Optimal Social Laws

Thomas Ågotnes[*]
Dept of Information Science and Media Studies
University of Bergen
PB. 7802, 5020 Bergen
Norway
thomas.agotnes@infomedia.uib.no

Michael Wooldridge
Dept of Computer Science
University of Liverpool
Liverpool L69 7ZF
UK
mjw@liv.ac.uk

ABSTRACT

Social laws have proved to be a powerful and theoretically elegant framework for coordination in multi-agent systems. Most existing models of social laws assume that a designer is attempting to produce a set of constraints on agent behaviour which will ensure that some single overall desirable objective is achieved. However, this represents a gross simplification of the typical situation, where a designer may have multiple (possibly conflicting) objectives, with different priorities. Moreover, social laws, as well as bringing benefits, also have implementation costs: imposing a social law often cannot be done at zero cost. We present a model of social laws that reflects this reality: it takes into account both the fact that the designer of a social law may have multiple differently valued objectives, and that the implementation of a social law is not cost-neutral. In this setting, designing a social law becomes an optimisation problem, in which a designer must take into account both the benefits and costs of a social law. We investigate the issue of representing a designer's objectives, characterise the complexity of the optimal social law design problem, and consider possible constraints that lead to reductions in computational complexity. We then show how the problem of designing an optimal social law can be formulated as an integer linear program.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; I.2.4 [**Knowledge representation formalisms and methods**]

General Terms

Theory

Keywords

social laws, normative systems, logic, optimisation, complexity

1. INTRODUCTION

Social laws, or normative systems, have proved to be an attractive approach to coordination in multi-agent systems [13, 15, 1, 3, 4]. The basic idea is to manage a social system by placing restrictions on the activities of the agents within the system; the purpose of these restrictions is typically to prevent some destructive interaction from taking place, or to facilitate some positive interaction. In the original framework of Shoham and Tennenholtz [13], the aim of a social law was to restrict the activities of agents so as to ensure that individual agents were not prevented from accomplishing their personal goals. In [15], this idea was generalised to allow for the objective of a social law (i.e, what the designer intends to accomplish with the social law) to be specified as a logical formula. Variations on the same theme have subsequently been explored in a number of papers, e.g., [1, 2]. While these frameworks have proved to be powerful and valuable as a means to explore the computational aspects of social laws in particular, they suffer from several key limiting factors:

- First, it is typically assumed that a social law can be imposed *at no cost*. In most real systems, of course, this is completely unrealistic: different social laws will vary wildly in the cost of their implementation, and this will be a key factor in assessing the relative merits of different social laws.
- Second, it is typically assumed that the designer of a social law has a single overall objective to be achieved, or that if a designer has multiple goals, then these are of equal value. Again, this seems unrealistic in many real-world settings.

Our aim in this paper is to develop and investigate a model of social laws which allows for both the fact that different social laws have different implementation costs, and that the designer of a social law may have multiple differently valued objectives. In this setting, the design of social laws is an *optimisation problem*, where we must take into account both the benefits and costs of different possible social laws: the aim is to find an *optimal social law*.

In the next section we briefly review the models and other formalisms we employ in the paper: systems are modelled using Kripke structures; social laws are modelled as restrictions on such structures; and Computation Tree Logic (CTL) is used to express properties of such structures. We augment Kripke structures with costs on edges in order to model the cost of implementing social laws. In Section 3 we introduce models of the utility of a normative system on the basis of costs and benefits. In particular, we use a *weighted formulas* representation, in the style of marginal contribution nets and related formalisms [8, 9, 14, 6, 14], in order to be able to represent different designer objectives and their values compactly. In Section 4 we characterise the complexity of the optimal social law design problem. The problem is, in general, computationally hard. We consider this from two angles. First, we look at possible constraints that lead to reductions in computational complexity. Second, in Section 5, we look at a way to solve the (general) problem in practice: *integer programming*. Integer programming is one of the most successful and widely-used approaches to solving computationally hard optimisation problems. We show how the problem

[*] Also affiliated with Bergen University College.

Cite as: Optimal Social Laws, Thomas Ågotnes and Michael Wooldridge, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 667-674

of designing an optimal social law can be formulated as an integer program.

2. THE FORMAL FRAMEWORK

The model of social laws we use here is that of [15, 1]; we give a complete but terse summary of the model, referring to the above cited papers for more details.

Weighted Kripke Structures: We use *weighted Kripke structures* as our semantic model for multi-agent systems, which extend conventional Kripke structures for branching-time temporal logic (see, e.g., [7]), with costs. A conventional Kripke structure over a set of Boolean variables Φ is a structure $\langle S, s_0, R, \pi\rangle$, where S is a set of states, $s_0 \in S$ is the initial state, $R \subseteq S \times S$ is a binary transition relation on S, and $\pi : S \to \mathbf{2}^{\Phi}$ is a labelling function, associating with each state in S the set of Boolean variables that are true in that state. We extend these models first with a set A of *agents*, and a function $\alpha : R \to A$, which associates an agent with each transition in R. Intuitively, if we think of an edge $(s, s') \in R$ as corresponding to an action that transforms state s into state s', then $\alpha(s, s')$ is the agent that performs the action. Finally, we add a *cost function*, $c : R \to \mathbb{R}_+$ (where $\mathbb{R}_+$ is the non-negative real numbers), which associates a numeric cost with every transition in the Kripke structure. The intuition is that the cost $c(s, s')$ of an edge (s, s') represents how much it would cost to *remove* the edge. We do not demand a single interpretation for such costs, but several are possible:

- First, we can interpret a cost $c(s, s')$ as representing how much it would cost to *engineer out* the transition (s, s'), i.e., how much it would cost to re-engineer the system so that (s, s') was no longer present.
- Second, we could interpret the cost $c(s, s')$ as representing how much it would cost to "police" the system to ensure that the transition (s, s') was never enacted.

However, we emphasise again that no specific interpretation is required for the purposes of this paper: we simply assume the cost function c is given as part of a weighted Kripke structure.

Formally, a *weighted Kripke structure* (over Φ) is a 7-tuple $K = \langle S, s_0, R, A, \alpha, c, \pi\rangle$ where:

- S is a finite, non-empty set of *states*;
- $s_0 \in S$ is the initial state;
- $R \subseteq S \times S$ is a total (i.e., for every $s \in S$ there is a $t \in S$ such that $(s, t) \in R$) relation on S, which we refer to as the *transition relation*;
- $A = \{1, \ldots, n\}$ is a *set of agents*;
- $\alpha : R \to A$ labels each transition in R with an agent;
- $c : R \to \mathbb{R}_+$ is a *cost* function; and
- $\pi : S \to 2^{\Phi}$ is a valuation function.

In the interests of brevity, we shall sometimes refer to a weighted Kripke structure simply as a *Kripke structure*. Where R is a transition relation and s is a state, let $next(s, R) = \{s' : (s, s') \in R\}$. Let $rch(s, R)$ denote the set of states reachable from state s in transition relation R, i.e., $rch(s, R) = next(s, R^*)$ where R^* is the reflexive transitive closure of R. When R is clear from context, we simplify notation and write $next(s)$ and $rch(s)$. A *path* over a transition relation R is an infinite sequence of states $\tau = s_0, s_1, \ldots$ which must satisfy the property that $\forall u \in \mathbb{N}$: $s_{u+1} \in next(s_u)$. If $u \in \mathbb{N}$, then we denote by $\tau[u]$ the component indexed by u in τ (thus $\tau[0]$ denotes the first element, $\tau[1]$ the second, and so on). A path τ such that $\tau[0] = s$ is an *s-path*. Let $paths_R(s)$ denote the set of s-paths over R; we often omit reference to R, and simply write $paths(s)$. We will refer to and think of an s-path as a possible computation, or system evolution, from s.

