Social Laws for Multi-Agent Systems: Logic and Games

Lecture 5: Social laws design as an optimisation problem, and as amechanism design problem

Thomas Ågotnes[1]

[1] Department of Information Science and Media Studies
University of Bergen, Norway

NII Tokyo 17 January 2012

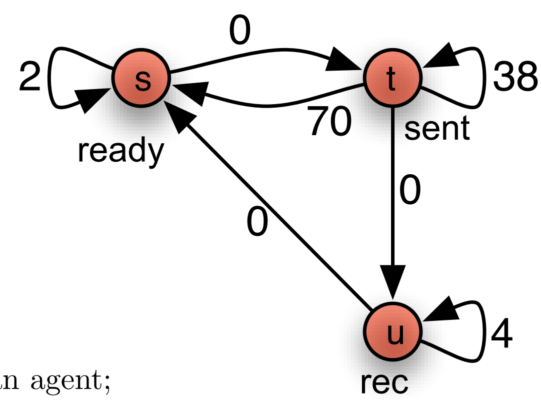
Contents

Optimal Social Laws

- Let us consider the following natural aspects of social laws:
    1. social laws have implementation costs
    2. the designer might have several objectives, with different priorities
- Finding a social law then becomes an optimisation problem
- Issues: representation; computational complexity; practical solving

Adding costs

$$K = \langle S, s_0, R, A, \alpha, c, \pi \rangle$$



- 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) *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.

Idea: the cost of removing the transition

The **value** of social laws

- Assumption: the designer has a valuation function

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

$$\hat{K} = \{K \dagger \eta : \eta \text{ is a social law over } K\}$$

Monday, January 16, 12

The **value** of social laws

- Assumption: the designer has a valuation function

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

- The utility of implementing a social law:

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

$$\hat{K} = \{K \dagger \eta : \eta \text{ is a social law over } K\}$$

Monday, January 16, 12

The **optimal social law** problem

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

The **optimal social law** problem

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

- The optimal social law problem: find:

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

Compact representation of the valuation function

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

Compact representation of the valuation function

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

- Explicit representation: typically *exponential* in the number of states
- Unrealistic; a more compact representations are needed
- We use weighted formulae, in the style of (e.g.) marginal contribution nets

Compact representation of the valuation function

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

- Explicit representation: typically *exponential* in the number of states
- Unrealistic; a more compact representations are needed
- We use weighted formulae, in the style of (e.g.) marginal contribution nets
- A feature set is a set of CTL formula/value pairs

Monday, January 16, 12

Compact representation of the valuation function

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

- Explicit representation: typically *exponential* in the number of states
- Unrealistic; a more compact representations are needed
- We use weighted formulae, in the style of (e.g.) marginal contribution nets
- A feature set is a set of CTL formula/value pairs

$$\mathcal{F} = \{(\phi_1, x_1), \ldots, (\phi_k, x_k)\}$$

Monday, January 16, 12

Compact representation of the valuation function

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

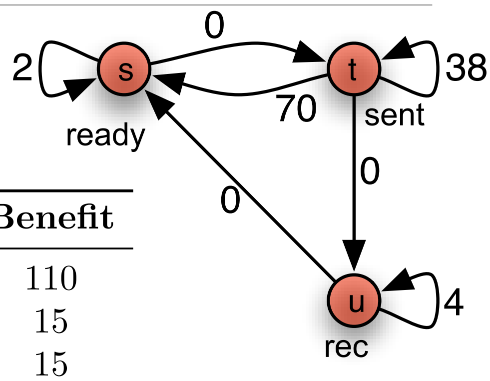
- Explicit representation: typically *exponential* in the number of states
- Unrealistic; a more compact representations are needed
- We use weighted formulae, in the style of (e.g.) marginal contribution nets
- A feature set is a set of CTL formula/value pairs

$$\mathcal{F} = \{(\phi_1, x_1), \ldots, (\phi_k, x_k)\}$$

- Represents:

