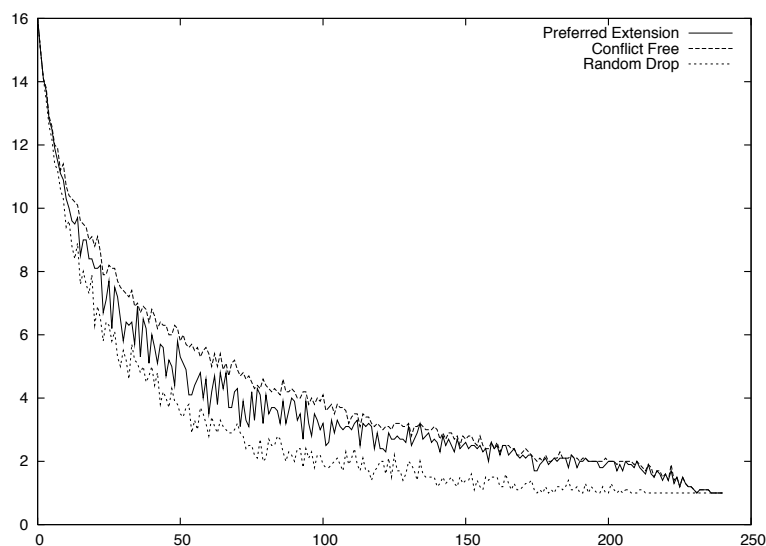
Computationally, it is well known that calculating the preferred extension of an arbitrary argument framework is difficult [8]. This is, of course, a worst case result; algorithms have been proposed that work well in typical situations [9].

While our normative model does not contain the notion of time, an agent could repeatedly make use of these heuristics to determine which norms to ignore at any time. Limited integration with a BDI type architecture is possible by allowing an agent to dynamically alter its social context and norm type pruning preferences based on its changing beliefs, desires and intentions. A closer integration between the work described here and the BDI model will be pursued in future work.

**Example** Looking at Alice’s final normative conflict graph (as shown in the rightmost graph of Figure 3), we know that the sets  $\{a\}$ ,  $\{e\}$ ,  $\{a, e\}$  and  $\{b, e\}$  are all admissible. Two maximal admissible sets, i.e. preferred extensions exist, namely  $\{a, e\}$  and  $\{b, e\}$ . Alice thus equally prefers to go to the theatre and ignore the paper, or to visit her sick mother and ignore the paper. Note that even in the latter case, the paper is ignored; this is due to Alice’s stated preference of permissions over obligations.

## 4 Evaluation

The three heuristics were compared by examining how many norms each was able to keep in the conflict free normative conflict graph when run over sets with different numbers of normative conflicts. We examined a system of 16 norms, and between 0 and 241 conflicts. For each different number of conflicts, we ran the simulation 10



**Fig. 6.** An evaluation of the preferred extension based norm conflict resolution heuristic (solid line), compared to the size of the largest conflict free set (dashed line) and the size of the set of norms obtained when dropping the dominant conflicting norm in random order (dotted line). These results are averaged over 10 runs. The y-axis shows the number of norms retained, while the x-axis indicates the number of normative conflicts.

times, and averaged the results. During each run, a random normative conflict graph was generated, containing random conflicts between nodes. The various heuristics were evaluated on this graph, and the number of retained nodes was recorded.

Figure 6 illustrates the results of this evaluation. As expected, the number of retained norms decreases as the number of conflicts increases. More interestingly, the *norm conflict resolution heuristic* based on the preferred extension performs far better than the *random drop heuristic*, often equalling the *conflict free heuristic* which, by definition, contains the maximal number of norms that may be retained to maintain a conflict free set. When very few or very many conflicts exist, the three heuristics converge; this is because very little freedom exists when selecting which norms to drop in such cases. As more choices become available, the three heuristics increasingly diverge.

## 5 Discussion

The links between norms and defeasible reasoning are well known [10], and our application of argumentation makes sense in such a context. Other approaches to conflict resolution include the work of Vasconcelos et al. [11]. However, their approach is very different; by having a more detailed logic, which includes temporal concepts, they shrink, or expand the effects of norms until no overlaps exist. Thus, unlike the work described in this paper, no norms are dropped in their approach.