In the following we will frequently treat weighted Kripke structures as conventional Kripke structures; obviously, this is done by disregarding the additional components. For example, we presently define the notion of bisimilar Kripke structures, and this notion also applies to weighted structures by viewing them as conventional.

A *bisimulation relation* between two (conventional) Kripke structures $K = \langle S, s_0, R, \pi\rangle$ and $K' = \langle S', s_0', R', \pi'\rangle$ is a binary relation $Z \subseteq S \times S'$ such that for all s and s' such that sZs', (i) $\pi(s) = \pi'(s')$, (ii) for any s_1 such that sRs_1 there is a s_1' such that $s'R's_1'$ and s_1Zs_1' and (iii) for any s_1' such that $s'R's_1'$ there is a s_1 such that sRs_1 and s_1Zs_1'. Two structures are *bisimulation equivalent*, $K \equiv K'$, if there exists a bisimulation relation Z between K and K' such that s_0Zs_0'.

Computation Tree Logic (CTL): CTL is a branching time temporal logic intended for representing the properties of Kripke structures [7]; since CTL is widely documented in the literature, our presentation will be somewhat terse. The syntax of CTL is defined by the following BNF grammar, where $p \in \Phi$:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \lor \varphi \mid \mathsf{E}\bigcirc\varphi \mid \mathsf{E}(\varphi\,\mathcal{U}\,\varphi) \mid \mathsf{A}\bigcirc\varphi \mid \mathsf{A}(\varphi\,\mathcal{U}\,\varphi)$$

The semantics of CTL are given with respect to the satisfaction relation "$\models$", which holds between pairs of the form K, s, (where K is a Kripke structure and s is a state in K), and formulae:

$K, s \models \top$;

$K, s \models p$ iff $p \in \pi(s)$ (where $p \in \Phi$);

$K, s \models \neg\varphi$ iff not $K, s \models \varphi$;

$K, s \models \varphi \lor \psi$ iff $K, s \models \varphi$ or $K, s \models \psi$;

$K, s \models \mathsf{A}\bigcirc\varphi$ iff $\forall\tau \in paths(s) : K, \tau[1] \models \varphi$;

$K, s \models \mathsf{E}\bigcirc\varphi$ iff $\exists\tau \in paths(s) : K, \tau[1] \models \varphi$;

$K, s \models \mathsf{A}(\varphi\,\mathcal{U}\,\psi)$ iff $\forall\tau \in paths(s), \exists u \in \mathbb{N}$, s.t. $K, \tau[u] \models \psi$ and $\forall v, (0 \leq v < u) : K, \tau[v] \models \varphi$

$K, s \models \mathsf{E}(\varphi\,\mathcal{U}\,\psi)$ iff $\exists\tau \in paths(s), \exists u \in \mathbb{N}$, s.t. $K, \tau[u] \models \psi$ and $\forall v, (0 \leq v < u) : K, \tau[v] \models \varphi$

The remaining classical logic connectives ("$\land$", "$\to$", "$\leftrightarrow$") are assumed to be defined as abbreviations in terms of $\neg$, $\lor$, in the conventional manner. The remaining CTL temporal operators are defined: $\mathsf{A}\Diamond\varphi \equiv \mathsf{A}(\top\,\mathcal{U}\,\varphi)$; $\mathsf{E}\Diamond\varphi \equiv \mathsf{E}(\top\,\mathcal{U}\,\varphi)$; $\mathsf{A}\Box\varphi \equiv \neg\mathsf{E}\Diamond\neg\varphi$; $\mathsf{E}\Box\varphi \equiv \neg\mathsf{A}\Diamond\neg\varphi$.

We say φ is *satisfiable* if $K, s \models \varphi$ for some Kripke structure K and state s in K; φ is *valid* if $K, s \models \varphi$ for all Kripke structures K and states s in K. The problem of checking whether $K, s \models \varphi$ for given K, s, φ (*model checking*) can be done in deterministic polynomial time, while checking whether a given φ is satisfiable or whether φ is valid is EXPTIME-complete [7]. We write $K \models \varphi$ if $K, s_0 \models \varphi$, and $\models \varphi$ if $K \models \varphi$ for all K.

Expressiveness of CTL is characterised by bisimulation equivalence: for any K, K', $K \equiv K'$ iff for all φ, $K \models \varphi$ iff $K' \models \varphi$ [5] (note that, unlike for many other modal logics, the implication holds in both directions here).

Social Laws: For our purposes, a *social law*, or a *normative system*, is simply *a set of constraints on the behaviour of agents in a system* [1]. More precisely, a social law defines, for every possible system transition, whether or not that transition is considered to be legal. Formally, a social law η (w.r.t. a Kripke structure $K = \langle S, s_0, R, A, \alpha, c, \pi\rangle$) is a subset of R, such that $R \setminus \eta$ is a total relation. The latter is a *reasonableness* constraint: it prevents social laws which lead to states with no successor. Let $N(R) = \{\eta : (\eta \subseteq R) \text{ and } (R \setminus \eta \text{ is total})\}$ be the set of social laws over R. The intended interpretation of a social law η is that $(s, s') \in \eta$ means transition (s, s') is forbidden in the context of η; hence $R \setminus \eta$ denotes the *legal* transitions of η.

Implementing Social Laws: The effect of *implementing* a social law on a Kripke structure is to eliminate from it all transitions that are forbidden according to this social law (see [15, 1]). If K is a Kripke structure, and η is a social law over K, then $K \dagger \eta$ denotes the Kripke structure obtained from K by deleting transitions forbidden in η. Formally, if $K = \langle S, s_0, R, A, \alpha, c, \pi \rangle$, and $\eta \in N(R)$, then $K \dagger \eta = K'$ is the Kripke structure $K' = \langle S, s_0, R', A, \alpha, c, \pi \rangle$ such that $R' = R \setminus \eta$ and all other components are as in K. We denote by $\hat{K}$ the set of Kripke structures that may be obtained by implementing some social law on K, i.e.,

$$\hat{K} = \{\langle S, s_0, R', A, \alpha, c, \pi \rangle : R' \subseteq R \text{ and } R' \text{ is total}\}.$$

3. OPTIMAL SOCIAL LAWS

The aim of the designer of a social law will typically be to *optimise* the system in some way. For example, the designer may wish to ensure that certain undesirable situations never arise in the system, or that certain positive situations *do* arise. We can think about the preferences of a social law designer over a given system K as being captured by a *valuation function*, which gives a value of every possible sub-system of K:

$$v : \hat{K} \to \mathbb{R}_+.$$

However, while social laws may bring benefits (in terms of the desirable properties they bring about), they also have costs, as captured in the cost function c. The *utility* of a social law η with respect to a Kripke structure K and valuation function v, which we denote by $u(\eta, K, v)$, is then the difference between the value brought by the social law and the cost of implementing it:

$$u(\eta, K, v) = \underbrace{v(K \dagger \eta)}_{\text{benefit}} - \underbrace{\sum_{(s,s') \in \eta} c(s, s')}_{\text{cost}}.$$

From the point of view of a designer with valuation function v, the *optimal* social law $\eta^*(K, v)$ with respect to Kripke structure K and valuation function v will be one that maximises the value of the function u (if there are several maximising systems we let η^* chose arbitrarily among them) :

$$\eta^*(K, v) = \arg \max_{\eta \in N(R)} u(\eta, K, v).$$

The OPTIMAL SOCIAL LAW problem is the problem of computing $\eta^*(K, v)$ – notice that this is a function problem, not a decision problem.