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

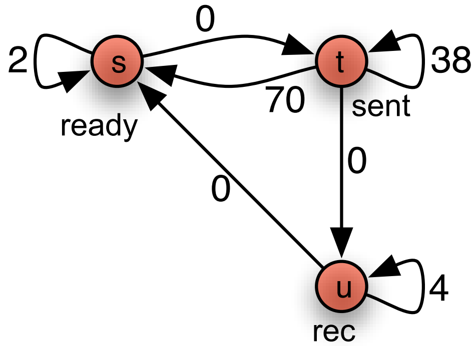
Example



Property	Benefit
$\phi_1 = \mathsf{E}\Diamond(rec \wedge \mathsf{E}\Diamond ready)$	110
$\phi_2 = \mathsf{A}\,\Box(rec \rightarrow \mathsf{A}\Diamond ready)$	15
$\phi_3 = \mathsf{A}\,\Box\mathsf{A}\Diamond ready$	15
$\phi_4 = \mathsf{A}\,\Box\mathsf{A}\Diamond sent$	18
$\phi_5 = \mathsf{A}\,\Box(sent \rightarrow \mathsf{A}\Diamond rec)$	10
$\phi_6 = \mathsf{A}\,\Box(ready \rightarrow \mathsf{A}\Diamond(sent \wedge \mathsf{A}\Diamond ready))$	23
$\phi_7 = \mathsf{A}\,\Box(\mathsf{A}\Diamond(rec \wedge \mathsf{A}\Diamond ready))$	25

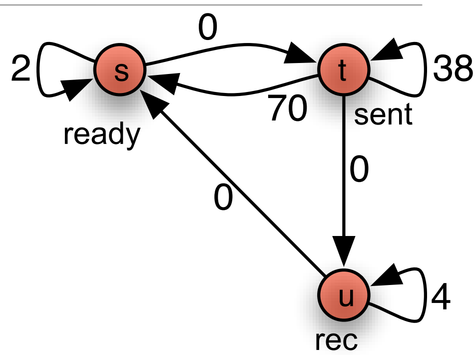
Example

Property	**Benefit**
$\phi_1 = \mathsf{E}\Diamond(rec \wedge \mathsf{E}\Diamond ready)$	110
$\phi_2 = \mathsf{A}\,\Box(rec \rightarrow \mathsf{A}\Diamond ready)$	15
$\phi_3 = \mathsf{A}\,\Box\mathsf{A}\Diamond ready$	15
$\phi_4 = \mathsf{A}\,\Box\mathsf{A}\Diamond sent$	18
$\phi_5 = \mathsf{A}\,\Box(sent \rightarrow \mathsf{A}\Diamond rec)$	10
$\phi_6 = \mathsf{A}\,\Box(ready \rightarrow \mathsf{A}\Diamond(sent \wedge \mathsf{A}\Diamond ready))$	23
$\phi_7 = \mathsf{A}\,\Box(\mathsf{A}\Diamond(rec \wedge \mathsf{A}\Diamond ready))$	25



Example

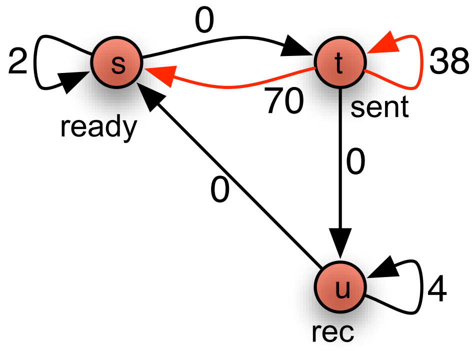
Property	**Benefit**
$\phi_1 = \mathsf{E}\Diamond(rec \wedge \mathsf{E}\Diamond ready)$	110
$\phi_2 = \mathsf{A}\,\Box(rec \rightarrow \mathsf{A}\Diamond ready)$	15
$\phi_3 = \mathsf{A}\,\Box\mathsf{A}\Diamond ready$	15
$\phi_4 = \mathsf{A}\,\Box\mathsf{A}\Diamond sent$	18
$\phi_5 = \mathsf{A}\,\Box(sent \rightarrow \mathsf{A}\Diamond rec)$	10
$\phi_6 = \mathsf{A}\,\Box(ready \rightarrow \mathsf{A}\Diamond(sent \wedge \mathsf{A}\Diamond ready))$	23
$\phi_7 = \mathsf{A}\,\Box(\mathsf{A}\Diamond(rec \wedge \mathsf{A}\Diamond ready))$	25



