Norms as Objectives: Revisiting Compliance Management in Multi-agent Systems

Aditya Ghose[1] and Tony Bastin Roy Savarimuthu[2]

[1] Decision Systems Laboratory
School of Computer Science and Software Engineering
University of Wollongong, NSW 2522 Australia
aditya@uow.edu.au
[2] Dept. of Information Science
University of Otago

Abstract. This paper explores a hitherto largely ignored dimension to norms in multi-agent systems: the normative role played by optimization objectives. We introduce the notion of *optimization norms* which constrain agent behaviour in a manner that is significantly distinct from norms in the traditional sense. We argue that optimization norms underpin most other norms, and offer a richer representation of these. We outline a methodology for identifying the optimization norms that underpin other norms. We then define a notion of compliance for optimization norms, as well as a notion of consistency and inconsistency resolution. We offer an algebraic formalization of *valued optimization norms* which allows us to explicitly reason about degrees of compliance and graded sanctions. We then outline an approach to decomposing and distributing sanctions amongst multiple agents in settings where there is joint responsibility.

1 Introduction

The connection between norms and preferences and between norms and optimization objectives has received relatively little attention in the literature. Boer et al [1] have argued that legal norms may be viewed as statements of ceteris paribus preference. van der Torre and Tan [2] have defined a preference-based semantics for norms which have been leveraged by Dignum et al [3] to develop a semantic account of how goals and intentions are obtained from desires via a process constrained by norms and obligations (utilities are also mentioned in [4] but not related to norms).

The connections, however, are deeper and merit closer scrutiny. Let us consider the connections with preferences and optimization objectives first. Optimization problems are traditionally formulated via a set of *decision variables* (a complete assignment of values to these constitutes a *solution* to the problem), a set of *constraints* on these variables and an *objective function* formulated from these variables whose value we seek to *optimize* (i.e., maximize or minimize). Solutions which satisfy all of the applicable constraints are called *feasible* solutions,

J. Simão Sichman and H. Aldewereld (Eds.): COIN 2012, LNAI 7756, pp. 105–122, 2013.

while those which optimize the value of the objective function are called *optimal* solutions. An objective function may be viewed as generating a preference relation on the set of feasible solutions, such that optimal solutions are the most preferred of the feasible solutions under this relation. Given a set of alternatives (in the this instance, the set of feasible solutions) an objective function may be viewed as an intensional representation of the underlying preference relation.

Imagine a future where organizations may be referred to "carbon tribunals" for non-compliance with the carbon-mitigation norm. We would argue that in its purest sense, this norm must be represented as the optimization objective *minimize carbon-footprint*. One might argue that in real-life, such norms are manifested as simpler numeric carbon mitigation targets for organizations (e.g. "reduce your cumulative emissions by n tons"). However, this representation of the norm is a compromise, where some regulatory authority has sought to present a simpler (to evaluate and understand) target for organizations to meet, by trading off the need to impose a target that achieves real emissions reduction against the need to not impose a norm that would be infeasible for organizations to meet. We would argue that such norms (with numerically specified targets) must be accorded a status befitting their true nature: as simplifications and imperfect, incomplete compromises between the true intent of the norm and the current business reality. The numeric target is a fragile construct, contingent on the perceived business context at the time the norm was formulated. The introduction of new technology may actually make a higher carbon-mitigation target feasible, but such a change would be hard to reflect dynamically on the norm (which would typically be reviewed and revised infrequently, together with other norms, by a regulatory authority). Similarly, difficult market conditions might oblige us to revise the carbon-mitigation targets downwards. Clearly, this constant need for revising the norm could be avoided if the norm were represented in its natural form (as the optimization objective *minimize carbon-footprint*).

It may appear at first blush that a notion of compliance with such norms is difficult to define. We will present evidence to the contrary. Imagine the organization charged with non-compliance with the carbon-mitigation offering its defence to the tribunal by presenting a log of all its key organizational decisions over an audit period and establishing that each choice of the organization from the available options was in fact the optimal one with respect to the *minimize carbon-footprint* objective. Intuitively, this is a reasonable defence, underpinned by the argument that the organization "did the best that it could" under the circumstances. We shall formalize a notion of compliance with norms that are optimization objectives along these lines.

Based on these considerations, we argue that there is a need to extend our ontology of norms with a new class of *optimization norms*. These are norms represented in the form of optimization objectives or, as we shall see later, as the preference relations that underpin them. Optimization norms do not admit boolean evaluation. We shall distinguish them from the more traditional conception of a norm (which does admit boolean evaluation) by referring to the latter as *boolean norms*. Our approach bears some similarity in underlying intuitions to studies of supererogation in deontic logic (see for instance [5]) but uses entirely different machinery.

Note that the use of optimization norms does not imply that there must exist a unique optimal state (or set of states) in the absolute sense. As the motivating example above illustrates, and as we shall discuss in greater detail later in this paper, optimization norms merely inform choice amongst the available feasible alternatives. That set of available feasible alternatives might change from context to context. This approach thus does not preclude dynamic contexts or norm evolution.

Part of our premise here is that “teasing out” the objective underpinning a norm and bringing it to bear on the reasoning process is important, for the following reasons. First, encoding norms as objective functions offers a more accurate and richer representation of norms. Second, it provides an opportunity to explicitly bring a mature body of results from the field of optimization to bear on norm-driven reasoning problems, opening up the possibility of significantly faster reasoners. Third, it permits us to relate norm-compliance to the notion of *satisficing* [6] of optimization objectives. Fourth, it enables us to define a notion of degrees of norm compliance, and correspondingly, graded sanctions.