3.1 Feature Sets for Valuation Functions

A key issue from a computational perspective is that of *representing* a valuation function v. The "obvious" representation of a function v is as a set of input/output pairs, i.e., we represent v via the set $\{(K \dagger \eta, v(K \dagger \eta)) : \eta \in N(R)\}$. The problem with this representation is that the number of social laws in $N(R)$ will typically be exponential in the number of system states $|S|$. We thus require a *compact* representation of valuation functions v.

The approach we propose in this paper is to represent a valuation function via *weighted formulae*, in the style of marginal contribution nets and related formalisms [8, 9, 6, 14]. The idea is that a valuation function v is additively decomposed into a set $\mathcal{F}$ of *features*, where a *feature* is a pair (φ, x), with φ being a CTL formula characterising the feature, and $x \in \mathbb{R}_+$ indicating the value of the feature. A *feature set* $\mathcal{F}$ is a set of features:

$$\mathcal{F} = \{(\varphi_1, x_1), \ldots, (\varphi_k, x_k)\}.$$

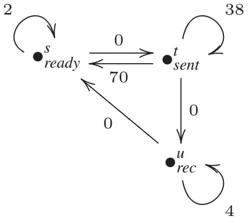
The valuation function $v_{\mathcal{F}}$ induced by a feature set $\mathcal{F}$ is formally defined as follows:

$$v_{\mathcal{F}}(K') = \sum_{(\varphi_i, x_i) \in \mathcal{F}, K' \models \varphi_i} x_i$$

Notice that there is no requirement for features in a feature set to be mutually consistent. Of course, if two features are not mutually consistent (or cannot be simultaneously satisfied in the relevant Kripke structure) then they cannot simultaneously be realised by any social law.

EXAMPLE 1. *A scientist is collecting data in the field, in a remote location. She has a system for transmitting data back to her lab, consisting of three agents: the* sender *at the remote location, the* receiver *at the lab which receives the data, and the* communication channel *which transmits data between the sender and receiver. The system is not perfect. Whenever the sender is* ready*, it is supposed to send a new message to the communication channel. However, it may occasionally fail, and idle in the ready state. The communication channel can deliver the message to the receiver immediately or after a delay, but may occasionally idle. The receiver is supposed to acknowledge the reception of a message immediately to the sender (for simplicity we assume that acknowledgements are transmitted safely). But like the sender, the receiver may occasionally idle. The sender will not be* ready *again until it receives an acknowledgment of the previous message. However, the communication channel may occasionally fail to deliver the message to the receiver and erroneously give the sender an acknowledgment message instead. The scenario is formalised in the model K_T in Figure 1, where proposition* ready *means that the sender is in the ready state,* sent *means that a message has been sent from the sender to the communication channel, and* rec *that a message has been received by the receiver but no acknowledgment has been sent yet. Even with its shortcomings, the system works well most of the time, but the scientist would still prefer if it worked perfectly all of the time. The following are some of the properties she would like the system to be guaranteed to have:*

- $\varphi_1 = \mathsf{E}\Diamond(rec \land \mathsf{E}\Diamond ready)$. *The system* can *transmit successfully and get back to the ready state. This property does not guarantee that the system* always will *transmit successfully, only that it is possible (if the property does not hold, the system will* never *transmit successfully).*
- $\varphi_2 = \mathsf{A}\Box(rec \to \mathsf{A}\Diamond ready)$. *It is always the case that when a message is delivered, the sender will be able to transmit again.*
- $\varphi_3 = \mathsf{A}\Box\mathsf{A}\Diamond ready$. *The sender is ready infinitely often.*
- $\varphi_4 = \mathsf{A}\Box\mathsf{A}\Diamond sent$. *A message is sent infinitely often.*
- $\varphi_5 = \mathsf{A}\Box(sent \to \mathsf{A}\Diamond rec)$. *Every sent message will be received.*
- $\varphi_6 = \mathsf{A}\Box(ready \to \mathsf{A}\Diamond(sent \land \mathsf{A}\Diamond ready))$. *Whenever the sender is ready, it will always be able to eventually send and after that go back to the ready state.*
- $\varphi_7 = \mathsf{A}\Box(\mathsf{A}\Diamond(rec \land \mathsf{A}\Diamond ready))$. *It will always be the case that a new message is eventually received and after that the sender becomes ready again.*



Property	Benefit
φ_1	110
φ_2	15
φ_3	15
φ_4	18
φ_5	10
φ_6	23
φ_7	25

Figure 1: Weighted Kripke structure K_T and feature set $\mathcal{F}_T$ of the transmission example. $s_0 = s$. Transitions are labeled by the cost function. Some of the transitions have zero cost; for example, it will cost nothing to disable the sender.

But she is not willing to improve the system at any cost. The feature set $\mathcal{F}_T$ (Fig. 1) shows our scientist's valuation (in North Pole dollars) of each of the properties.

The cost of "engineering out" behaviours of the system is described on the transitions in Figure 1. For example, the cost of re-engineering the sender agent so that it will always work correctly is 2, and fixing the acknowledgment-without-delivery fault in the communication channel will cost 70.

There are 42 different possible social laws for this model. Which are optimal?

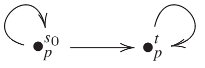
As we shall see in Section 4, finding optimal social laws is in general computationally very hard. The available space here prohibits an exhaustive description and comparison of all the 42 social laws, but with the current feature set we are able to take a shortcut. Observe, first, that it is necessary and sufficient for φ_1 that the transitions (s,t), (t,u) and (u,s) all are legal (not excluded by the social law), and, second, that the benefit of φ_1 alone exceeds the benefit of all the other features combined. From these observations we can see immediately that an optimal social law will never exclude any of (s,t), (t,u) and (u,s). Thus, in this special case we can restrict our attention to the 16 social laws that are subsets of the other four transitions. These are described in Table 1. We see that there are two optimal social laws:

- $\eta_9 = \{(s,s),(u,u)\}$. *The sender and the receiver behaves correctly.*
- $\eta_{13} = \{(s,s),(t,t),(u,u)\}$. *The sender and the receiver behaves correctly, and one of the problems with the communication channel is removed.*

Thus, it is optimal to change the behaviour of the sender and the receiver, as is in addition fixing the idle problem in the communication channel. In any case, it is not optimal to fix the acknowledgment-without-delivery fault in the communication channel. Many of the other options are not only sub-optimal, but are also worse than not implementing any social law at all (i.e., "implementing" η_0).