$$\eta = \{(t,t),(t,s)\}$$

Example

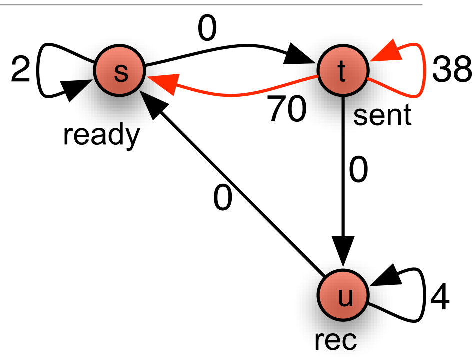
Property	Benefit
$\phi_1 = \mathsf{E}\Diamond(rec \wedge \mathsf{E}\Diamond ready)$	110
$\phi_2 = \mathsf{A}\,\Box(rec \rightarrow \mathsf{A}\Diamond ready)$	15
$\phi_3 = \mathsf{A}\,\Box\mathsf{A}\Diamond ready$	15
$\phi_4 = \mathsf{A}\,\Box\mathsf{A}\Diamond sent$	18
$\phi_5 = \mathsf{A}\,\Box(sent \rightarrow \mathsf{A}\Diamond rec)$	10
$\phi_6 = \mathsf{A}\,\Box(ready \rightarrow \mathsf{A}\Diamond(sent \wedge \mathsf{A}\Diamond ready))$	23
$\phi_7 = \mathsf{A}\,\Box(\mathsf{A}\Diamond(rec \wedge \mathsf{A}\Diamond ready))$	25



$$\eta = \{(t,t),(t,s)\}$$

Example

Property	Benefit
$\phi_1 = \mathsf{E}\Diamond(rec \wedge \mathsf{E}\Diamond ready)$	110
$\phi_2 = \mathsf{A}\,\Box(rec \rightarrow \mathsf{A}\Diamond ready)$	15
$\phi_3 = \mathsf{A}\,\Box\mathsf{A}\Diamond ready$	15
$\phi_4 = \mathsf{A}\,\Box\mathsf{A}\Diamond sent$	18
$\phi_5 = \mathsf{A}\,\Box(sent \rightarrow \mathsf{A}\Diamond rec)$	10
$\phi_6 = \mathsf{A}\,\Box(ready \rightarrow \mathsf{A}\Diamond(sent \wedge \mathsf{A}\Diamond ready))$	23
$\phi_7 = \mathsf{A}\,\Box(\mathsf{A}\Diamond(rec \wedge \mathsf{A}\Diamond ready))$	25

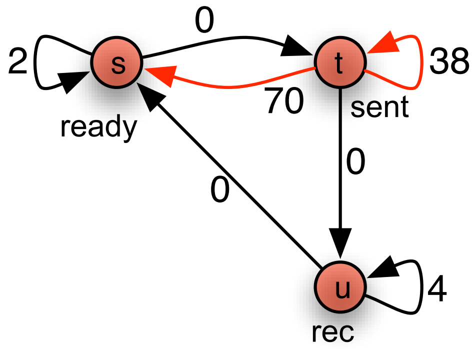


$$\eta = \{(t,t),(t,s)\}$$

Cost: 108

Example

Property	Benefit
$\phi_1 = \mathsf{E}\Diamond(rec \wedge \mathsf{E}\Diamond ready)$	110
$\phi_2 = \mathsf{A}\,\Box(rec \rightarrow \mathsf{A}\Diamond ready)$	15
$\phi_3 = \mathsf{A}\,\Box\mathsf{A}\Diamond ready$	15
$\phi_4 = \mathsf{A}\,\Box\mathsf{A}\Diamond sent$	18
$\phi_5 = \mathsf{A}\,\Box(sent \rightarrow \mathsf{A}\Diamond rec)$	10
$\phi_6 = \mathsf{A}\,\Box(ready \rightarrow \mathsf{A}\Diamond(sent \wedge \mathsf{A}\Diamond ready))$	23
$\phi_7 = \mathsf{A}\,\Box(\mathsf{A}\Diamond(rec \wedge \mathsf{A}\Diamond ready))$	25



