JAWS2011

A preliminary investigation of decentralized decision making with bounded resources

Daniel Laszlo KOVACS
Department of Measurement and Information Systems
Budapest University of Technology and Economics
Budapest, Hungary, H-1117
dkovacs@mit.bme.hu, http://www.mit.bme.hu/~dkovacs

Naoki FUKUTA
Department of Computer Science
Shizuoka University
Hamamatsu, Japan, 432-8011
fukuta@cs.inf.shizuoka.ac.jp, http://whitebear.cs.inf.shizuoka.ac.jp

Takashi WATANABE
Department of Computer Science
Shizuoka University
Hamamatsu, Japan, 432-8011
watanabe@inf.shizuoka.ac.jp, http://aurum.cs.inf.shizuoka.ac.jp/english

keywords: decentralized, resource-bounded, decision making, rationality, AIXI

Summary

This paper is a preliminary theoretical investigation of decentralized decision making with bounded resources. We build upon a feasible rationality concept for single agents to enable the formal investigation of the individual and group-level rationality of possibly heterogeneous, decentralized decision makers. We conjecture that it is more realistic than the centralized setting. Compared to other recent decentralized rationality notions (e.g. optimal decentralized metareasoning) our concept is stronger and not restricted to special cases (e.g. just collaborative agents). We introduce our model in detail, discuss its merits and limitations, connect it to the single-agent case, show that it is implicitly a decentralized extension of the recently successful AIXI, and provide a few application examples.

1. Introduction

This paper presents a decentralized extension and refinement of the original, single-agent concept of bounded-optimality, which is a rationality criterion for intelligent agents similarly to perfect, calculative or metalevel rationality [Russell & Subramanian, 1995]. We decided to extend it because it "seems to offer the best hope for a strong theoretical foundation for AI" [Russell & Norvig, 2010, Ch. 27, pp. 1050]. To our knowledge currently there are no general means for designing such bounded-optimal agents. We hope to advance in this direction by introducing our model, which is the main contribution of this paper.

A decentralized extension is necessary – although other agents may as well be modeled implicitly as (hidden) parts of a dynamic environment in the single-agent case – because it allows explicit representation and reasoning about other, possibly heterogeneous agents' structure (e.g. architecture, sensors, effectors, resources, goals, beliefs, utilities, properties) and behavior (e.g. program, strategy, rationality, faultiness, competition, coordination, cooperation, interaction, selfishness, altruism), which can be more effective. It enables examination of rationality of agents both at an individual and group level in connection.

When using the proposed model one must be careful with the definition of agents' utility. Every agent may have a different (or even the same) user/Designer-defined utility. *If the utility of an agent is ill defined, then the interpretation of its rationality may be inappropriate*. For example if there is a zero-sum situation (e.g. a Poker game), then if we define an agent's utility as the sum of all the agents' individual payoff (which is by definition zero), then the value of this agent's utility will be zero for every possible outcome, which either means that we are indifferent toward its behavior (everything it does is considered rational), or we defined its utility function inappropriately.[1] Eventually in this paper we try to answer the following question: *given several, possibly heterogeneous decentralized, utility-driven agents with bounded resources, when are they rational?*

[1] For zero-sum situations there is no point of defining a social welfare utility [Pattanaik, 2008] summing up agents' individual valuation, since it would make no distinction between outcomes.

The paper is structured as follows. Section 2 reflects on related work. Section 3 introduces the fundamentals of bounded-optimality and AIXI [2] [Hutter, 2005], which is another recently successful concept of rationality. We prove the convergence of AIXI first to perfect rationality, then (in case of more realistic assumptions) to bounded-optimality to stress its importance even further. Section 4 contains our main contribution: a decentralized extension of bounded-optimality. Section 5 gives examples of this extension. Section 6 concludes with a discussion of the results, limitations and an outline of future research directions.

2. Related work

There are several predecessors of our model [Russell & Subramanian, 1995]. *Perfect rationality*, for example, one of the earliest, classical single-agent rationality concepts, states that an agent should act so as to maximize its expected utility at every instant. This concept is not feasible in general, since it takes no account of the limited computational resources of the agent and the time needed for deliberation. *Calculative rationality* relaxes this assumption by allowing the agent to eventually return what would have been a perfectly rational decision at the beginning of its deliberation. This concept is more interesting in-principle, but it is of less value in practice, since the actual behavior of such an agent may be far from optimal. *Metalevel rationality*, e.g. *optimal metareasoning* [Cox & Raja, 2011] responds to the previous problems by trying to optimize not only over (ground-level) actions, but also to find an optimal trade-off between the cost and value of the (object-level) computation generating these actions. The drawback of this approach is that such meta-level decision problems are often more difficult than the original (object-level) decision problems, thus optimality can only be guaranteed in special cases, e.g. when the agent's utility is a function of the *time* spent for deliberation and the decision making procedure is e.g. an anytime algorithm. *Bounded-optimality* on the other hand makes no assumptions about the structure of agents' program (it is not required that the agent itself be engaged in any form of metareasoning). It only requires that "[the agent's] program is a solution to the constrained optimization problem presented by its architecture and the task environment" [Russell & Subramanian, 1995]. This means that for instance even a simple reactive agent-program based on random decisions can be bounded-optimal in a given task environment iff it yields the highest expected utility among the programs runnable on the agent's architecture. "This is a stronger guarantee than optimal metareasoning, but it is also harder to achieve" [Carlin & Zilberstein, 2011].