3.1.1 Representation theorem

We say a feature set $\mathcal{F} = \{(\varphi_1, x_1), \ldots, (\varphi_k, x_k)\}$ *represents* a valuation function v over K whenever $v(K') = v_{\mathcal{F}}(K')$ for all $K' \in \hat{K}$. This raises a natural question: *which* valuation functions can be represented as feature sets? Not all. As an example, let K be the following structure:



If $v(K \dagger \{(s_0,s_0)\}) \neq v(K \dagger \{s_0,t\})$, then there is no feature set representing v, because $K \dagger \{(s_0,s_0)\} \models \varphi$ iff $K \dagger \{(s_0,t)\} \models \varphi$ for any φ. It is clear that if the valuation function gives a different value for structures which cannot be discerned by logical formulae, then it cannot be represented. But the answer to the question is in fact that the valuation functions that can be represented are exactly those that do not discern between bisimulation equivalent structures.

	s,s	t,t	t,s	u,u	φ_1	φ_2	φ_3	φ_4	φ_5	φ_6	φ_7	**Cost**	**Benefit**	**Utility**
η_0	-	-	-	-	+	-	-	-	-	-	-	0	110	110
η_1	-	-	-	+	+	+	-	-	-	-	-	4	125	121
η_2	-	-	+	-	+	-	-	-	-	-	-	70	110	40
η_3	-	-	+	+	+	+	-	-	-	-	-	74	125	51
η_4	-	+	-	-	+	-	-	-	-	-	-	38	110	72
η_5	-	+	-	+	+	+	+	-	-	-	-	42	140	98
η_6	-	+	+	-	+	-	-	-	+	-	-	108	120	12
η_7	-	+	+	+	+	+	+	-	+	-	-	112	150	38
η_8	+	-	-	-	+	-	-	-	-	-	-	2	110	108
$\boldsymbol{\eta_9}$	+	-	-	+	+	+	-	+	-	-	-	6	143	**137**
η_{10}	+	-	+	-	+	-	-	-	-	-	-	72	110	38
η_{11}	+	-	+	+	+	+	-	+	-	-	-	76	143	67
η_{12}	+	+	-	-	+	-	-	-	-	-	-	40	110	70
$\boldsymbol{\eta_{13}}$	+	+	-	+	+	+	+	+	-	+	-	44	181	**137**
η_{14}	+	+	+	-	+	-	-	-	+	-	-	110	120	10
η_{15}	+	+	+	+	+	+	+	+	+	+	+	114	216	102

Table 1: The transmission example: social laws. For the transitions, "+" means that the transition is included in the social law (i.e., that it is *illegal* according to the social law); "−" that the transition is *legal*. For the formulae, "+" ("−") means that the formula is satisfied (not satisfied) if the social law is implemented.

THEOREM 1. *Let K be a (finite) Kripke structure and v an evaluation function over K. There is a feature set $\mathcal{F}$ representing v iff for all K_1, $K_2 \in \hat{K}$:*

$$K_1 \equiv K_2 \Rightarrow v(K_1) = v(K_2)$$

PROOF. For the "if" direction, let $N = \{\eta_1, \ldots, \eta_k\}$ be a "representative" set of social laws over K, such that for any social law η over K there is a $\eta_j \in N$ such that $K \dagger \eta \equiv K \dagger \eta_j$ and such that $K \dagger \eta_j \not\equiv K \dagger \eta_l$ when $j \neq l$ (N is finite since K is). From the latter condition (and the fact that if two structures are not bisimulation equivalent they do not satisfy the same CTL formulae) we have that for any $j, l \leq k, j \neq l$, there is a formula $\varphi_{j,l}$ such that $K \dagger \eta_j \models \varphi_{j,l}$ and $K \dagger \eta_l \not\models \varphi_{j,l}$. We define $\mathcal{F} = \{(\varphi_1, x_1), \ldots, (\varphi_k, x_k)\}$ as follows, for $1 \leq j \leq k$:

$$\varphi_j = \bigwedge_{l \neq j} \varphi_{j,l} \qquad x_j = v(K \dagger \eta_j)$$

It is easy to see that $K \dagger \eta_j \models \varphi_l$ iff $j = l$, and it follows that for $\eta_j \in N$, $v(K \dagger \eta_j) = x_j = \sum_{(\varphi_i,x_i) \in \mathcal{F}: K \dagger \eta_j \models \varphi_i} x_i = v_{\mathcal{F}}(\eta_j)$. For $\eta \notin N$, there is some $\eta_j \in N$ such that $K \dagger \eta \equiv K \dagger \eta_j$, and it follows that $v(K \dagger \eta) = v(K \dagger \eta_j) = x_j = \sum_{(\varphi_i,x_i) \in \mathcal{F}: K \dagger \eta_j \models \varphi_i} x_i = \sum_{(\varphi_i,x_i) \in \mathcal{F}: K \dagger \eta \models \varphi_i} x_i = v_{\mathcal{F}}(\eta)$.

The "only if" direction is immediate: if $v(K_1) \neq v(K_2)$ when $K_1 \equiv K_2$, then K_1 and K_2 satisfy exactly the same formulae and v cannot be represented. □

Thus, as long as the valuation function does not discern between social laws which give the same logical properties after implementation (not an unreasonable property), then it can be represented.

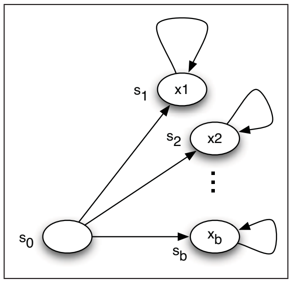


Figure 2: The reduction for Theorem 2.

4. COMPLEXITY

Observe that, given a Kripke structure K and a feature set $\mathcal{F}$, computing $v_{\mathcal{F}}(K)$ can be done in polynomial time. We now have a compact, expressive, and, we believe, quite natural representation for valuation functions, and given this representation, the following question arises: How hard is it to find an optimal social law? In fact, we can precisely characterise the complexity of this problem (see, e.g., [10] for a discussion of the complexity class FP^{NP}).

THEOREM 2. *The* OPTIMAL SOCIAL LAW *problem for the feature set representation of valuation functions is* FP^{NP}*-complete.*

PROOF. For membership in FP^{NP}, note that the associated decision problem (does there exist a social law with utility at least k) is NP-complete, since it subsumes the feasibility of social law design [15, Theorem 2]. All optimization problems whose decision problem is in NP are in FP^{NP} [10, p.416]. For hardness, we reduce the optimization problem MAX WEIGHT SAT [10, p.416]. An instance of MAX WEIGHT SAT is given by a set of propositional clauses $\psi_1, \ldots, \psi_a$, over Boolean variables $x_1, \ldots, x_b$, together with integer weights $w_1, \ldots, w_a$ for each clause. The aim is to find the valuation $\xi \subseteq \{x_1, \ldots, x_b\}$ that maximises the sum of weights of clauses satisfied by the valuation. We construct a Kripke structure K with $l+1$ states, and a transition relation R defined as in Figure 2: s_0 is the only initial state, and inside each state we illustrate the propositional variables true in that state (thus x_1 is only true in state s_1, etc). Notice that no social law can forbid any of the self loops (s_i, s_i), since this would violate the reasonableness requirement for social laws. We create a feature set with a features, one feature for every clause. For a clause $\psi_i = \{\ell_{i_1}, \ldots, \ell_{i_j}\}$ (where each ℓ is either a propositional variable or the negation of a propositional variable) with weight w_i, we create a feature (ψ_i^*, w_i) with ψ_i^* defined as follows:

$$\psi_i^* = \bigvee_{\ell \in \psi_i} \tau(\ell) \qquad \tau(\ell) = \begin{cases} (\mathsf{E}\Diamond x) & \text{if } \ell = x \\ (\neg\mathsf{E}\Diamond x) & \text{if } \ell = \neg x \end{cases}$$

We set costs $c(s, s') = 0$ for all edges (s, s'). Now, any social law will define a valuation under which a variable x_i is considered true if the arc (s_0, s_i) is permitted under that social law. Clearly, any optimal social law will correspond to a valuation that maximises the total weight of clauses satisfied in the MAX WEIGHT SAT instance, and vice versa. □

This is of course a negative result: it basically says that in the worst case we need a polynomial number of queries to an NP oracle in order to compute the optimal social law. Can we do better? Can we identify a class of instances of lower complexity? Let us say an instance $\langle K, \mathcal{F}\rangle$ of the optimal social law problem is *simple* if all feature values in $\mathcal{F}$ are the same, and the cost $c(s, s')$ of every arc (s, s') is 0. Notice that even in this very restricted case, the decision version of the problem is NP-complete, since it subsumes the feasibility of social law design [15, Theorem 2]. However, we have a somewhat more positive complexity result for the optimisation problem:

THEOREM 3. *The* OPTIMAL SOCIAL LAW *problem for simple instances is* $\text{FP}^{\text{NP}[\log_2 |\mathcal{F}|]}$*-complete.*

PROOF. Notice that for simple instances, the optimal social law η^* is simply the one that maximises the number of features in $\mathcal{F}$ that are realised. To show that the function is in $\text{FP}^{\text{NP}[\log_2 |\mathcal{F}|]}$, we must show that it can be computed by a deterministic polynomial time algorithm that is permitted $\log_2 |\mathcal{F}|$ queries to an NP oracle. The idea is to use a binary search to find the exact value for the number of features that can be realised. If $|\mathcal{F}| = c$, then we start by invoking with a bound $\lceil \frac{c+1}{2} \rceil$, and so on, until we converge on the actual value. We will only require $\log_2 |\mathcal{F}|$ queries in order to find the maximum number of features than can be realised. For hardness, we reduce the problem MAX SAT, the unweighted version of MAX WEIGHT SAT [10, p.423]. An instance of MAX SAT is given by a set of clauses $\psi_1, \ldots, \psi_a$, over Boolean variables $x_1, \ldots, x_b$, and we are simply asked to find the assignment that maximises the number of clauses satisfied by the assignment. The reduction is based on that of Theorem 2, except that we simply set all feature weights to 1, and we need to change the definition of ψ^* so that no feature is satisfied in the initial Kripke structure. We can do this by creating another state s', with arcs (s_0, s') and (s', s'), a dummy variable d true in state s', and finally conjoining the formula $(\neg\mathsf{E}\bigcirc d)$ to feature formulae. □

The assumption that all costs are 0, of course, is rather strong. Let us say an instance $\langle K, \mathcal{F}\rangle$ of the optimal social law problem is *homogeneous* if all if all feature values in $\mathcal{F}$ are the same and all costs are the same. We have:

THEOREM 4. *The* OPTIMAL SOCIAL LAW *problem for homogeneous instances is in* $\text{FP}^{\text{NP}[|R| \log_2 |\mathcal{F}|]}$ *(where R is the transition relation in the input instance).*

PROOF. Let $\langle K, \mathcal{F}\rangle$ be the homogeneous problem instance, let x be the edge weight on the input instance, and y be the cost weight. Now, define a simple instance $\langle K', \mathcal{F}\rangle$ that is identical to $\langle K, \mathcal{F}\rangle$ except that all edge costs are 0. We use the fact that, given some natural number f, $1 \leq f \leq |R|$ and simple instance $\langle K', \mathcal{F}\rangle$, the problem of computing the optimal social law η^* such that $|\eta^*| = f$ for a simple instance is $\text{FP}^{\text{NP}[\log |\mathcal{F}|]}$-complete, by essentially the same argument as in Theorem 3. Now, for each f, $1 \leq f \leq |R|$, we ask for the optimal social law containing exactly f edges, giving us a sequence of social laws $\eta_1^*, \ldots, \eta_f^*$; so now we simply find which of these maximises utility. The overall optimal social law for $\langle K, \mathcal{F}\rangle$ will be in this sequence. Since $f = |R|$, the overall number of queries to the NP oracle required is thus $|R| \log_2 |\mathcal{F}|$. □

4.1 Tractable Instances

In this section we discuss instances of the optimal social law problem which can be easily solved.

Dichotomous Valuations: Consider the case that the designer of the system can identify a set of "*bad*" states, and that his (only) goal is to prevent the system from going into a bad state without excluding *any* of the "*good*" states (i.e., the states that are not "bad")[1].

[1] The notion of good/bad, or red/green, states can be seen as an alternative, or complement, to the notion of legal/illegal transitions for modelling normative systems [12].

Formally, given a weighted Kripke structure K, we say that a valuation function v is *dichotomous* if there is a set of states $B \subseteq S$ (the "bad" states) such that for any K' (recall that $rch(s)$ denotes the set of states reachable from s)

$$v(K') = \begin{cases} a & rch(s_0) = S \setminus B \\ 0 & \text{otherwise} \end{cases}$$

for some value $a \geq 0$, effectively assigning the benefit a to any social law excluding exactly the bad states B and zero benefit to all other (not eliminating *all* "bad" states, or eliminating some of the "good" states, or both). In the OPTIMAL SOCIAL LAW problem for dichotomous valuations, we assume that the valuation function v is represented by the set B and the value $a \geq 0$.

THEOREM 5. *The* OPTIMAL SOCIAL LAW *problem for dichotomous valuations can be decided in polynomial time.*

PROOF. Let K, B, a be given, and let $\overline{B} = S \setminus B$. Let $\eta' = \{(s,t) \in R : s \in \overline{B}, t \in B\}$. Define a social law $\hat{\eta}$ over K:

$$\hat{\eta} = \begin{cases} \eta' & \text{if } \eta' \in N(R) \text{ and } u(\eta', K, v) > 0 \\ \emptyset & \text{otherwise} \end{cases}$$

It is immediate from the definition that $\hat{\eta} \in N(R)$. We argue that $\hat{\eta}$ is an optimal social law. Assume otherwise, i.e., that there exists a $\eta \in N(R)$ such that $u(\eta, K, v) > u(\hat{\eta}, K, v)$. We first argue that

$$rch(s_0, R \setminus \eta) = \overline{B} \Rightarrow \eta' \subseteq \eta. \tag{1}$$

If there is a $(s,s') \in R$ such that (i) $(s,s') \in \eta'$ and (ii) $(s,s') \notin \eta$, then $s \in \overline{B}$ by (i) and if $s \in rch(s_0, R \setminus \eta)$ then also $s' \in rch(s_0, R \setminus \eta)$ by (ii) – but $s' \notin \overline{B}$ by (i).

The main argument is by cases in the definition of $\hat{\eta}$; $\eta' \in N(R)$ and $u(\eta', K, v) > 0$. Assume the first case. Since the costs are non-negative, $u(\eta', K, v) > 0$ entails that $v(K \dagger \eta') = a$ and $a > 0$, and from $u(\eta, K, v) > u(\hat{\eta}, K, v)$ it follows that also $v(K \dagger \eta) = a$. Thus, $\sum_{(s,s') \in \eta'} c(s,s') > \sum_{(s,s') \in \eta} c(s,s')$. It follows that $\eta' \not\subseteq \eta$, and thus that $rch(s_0, R \setminus \eta) \neq \overline{B}$ by (1). But from the facts that $v(K \dagger \eta) = a$ and that $a > 0$ it follows that $rch(s_0, R \setminus \eta) = \overline{B}$; a contradiction.