$$K \dagger \eta \models \phi_1$$
$$K \dagger \eta \models \phi_5$$

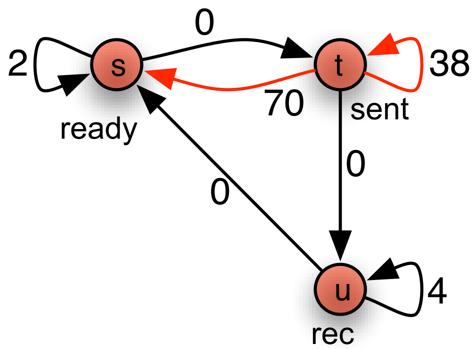
$$\eta = \{(t,t),(t,s)\}$$

Benefit: 120

Cost: 108

Example

Property	Benefit
$\phi_1 = \mathsf{E}\Diamond(rec \wedge \mathsf{E}\Diamond ready)$	110
$\phi_2 = \mathsf{A}\,\Box(rec \rightarrow \mathsf{A}\Diamond ready)$	15
$\phi_3 = \mathsf{A}\,\Box\mathsf{A}\Diamond ready$	15
$\phi_4 = \mathsf{A}\,\Box\mathsf{A}\Diamond sent$	18
$\phi_5 = \mathsf{A}\,\Box(sent \rightarrow \mathsf{A}\Diamond rec)$	10
$\phi_6 = \mathsf{A}\,\Box(ready \rightarrow \mathsf{A}\Diamond(sent \wedge \mathsf{A}\Diamond ready))$	23
$\phi_7 = \mathsf{A}\,\Box(\mathsf{A}\Diamond(rec \wedge \mathsf{A}\Diamond ready))$	25



$$K \dagger \eta \models \phi_1$$
$$K \dagger \eta \models \phi_5$$

$$\eta = \{(t,t),(t,s)\}$$

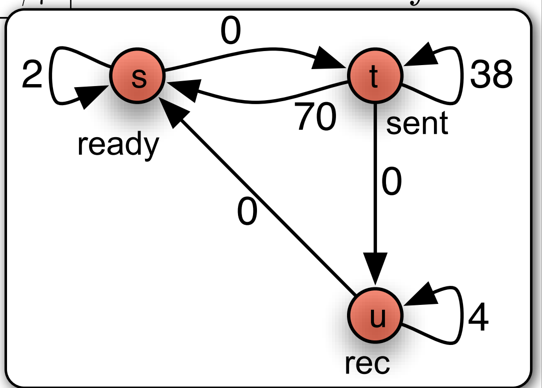
Benefit: 120 **Utility:** 12 Cost: 108

Example

	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
η_9	+	-	-	+	+	+	-	+	-	-	-	6	143	137
η_{10}	+	-	+	-	+	-	-	-	-	-	-	72	110	38
η_{11}	+	-	+	+	+	+	-	+	-	-	-	76	143	67
η_{12}	+	+	-	-	+	-	-	-	-	-	-	40	110	70
η_{13}	+	+	-	+	+	+	+	+	-	+	-	44	181	137
η_{14}	+	+	+	-	+	-	-	-	+	-	-	110	120	10
η_{15}	+	+	+	+	+	+	+	+	+	+	+	114	216	102

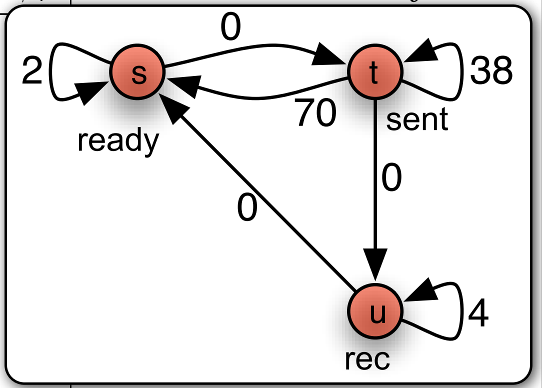
Example

	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	-	-	-	-	+	-	-	-	-	-				
η_1	-	-	-	+	+	+	-	-	-	-				
η_2	-	-	+	-	+	-	-	-	-	-				
η_3	-	-	+	+	+	+	-	-	-	-				
η_4	-	+	-	-	+	-	-	-	-	-				
η_5	-	+	-	+	+	+	+	-	-	-				
η_6	-	+	+	-	+	-	-	-	+	-				
η_7	-	+	+	+	+	+	+	-	+	-				
η_8	+	-	-	-	+	-	-	-	-	-				
η_9	+	-	-	+	+	+	-	+	-	-	-	6	143	137
η_{10}	+	-	+	-	+	-	-	-	-	-	-	72	110	38
η_{11}	+	-	+	+	+	+	-	+	-	-	-	76	143	67
η_{12}	+	+	-	-	+	-	-	-	-	-	-	40	110	70
η_{13}	+	+	-	+	+	+	+	+	-	+	-	44	181	137
η_{14}	+	+	+	-	+	-	-	-	+	-	-	110	120	10
η_{15}	+	+	+	+	+	+	+	+	+	+	+	114	216	102