This is why *asymptotic bounded-optimality* was proposed [Russell & Subramanian, 1995]. It requires only that the agent's program is not worse than any other program on its current architecture provided with a constant-times faster architecture (or with constant-times more capacity). In this sense bounded-optimality is similar to AIXI, which is also a universal rationality measure that is only asymptotically computable in practice. Nevertheless a computationally feasible, direct Monte-Carlo approximation [Veness et al., 2011] was provided for it recently, whose main ideas stem form POMCP [Silver & Veness, 2010], which is part of POMDPX_NUS [Ong et al., 2010], the planner winning the probabilistic Boolean POMDP track of this year's International Planning Competition (ICAPS IPC-2011).

An AIXI agent is dual in comparison to a bounded-optimal agent in that the former first acts and then receives a percept and a reward as a result (general reinforcement learning scheme), while the latter first perceives the actual state of its environment and then acts according to its current percept history (intelligent/rational agent scheme). The two schemes can be connected as shown in [Hutter, 2005, Ch. 6], but we can also prove that AIXI converges to bounded-optimality (c.f. Section 3.2) and thus e.g. the above mentioned AIXI approximation is effectively an approximation of bounded-optimality.

All of the previous concepts are important measures of agents' intelligence, but all of them rely on the simplifying assumption of a single agent. Extending them "to multi-agent settings is hard" [Carlin & Zilberstein, 2011], but it became a current topic of research. For example such an extension of optimal metareasoning was given in [Carlin & Zilberstein, 2011], but it actually works only for two collaborative agents, and in general has the same drawbacks as single-agent metareasoning. Moreover it should give rise to the problem of infinite regress in reciprocal (higher order) beliefs of agents as they start to reason about each other's reasoning (which may be necessary for optimality).

In case of perfect rationality Game Theory [Neumann & Morgenstern, 1944] is an appropriate extension to decentralized decision making, but it inherently has the same problems as perfect rationality. Nonetheless it overcomes the issue of infinite regress of reciprocal expectations by assuming common knowledge of the game (and – in case of incomplete information – common priors).

AIXI has no direct extension to the decentralized setting yet, and it is not trivial, since it would require consideration of several agents' deliberation and actions, of which there is currently no trace (or place for) in the model. But since AIXI converges to bounded-optimality, our extension of it can be seen as an indirect decentralized extension of AIXI.

3. Preliminaries

As we have already mentioned in the introduction, the definition of agents' utility is central to their definition of

[2] AIXI stands for "Artificial Intelligence ξ", where ξ is Solomonoff's universal a priori probability distribution over the possible true environments [Hutter, 2005] tending to converge to the *initially unknown*, true a priori probability distribution μ.

rationality. A *perfectly rational agent* according to [Russell & Subramanian, 1995] corresponds to an agent function f_{opt} such that

$$f_{\text{opt}} = \text{argmax}_f\big(V(f,\mathbf{E})\big) \quad (1)$$

An agent function, $f\colon \mathbf{O}^t \to \mathbf{A}$, is a mapping from the finite set of percept history prefixes $\mathbf{O}^t = \{O^t | t \in \mathbf{T}, O^\mathbf{T} \in \mathbf{O}^\mathbf{T}\}$ to the finite set $\mathbf{A}$ of actions; $\mathbf{T}$ is the finite, totally ordered set of time instants (with a unique least element, *0*); $\mathbf{O}$ is the finite set of the agent's possible percepts; and $\mathbf{O}^\mathbf{T} = \{O^\mathbf{T}\colon \mathbf{T} \to \mathbf{O}\}$ is the set of all possible percept histories, where $O^\mathbf{T}$ is a particular percept history. Thus a perfectly rational agent corresponds to a function that maximizes $V(f,\mathbf{E})$, where $\mathbf{E}$ is a finite set of possible environments with a probability distribution p over them. An environment, $E \in \mathbf{E}$, consists of a set $\mathbf{X}$ of states with a distinguished initial state X_0, a transition function f_e and a perceptual filter function f_p such that $X^\mathbf{T}(0) = X_0$, $O^\mathbf{T}(t) = f_p\big(X^\mathbf{T}(t)\big)$, $A^\mathbf{T}(t) = f(O^t)$ and $X^\mathbf{T}(t+1) = f_e\big(A^\mathbf{T}(t), X^\mathbf{T}(t)\big)$ holds for $\forall t \in \mathbf{T}$, where $X^\mathbf{T}\colon \mathbf{T} \to \mathbf{X}$ is the state trajectory, and $A^\mathbf{T}\colon \mathbf{T} \to \mathbf{A}$ is the action history produced by the agent.