This latter point is of particular importance, and can be analyzed from two perspectives. The first involves the notion of *graded compliance.* Many commonly occurring compliance requirements are stated in an imprecise fashion. Consider, for instance, the requirement *quarterly activity statements must be filed within a reasonable time frame* [7]. It is difficult to determine in a categorical fashion whether this requirement has been satisfied, given the ambiguity associated with determining whether a certain time frame is ”‘reasonable”’. One way of dealing with the problem is to ”‘contextualize”’ such requirements through a (largely human-mediated) exercise of transforming these into crisp requirements by adding elements to the specific context (such as a definition of ”‘reasonable”’ in a particular application context) [8]. Alternatively, one can make (potentially subjective) assessments of degrees of compliance. In the spirit of satisficing optimization objectives, thresholds on these degrees of compliance can be used to determine, for instance, whether the operations of an organization are sufficiently compliant, even in the absence of boolean assessments of compliance.

The second involves the related notion of *graded sanctions.* The use of formal reasoning tools to model, analyze and monitor contracts is becoming increasingly important [9] [10]. A particularly hard problem in this space is the formalization of sub-contracting and outsourcing (both increasingly common business practices). In particular, the decomposition of a set of penalties or sanctions amongst a set of sub-contractors is difficult to formalize. Intuitively, when a norm or contractual obligation is violated, we would expect the applicable penalties to be distributed amongst the sub-contractors in direct proportion to the extent of their contribution to (or responsibility for) the violation. While Villatoro et al [11] and Boella et al [12] have considered the problem of sanctions in multi-agent contexts, they have not described any machinery for decomposing a sanction to obtain individual agent-specific graded sanctions in settings where agents have joint responsibility (and hence graded levels of norm violation).

The remainder of this paper is structured as follows. In Section 2, we discuss how optimization norms underpin most traditional conceptions of norms (i.e., *boolean norms*), and provide methodological guidelines for how these might be identified/extracted from boolean norms. In Section 3, we motivate and define an algebraic formalization of optimization norms in the form of *valued optimization norms*. In Section 4, we identify two alternative notions of compliance with optimization norms, based on the extent of the horizon over which compliance is assessed. In Section 5, we identify alternative notions of consistency (both between optimization norms and between optimization and boolean norms). We also identify alternative approaches to the resolution of these inconsistencies. In Section 6, we address the problem of sanction management, and in particular, how sanctions might be decomposed, or distributed amongst a collection of agents that had shared responsibility for a norm (the violation of which leads to the sanctions in question). Section 7 involves a discussion of some related implementations that offer pointers to how a machinery for optimization norm enforcement might be implemented. We present concluding remarks in Section 8.

2 Identifying Optimization Norms

In a very intuitive manner, it is possible to articulate an objective function underpinning every norm. Consider the social norm that one should not litter. Given that there are many plausible extenuating circumstances where littering may in fact be permissible (e.g., one drops one's bag of sandwiches on the park bench to go prevent a child from stepping into traffic), the underpinning optimization objective is to *minimize* the extent of littering. In a social context, the articulation of the prohibition of littering may be viewed as a necessary simplification of the more complex (and more nuanced) underpinning social objective. In a similar vein, the social norm that prohibits delays in the payment of outstanding invoices is underpinned by the optimization objective to minimize the delay between the receipt of the invoice and payment. The social norm that obliges us to consult widely prior to taking decisions in organizational settings may be viewed as being underpinned by the optimization objective to maximize the extent of consultation prior to taking decisions. More generally, we could posit that:

- Prohibitions are underpinned by *minimization* objectives.
- Obligations are underpinned by *maximization* objectives.

The duality of maximization and minimization objectives (every maximization objective can be represented by a corresponding minimization objective and vice versa) is reflected by the duality of prohibitions and obligations [13]. In addition, fairness norms [14] can be encoded as load-balancing objectives.

In many cases, the boolean proposition associated with the norm can be transformed into a continuous valued variable. Thus, the boolean proposition *littering* is transformed into the variable *extent-of-littering*, the proposition *consultation* to the variable *extent-of-consultation* and so on in the examples above. In general, the following simple procedure might be used to identify the optimization objective underpinning a norm:

– Identify the action that the norm seeks to constrain. In our examples above, these would be "littering", "consultation" and "carbon emission".
– Identify a measure that is applicable on the task that norm makes reference to. In our examples above, these would be *extent-of-littering*, *extent-of-consultation* or *extent-of-carbon-emission*. These become the variables that the eventual objective function would refer to. In the following, we shall refer to this as the *task measure*.
– Identify whether the norm seeks to maximize or minimize the measure defined in the previous step (load balancing objectives can also be represented as maximization or minimization objectives). This gives us the final objective function.

Note that articulating an objective function does not automatically give us a fully formulated optimization problem (that would require the typically hard task of modeling constraints). However, as we shall see below, the optimality of choices with respect to an optimization objective can be evaluated using a variety of means even if the problem has not been formulated as an optimization problem.

It is important to note that our conception of optimization norms is not specifically intended for modeling social norms. The machinery we develop is applicable to both individual objectives (and preferences) as well as social norms and the preferences that underpin them.

It is also important to note that, our observations in this section notwithstanding, practical applications will likely involve both boolean and optimization norms. It is therefore important to develop machinery that can handle both (our discussion of norm consistency later in the paper will address the question of consistency between boolean and optimization norms).

3 An Alternative Formalization

In this section, we provide an alternative formalization of optimization norms in terms of an algebraic framework for preference handling and show how it offers a sophisticated machinery for dealing with *graded compliance* and correspondingly *graded sanctions*.

It is useful to examine first the space of alternative means for formalizing optimization norms. The most obvious is to represent optimization norms in the form of *objective functions* in the sense understood in the literature on operations research, or as *utility functions* in the sense of the literature on decision theory. One might also formalize these using preference relations of various kinds as formalized in the literature on preference handling. To the extent that an optimization norm needs to encode preference over states of affairs (or solutions) each of these approaches turn out to be equally viable (recall the discussion on the interplay between objective functions and preference in Section 1). Ultimately, our choice was guided by two additional requirements. First, the representation scheme needed to support an explicit notion of *degree of compliance*. Objective functions (or utility functions) arguably meet this requirement - the value of a (maximization) objective function for a given solution can be viewed as an indicator of how good that solution might be. Representations based on preference relations do not naturally lend themselves to analysis of degree of compliance. Second, the representation scheme had to be general enought to admit assessments of preference on multiple heterogeneous scales (as is likely to be the case in real-life applications) including both quantitative and qualitative scales. Both objective/utility functions and preference relation based approaches fall short relative to this requirement.