Assume now the second case in the definition of $\hat{\eta}$. $\hat{\eta} = \emptyset$. We have that (a) $u(\eta, K, v) = v(K \dagger \eta) - \sum_{(s,s') \in \eta} c(s,s') > u(\hat{\eta}, K, v) = v(K)$, and the only possibility is that $v(K \dagger \eta) = a$ and $a > 0$, and it follows that $rch(s_0, R \setminus \eta) = \overline{B}$ and from (1) that $\eta' \subseteq \eta$. First assume that $\eta' \notin N(R)$. That means that $R \setminus \eta'$ is not total, i.e., that there is a state $s \in \overline{B}$ such that for all $(s,t) \in R$, $t \in B$. It cannot be the case that $rch(s_0, R) = \overline{B}$, because $s \in \overline{B}$ and every successor of s is not in $\overline{B}$. Thus, $v(K) = 0$, and $v(K \dagger \eta) > \sum_{(s,s') \in \eta} c(s,s')$. The only possibility is that $v(K \dagger \eta) = a$, and $rch(s_0, R \setminus \eta) = \overline{B}$. Since $s \in \overline{B}$ and $t \notin \overline{B}$ for every $(s,t) \in R$, we must have that $(s,t) \in \eta$ for every $(s,t) \in R$ – but then η is not total, a contradiction. Second, assume that $\eta' \in N(R)$ and $u(\eta', K, v) \leq 0$, i.e., that (b) $v(K \dagger \eta') \leq \sum_{(s,s') \in \eta'} c(s,s')$. By the fact that $\eta' \subseteq \eta$ we have that (c) $\sum_{(s,s') \in \eta'} c(s,s') \leq \sum_{(s,s') \in \eta} c(s,s')$. It follows from (a) that $v(K \dagger \eta) - v(K) > \sum_{(s,s') \in \eta} c(s,s')$, and thus from (b) and (c) that $v(K \dagger \eta) - v(K) > v(K \dagger \eta')$. Since $v(K \dagger \eta) = a$, the only possibility is that $v(K \dagger \eta') = v(K) = 0$. But since $rch(s_0, R \setminus \eta') = \overline{B}$ (see start of the second case), that means that $a = 0$. But that contradict the fact that $a = v(K \dagger \eta) > v(K \dagger \eta')$.

It is easy to see that η' can be constructed in $O(|R| \times |B|)$ time, and $\eta' \in N(R)$ checked in $O(|S| \times |R| \times |B|)$ time. $u(\eta', K, v)$ is found by constructing $rch(s_0, R \setminus \eta')$ and $\sum_{(s,s') \in \eta'} c(s,s')$. $rch(s_0, R \setminus \eta')$ can be constructed in polynomial time by unravelling the Kripke structure to a tree while only expanding each state once. □

Composite Valuations: A very simple special case is when the benefit of a social law can be seen as being composed of, or being the sum of, the benefit of removing each transition in the social law (a positive correspondent to the cost function), i.e., when there is a function $v' : R \to \mathbb{R}_+$ such that for all η

$$v(K \dagger \eta) = \sum_{(s,s') \in \eta} v'(s,s')$$

We call such valuation functions v *composite* valuations. We have that $u(\eta, K, v) = \sum_{(s,s') \in \eta} (v'(s,s') - c(s,s'))$. Note that $v'(s,s') - c(s,s')$ might be *negative*. If negative *costs* were allowed (which they are not in our general setting), this setting would be equivalent to one without a valuation function ($v(K') = 0$ for all K'). The OPTIMAL SOCIAL LAW problem for composite valuations takes as input K and the function $v' : R \to \mathbb{R}_+$. It is easy to see that:

THEOREM 6. *The* OPTIMAL SOCIAL LAW *problem for composite valuations can be decided in polynomial time.*

5. ILP FOR OPTIMAL SOCIAL LAWS

Finding an optimal social law is an NP-hard optimisation problem. This suggests that approaches for solving such optimisation problems may usefully be applied to the problem of synthesising optimal social laws. *Integer programming* is one of the most successful and widely-used approaches to solving computationally hard optimisation problems. In this section, we will show the optimal social law problem with the feature set representation can be solved through integer programming. Formally, given an instance of the OPTIMAL SOCIAL LAW problem, we produce an *Integer-Linear Program* (ILP) such that solutions to the ILP define solutions to the given OPTIMAL SOCIAL LAW instance.

Before we start, we must state some assumptions and give some auxiliary definitions. One assumption is that all CTL formulae given in the input instance have been re-written so that the only Boolean connectives used are $\neg$ and $\vee$, and the only temporal operators are $\mathsf{E}\bigcirc$, $\mathsf{E}\mathcal{U}$, and $\mathsf{E}\square$. We emphasise that these connectives provide a complete basis for CTL, and so this assumption does not in any way represent a restriction on input instances; but it greatly simplifies subsequent presentation.

Intuitively, the ILP we define labels states in the Kripke structure with the formulae that are true at these states. The ILP construction we use to label states with the formulae true in those states is derived from the semantics of CTL formulae. The basis of the labeling is provided by the valuation function π, which tells us what atomic propositions are true in what states; the labeling for a formula φ in a state s is then obtained from the labeling of sub-formulae of φ in s and other states. The key idea in the construction is show how this labeling can be encoded in a ILP. One issue is that of dealing with the temporal connectives. The idea we use is to exploit the *fixpoint* nature of this operators. For example, the following is a well-known equivalence in CTL, which tells us that $\mathsf{E}(\psi \mathcal{U} \chi)$ is defining a least fixpoint [7, p.1040]:

$$\mathsf{E}(\psi \mathcal{U} \chi) \leftrightarrow (\chi \vee (\psi \wedge \mathsf{E}\bigcirc \mathsf{E}(\psi \mathcal{U} \chi))).$$

The labeling of formulae $\mathsf{E}(\psi \mathcal{U} \chi)$ in a state s in the ILP is thus derived from the labeling of the formulae χ and $\psi \wedge \mathsf{E}\bigcirc \mathsf{E}(\psi \mathcal{U} \chi)$ in state s.

Let us denote the *closure* of a CTL formula φ by $cl(\varphi)$, and define the function $cl(\cdots)$ as follows:

$$cl(\varphi) = \{\varphi\} \cup cl_0(\varphi)$$

maximize:

$$\sum_{(\varphi_i,x_i)\in\mathcal{F}} \tau(\varphi_i,s_0)\cdot x_i - \sum_{(s,s')\in R} \eta(s,s')\cdot c(s,s') \tag{2}$$

subject to constraints:

$$\begin{array}{rcll}
\tau(\psi,s) & \in & \{0,1\} & \\
 & & \forall\psi\in cl(\mathcal{F}),\, s\in S & (3)\\
\eta(s,s') & \in & \{0,1\} & \\
 & & \forall(s,s')\in R & (4)\\
\sum_{s'\in next(s)} \big(1-\eta(s,s')\big) & \geq & 1 & \\
 & & \forall s\in S & (5)\\
\tau(p,s) & = & \begin{cases} 1 & \text{if } p\in\pi(s)\\ 0 & \text{otherwise}\end{cases} & \\
 & & \forall p\in\Phi\cap cl(\mathcal{F}),\, s\in S & (6)\\
\tau(\neg\psi,s) & = & 1-\tau(\psi,s) & \\
 & & \forall\neg\psi\in cl(\mathcal{F}),\, s\in S & (7)\\
\tau(\psi\vee\chi,s) & \leq & \tau(\psi,s)+\tau(\chi,s) & \\
 & & \forall\psi\vee\chi\in cl(\mathcal{F}),\, s\in S & (8)\\
\tau(\psi\vee\chi,s) & \geq & \tau(\psi,s) & \\
 & & \forall\psi\vee\chi\in cl(\mathcal{F}),\, s\in S & (9)\\
\tau(\psi\vee\chi,s) & \geq & \tau(\chi,s) & \\
 & & \forall\psi\vee\chi\in cl(\mathcal{F}),\, s\in S & (10)
\end{array}$$

Figure 3: ILP for the Optimal Social Law problem (1/4).

where

$$cl_0(\varphi) = \begin{cases} cl(\psi)\cup cl(\chi) & \text{if } \varphi=\psi\vee\chi \text{ or } \varphi=\mathsf{E}(\psi\,\mathcal{U}\,\chi)\\ cl(\psi) & \text{if } \varphi=\neg\psi \text{ or } \varphi=\mathsf{E}\bigcirc\psi \text{ or } \varphi=\mathsf{E}\square\psi\\ \{\varphi\} & \text{if } \varphi\in\Phi. \end{cases}$$

Where $\mathcal{F}=\{(\varphi_1,x_1),\ldots,(\varphi_k,x_k)\}$ is a feature set, we let:

$$cl(\mathcal{F}) = cl(\varphi_1)\cup\cdots\cup cl(\varphi_k).$$

Intuitively, $cl(\mathcal{F})$ is the set of formulae whose truth or falsity we must label against states in the ILP.