Now $V(f,\mathbf{E})$ can be defined as the *expected utility* of f in $\mathbf{E}$ with a probability distribution p over $\mathbf{E}$ as follows.

$$V(f,\mathbf{E}) = \textstyle\sum_{E\in\mathbf{E}} p(E)\cdot V(f,E) \quad (2)$$

Here $V(f,E)$ denotes the *utility* of f in $E \in \mathbf{E}$.

$$V(f,E) = U\big(\mathit{effects}(f,E)\big) \quad (3)$$

In Eq. 3 $\mathit{effects}(f,E) \in \mathbf{X}^\mathbf{T}$ stands for the state trajectory generated by f in E; and so $U\colon \mathbf{X}^\mathbf{T} \to \mathbb{R}$ denotes the *utility function* of the agent, a mapping from the set of state trajectories, $\mathbf{X}^\mathbf{T}$, to real numbers.

3.1 Convergence of AIXI to perfect rationality

One of our results is in showing that Hutter's AIXI converges to a special case of perfect rationality based on the claim in [Hutter, 2005, p146] that AIξ converges to AIμ as defined in [Hutter, 2005, p130, Def. 4.4], and showing that AIμ corresponds to a special case of perfect rationality.

Definition 1 (The AIμ model). *The AIμ model is the agent with policy p^μ that maximizes the μ-expected total reward $r_1 + \cdots + r_m$, i.e. $p^* \equiv p^\mu := argmax_p V_\mu^p$. Its value is $V_\mu^* := V_\mu^{p^\mu}$.* [Hutter, 2005, p130]

Policy p^* corresponds to agent function f_{opt} in Eq. 1, and generally a policy p corresponds to an agent function f. The *expected utility* (called value function in AIXI) defined as $V_\mu^p \equiv V_{1m}^{p\mu} := \sum_q \mu(q)\, V_{1m}^{pq}$ [Hutter, 2005, p130] corresponds to Eq. 2 with $V_\mu^p \equiv V_{1m}^{p\mu}$ corresponding to $V(f,\mathbf{E})$; environment q corresponds to environment E; probability $\mu(q)$ corresponds to $p(E)$ in Eq. 2; and total utility V_{1m}^{pq} corresponds to utility $V(f,E)$ as given in Eq. 3.

V_{1m}^{pq} is defined as $V_{1m}^{pq} := \sum_{i=1}^m r\big(x_i^{pq}\big)$ in [Hutter, 2005, p129], where m denotes the lifetime of the agent, and corresponds to $|\mathbf{T}|$; $r\big(x_i^{pq}\big) = r_i$ is the reinforcement of an AIXI agent in cycle i ($i = 1,2,\ldots,m$) as seen in Def. 1; and $x_i^{pq} \in \mathcal{X}$ is the perception (input) of the AIXI agent in cycle i (in case of p and q) corresponding to $O^\mathbf{T}(i) \in \mathbf{O}$, so $\mathcal{X}$ corresponds to $\mathbf{O}$. According to Eq. 3 this means that $\sum_{i=1}^m r\big(x_i^{pq}\big)$ corresponds to $U\big(\mathit{effects}(f,E)\big)$, i.e. AI$\mu$ corresponds to a *special case of perfect rationality*, where the utility U of the state trajectory $\mathit{effects}(f,E)$ – which corresponds to the unique I/O sequence $\omega^{pq} := y_1^{pq}x_1^{pq}\ldots y_m^{pq}x_m^{pq}$ in AIXI – is calculated as the sum of rewards r_i extracted from percepts at every time instant. Moreover, cycle i in AIXI can be considered to correspond to time-interval $[i-1, i]$, so $y_i^{pq} \in \mathcal{Y}$ corresponds to $A^\mathbf{T}(i-1) \in \mathbf{A}$, $\mathcal{Y}$ corresponds to $\mathbf{A}$, and $O^\mathbf{T}(0) = f_p(X_0)$ is an empty perception/string, ϵ.

From the above we see that AIμ in Def. 1 corresponds to a special case of perfect rationality defined in Eq. 1 (with a simple linear utility[3], an empty initial percept, and a computable environment[4]), implying that if AIξ converges to AIμ, then AIξ eventually converges to this special case. The problem with this is only that perfect rationality is not feasible, which implies that in general AIXI is not feasible. To overcome this resource bounds need to be introduced.