The c-semiring framework [15] was chosen for our formalization of optimization norms primarily because it satisfies all of the requirements discussed above. The framework was originally developed for defining soft constraints as preferences over assignments of values to decision variables in (potentially overconstrained) constraint satisfaction problems. For our purposes, the c-semiring framework enables abstract encodings of preference over multiple heterogeneous scales (which could be both qualitative and quantitative). Multiple distinct c-semirings, each encoding a distinct dimension over which preference is specified (including mixes of qualitative and quantitative dimensions) can be combined in a simple fashion to obtain a single c-semiring [15] - thus providing a modular framework in which preference dimensions could be added or removed while leaving much of the reasoning machinery intact.

We start with the definition of a c-semiring [15].

Definition 1. [15]: A c-semiring is a 5-tuple $\langle A, \oplus, \otimes, \mathbf{0}, \mathbf{1} \rangle$ such that:

- A is a set of abstract preference values with $\mathbf{0}, \mathbf{1} \in A$ ($\mathbf{0}$ represents the "'worst"' preference value while $\mathbf{1}$ represents the "'best"' preference value);
- $\oplus$ is a binary operator which is closed (i.e. if $a, b \in A$, then $a \oplus b \in A$), commutative (i.e. $a \oplus b = b \oplus a$), associative (i.e. $a \oplus (b \oplus c) = (a \oplus b) \oplus c$), idempotent (i.e. if $a \in A$, then $a \oplus a = a$), has $\mathbf{0}$ as a unit element (i.e. $a \oplus \mathbf{0} = a = \mathbf{0} \oplus a$), and with $\mathbf{1}$ as an absorbing element (i.e. $a \oplus \mathbf{1} = \mathbf{1} = \mathbf{1} \oplus a$);
- $\otimes$ is a binary operator which is closed (i.e. if $a, b \in A$, then $a \otimes b \in A$), commutative (i.e. $a \otimes b = b \otimes a$), associative (i.e. $a \otimes (b \otimes c) = (a \otimes b) \otimes c$), has $\mathbf{1}$ as a unit element (i.e. $a \otimes \mathbf{1} = a = \mathbf{1} \otimes a$), and $\mathbf{0}$ as an absorbing element (i.e. $a \otimes \mathbf{0} = \mathbf{0} = \mathbf{0} \otimes a$);
- $\otimes$ distributes over $\oplus$ (i.e. $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$).

Intuitively, $\oplus$ is used to compare preference values, while $\otimes$ is used to combine preference values. The $\oplus$ operator generates a partial order $\preceq$ on A as follows: for any $v_1, v_2 \in A$, $v_1 \preceq v_2$ (read this as v_1 is "at least as good as" v_2) if $v_1 \oplus v_2 = v_1$.

In the constraint satisfaction literature, several useful instantiations of c-semirings have been discussed: boolean (where $A = \{T, F\}$), fuzzy (where $A = [0, 1]$), weighted (where $A = \mathcal{R}^+$) etc. Qualitative c-semirings can be of interest, where the $\otimes$ and $\oplus$ operators are defined extensionally. Consider the following c-semiring, where l represent *low*, m represents *medium* and h represents *high*: $Q = \langle\{l, m, h\}, \oplus, \otimes, l, h\rangle$ where $l \oplus m = m$, $m \oplus h = h$, $l \oplus h = h$, $l \otimes m = l$, $m \otimes h = m$, and $l \otimes h = l$. To rephrase this, this c-semiring allows us to use 3 preference values - *low*, *medium* and *high* - with *low* as the designated "worst" value (the element denoted by **0** in the c-semiring) and *high* as the designated "best" value (the element denoted by **1** in the c-semiring). If we were to compare these values using the $\oplus$ operator, the comparison of *low* and *medium* would yield *medium*, the comparison of *medium* and *high* would yield *high* and so on. Similarly, if we were to combine these values using the $\otimes$ operator, then the combination of *low* and *medium* would generate *low*, the combination of *medium* and *high* would generate *medium* and so on.

Notice that the set of preference values in each of the c-semiring instances discussed above offers a scale on which degree of compliance might be assessed. This is also true for c-semirings consisting only of abstract preference values, and for composite c-semirings obtained by combining several component c-semirings using the technique described in [15] (ommitted here for brevity). Thus, if we were using the qualitative c-semiring discussed above with preference values $\{l, m, h\}$, we obtain a vocabulary for describing degree of compliance that permits us to assert that a given state of affairs has a low (l) degree of compliance, or a medium (m) degree of compliance and so on. Similarly, one might conceive of a composite c-semiring consisting of this qualitative c-semiring combined with the fuzzy c-semiring discussed above which would allow us to assess preference on two separate dimensions using these two distinct scales, where degrees of compliance would be represented using pairs where the first element is a preference value from the qualitative c-semiring and the second element a value from the fuzzy c-semiring (e.g. $\langle l, 0.7\rangle$ or $\langle m, 0.55\rangle$).

In the following, we will take a state to be a complete assignment of values to a set of variables V (these do not necessarily have to be propositional, thus permitting us to also view solutions to constraint satisfaction or optimization problems as states). We define a *valued optimization norm* (so called because these norms associate states with specific preference and sanction values) as a mapping from a state to a semiring valuation of that state. In addition, we define a real-valued penalty associated with each state. We also constrain the penalties so that any penalty associated with a more preferred state has to be lower than a penalty associated with a less preferred state. If two states are equally preferred, then their penalties must be equal. Also, the penalty associated with the most preferred state must be 0.