The ILP we produce from an Optimal Social Law instance $\langle K,\mathcal{F}\rangle$ is defined in Figures 3–6. Let $soln(K,\mathcal{F})$ denote the set of solutions for the ILP defined in Figures 3–6 for Optimal Social Law instance $\langle K,\mathcal{F}\rangle$. Now, solutions $\sigma\in soln(K,\mathcal{F})$ define values for the variables $\eta(s,s')$ for all transitions (s,s') in K. Where $\sigma\in soln(K,\mathcal{F})$, define a social law η_σ as follows: $(s,s')\in\eta_\sigma$ iff $\eta(s,s')=1$. We then have the following:

Theorem 7. *The* ILP *defined in Figures 3–6 correctly computes solutions to the* Optimal Social Law *problem. Formally, let $\langle K,\mathcal{F}\rangle$ be an instance of the* Optimal Social Law *problem. Then $\sigma\in soln(K,\mathcal{F})$ iff the social law η_σ is a solution to the* Optimal Social Law *problem $\langle K,\mathcal{F}\rangle$.*

Proof. Let $\langle K,\mathcal{F}\rangle$ and $\sigma\in soln(K,\mathcal{F})$ be as stated in the proposition. The ILP makes use of the following key variables:

- For each $\varphi\in cl(\mathcal{F})$ and state s in K, the variable $\tau(\varphi,s)\in\{0,1\}$ will indicate whether formula φ is true ($\tau(\varphi,s)=1$) or false ($\tau(\varphi,s)=0$) in state s of $K\dagger\eta_\sigma$.

$$\begin{array}{rcll}
d(\psi,s,s') & \in & \{0,1\} & \\
 & & \forall\mathsf{E}\bigcirc\psi\in cl(\mathcal{F}),\, s\in S,\, s'\in next(s) & (11)\\
d(\psi,s,s') & \geq & \tau(\psi,s')-\eta(s,s') & \\
 & & \forall\mathsf{E}\bigcirc\psi\in cl(\mathcal{F}),\, s\in S,\, s'\in next(s) & (12)\\
d(\psi,s,s') & \leq & \tau(\psi,s') & \\
 & & \forall\mathsf{E}\bigcirc\psi\in cl(\mathcal{F}),\, s\in S,\, s'\in next(s) & (13)\\
d(\psi,s,s') & \leq & 1-\eta(s,s') & \\
 & & \forall\mathsf{E}\bigcirc\psi\in cl(\mathcal{F}),\, s\in S,\, s'\in next(s) & (14)\\
\tau(\mathsf{E}\bigcirc\psi,s) & \geq & d(\psi,s,s') & \\
 & & \forall\mathsf{E}\bigcirc\psi\in cl(\mathcal{F}),\, s\in S,\, s'\in next(s) & (15)\\
\tau(\mathsf{E}\bigcirc\psi,s) & \leq & \sum_{s'\in next(s)} d(\psi,s,s') & \\
 & & \forall\mathsf{E}\bigcirc\psi\in cl(\mathcal{F}),\, s\in S & (16)
\end{array}$$

Figure 4: ILP for the Optimal Social Law problem (2/4).

- For each edge (s,s') in the transition relation of K, the variable $\eta(s,s')\in\{0,1\}$ will be used to indicate whether the edge (s,s') is forbidden in the optimal social law ($\eta(s,s')=1$) or not forbidden ($\eta(s,s')=0$) in η_σ.

Notice that the variables $\tau(\varphi,s)$ and $\eta(s,s')$ take values from $\{0,1\}$. In the objective function, we also use the edge costs $c(s,s')$ from the Kripke structure and feature values x_i from the feature set $\mathcal{F}$: these are constants in the ILP, so linearity is not violated.

First, we note that the social law η_σ defined by σ is indeed a social law (the "coherence" requirement): this is by constraint (5).

Next, we claim that the variables $\tau(\cdots)$ correctly label states with the formulae that are true in those states in the Kripke structure $K\dagger\eta_\sigma$, i.e., $\forall\varphi\in cl(\mathcal{F})$ and $s\in S$ we have $\tau(\varphi,s)=1$ iff $K\dagger\eta_\sigma,s\models\varphi$. The proof is by induction on the structure of formulae. The inductive base is for atomic propositions Φ, and follows from constraint (6). For the inductive step, we reason by cases:

- $\varphi=\neg\chi$: from constraint (7).
- $\varphi=\psi\vee\chi$: from constraints (8)–(10). Constraint (8) ensures that if both disjuncts are false in state s, then $\tau(\psi\vee\chi,s)=0$, while constraints (9) and (10) ensure that if either disjunct is true in state s, then $\tau(\psi\vee\chi,s)=1$.
- $\varphi=\mathsf{E}\bigcirc\psi$: from constraints (11)–(16). These constraints use subsidiary variables $d(\psi,s,s')$, such that $d(\psi,s,s')=1$ iff $\tau(\psi,s')=1$ and $\eta(s,s')=0$.
- $\varphi=\mathsf{E}(\psi\,\mathcal{U}\,\chi)$: from constraints (17)–(27). These constraints make use of subsidiary variables $e(\psi,\chi,s)\in\{0,1\}$ and $f(\psi,\chi,s,s')\in\{0,1\}$. These variables are defined s.t.:
  - $e(\psi,\chi,s)=1$ iff $K\dagger\eta_\sigma,s\models\psi\wedge\mathsf{E}\bigcirc\mathsf{E}(\psi\,\mathcal{U}\,\chi)$: constraints (25)–(27)
  - $f(\psi,\chi,s,s')=1$ iff both $K\dagger\eta_\sigma,s'\models\mathsf{E}(\psi\,\mathcal{U}\,\chi)$ and $\eta(s,s')=0$: constraints (22)–(24).
- $\varphi=\mathsf{E}\square\psi$: from constraints (28)–(37). These constraints make use of subsidiary variables $g(\psi,s)\in\{0,1\}$ and $h(\psi,s,s')\in\{0,1\}$. These variables are defined s.t.:
  - $g(\psi,s)=1$ iff $K\dagger\eta_\sigma,s\models\mathsf{E}\bigcirc\mathsf{E}\square\psi$: constraints (36)–(37)
  - $h(\psi,s,s')=1$ iff both $K\dagger\eta_\sigma,s'\models\mathsf{E}\square\psi$ and $\eta(s,s')=0$: constraints (33)–(35).