3.2 Convergence of AIXI to bounded-optimality

In general resource bounded computation can be modeled with time- and/or space-bounded Turing machines. So from now on we model agents' resource bounded architectures that interpret and run their programs with Turing machines. Let $M\colon \mathcal{L}_M \times \mathbf{I} \times \mathbf{O} \to \mathbf{I} \times \mathbf{A}$ denote an *agent's architecture* [Russell & Subramanian, 1995], a fixed interpreter for programs, where $\langle I^\mathbf{T}(t+1), A^\mathbf{T}(t)\rangle = M\big(\ell, I^\mathbf{T}(t), O^\mathbf{T}(t)\big)$. M has 3 inputs and 2 outputs. The inputs are: **(1)** the *agent's* program $\ell \in \mathcal{L}_M$, where $\mathcal{L}_M$ denotes the finite programming language of M; **(2)** the current inner state of the agent at time *t*, $I^\mathbf{T}(t) \in \mathbf{I}$, drawn from the set of possible inner states $\mathbf{I}$ (with initial state i_0), where $I^\mathbf{T}\colon \mathbf{T} \to \mathbf{I}$ denotes the internal state history of the agent; and **(3)** the current percept of the agent at time *t*, $O^\mathbf{T}(t) \in \mathbf{O}$. The outputs of M are: **(1)** the next inner state of the agent at time *t+1*, $I^\mathbf{T}(t+1) \in \mathbf{I}$; and **(2)** the current action of the agent at time *t*, $A^\mathbf{T}(t)$.

We can now define the subset of agent functions, that can be implemented on a given architecture M as follows: $\mathit{Feasible}(M) = \{f | \exists \ell \in \mathcal{L}_M, f = \mathit{Agent}(\ell, M)\}$, where

[3] Though simple, this utility function can be used for on-line reinforcement learning by maximizing expected future reward.

[4] The computability of the environment q is a requirement in AIXI, but it is not mentioned in [Russell & Subramanian, 1995].

$Agent(\ell, M)$ denotes the agent-function implemented by ℓ running on M, if for any E, M generates an action history $A^{\mathbf{T}}$ for which $f(O^t) = A^{\mathbf{T}}(t)$ holds. It would be pointless to discuss feasibility if M wasn't time- and/or space-bounded.

Now a *bounded-optimal agent* with an architecture M for a finite set $\mathbf{E}$ of environments has a program ℓ_{opt} such that

$$\ell_{\text{opt}} = argmax_{\ell \in \mathcal{L}_M}\big(V(\ell, M, \mathbf{E})\big) \quad (4)$$

We can calculate $V(\ell, M, \mathbf{E})$, the expected utility of a program ℓ running on M in case of $\mathbf{E}$ similarly to Eq. 2.

$$V(\ell, M, \mathbf{E}) = \sum_{E \in \mathbf{E}} p(E) \cdot V(\ell, M, E) \quad (5)$$

Finally, by instantiating $f = Agent(\ell, M)$ in Eq. 3 we get $V(\ell, M, E)$, the utility of ℓ running on M in E.

$$V(\ell, M, E) = V(Agent(\ell, M), E) \quad (6)$$

The proposed correspondence with AIXI (if not as obvious as before[5]) is mainly the same as in Section 3.1 except the following differences: policy p^* (c.f. Def. 1) now corresponds to the bounded-optimal agent-program ℓ_{opt}, which codes a Turing machine $M_{\ell_{\text{opt}}}$ that is simulated by M on input $\langle I^{\mathbf{T}}(t), O^{\mathbf{T}}(t)\rangle$ for $\forall t \in \mathbf{T}$, thus M is a universal Turing machine (in accordance with [Hutter, 2005, Ch. 1.7.1, p21]), and generally every AIXI policy p should correspond to a program ℓ in a similar fashion. Expected utility $V_\mu^p \equiv V_{1m}^{p\mu} \coloneqq \sum_q \mu(q)\, V_{1m}^{pq}$ corresponds to Eq. 5, and total utility V_{1m}^{pq} corresponds to $V(\ell, M, E)$ in Eq. 6.

Now it would be too early to claim that AIXI converges to bounded-optimality, since we need to take limited resources into account. In AIXI both p and q are modeled with Turing machines which can be time- and/or space-bounded, but we will focus only on policies and programs, since bounded-optimality assumes no bounds on E. An AIXI policy p is time-bounded if it calculates its output in time $\leq t$ per cycle and space-bounded if its length[6] is $\leq l$. This approach is labeled AIXI*tl* in [Hutter, 2005, Ch. 7.2] and it is not conventional since usually space-bounds apply to the capacity of M_p, the machine corresponding to p, and not to the universal Turing machine running p. So if in bounded-optimality a program ℓ corresponds to a t time- and/or l space-bounded AIXI policy p, then M_ℓ should be t time-bounded and/or ℓ should be $\leq l$ bit long. This will be further refined in frames of our model (c.f. Section 4), but before that we need to address one more topic: *semantics of time*, which wasn't a problem in Section 3.1, but now after the introduction of architectures it shall be considered.

AIXI measures time in cycles, while bounded-optimality in time steps. These two can correspond to each other as shown in Section 3.1, but then neither is a measure of computation time, just an indicator of I/O cycles. In AIXI this is made explicit: cycles are different than the computation time of policies. Section 4 in [Russell & Subramanian, 1995] on the other hand suggests that time steps should correspond to computation time, while they permit also our cycle-based interpretation. This ambiguity is also clarified in our model (c.f. next section).