Definition 2. *Given a c-semiring $P = \langle A, \oplus, \otimes, \mathbf{0}, \mathbf{1}\rangle$ and a set of states S (with penalties represented as elements of the set of reals $\mathcal{R}$), a* **valued optimization norm** *n_P is defined as $n_P : S \rightarrow A \times \mathcal{R}$, such that: (1) For any s such that $n_P(s) = \langle \mathbf{1}, r\rangle$, $r = 0$ and (2) For any s_1 and s_2, if $n_P(s_1) = \langle v_1, r_1\rangle$ and $n_P(s_2) = \langle v_2, r_2\rangle$ where $s_1, s_2 \in S$, $v_1, v_2 \in A$ and $r_1, r_2 \in \mathcal{R}$ and $v_1 \oplus v_2 = v_1$, then $r_1 \leq r_2$ (if $v_1 = v_2$, then $r_1 = r_2$).*

Example 1. Consider a simple example where our propositional vocabulary consists of 2 letters, p and q and we prefer the state $p \wedge q$ the most and the state

$\neg p \wedge \neg q$ the least. The states $\neg p \wedge q$ and $p \wedge \neg q$ are in-between most and least preferred and are equally preferred. A valued optimization norm n_Q conforming to the definition above could be defined as follows, using the c-semiring $Q = \langle \{l, m, h\}, \oplus, \otimes, l, h \rangle$ (together with its associated extensional definitions of $\oplus$ and $\otimes$) as in the discussion above: $n_Q(p \wedge q) = \langle h, 0 \rangle$, $n_Q(\neg p \wedge q) = \langle m, 5 \rangle$, $n_Q(p \wedge \neg q) = \langle m, 5 \rangle$, $n_Q(\neg p \wedge \neg q) = \langle l, 10 \rangle$.

Note that the definition above provides the basis for two important dimensions to practical compliance management. The first is *graded compliance*. Valued optimization norms permit us to associate an explicit *degree of compliance* value (effectively the corresponding c-semiring value) with each state of affairs. They also permit us to associate finer-grained sanctions (as opposed to a single sanction for the violation of a boolean norm) with different degrees of compliance.

It is easy to see how contrary-to-duty obligations can be represented in this formalization. Consider a reparation sequence (one way to represent a contrary-to-duty obligation) in FCL [16] ($\times$ is used in the following to mean "'else"') : $O_1 \times O_2 \times \ldots \times O_m$. The reparation sequence above is to be read as follows: obligation O_1 holds, failing which obligation O_2 holds and so on. Let $M(\phi)$ represent the set of states that satisfy ϕ (i.e., the models of ϕ). This reparation sequence can be represented by any norm n_P satisfying the following property: For any $s \in M(O_i)$ and any $s' \in M(O_j)$ where $i < j$ and $n_P(s) = \langle v, r \rangle$ and $n(s') = \langle v', r' \rangle$, $v \oplus v' = v$ and $r < r'$. Consider an FCL[16] reparation sequence: $(Op \wedge q) \times (O\neg p \wedge q) \times (O\neg p \wedge \neg q)$, where $O\phi$ represents the obligation to make ϕ true. n_Q defined in the example above is one instance of a valued optimization norm that encodes this reparation sequence. In general, a given reparation sequence could be encoded by a number of distinct valued optimization norms.

In the discussion in the subsequent sections, some of the development will be presented in terms of the abstract notion of optimization norms, without specifically referring to valued optimization norms. Some concepts will, however, be illustrated using valued optimization norms. The discussion on sanction management later in the paper relies entirely on our formalization of valued optimization norms.

4 Complying with Optimization Objectives

We first consider in abstract terms the machinery that we might use to select amongst alternative options. These options could be plans in BDI agents (often called *option selection* [17]) or intentions (the *intention selection* problem [17]) or, more generally, states that an agent might seek to realize. In general terms, the machinery we require must be able to make ordinal comparisons between states resulting from the optional courses of action that an agent might have to select from. For instance, given two states s_1 and s_2 and an optimization norm *maximize* x articulated in terms of a task measure x, the machinery must be able to make a determination on which of s_1 and s_2 lead to a higher value of x (without necessarily computing that value), or whether they lead to identical values of x.

As discussed in the introduction, one intuitive approach to establishing that an agent has complied with a optimization norm is to establish that its decisions sought to optimize the norm. This can be formalized in two ways:

– *Global compliance:* This requires that the final state achieved by an agent be the optimal (relative to the optimization norm) of all the states that were feasible for the agent to achieve. Note that this approach is only applicable when the agent performs bounded computation and a clear notion of final state exists. It also requires the ability to compute the set of all possible final states that would be feasible for the agent to achieve. It might also mean that agents would make local choices that not optimal with respect to the objective (in the interests of arriving at the globally optimal final state).
– *Local compliance:* This requires that every (local) choice made by the agent be optimal with respect to the objective. Note that this might mean that the final state arrived at by an agent (performing bounded computation) is sub-optimal. However, this notion of compliance can be used in agents performing unbounded computation.

5 Managing Norm Conflict

We address the question of norm conflict detection and resolution in this section. We begin by considering a well-known example of the interplay between norms and preferences.

Sen's Example. Sen [18] offers an example where the interplay between norms and objectives generates results that are contrary to what preference maximization would generate. Consider a situation where an agent prefers option x to y and y to z. Assume that the agent selects y from $\{x, y, z\}$ and selects z from $\{y, z\}$ (both choices run counter to the preference maximization principle). Sen offers an account involving norms that explains such behaviour. Assume that the options are differently-sized slices of a cake, and that x is the biggest slice, y the next in size and z the smallest. The agent has a preference for larger slices of the cake. To explain the behaviour of the agent, Sen brings to bear a "politeness" norm, which requires the agent to not select the largest slice when asked to choose from a set of cake slices. In selecting amongst $\{x, y, z\}$, the agent uses this norm to rule out option x and then selects the preferred option from the remainder (i.e., option y). Similarly, in selecting amongst $\{y, z\}$, the agent rules out y as an "impolite" choice (it is the largest of the 2 choices) and selects the remaining option (z).