Finally, we claim that the social law η_σ maximises utility: this follows from the objective function (2). □

$$
\begin{aligned}
e(\psi,\chi,s) &\in \{0,1\} \quad \forall \mathsf{E}(\psi\,\mathcal{U}\,\chi) \in cl(\mathcal{F}), s \in S && (17)\\
f(\psi,\chi,s,s') &\in \{0,1\}\\
&\qquad \forall \mathsf{E}(\psi\,\mathcal{U}\,\chi) \in cl(\mathcal{F}), s \in S, s' \in next(s) && (18)\\
\tau(\mathsf{E}(\psi\,\mathcal{U}\,\chi),s) &\geq \tau(\chi,s)\\
&\qquad \forall \mathsf{E}(\psi\,\mathcal{U}\,\chi) \in cl(\mathcal{F}), s \in S && (19)\\
\tau(\mathsf{E}(\psi\,\mathcal{U}\,\chi),s) &\geq e(\psi,\chi,s)\\
&\qquad \forall \mathsf{E}(\psi\,\mathcal{U}\,\chi) \in cl(\mathcal{F}), s \in S && (20)\\
\tau(\mathsf{E}(\psi\,\mathcal{U}\,\chi),s) &\leq \tau(\psi,s) + e(\psi,\chi,s)\\
&\qquad \forall \mathsf{E}(\psi\,\mathcal{U}\,\chi) \in cl(\mathcal{F}), s \in S && (21)\\
f(\psi,\chi,s,s') &\geq \tau(\mathsf{E}(\psi\,\mathcal{U}\,\chi),s') - \eta(s,s')\\
&\qquad \forall \mathsf{E}(\psi\,\mathcal{U}\,\chi) \in cl(\mathcal{F}), s \in S, s' \in next(s) && (22)\\
f(\psi,\chi,s,s') &\leq \tau(\mathsf{E}(\psi\,\mathcal{U}\,\chi),s')\\
&\qquad \forall \mathsf{E}(\psi\,\mathcal{U}\,\chi) \in cl(\mathcal{F}), s \in S, s' \in next(s) && (23)\\
f(\psi,\chi,s,s') &\leq 1 - \eta(s,s')\\
&\qquad \forall \mathsf{E}(\psi\,\mathcal{U}\,\chi) \in cl(\mathcal{F}), s \in S, s' \in next(s) && (24)\\
e(\psi,\chi,s) &\leq \tau(\psi,s)\\
&\qquad \forall \mathsf{E}(\psi\,\mathcal{U}\,\chi) \in cl(\mathcal{F}), s \in S && (25)\\
e(\psi,\chi,s) &\leq \sum_{s' \in next(s)} f(\psi,\chi,s,s')\\
&\qquad \forall \mathsf{E}(\psi\,\mathcal{U}\,\chi) \in cl(\mathcal{F}), s \in S, s' \in next(s) && (26)\\
e(\psi,\chi,s) &\geq 1 - ((1-\tau(\psi,s)) + (1-(f(\psi,\chi,s,s'))))\\
&\qquad \forall \mathsf{E}(\psi\,\mathcal{U}\,\chi) \in cl(\mathcal{F}), s \in S, s' \in next(s) && (27)
\end{aligned}
$$

Figure 5: ILP for the OPTIMAL SOCIAL LAW problem (3/4).

6. CONCLUSIONS AND FURTHER WORK

In this paper, we have modelled the trade-offs between the costs and benefits of implementing a social law, and we have formulated the problem of designing an optimal social law as an optimisation problem. We have characterised the computational complexity of the problem, and have shown how the optimisation problem can be solved using integer linear programming, an approach that has been successfully and widely used to tackle computationally hard problems in other domains. We have also identified some tractable instances, and characterised the expressiveness of using a compact logical representation of social law features.

An opportunity for further research is to investigate the relationship between particular classes of weighted Kripke structures and feature sets on the one hand, and known special types of integer programming problems on the other. In particular, we are interested in whether there are interesting classes that correspond to integer programming problems known to be efficiently solvable [11].

$$
\begin{aligned}
g(\psi,s) &\in \{0,1\} \quad \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}), s \in S && (28)\\
h(\psi,s,s') &\in \{0,1\}\\
&\qquad \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}), s \in S, s' \in next(s) && (29)\\
\tau(\mathsf{E}\Box\psi,s) &\leq \tau(\psi,s)\\
&\qquad \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}), s \in S && (30)\\
\tau(\mathsf{E}\Box\psi,s) &\leq g(\psi,s)\\
&\qquad \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}), s \in S && (31)\\
\tau(\mathsf{E}\Box\psi,s) &\geq 1 - ((1-\tau(\psi,s)) + (1-g(\psi,s)))\\
&\qquad \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}), s \in S && (32)\\
h(\psi,s,s') &\leq 1 - \eta(s,s')\\
&\qquad \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}), s \in S, s' \in next(s) && (33)\\
h(\psi,s,s') &\leq \tau(\mathsf{E}\Box\psi,s')\\
&\qquad \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}), s \in S, s' \in next(s) && (34)\\
h(\psi,s,s') &\leq 1 - (\eta(s,s') + (1-\tau(\mathsf{E}\Box\psi,s')))\\
&\qquad \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}), s \in S, s' \in next(s) && (35)\\
g(\psi,s) &\leq \sum_{s' \in next(s)} h(\psi,s,s')\\
&\qquad \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}), s \in S && (36)\\
g(\psi,s) &\geq h(\psi,s,s')\\
&\qquad \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}), s \in S, s' \in next(s) && (37)
\end{aligned}
$$

Figure 6: ILP for the OPTIMAL SOCIAL LAW problem (4/4).

7. REFERENCES

[1] T. Ågotnes, W. van der Hoek, J. A. Rodriguez-Aguilar, C. Sierra, and M. Wooldridge. On the logic of normative systems. In *Proceedings of IJCAI-07*, 2007.

[2] T. Ågotnes, W. van der Hoek, and M. Wooldridge. Normative system games. In *Proceedings of AAMAS-07*, 2007.

[3] G. Boella and L. Torre. Delegation of power in normative multiagent systems. In *Proceedings of DEON-06*, 2006.

[4] G. Boella and L. Torre. Institutions with a hierarchy of authorities in distributed dynamic environments. *Artificial Intelligence and Law*, 16(1):53–71, 2008.

[5] M. C. Browne, E. M. Clarke, and O. Grümberg. Characterizing finite Kripke structures in propositional temporal logic. *Theoretical Computer Science*, 59, 1988.

[6] E. Elkind, L. A. Goldberg, P. Goldberg, and M. Wooldridge. A tractable and expressive class of marginal contribution nets and its applications. *Mathematical Logic Quarterly*, 55(4):362–376, 2009.

[7] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science Volume B: Formal Models and Semantics*, pages 996–1072. Elsevier Science Publishers B.V.: Amsterdam, 1990.

[8] S. Ieong and Y. Shoham. Marginal contribution nets: A compact representation scheme for coalitional games. In *Proceedings of Electronic Commerce (EC-05)*, 2005.

[9] J. Lang, U. Endriss, and Y. Chevaleyre. Expressive power of weighted propositional formulas for cardinal preference modelling. In *Proceedings of KR-06*, 2006.

[10] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley: Reading, MA, 1994.

[11] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization*. Prentice Hall International, England, 1982.

[12] M. Sergot and R. Craven. The deontic component of action language *nc*+. In L. Goble and J.-J. Meyer, editors, *Deontic Logic and Artificial Normative Systems*, *LNCS* 4048, 2006.

[13] Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies. In *Proc. of AAAI-92*.

[14] J. Uckelman, Y. Chevaleyre, U. Endriss, and J. Lang. Representing utility functions via weighted goals. *Mathematical Logic Quarterly*, 55(4):341–361, 2009.

[15] W. van der Hoek, M. Roberts, and M. Wooldridge. Social laws in alternating time: Effectiveness, feasibility, and synthesis. *Synthese*, 156(1):1–19, 2007.