For now we should conclude that bounded-optimality corresponds to AIμ with resource bounded policies, and that it is the limit to which AIXI with resource bounded policies converges. So eventually since every real implementation is resource bounded, any approximation of AIXI (e.g. [Veness et al., 2011]) is an approximation of bounded-optimality.

4. Decentralized extension of bounded-optimality

Now our main contribution, the decentralized extension of bounded-optimality is presented. Let $\mathbf{N} = \{0,1,2,\ldots,n\}$ be the set of agents in the environment, and let $\mathbf{A}_i$ be the finite set of possible actions of agent $i \in \mathbf{N}$. Similarly to Section 3 we introduce a set of time instants, $\mathbf{T}$, a totally-ordered, finite set of non-negative integers (including 0). In our model $\boldsymbol{\Omega}$ denotes the states of an environment, so the set of possible state trajectories is defined as $\boldsymbol{\Omega}^{\mathbf{T}} = \{\Omega^{\mathbf{T}}: \mathbf{T} \to \boldsymbol{\Omega}\}$, where $\Omega^{\mathbf{T}}$ is a state trajectory, a mapping from time instants to states. The prefix of a state trajectory $\Omega^{\mathbf{T}} \in \boldsymbol{\Omega}^{\mathbf{T}}$ till time t is $\Omega^t = \{\Omega^{\mathbf{T}}(u) | u \in \mathbf{T}, 0 \leq u \leq t\}$ and so the set of possible state trajectory prefixes is $\boldsymbol{\Omega}^* = \{\Omega^t | t \in \mathbf{T}, \Omega^{\mathbf{T}} \in \boldsymbol{\Omega}^{\mathbf{T}}\}$. We can also define the set of action histories of agent *i*, $\mathbf{A}_i^{\mathbf{T}} = \{A_i^{\mathbf{T}}: \mathbf{T} \to \mathbf{A}_i\}$, where $A_i^{\mathbf{T}}$ is an action history of agent *i*. The prefix of an action history $A_i^{\mathbf{T}} \in \mathbf{A}_i^{\mathbf{T}}$ of agent *i* till time *t* is $A_i^t = \{A_i^{\mathbf{T}}(u) | u \in \mathbf{T}, 0 \leq u \leq t\}$ and $\mathbf{A}_i^* = \{A_i^t | t \in \mathbf{T}, A_i^{\mathbf{T}} \in \mathbf{A}_i^{\mathbf{T}}\}$ is the set of possible action history prefixes of agent *i*.

4.1 Specification of agents and environments

An agent is typically described as an abstract mapping (the agent function) from percept sequences to actions, but eventually those percept sequences arise from state sequences perceived by the agent, and this perception mechanism should also be part of the agent function, i.e. $f_i: \boldsymbol{\Omega}^* \to \mathbf{A}_i$ $(i \in \mathbf{N})$, mapping directly from state sequences to actions in order to capture the "complete physical functionality" of the agent. Moreover $A_i^{\mathbf{T}}(t) = f_i(\Omega^t)$ must also hold for $\forall t \in \mathbf{T}$.

To model non-deterministic environments without loss of generality we assume (in accordance with extensive-form games) that an agent *0* represents *chance*, having an agent-function $f_0: \boldsymbol{\Omega}^* \to \mathbf{A}_0$, where $f_0(\Omega^t)$ is chosen randomly according to probability distribution $\pi_0(\Omega^t)$, where $\pi_0: \boldsymbol{\Omega}^* \to \Delta(\mathbf{A}_0)$ denotes a random action policy, a function that maps from the set of state trajectory prefixes to the set

[5] An AIXI policy p could also correspond e.g. to an agent-function $Agent(\ell, M)$ implemented by a program ℓ on an architecture M, or just to an $\langle \ell, M \rangle$ pair, but it wouldn't be as clear as above (c.f. bounded resources), yet it wouldn't change the overall result.

[6] Policy p is an input of a universal Turing machine in AIXI.

of probability distributions over $\mathbf{A}_0$, $\Delta(\mathbf{A}_0)$. The effects of agent *0* are incorporated in the following definition of a state transition relation, $\mathbf{R} \subseteq \mathbf{\Omega} \times (\times_{i=0}^{n} \boldsymbol{A}_i) \times \mathbf{\Omega}$, which is similar to Def. 1 in [Bowling et al., 2002]. The set of actions that agent *i* can perform in $\omega \in \mathbf{\Omega}$ are implied by **R**, and are denoted with $\mathrm{ACT}_i(\omega) \subseteq \mathbf{A}_i$ (the *do-nothing* action should be part of $\mathrm{ACT}_i(\omega)$ for every agent and state), and **R** should satisfy the following two conditions:

1. For $\forall \omega \in \mathbf{\Omega}$ state and $\forall a \in \times_{i=0}^{n} \mathrm{ACT}_i(\omega)$ action-combination $\exists!\, \omega' \in \mathbf{\Omega}$ that $(\omega, a, \omega') \in \mathbf{R}$.
2. If for a given $\omega \in \mathbf{\Omega}$ and $a \in \times_{i=0}^{n} \mathbf{A}_i \quad a \notin \times_{i=0}^{n} \mathrm{ACT}_i(\omega)$ holds, then $\neg\exists \omega' \in \mathbf{\Omega}$: $(\omega, a, \omega') \in \mathbf{R}$.