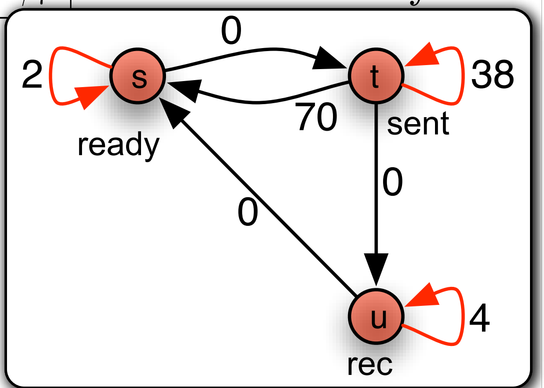
Example

	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	-	-	-	-	+	-	-	-	-	-				
η_1	-	-	-	+	+	+	-	-	-	-				
η_2	-	-	+	-	+	-	-	-	-	-				
η_3	-	-	+	+	+	+	-	-	-	-				
η_4	-	+	-	-	+	-	-	-	-	-				
η_5	-	+	-	+	+	+	+	-	-	-				
η_6	-	+	+	-	+	-	-	-	+	-				
η_7	-	+	+	+	+	+	+	-	+	-				
η_8	+	-	-	-	+	-	-	-	-	-				
η_9	+	-	-	+	+	+	-	+	-	-	-	6	143	137
η_{10}	+	-	+	-	+	-	-	-	-	-	-	72	110	38
η_{11}	+	-	+	+	+	+	-	+	-	-	-	76	143	67
η_{12}	+	+	-	-	+	-	-	-	-	-	-	40	110	70
η_{13}	+	+	-	+	+	+	+	+	-	+	-	44	181	137
η_{14}	+	+	+	-	+	-	-	-	+	-	-	110	120	10
η_{15}	+	+	+	+	+	+	+	+	+	+	+	114	216	102



Example

	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	-	-	-	-	+	-	-	-	-	-				
η_1	-	-	-	+	+	+	-	-	-	-				
η_2	-	-	+	-	+	-	-	-	-	-				
η_3	-	-	+	+	+	+	-	-	-	-				
η_4	-	+	-	-	+	-	-	-	-	-				
η_5	-	+	-	+	+	+	+	-	-	-				
η_6	-	+	+	-	+	-	-	-	+	-				
η_7	-	+	+	+	+	+	+	-	+	-				
η_8	+	-	-	-	+	-	-	-	-	-				
η_9	+	-	-	+	+	+	-	+	-	-	-	6	143	137
η_{10}	+	-	+	-	+	-	-	-	-	-	-	72	110	38
η_{11}	+	-	+	+	+	+	-	+	-	-	-	76	143	67
η_{12}	+	+	-	-	+	-	-	-	-	-	-	40	110	70
η_{13}	+	+	-	+	+	+	+	+	-	+	-	44	181	137
η_{14}	+	+	+	-	+	-	-	-	+	-	-	110	120	10
η_{15}	+	+	+	+	+	+	+	+	+	+	+	114	216	102



The complexity of finding optimal social laws

Theorem *The* OPTIMAL SOCIAL LAW *problem for the feature set representation of valuation functions is* $\mathrm{FP}^{\mathrm{NP}}$*-complete.*

Simple instances

- A **simple instance**:
  - all values in the feature set are the same
  - the cost of every transition is 0

Simple instances

- A **simple instance**:
  - all values in the feature set are the same
  - the cost of every transition is 0

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

Homogeneous instances

- A **homogenous instance**:
  - all values in the feature set are the same
  - the cost of every transition is the same

Theorem *The* OPTIMAL SOCIAL LAW *problem for homogenous instances is in* $\mathrm{FP}^{\mathrm{NP}[|R| \log_2 |\mathcal{F}|]}$.