We argue that viewing the politeness norm as a preference ordering (which prefers any state where a non-largest slice of cake is selected over a state where the largest slice is selected), together with a prioritization on objectives (where the optimization norm derived from the politeness norm has priority over the optimization norm to maximize the size of the cake slice selected), offers an equally valid explanation of the agent's behaviour. Indeed, our approach supports finer grained reasoning, for instance, by permitting us to explore the pareto-frontier with respect to these two objectives (if they were treated as having equal priority).

We need, in the first place, a formal definition of conflict for optimization norms (with other optimization norms as well as with boolean norms). We will define consistency for optimization norms by viewing them as preference relations on the set of feasible solutions (recall the discussion of these issues in the introduction). We can then define three distinct notions of consistency for optimization norms.

- *Absolute consistency:* Under this notion, a pair of optimization norms o_1 and o_2 are deemed to be inconsistent if and only if there exists a pair of solutions (or options) s and s' such that s is preferred over s' under o_1 and s' is preferred over s under o_2. Absolute consistency is arguably an overly stringent notion. Our intent here is to identify situations where objectives might "pull in different directions"'. Under this notion of consistency, we would deem a pair of optimization norms to be inconsistent even if the pair s and s' whose relative ordering they disagreed on were actually not in the set of options currently being deliberated on. In other words, we would deem the objectives to be inconsistent even if they were not "pulling in different directions" in the current instance. Note also that checking consistency in this mode requires that we extensionally elaborate the preference relation corresponding to the optimization norm with reference to the set of all possible options over which preferences might be specified - but that set is potentially unbounded and cannot in general be predicted. Checking for absolute inconsistency is therefore impractical.
- *Contextual consistency:* Under this notion, a pair of optimization norms o_1 and o_2 are deemed to be inconsistent in a given context C (defined as a set of options/solutions from which a choice has to be made) if and only if there exists a pair of solutions (or options) $s, s' \in C$ such that s is preferred over s' under o_1 and s' is preferred over s under o_2. The notion of contextual consistency helps us determine whether a pair of objectives would lead to conflicting preferences in a particular context, given a particular set of alternatives. It could be argued that this too is a somewhat stringent notion, since conflicting preferences might not manifest themselves in actual conflicting choices if the conflicting preferences involve options that are not the top choices (i.e., the most preferred options) under the two preference orderings. In other words, an agent could "tolerate" optimization norms which are deemed to be both absolutely inconsistent and contextually inconsistent, as long as this does not lead to conflicting choices. On the other hand, if the context changes infrequently, this notion of consistency can be useful (if the relevant optimization norms are - or can be made - consistent for that context, no further inconsistency handling will be required while that context remains unchanged). Checking for consistency over a set of optimization norms involves a straightforward generalization of the pair-wise check mentioned above.

– *Choice consistency:* Under this notion, a pair of optimization norms o_1 and o_2 are deemed to be inconsistent in a given context C (defined as a set of options/solutions from which a choice has to be made) if and only if S is the set of strictly non-dominated solutions (or options) under o_1, S' is the set of strictly non-dominated solutions (or options) under o_2, $S \subseteq C$, $S' \subseteq C$ and $S \cap S' = \emptyset$. Here, the set of strictly non-dominated solutions consists of those solutions for which there exists no other that is strictly more preferred under the given ordering.

We now consider the question of conflict between optimization norms and boolean norms. Optimization norms and boolean norms do not conflict in an absolute sense, i.e., independent of a given context (given by set of available alternatives). To understand why, we need to recall a commonly used definition of inconsistency: two assertions are inconsistent if there does not exist a model which satisfies both. Note that the notion of a model "satisfying"' an optimization objective is undefined. Viewed in terms of preferences over models or states, the notion of compliance with an optimization norm that we have defined above requires not just the preferred state/model but all of the other states/models that were available as options before we can determine compliance with an optimization norm. We cannot therefore speak of the inconsistency of an optimization norm with a boolean norm, independent of richer contextual information, in any meaningful sense.

However, an optimization norm o will be deemed to be inconsistent with a boolean norm n in a given context C if and only if every element of the set S (where $S \subseteq C$) of strictly non-dominated (most preferred) options according to o violates n. We assume, as before, that the options in question are actions/tasks, or plans or states of affairs, so that in each case we are able to check for the violation of boolean norms. Note that we can only define inconsistency using intuitions similar to those for *choice consistency* for optimization norms.

We will now consider an example that illustrates these notions of inconsistency, using valued optimization norms. Note that the focus in the example is on the c-semiring valuations associated with states of affair - the associated penalties/sanctions do not play a role in the example (but become critical in the later discussion on sanction management).

Example 2. Recall, from Example 1, the valued optimization norm n_Q using the c-semiring $Q = \langle \{l, m, h\}, \oplus, \otimes, l, h \rangle$ (together with its associated extensional definitions of $\oplus$ and $\otimes$): $n_Q(p \wedge q) = \langle h, 0 \rangle$, $n_Q(\neg p \wedge q) = \langle m, 5 \rangle$, $n_Q(p \wedge \neg q) = \langle m, 5 \rangle$, $n_Q(\neg p \wedge \neg q) = \langle l, 10 \rangle$. Consider another valued optimization norm $n2_Q$ defined (using the same c-semiring Q) as: $n2_Q(p \wedge q) = \langle h, 0 \rangle$, $n2_Q(\neg p \wedge q) = \langle m, 5 \rangle$, $n2_Q(\neg p \wedge \neg q) = \langle m, 5 \rangle$, $n2_Q(p \wedge \neg q) = \langle l, 10 \rangle$. Under the notion of absolute inconsistency, n_Q and $n2_Q$ are inconsistent, since the state $p \wedge \neg q$ is preferred over $\neg p \wedge \neg q$ under n_Q while the opposite preference holds under $n2_Q$. If the current context is defined by these two states, then this also illustrates contextual inconsistency. Consider a third valued optimization norm $n3_Q$ defined on the same c-semiring Q: $n_Q(p \wedge \neg q) = \langle h, 0 \rangle$, $n_Q(p \wedge q) = \langle m, 5 \rangle$, $n_Q(\neg p \wedge \neg q) = \langle m, 5 \rangle$, $n_Q(p \wedge q) = \langle l, 10 \rangle$. If the context is defined by states satisfying $p \wedge q$

and $p \wedge \neg q$, then the norms n_Q and $n3_Q$ are inconsistent under the notion of choice consistency (the strictly non-dominated states under the two norms lead to incosistent states). In the same context, the boolean norm $p \wedge \neg q$ contradicts the valued optimization norm n_Q (note that the set of strictly non-dominated states under n_Q consists of a single state: $p \wedge q$).