The first condition is about *completeness* and *determinism* (for every state and executable action-combination there should be just one resulting state defined by **R**), and the second condition is about *consistency* (if an action-combination can't be executed in a given state, then **R** should not define any state resulting from its execution).

Definition 2 (Environment). An *environment*[7] is a 6-tuple, $E = (\mathbf{N}, \mathbf{\Omega}, \omega_0, \mathbf{T}, \{\mathbf{A}_i\}_{i\in\mathbf{N}}, \mathbf{R})$, such that $\Omega^{\mathbf{T}}(0) = \omega_0$, $A_i^{\mathbf{T}}(t) = f_i(\Omega^t)$, and $\left(\Omega^{\mathbf{T}}(t), \left(\times_{i=0}^{n} A_i^{\mathbf{T}}(t)\right), \Omega^{\mathbf{T}}(t+1)\right) \in \mathbf{R}$.

State history $\Omega^{\mathbf{T}}$ is determined by *E* and agent functions f_i $(i \in \mathbf{N})$ which altogether are denoted by $f: \mathbf{\Omega}^* \to \times_{i\in\mathbf{N}} \mathbf{A}_i$, the collective agent function, where $f(\Omega^t) = \left(f_0(\Omega^t), f_1(\Omega^t), \ldots, f_n(\Omega^t)\right)$ for $\forall t \in \mathbf{T}$. We can also define the functionality of all agents except agent *i* as $f_{-i}(\Omega^t) = \left(f_0(\Omega^t), \ldots, f_{i-1}(\Omega^t), f_{i+1}(\Omega^t), \ldots, f_n(\Omega^t)\right)$, and for short we can write $f = (f_i, f_{-i})$. Similarly A^t denotes the collective prefix of action histories $A_i^{\mathbf{T}} \in \mathbf{A}_i^{\mathbf{T}}$ till time *t*, $A^t = \left\{\left(A_i^{\mathbf{T}}(u)\right)_{i\in\mathbf{N}} | u \in \mathbf{T}, 0 \le u \le t\right\}$, and A_{-i}^t denotes the collective prefix of action histories $A_j^{\mathbf{T}} \in \mathbf{A}_j^{\mathbf{T}}$ till time *t* except agent *i* similarly to f_{-i}. For short: $A^t = (A_i^t, A_{-i}^t)$. Based on this *effects*(f, E) denotes the state history generated by a collective agent function f operating in *E*.

4.2 Implementation of agents in environments

We will consider an agent (except agent *0*) to consist of an architecture and a program (like hardware and software). The main difference between the original concept and our approach is that we model agents' sensors and actuators as a part of their architecture, not the environment. Let $\mathbf{O}_j$ denote the set of percepts of agent *j* $(j = 1..n)$, so $\mathbf{O}_j^{\mathbf{T}} = \left\{O_j^{\mathbf{T}}: \mathbf{T} \to \mathbf{O}_j\right\}$ is the *set of* percept histories of agent *j*, where $O_j^{\mathbf{T}}$ is a percept history of agent *j*. The prefix of a percept history $O_j^{\mathbf{T}} \in \mathbf{O}_j^{\mathbf{T}}$ of agent *j* till time *t* is denoted with $O_j^t = \left\{O_j^{\mathbf{T}}(u) | u \in \mathbf{T}, 0 \le u \le t\right\}$, and the set of possible percept history prefixes of *j* is $\mathbf{O}_j^* = \left\{O_j^t | t \in \mathbf{T}, O_j^{\mathbf{T}} \in \mathbf{O}_j^{\mathbf{T}}\right\}$. Now the architecture of an agent can be defined as follows.