Dichotomous valuations

- Assume that
  - the designer is able to partition the state space into **bad** states **B** and **good** states **S\B**
  - and that she gets some positive value **a** from a social law preventing the system from going into bad states, while not excluding any good states

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

- Such *dichotomus valuations* can be represented by (B,a)

Monday, January 16, 12

Dichotomous valuations

- Assume that
  - the designer is able to partition the state space into **bad** states **B** and **good** states **S\B**
  - and that she gets some positive value **a** from a social law preventing the system from going into bad states, while not excluding any good states

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

- Such *dichotomus valuations* can be represented by (B,a)

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

Monday, January 16, 12

Integer linear programming

- Integer programming is one of the most successful and widely used approaches to solving computationally hard optimisation problems
- We show how the optimal social law problem can be formulated as an integer linear program (ILP)
- The ILP correctly synthesises optimal social laws

ILP

maximize:

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

subject to constraints:

$$
\begin{array}{lcl}
\tau(\psi, s) & \in & \{0,1\} \\
 & & \forall \psi \in cl(\mathcal{F}),\ s \in S \\
\eta(s,s') & \in & \{0,1\} \\
 & & \forall (s,s') \in R \\
\sum_{s' \in next(s)} (1 - \eta(s,s')) & \geq & 1 \\
 & & \forall s \in S \\
\tau(p, s) & = & \begin{cases} 1 & \text{if } p \in \pi(s) \\ 0 & \text{otherwise} \end{cases} \\
 & & \forall p \in \Phi \cap cl(\mathcal{F}),\ s \in S \\
\tau(\neg\psi, s) & = & 1 - \tau(\psi, s) \\
 & & \forall \neg\psi \in cl(\mathcal{F}),\ s \in S \\
\tau(\psi \vee \chi, s) & \leq & \tau(\psi, s) + \tau(\chi, s) \\
 & & \forall \psi \vee \chi \in cl(\mathcal{F}),\ s \in S \\
\tau(\psi \vee \chi, s) & \geq & \tau(\psi, s) \\
 & & \forall \psi \vee \chi \in cl(\mathcal{F}),\ s \in S \\
\tau(\psi \vee \chi, s) & \geq & \tau(\chi, s) \\
 & & \forall \psi \vee \chi \in cl(\mathcal{F}),\ s \in S
\end{array}
$$

ILP

maximize:

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

subject to constraints:

$$
\begin{array}{lcl}
\tau(\psi, s) & \in & \{0,1\} \\
 & & \forall \psi \in c \\
\eta(s,s') & \in & \{0,1\} \\
 & & \forall (s,s' \\
\sum_{s' \in next(s)} (1 - \eta(s,s')) & \geq & 1 \\
 & & \forall s \in S \\
\tau(p,s) & = & \begin{cases} 1 & \text{i} \\ 0 & \text{c} \end{cases} \\
 & & \forall p \in \Phi \\
\tau(\neg\psi, s) & = & 1 - \tau(\psi \\
 & & \forall \neg\psi \in \\
\tau(\psi \vee \chi, s) & \leq & \tau(\psi, s) \\
 & & \forall \psi \vee \chi \\
\tau(\psi \vee \chi, s) & \geq & \tau(\psi, s) \\
 & & \forall \psi \vee \chi \\
\tau(\psi \vee \chi, s) & \geq & \tau(\chi, s) \\
 & & \forall \psi \vee \chi
\end{array}
$$

$$
\begin{array}{lcl}
g(\psi, s) & \in & \{0,1\} \quad \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}),\ s \in S \\
h(\psi, s, s') & \in & \{0,1\} \\
 & & \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}),\ s \in S,\ s' \in next(s) \\
\tau(\mathsf{E}\Box\psi, s) & \leq & \tau(\psi, s) \\
 & & \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}),\ s \in S \\
\tau(\mathsf{E}\Box\psi, s) & \leq & g(\psi, s) \\
 & & \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}),\ s \in S \\
\tau(\mathsf{E}\Box\psi, s) & \geq & 1 - ((1 - \tau(\psi, s)) + (1 - g(\psi, s))) \\
 & & \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}),\ s \in S \\
h(\psi, s, s') & \leq & 1 - \eta(s, s') \\
 & & \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}),\ s \in S,\ s' \in next(s) \\
h(\psi, s, s') & \leq & \tau(\mathsf{E}\Box\psi, s') \\
 & & \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}),\ s \in S,\ s' \in next(s) \\
h(\psi, s, s') & \leq & 1 - (\eta(s, s') + (1 - \tau(\mathsf{E}\Box\psi, s'))) \\
 & & \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}),\ s \in S,\ s' \in next(s) \\
g(\psi, s) & \leq & \sum_{s' \in next(s)} h(\psi, s, s') \\
 & & \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}),\ s \in S \\
g(\psi, s) & \geq & h(\psi, s, s') \\
 & & \forall \mathsf{E}\Box\psi \in cl(\mathcal{F}),\ s \in S,\ s' \in next(s)
\end{array}
$$