The resolution of inconsistency in any theory leads to changing the theory in some way. The logic of the theory change [19] argues that when we are obliged to make changes to a theory, we should try to minimize the extent of change, given that theories encode knowledge or intent or deontic constraints which we should alter as little as possible while still accommodating the change that needs to be implemented. Arguably, the same intuitions apply when we resolve inconsistencies amongst norms.

The three notions of consistency discussed above lead to three corresponding notions of resolution:

Resolving absolute inconsistency: Resolving this type of inconsistency requires that an assertion of the form $s' \prec s$ (*s is preferred to s'*) be removed from one of the pair of optimization norms that are found to be inconsistent. As discussed before, this notion of consistency is in general of little practical value, except in settings where the set of all alternatives that an agent might ever have to select amongst is known a priori (in which case it effectively reduces to contextual consistency).

Resolving contextual inconsistency: This involves the same machinery as in the case of absolute inconsistency. Observe that inconsistency resolution means that one or more of the currently applicable set of optimization norms is relaxed. In the event that inconsistency is detected over a set of optimization norms $O = \{o_1, o_2, \ldots, o_m\}$ where $O_s \subseteq O$ consists of optimization norms which prefer s over s' and $O_{s'} \subseteq O$ is the set of optimization norms which prefer s' over s, then we might use majority as the basis for deciding which of the preferences get relaxed. Thus, if $|O_{s'}| < |O_s|$, we might choose to remove $s \prec s'$ from each element of $O_{s'}$. In other settings, the criteria to determine which optimization norm to relax would be domain-dependent (in mixed-initiative reasoning settings, one might even ask the user for guidance on this). Observe that once inconsistency has been resolved, we could take the union of the resulting set of preference relations to guide choice. A range of other intuitions are explored in computational social choice theory (see [20] for a survey).

Resolving choice inconsistency: In the case of a set of optimization norms generating sets of strictly non-dominated (most preferred) options that do not intersect, we would have to pick the "winning" set of norms (whose top choices would determine an agent's selections). As with contextual inconsistency, we could use majority as the basis for deciding the winners - alternatively, the criteria would be domain-dependent.

In the account of inconsistency resolution above, we had to rely on a representation of an optimization norm in the form of a preference relation. We can explore the resolution of inconsistency between optimization norms using two additional intuitions, which do not require this reduction to a preference relation:

Pareto-optimal solutions: The notion of pareto-optimality, frequently used in decision theory, provides an alternative basis for resolving inconsistency amongst optimization norms. In the following, we will use $o(s)$ to denote the value of the objective o for option s. Given a set of optimization norms $O = \{o_1, o_2, \ldots, o_n\}$, which are viewed uniformly, and without loss of generality, as maximization objectives and a set of options $S = \{s_1, s_2, \ldots, s_m\}$, a *pareto-optimal solution* is typically defined as some $s_i \in S$ such that there exists no $s_j \in S$ for which $o_k(s_j) > o_k(s_i)$, for at least one $o_k \in O$, and for all other $o_i \in O$ where $i \neq k$, $o_i(s_j) \geq o_i(s_i)$. In other words, a pareto-optimal option is one for which there exists no other feasible option which performs strictly better on one objective and at least as well on all of the others. The term *pareto-frontier* is often used to describe the set of all pareto-optimal solutions. In our setting, the pareto-frontier can be viewed as consisting of alternative resolutions of optimization norm inconsistency. These can be presented as choices to users in a mixed-initiative reasoning setting (the system only filters, but does not make the final choice in such settings).

Prioritization of objectives: A well-known approach to dealing with multiple objectives in operations research is to create a weighted sum of these objectives, with the weights reflecting the relative priorities of the corresponding objectives. There is a critical assumption here that the objective functions map options to commensurate scales (what would happen if one objective measured cost in dollars and another measured time in seconds?). If all of the objectives are equally weighted, then each solution to the resulting optimization problem represents an element of the pareto-frontier. In our alternative account of Sen's cake-choice example, let o_{NL} represent the optimization norm that makes us prefer states where we haven't eaten the largest slice of cake to states where we have. Let o_L represent our preference for eating larger slices of a cake. We simplify our discussion by avoiding the formalization of the commensurate scales on which the two objectives would evaluate options. The weighted combination of the two objectives would be of the form $w_{NL}.o_{NL} + w_L.o_L$ where $w_{NL} > w_L$ (for the purposes of our example, the actual weights are immaterial as long as this inequality holds).

We now need to address the question of resolving inconsistencies between optimization norms and boolean norms. The solution here is fairly simple: optimization norms can be relaxed while boolean norms cannot. Recall that an optimization norm o will be deemed to be inconsistent with a boolean norm n in a given context C if and only if every element of the set S (where $S \subseteq C$) of strictly non-dominated (most preferred) options violates n. This inconsistency can be resolved if we are able to promote at least one n-satisfying state (say s) to become a member of S. Given our earlier discussion on the need to minimize change to the original specification of a norm, we could explore several intuitions on the specific changes required in the underlying preference relation (as before, viewed as a set of preference assertions). The general approach would be to identify an o' which satisfies the following conditions: (1) There exists at least one n-satisfying element of the set of strictly non-dominated options under o', (2) There exists no o'' where $(o \Delta o'') \subseteq (o \Delta o')$ which also satisfies condition (1) (here Δ refers to the symmetric set difference operator).

An important question to address is the distinction between the *violation* and *relaxation* of optimization norms. We *relax* an optimization norm to ensure consistency with other norms. The notion of norm compliance discussed earlier continues to provide a clear yardstick for deciding whether an optimization norm (or a set of such norms) has been violated.