Definition 3 (Architecture). *Architecture M_j of agent $j \in \{1,2,\ldots,n\}$ is an $s(\cdot)$ space-bounded*[8] *and/or $t(\cdot)$ time-bounded k_j -tape Embedded Universal Turing Machine, $M_j = \left(\mathbf{I}_j, \Lambda_j, \varepsilon_j, \Sigma_j, i_{0j}, F_j, \delta_j, f_{pj}, f_{aj}\right)$ with a dedicated half-infinite read-only input and half-infinite readable/writeable output tape $\left(k_j \ge 2, k_j \in \mathbb{Z}^+\right)$, where*

- $\mathbf{I}_j$ *is the finite set of internal states of agent j;*
- Λ_j *is a finite, non-empty set of the tape symbols;*
- $\varepsilon_j \in \Lambda_j$ *is the blank symbol;*
- $\Sigma_j \subseteq \Lambda_j \backslash \{\varepsilon_j\}$ *is the non-empty set of input symbols;*
- $i_{0j} \in \mathbf{I}_j$ *is the initial state of the machine;*
- $F_j \subseteq \mathbf{I}_j$ *is the set of final/accepting states;*
- $\delta_j: \mathbf{I}_j \times \Lambda_j^{k_j} \to \mathbf{I}_j \times \left(\Lambda_j \times \{\leftarrow, \rightarrow, \downarrow\}\right)^{k_j}$ *is the transition function, where ← is left shift, → is right shift, and ↓ is no shift of a head over a tape;*
- $f_{pj}: \mathbf{\Omega} \to \Sigma_j^*$ *is the many-to-one perceptual filter function of agent j, where Σ_j^* is the set of finite, unbounded sequences of input symbols. The image of f_{pj} is $Im(f_{pj}) = \mathbf{O}_j \subseteq \Sigma_j^*$, thus f_{pj} models agent j's sensors; it encodes its percepts on the input tape. The inverse of f_{pj} is the information function of agent j, $f_{pj}^{-1}: \Sigma_j^* \to 2^{\mathbf{\Omega}}$;*[9]
- $f_{aj}: \Lambda_j^* \to \mathbf{A}_j$ *is the surjective action function of agent j, where Λ_j^* is the set of finite, unbounded sequences of tape symbols, thus f_{aj} models agent j's actuators; it decodes the executable actions of agent j from the output tape.*

The content of the input and output tape of M_j at any $t \in \mathbf{T}$ is denoted by $input_j(t)$ and $output_j(t)$ respectively, where $input_j: \mathbf{T} \to \Sigma_j^*$ and $output_j: \mathbf{T} \to \Lambda_j^*$. Initially (at $t = 0$) the content of the input tape is $\ell_j \# f_{pj}\left(\Omega^{\mathbf{T}}(0)\right)$, the output tape is completely blank, and the heads are at the beginning of the tapes, over the first symbol. Let $heads_j(t)$ denote the sequence of symbols under the heads at any $t \in \mathbf{T}$, where $heads_j: \mathbf{T} \to \Lambda_j^{k_j}$. The language recognized by M_j is $\mathcal{L}_{M_j} = \left\{\ell_j \# o_j | \ell_j \in \mathcal{P}_{M_j} \subseteq \Sigma_j^*, o_j \in \mathbf{O}_j \subseteq \Sigma_j^*\right\}$, where $\ell_j \in \mathcal{P}_{M_j} \subseteq \Sigma_j^*$ is a *program* and $\mathcal{P}_{M_j} \subseteq \Sigma_j^*$ is the programming language of agent *j*. ℓ_j is interpreted by M_j, so that there is a corresponding M_{ℓ_j} Turing machine, which is simulated by M_j, i.e. M_j accepts, rejects or falls into an infinite loop given the input $\ell_j \# o_j$ iff M_{ℓ_j} accepts, rejects or falls into an infinite loop given the input $o_j \in \Sigma_j^*$

[7] This definition of the environment is discrete and could be made continuous, but that would introduce unnecessary complexity and it is not essential for our results, so we omit its discussion.

[8] Space-bound is a limit on the sum of non-empty cells on k_j tapes.

[9] In accordance with [Aumann, 1976].

respectively and $f_{M_j}(\ell_j \# o_j) = f_{M_{\ell_j}}(o_j)$ for every $o_j \in \Sigma_j^*$, where $f_{M_j}: \Sigma_j^* \to \Lambda_j^*$ and $f_{M_{\ell_j}}: \Sigma_j^* \to \Lambda_j^*$ denote the input-output functions implemented by Turing machines M_j and M_{ℓ_j} respectively. Beyond that $\delta_j\left(I_j^{\mathbf{T}}(t), heads_j(t)\right) = \langle I_j^{\mathbf{T}}(t+1), \langle write_{j,k}(t), move_{j,k}(t)\rangle_{k=1}^{k_j}\rangle$, $I_j^{\mathbf{T}}(0) = i_{0j}$, $O_j^{\mathbf{T}}(t) = f_{pj}(\Omega^{\mathbf{T}}(t))$, $input_j(t) = \ell_j \# f_{pj}(\Omega^{\mathbf{T}}(t))$, [10] and $A_j^{\mathbf{T}}(t) = f_{aj}\left(output_j(t)\right)$ should hold[11], where $write_{j,k}(t)$ denotes the symbol written on tape k of M_j at time t, $move_{j,k}(t)$ is the following movement of the head over tape k, $I_j^{\mathbf{T}} \in \mathbf{I}_j^{\mathbf{T}} = \{I_j^{\mathbf{T}}: \mathbf{T} \to \mathbf{I}_j\}$ is the internal state history with $\mathbf{I}_j^{\mathbf{T}}$ being the set of possible internal state histories of j.