Given a strict total ordering over social contexts, our normative conflict graph is acyclic. In such a situation, the preferred extension heuristic is optimal. This result may in fact be strengthened; it is well known that odd-length cycles cause problems with preferred semantics in Dung’s framework [12]; a preferred extension in this case will omit sets of arguments that may sometimes be considered consistent. Thus, given a graph with no such cycles, the maximal preferred extension returns the maximal set of most important norms that the agent should satisfy. Knowing this, two questions may be asked. First, given these problems with the preferred extension semantics, we may ask how the heuristic’s performance may be improved. Second, we may ask whether some other heuristic (such as the conflict-free set heuristic described above) may be modified to yield correct results.

Consider the argument framework  $(\{a, b, c\}, \{(a, b), (b, c), (c, a)\})$ . This argument framework consists of one odd length cycle of attacks, and its only preferred extension is the empty set. This makes sense for argument semantics: given a set of three arguments, each of which defeats another, a credulous reasoner is unable to select between them. However, mapping this argument framework to the normative case means that a hypothetical normative agent is unable to choose between these three norms, and thus decides to comply with none of them. Clearly, the correct course of action for the agent would be to drop one of the norms, and then re-evaluate the preferred extension of the resulting set. However, it is difficult to select which norm to drop in such cases, as the norm may interact with other norms outside this cycle. Our first priority for future work involves investigating how we may enhance the heuristic to cater for such situations: by detecting the interactions amongst odd-length cycles, and dropping the appropriate norm(s), we aim to create a better performing heuristic.

A number of other extensions have been defined [7, 13]. The *grounded* extension, for example, is the least fixed point of the function  $F_{AS}(\{\})$  where

$$F_{AS}(S) = \{A \mid A \text{ is acceptable with respect to } S\}$$

for an argument system  $AS$ . The grounded extension is a very aggressive *sceptical* extension, in the sense that any arguments within the grounded extension should be deemed consistent by any rational agent; an argument system has only one grounded extension, and its arguments defend it from any attack. Another type of sceptical semantics consists of the arguments found in the intersection of all preferred extensions.

Sceptical extensions are useful in situations in which an agent must decide what norms to assign to another agent, or when it attempts to predict the other agent’s behaviour. Assuming that the assigning agent is aware of the assignee’s norms and preferences, it may compute the sceptical extension of its normative conflict graph. The assigning agent could then be guaranteed that the assignee will attempt to comply with those norms found in this extension. Based on this information, it could decide whether to assign additional norms to the agent.

Apart from the extensions to the heuristic discussed above, we hope to extend the representational power of our approach in a number of important ways. For example, consider the situation where an agent made a promise to their boss, and two promises to friends. To evaluate which the agent should comply with, we may make use of social contexts. Currently, we are unable to accrue these contexts. Thus, if the promise to

the boss outweighs the promise to a single friend, it will outweigh any number of such promises (unless all promises are combined into one context, in which case difficulties are encountered if one of the norms existing within the sub contexts is dropped). Existing work on accrual in argumentation, such as [14], may prove applicable in dealing with these issues, and extending the framework to deal with such situations will allow for the modelling of more complex normative domains.

At the moment, we only implicitly consider the sanctions and rewards an agent obtains when violating, or meeting obligations, via preferences over social contexts. Adding the notion of utility to the framework, and allowing the agent to reason with this concept should also enhance the representational capabilities of the framework. Such utilities impose an additional layer of preferences over norms, and the work carried out in [15] may be applicable to such an enhancement.

Finally, it can be seen that in some situations, norms support one another. For example, groups of norms may have to be dropped together (e.g. in the case of a contract violation), or if one obligation is not met, another obligation may come into force (as in contrary to duty violations). The use of abstract argument frameworks containing the notion of support [16] may allow us to extend our approach to represent such concepts.

## 6 Conclusions

In this paper we have proposed an argumentation theory based heuristic allowing an agent to determine which norms to maintain and drop in cases where normative conflicts arise. Since norms may be assigned to an agent by forces outside its control, even the most careful of agents may be unable to prevent normative conflict from occurring. Thus, being able to reason about which norms to comply with is important for most normative agents.