6 Sanction Management

We now consider the problem of norm compliance in multi-agent systems, and in particular the problem of how graded sanctions (corresponding to graded degrees of non-compliance) might be *decomposed* and assigned to the (possibly many) agents responsible for a (graded) violation. Social norms may need to be decomposed to obtain agent-specific obligations in a multi-agent context. As discussed earlier, this is a problem of practical importance whenever agents delegate responsibility to other agents. In a more general business setting, a formal understanding of the decomposition of sanctions is critical in managing outsourcing and in formalizing the relationship between a contract and sub-contracts whenever sub-contracting is involved. A formalization is complicated by the fact that the sanctions associated with different levels of compliance/violation are contextually determined, specifically by the number of agents involved in satisfying the norm in question, and the level to which each of these comply with or violate the norm.

In the following, given a set of agents Ag and a set of variables Var, we will use the function $\theta : Ag \rightarrow 2^{Var}$ to map an agent to a set of variables that the agent is responsible for. Also in the following, given a norm n_P which specifies preferences using an underlying semiring P, we will use $bestval_{n_P}(ag)$, to denote the set of preference values assigned to the set of value assignments (states) that include the current assignments of values to variables in $\theta(ag)$ by n_P such that there exists no state s that also includes the current assignment of values to $\theta(ag)$ where $n_P(s) = \langle v', r' \rangle$ and $v' \prec v$ for some $v \in bestval_{n_P}(ag)$ (here $\prec$ is the strict version of the partial order $\preceq$ associated with the c-semiring P).

Definition 3. *Given a multi-agent norm context $\langle n_P, Var, Ag, \theta \rangle$ where n_P is the norm in question, Var is the set of variables over which the states that the norm refers to are defined, Ag is the set of agents jointly responsible for satisfying n_P and the $\otimes$ operator associated with the c-semiring P is idempotent, and given a complete assignment s of values to variables in V where $n_P(s) = \langle v, r \rangle$, the (real-valued) sanction applied on an agent ag_i is defined as:*

- *If $bestval_{n_P}(ag_i) = v$ then the sanction incurred by ag_i is r/m where $Ag_X = \{ag \mid bestval_{n_P}(ag) = v\}$ and $|Ag_X| = m$.*
- *If $bestval_{n_P}(ag_i) \neq v$ then sanction incurred by ag_i is 0.*

The idempotence property of the $\otimes$ operator, which states that $a \otimes a = a$ for any preference value a is key to understanding the definition above. Operators such as *min* and *max* are idempotent, while the arithmetic addition operator $+$ is not. With an idempotent combination operator such as *min*, the contribution of an agent to the final preference value is only of interest if that final preference value equals the best value assigned by the valued optimization norm to the portion of the state determined by that agent. If the best value happens to be higher, then the agent in question was not in fact responsible for the final value of that state (some other agent or agents whose component of the final state was evaluated to the eventual valuation of the state would have been responsible instead). Similarly, if the contribution of the agent had a lower valuation, then the whole state would have had that lower valuation, hence that case is not of interest either.

As an example, consider a compliance requirement that both p and q have to be completely satisfied. Assume that the c-semiring being used is Q as defined above and that both p and q can be satisfied at one of the following 3 levels: COMPLETELY (represented by the c-semiring value h), PARTIALLY (represented by m), NOT-AT-ALL (represented by l). Consider 2 agents: A_1 and A_2. A_1 is responsible for satisfying p. $A_2$is responsible for satisfying q. The compliance requirement is formalized by the valued optimization norm $n1_Q$ as follows: $n1_Q(\langle p = h, q = h\rangle) = (h, 0)$, $n1_Q(\langle p = h, q = m\rangle) = (m, 50)$, $n1_Q(\langle p = m, q = h\rangle) = (m, 50)$, $n1_Q(\langle p = m, q = m\rangle) = (m, 50)$, $n1_Q(\langle p = l, q = l\rangle) = (l, 100)$, $n1_Q(\langle p = h, q = l\rangle) = (l, 100)$ (note that we need a total of 9 such assertions - we do not list them all for brevity). Notice that the preference value assigned to each state is obtained by applying the $\otimes$ operator on the preference values associated with p and q.

– Scenario 1: The requirement is violated completely (incurring a penalty of 100) because A_2 fails to satisfy q completely, i.e. as a NOT-AT-ALL (even though A_1 satisfies p completely). Here A_2 pays a penalty of 100.
– Scenario 2: The requirement is violated partially (incurring a penalty of 50) because A_2 fails to satisfy q partially (even though A_1 satisfies p completely). Here A_1 pays a penalty of 50.
– Scenario 3: The requirement is violated partially (incurring a penalty of 50) because both A_1 and A_2 satisfy q and p (respectively) partially. Here A_1 and A_2 pay a penalty of 25 each

We do not list all of the scenarios here for brevity. The formalization above does not cover the complete sanction decomposition problem, but offers pointers on how the full problem might be solved using an approach like this. In the non-idempotent case, if the $\otimes$ operator performs arithmetic addition of cost, it is easy to decompose penalties to agents in direct proportion to their contribution to costs. In other cases, the complexity might arise due to evaluating the contribution of each agent to an eventual outcome.

7 Related Implementions of Optimization in Agent Deliberation

It is useful, at this point, to consider the implementation of optimization norms in agent systems. The theory of compliance for optimization norms discussed in this paper has been implemented in the CASO BDI agent programming language [21], which provides pointers on how optimization objectives might be integrated into agent deliberation (but it does not support the framework for inconsistency detection and resolution developed in this paper). The key idea is that CASO agents accept optimization objectives as events in their event queues. A CASO agent uses these objectives in option selection (choosing between competing plans)and intention selection. In the absence of a clear notion of termination (common to most BDI agent implementations), a CASO agent cannot look ahead to a final state (as required by the notion of global compliance defined in this paper). Instead, a CASO agent achieves local compliance by pruning the goal-plan tree obtained from its current options at a parametric depth, and then exploring all paths to the pseudo-leaf nodes obtained. In deciding which option to commit to, a CASO agent solves an optimization problem, using the currently applicable optimization objectives, once for each option (CASO plan contexts contain both constrain and non-constraint predicates, leading to a representation akin to constraint logic programming). A CASO agent then selects the option that offers the optimal value with respect to the currently applicable optimization norms. Intention selection in CASO uses similar machinery.