Conclusions

- Trade-offs between costs and benefits of implementing social laws -> optimisation problem
- With feature set representation: computationally hard
  - We show how to solve it using ILP
  - Identified some less complex instances
- Future work: classes of Kripke structures/feature sets vs. classes of ILPs known to be efficiently solvable

Contents

Mechanism Design

- Up to now we have assumed that some social law is designed or synthesised by a designer
- We now look at how we can let *the agents come up with a social law*
- Key question: given their individual preferences/goals, how can they decide on a social law that (in some sense) makes them collectively better off compared to the status quo?
- To answer this question, we turn to social choice theory, which have studied the design of mechanisms for collective desicion making
- Key propery of such mechanisms: incentive compatibility

Goal Model

- We use a more sophisticated goal model here
- Each agent has a set of goals in the form of weighted formulas

  $$\gamma = \{(\varphi_1, x_1), \ldots, (\varphi_k, x_k)\}$$

  where each x_i is a real number
- No consistency requirements

The value of social laws

A goal set:

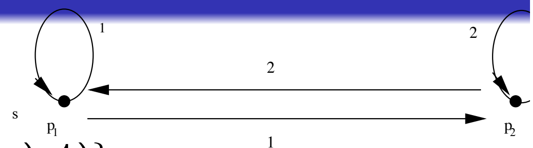
$$\gamma = \{(\varphi_1, x_1), \ldots, (\varphi_k, x_k)\}$$

Let:

$v_\gamma(K) = \sum x : (\varphi, x) \in \gamma \& K \models \varphi$ the value of K according to γ

$v_\gamma(K, \eta) = v_\gamma(K \dagger \eta) - v_\gamma(K)$ the increase of value an agent with γ as goals would get if η was implemented

Example



$$\begin{aligned}\gamma_1 &= \{(\mathrm{A}\bigcirc\mathrm{A}\Diamond p_1, 4), (\mathrm{E}\Box p_1, 2)\}\\ \gamma_2 &= \{(\neg\mathrm{A}\bigcirc\mathrm{A}\Diamond p_1, 3), (\mathrm{A}\Box(p_1 \rightarrow \mathrm{E}\Diamond p_2), 1)\}\end{aligned}$$

η_i	(s,s)	(s,t)	(t,s)	(t,t)	$v_1(K_0,\eta_i)$	$v_2(K_0,\eta_i)$
1	0	0	0	0	0	0
2	0	0	0	1	4	-3
3	0	0	1	0	0	0
4	0	1	0	0	4	-4
5	0	1	0	1	4	-4
6	0	1	1	0	4	-4
7	1	0	0	0	-2	0
8	1	0	0	1	2	-3
9	1	0	1	0	-2	0

Social choice rules

A social choice rule for social laws is a function f mapping a model K and one goal set γ_i for each agent i to a social law

$$f(K, \gamma_1, \ldots, \gamma_n)$$

Maximising Social Welfare

One possible social choice rule is maximising social welfare:

$$f_{sw}(K, \gamma_1, \ldots, \gamma_n) = \arg\max_\eta \sum_{i=1}^{n} v_i(K \dagger \eta)$$

Example cont.

$$\begin{array}{lcl} \gamma_1 & = & \{(A\bigcirc A\Diamond p_1, 4), (E\Box p_1, 2)\} \\ \gamma_2 & = & \{(\neg A\bigcirc A\Diamond p_1, 3), (A\Box(p_1 \rightarrow E\Diamond p_2), 1)\} \end{array}$$

$$f_{sw}(K, \gamma_1, \gamma_2) = \eta_2$$

η_i	(s,s)	(s,t)	(t,s)	(t,t)	$v_1(K_0, \eta_i)$	$v_2(K_0, \eta_i)$
1	0	0	0	0	0	0
2	0	0	0	1	4	-3
3	0	0	1	0	0	0
4	0	1	0	0	4	-4
5	0	1	0	1	4	-4
6	0	1	1	0	4	-4
7	1	0	0	0	-2	0
8	1	0	0	1	2	-3
9	1	0	1	0	-2	0

Computing max social welfare

How hard is it to compute f_{sw}?