Using a simple representation for norms, we were able to generate a normative conflict graph. We were then able to prune this graph based on various preferences the agent might hold. Even after pruning took place, normative conflict could still exist. We thus made use of the structural and semantic similarities between Dung's abstract argument frameworks and our normative conflict graph to create heuristics that allow an agent to choose which norms to drop, and which norms to comply with.

We evaluated our approach by comparing the number of norms the agent was able to comply with when making use of the preferred extension heuristic, as opposed to the number of norms it could comply with when randomly dropping norms (while still pruning the normative conflict graph), and the number of norms kept in the largest conflict free set. As expected, the preferred extension based heuristic performed better than the former, but worse than the latter.

While, as shown by our evaluation, our heuristic performs well, it is not yet guaranteed to find the optimal set of norms. We look forward to carrying out the future work proposed in Section 5, which will allow us to both enhance the heuristic, and, by extending our model, increase the number of situations in which it is applicable.

## Acknowledgements

This research is supported by the EU 6th Framework project CONTRACT (INFSO-IST-034418). The opinions expressed herein are those of the named authors only and should not be taken as necessarily representative of the opinion of the European Commission or CONTRACT project partners.

## References

1. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In Allen, J., Fikes, R., Sandewall, E., eds.: Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann publishers Inc.: San Mateo, CA, USA (1991) 473–484
2. Panzarasa, P., Norman, T.J., Jennings, N.R.: Modelling sociality in a bdi framework. In: 1st Asia-Pacific Conf. on Intelligent Agent Technology. (1999) 202–206
3. Dignum, F., Morley, D., Sonenberg, E.A., L, C.: Towards socially sophisticated bdi agents. In: Proc. of the fourth Int. Conf. on MultiAgent Systems (ICMAS-2000). (2000) 111–118
4. López y López, F.: Social Power and Norms: Impact on Agent Behaviour. PhD thesis, University of Southampton (2003)
5. Moses, Y., Tennenholtz, M.: Artificial social systems. *Computers and AI* **14**(6) (1995) 533–562
6. Horty, J.F.: Agency and obligation. *Synthese* **108** (1996) 269–307
7. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence* **77**(2) (1995) 321–357
8. Dunne, P.E., Bench-Capon, T.J.M.: Coherence in finite argument systems. Technical report, University of Liverpool (2001)
9. Verheij, B.: A labeling approach to the computation of credulous acceptance in argumentation. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India (2007) 623–628
10. Horty, J.: Nonmonotonic foundations for deontic logic. In Nute, D., ed.: *Defeasible Deontic Logic*. Kluwer Academic Publishers, Dordrecht (1997) 17–44
11. Vasconcelos, W.W., Kollingbaum, M.J., Norman, T.J.: Resolving conflict and inconsistency in norm-regulated virtual organizations. In: Proceedings of the Sixth International Conference on Autonomous Agents and Multiagent Systems, Honolulu, Hawaii, USA (2007) 632–639
12. Baroni, P., Giacomin, M.: Solving semantic problems with odd-length cycles in argumentation. In: Proceedings of the 2003 European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty. Volume 2711 of Lecture Notes in Artificial Intelligence., Aalborg, Denmark (2003) 440–451
13. Coste-Marquis, S., Devred, C., Marquis, P.: Prudent semantics for argumentation frameworks. In: Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence, Washington, DC, USA, IEEE Computer Society (2005) 568–572
14. Prakken, H.: A study of accrual of arguments, with applications to evidential reasoning. In: Proceedings of the 10th International Conference on Artificial Intelligence and Law. (2005) 85–94
15. Bench-Capon, T.: Value based argumentation frameworks. In: Proceedings of the 9th International Workshop on Nonmonotonic Reasoning, Toulouse, France (2002) 444–453
16. Oren, N.: An Argumentation Framework Supporting Evidential Reasoning with Applications to Contract Monitoring. Phd thesis, University of Aberdeen, Aberdeen, Scotland (2007)