In the BAOP agent programming language [22], both objective functions and (c-semiring) valued preferences are brought to bear on agent deliberation. The c-semiring preferences and specified with respect to domain states. The traditional AgentSpeak-style plans of CASO are therefore annotated with effects, and machinery defined to propagate effects over plans and sub-plans. As with CASO, BAOP provides pointers to how optimization norms might be integrated into agent deliberation, but does not make provision for inconsistency detection and resolution.

The rest of the machinery described in this paper has not been implemented. However, as the discussion above suggests, much of the conceptual framework presented should be possible to implement in a straightforward manner. On the one hand, the need to solve optimization problems during agent deliberation adds to its complexity. On the other hand, this framework provides tantalizing hints on how agent programming environments could be based entirely on (efficient) optimization technology.

8 Conclusions and Future Work

We have argued in this paper that optimization norms represent an important dimension to normative reasoning in multi-agent systems. We have provided a conceptual framework to support reasoning with optimization norms, and extended it to provide support for graded compliance, graded sanctions and sanction decomposition. The observations provided in this paper provide a starting point for an interesting strand of research. Ultimately, this raises the question: are the problems of compliance checking and non-compliance resolution instances of the more general optimization problem?

References

1. Boer, A., van Engers, T., Winkels, R.: Mixing legal and non-legal norms. In: Proceedings of the 2005 Conference on Legal Knowledge and Information Systems: JURIX 2005: The Eighteenth Annual Conference, Amsterdam, The Netherlands, pp. 25–36. IOS Press (2005)
2. van der Torre, L., Tan, Y.H.: Contrary-to-duty reasoning with preference-based dyadic obligations. Annals of Mathematics and AI 27, 49–78 (1989)
3. Frank Dignum, D.K., Sonenberg, L.: From desires, obligations and norms to goals. Cognitive Science Quarterly 2, 407–430 (2002)
4. Panagiotidi, S., Vazquez-Salceda, J.: Norm-aware planning: Semantics and implementation. In: 2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), vol. 3, pp. 33–36 (2011)
5. Mares, E.D., MacNamara, P.: Supererogation in deontic logic: Metatheory for dwe and some close neighbours. Studia Logica 57, 397–415 (1997)
6. Simon, H.A.: Models of bounded rationality. MIT Press, Cambridge (1982)
7. Morrison, E., Ghose, A.K., Koliadis, G.: Dealing with imprecise compliance requirements. In: Proceedings of the 2nd International Workshop on Dynamic and Declarative Business Processes (DDBP 2009). IEEE Computer Society Press (2009)
8. Koliadis, G., Desai, N., Narendra, N.C., Ghose, A.K.: Analyst-mediated contextualization of regulatory policies. In: Proceedings of the IEEE International Services Computing Conference (IEEE SCC 2010), Miami, USA, pp. 281–288. IEEE Computer Society Press (July 2010)
9. Governatori, G.: Representing business contracts in ruleml. International Journal of Cooperative Information Systems 14, 181–216 (2005)
10. Udupi, Y.B., Singh, M.P.: Contract enactment in virtual organizations: A commitment-based approach. In: Proceedings of the 21st National Conference on Artificial Intelligence, AAAI (July 2006)
11. Villatoro, D., Andrighetto, G., Sabater-Mir, J., Conte, R.: Dynamic sanctioning for robust and cost-efficient norm compliance. In: IJCAI, pp. 414–419 (2011)
12. Boella, G., van der Torre, L.W.N.: Negotiating the distribution of obligations with sanctions among autonomous agents. In: Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI 2004, pp. 13–17 (2004)
13. von Wright, G.H.: Deontic logic. Mind 60(237), 1–15 (1951)
14. Elster, J.: Fairness and norms. Social Research: An International Quarterly 73(2), 365–376 (2006)
15. Bistarelli, S., Montanari, U., Rossi, F., Schiex, T., Verfaillie, G., Fargier, H.: Semiring-based csps and valued csps: Frameworks, properties, and comparison. Constraints 4(3), 199–240 (1999)
16. Governatori, G., Milosevic, Z.: An approach for validating bcl contract specifications. In: 2nd EDOC Workshop on Contract Architectures and Languages (CoALA 2005). IEEE Digital Library (2005)
17. Rao, A.S.: AgentSpeak(L): BDI agents speak out in a logical computable language. In: Perram, J.W., Van de Velde, W. (eds.) MAAMAW 1996. LNCS, vol. 1038, Springer, Heidelberg (1996)

18. Sen, A.K.: Internal consistency of choice. Econometrica 61, 495–521 (1993)
19. Alchourron, C., Gardednfors, P., Makinson, D.: On the logic of theory change: Partial meet functions for contraction and revision. Journal of Symbolic Logic 50, 510–530 (1985)
20. Chevaleyre, Y., Endriss, U., Lang, J., Maudet, N.: A Short Introduction to Computational Social Choice. In: van Leeuwen, J., Italiano, G.F., van der Hoek, W., Meinel, C., Sack, H., Plášil, F. (eds.) SOFSEM 2007. LNCS, vol. 4362, pp. 51–69. Springer, Heidelberg (2007)
21. Dasgupta, A., Ghose, A.K.: Implementing reactive bdi agents with user-given constraints and objectives. International Journal of Agent-Oriented Software Engineering 4(2), 141–154 (2010)
22. Dasgupta, A., Ghose, A.K.: BDI Agents with Objectives and Preferences. In: Omicini, A., Sardina, S., Vasconcelos, W. (eds.) DALT 2010. LNCS, vol. 6619, pp. 22–39. Springer, Heidelberg (2011)