Theorem

The following decision problem: given a system $\langle K, \gamma_1, \ldots, \gamma_n \rangle$ and a bound $k \in \mathbb{R}$, decide whether

$$\exists \eta \in N(R) : \sum_{i=1}^{n} v_i(K, \eta) \geq k$$

This problem is NP*-complete, even for a single agent with a single goal valued at 1, with $k = 1$.*

Computing max social welfare

Theorem

Computing f_{sw} is $\mathrm{FP}^{\mathrm{NP}}$*-complete.*

Other Social Choice Rules

- f_{nash}: maximising the Nash product, i.e., select the product-maximising social law
- f_{nash} is equally hard to compute as f_{sw}
- Both f_{sw} and f_{nash} are Pareto-optimal (there is no social law that is better for all agents compared to the social law selected by the rule)

Naive Mechanisms

Direct implementation of, e.g., $X = sw$:

1. Every agent $i \in A$ declares to the mechanisms a set of goals $\hat{\gamma}_i$.
2. The mechanism computes $\eta^* = f_X(K, \hat{\gamma}_1, \ldots, \hat{\gamma}_n)$.
3. The social η^* is chosen, and every agent $i \in A$ pays $v_i(K \dagger \eta^*)$ to the mechanism (if $v_i(K \dagger \eta^*) < 0$, then the payment is from the mechanism to the agent).

Naive Mechanisms

- It is easy to see that a direct implementation of (e.g.) f_{sw} is manipulable, in the sense that an agent can gain from misrepresenting her true goals
- Examples:
  - hiding goals
  - misrepresenting goal weight
  - phantom goals

Incentive Compatible Mechanisms

Variant of the Vickrey-Clarke-Groves (VCG) mechanism:

1. Every agent $i \in A$ declares a set of goals $\hat{\gamma}_i$.
2. Compute $\eta^* = f_{sw}(K, \hat{\gamma}_1, \ldots, \hat{\gamma}_i, \ldots, \hat{\gamma}_n)$
3. Social law η^* is chosen, and every agent $i \in A$ pays p_i computed as follows. First, let η' denote the social law that would have been chosen had agent i declared goals $\hat{\gamma}_i = \emptyset$ and every other agent had made the same declaration:

$$\eta' = f_{sw}(K, \hat{\gamma}_1, \ldots, \emptyset, \ldots, \hat{\gamma}_n)$$

$$p_i = sw_{-i}(K \dagger \eta', \hat{\gamma}_1, \ldots, \emptyset, \ldots \hat{\gamma}_n) - sw_{-i}(K \dagger \eta^*, \hat{\gamma}_1, \ldots, \hat{\gamma}_i, \ldots, \hat{\gamma}_n)$$

$$sw_{-i}(K, \gamma_1, \ldots, \gamma_n) = \sum_{j \in A (j \neq i)} v_j(K)$$

Incentive Compatible Mechanisms

Variant of the Vickrey-Clarke-Groves (VCG) mechanism:

1. Every agent $i \in A$ declares a set of goals $\hat{\gamma}_i$.
2. Compute $\eta^* = f_{sw}(K, \hat{\gamma}_1, \ldots, \hat{\gamma}_i, \ldots, \hat{\gamma}_n)$
3. Social law η^* is chosen, and every agent $i \in A$ pays p_i computed as follows. First, let η' denote the social law that would have been chosen had agent i declared goals $\hat{\gamma}_i = \emptyset$ and every other agent had made the same declaration:

$$\eta' = f_{sw}(K, \hat{\gamma}_1, \ldots, \emptyset, \ldots, \hat{\gamma}_n)$$

$$p_i = sw_{-i}(K \dagger \eta', \hat{\gamma}_1, \ldots, \emptyset, \ldots \hat{\gamma}_n) - sw_{-i}(K \dagger \eta^*, \hat{\gamma}_1, \ldots, \hat{\gamma}_i, \ldots, \hat{\gamma}_n)$$

Theorem

The above mechanism is incentive compatible. That is, using this mechanism, an agent $i \in A$ can do no better than declaring $\hat{\gamma}_i = \gamma_i$.

Contents

Some references I

T. Ågotnes and M. Wooldridge.
Optimal social laws.
In *Proceedings of the Ninth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, Toronto, Canada, May 2010.

Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge.
Normative system design as social choice.
Unpublished manuscript, 2009.