Now we can relate agents' programs to their functions: a program $\ell_j \in \mathcal{P}_{M_j}$ running on M_j *implements* an agent function $f_j = Agent(\ell_j, M_j, E)$ $(j = 1..n)$ in E iff $f_j(\Omega^t) = A_j^{\mathbf{T}}(t)$, $A_j^{\mathbf{T}}(t) = f_{aj}\left(output_j(t)\right)$, $\delta_j\left(I_j^{\mathbf{T}}(t), heads_j(t)\right) = \langle I_j^{\mathbf{T}}(t+1), \langle write_{j,k}(t), move_{j,k}(t)\rangle_{k=1}^{k_j}\rangle$, $O_j^{\mathbf{T}}(t) = f_{pj}(\Omega^{\mathbf{T}}(t))$ $input_j(t) = \ell_j \# O_j^{\mathbf{T}}(t)$, $\left(\Omega^{\mathbf{T}}(t), \left(\times_{i=0}^{n} A_i^{\mathbf{T}}(t)\right), \Omega^{\mathbf{T}}(t+1)\right) \in \mathbf{R}$, $\Omega^{\mathbf{T}}(0) = \omega_0$ and $I_j^{\mathbf{T}}(0) = i_{0j}$ holds.

Although every program ℓ_j induces a corresponding agent function $Agent(\ell_j, M_j, E)$ in E, [12] not every agent function has an implementation $\ell_j \in \mathcal{P}_{M_j}$ in case of a given environment E and an architecture M_j. We can define a subset of the set of agent functions f_j that are implementable on a given architecture M_j and programming language $\mathcal{P}_{M_j}$ in environment E as follows: $Feasible(M_j, E) = \{f_j | \exists \ell_j \in \mathcal{P}_{M_j}, f_j = Agent(\ell_j, M_j, E)\}$. [13]

4.3 Utility of decentralized agents

We define a real-valued *utility function* over state trajectories for every agent $i \in \{1,2,\ldots,n\}$ to measure their performance in the environment: $U_i: \boldsymbol{\Omega}^{\mathbf{T}} \to \mathbb{R}$.[14] It implicitly defines their goal set by their user or Designer. Several (or all) agents can have the same user or Designer and thus the same utility function. A combination of an environment and the utility functions is a *decentralized task environment*.

[10] The actual percept of agent j should be always updated by f_{pj}.

[11] Effectors of agent j should be driven by f_{aj} at every instant.

[12] Compared to single-agent bounded-optimality the difference is that the implemented agent function depends also on E, because the perceptual filter function is now part of agents' architecture. If the environment (e.g. state space) changes then the same perceptual filter function may not be valid anymore.

[13] The set of feasible agent functions also depends on E.

[14] Utility of agents may need to be calculated for state trajectory prefixes instead of whole trajectories (e.g. for on-line decision making). It may be calculated even for any time-interval $[t_1, t_2]$, $t_2 > t_1$, as the difference between utilities of $[0, t_2]$ and $[0, t_1]$.

Based on the above we can define the value of an agent function f_i in an environment E with other f_{-i} agents as the expected utility $\mathbb{E}[U_i(\cdot)]$ of state histories they generate:[15]

$$V_i((f_i, f_{-i}), E) = \mathbb{E}\left[U_i\left(effects((f_i, f_{-i}), E)\right)\right] \quad (7)$$

The value of agent i in a *set* $\mathbf{E}$ *of environments*[16] with a probability distribution p over them, and with other f_{-i} agents is the expected value of Eq. 7.[17]

$$V_i((f_i, f_{-i}), \mathbf{E}) = \sum_{E \in \mathbf{E}} p(E) \cdot V_i((f_i, f_{-i}), E) \quad (8)$$

Similarly the value of ℓ_i executed by M_i in E with other f_{-i} agents can be given simply by looking at the effects of the collective agent function $f = (Agent(\ell_i, M_i, E), f_{-i})$.

$$V_i(\ell_i, M_i, f_{-i}, E) = V_i((Agent(\ell_i, M_i, E), f_{-i}), E) \quad (9)$$

The value of ℓ_i run by M_i in $\mathbf{E}$ with other f_{-i} agents is:

$$V_i(\ell_i, M_i, f_{-i}, \mathbf{E}) = \sum_{E \in \mathbf{E}} p(E) \cdot V_i(\ell_i, M_i, f_{-i}, E) \quad (10)$$

4.4 Rationality of decentralized agents

A *perfectly rational* agent i in the above setting has an agent function f_i that maximizes $V_i((f_i, f_{-i}), \mathbf{E})$ over all agent functions of i, i.e. it has an agent function f_i^* such that

$$f_i^* = argmax_{f_i}\left(V_i((f_i, f_{-i}), \mathbf{E})\right) \quad (11)$$

As we see, other agents' f_{-i} functionality is explicitly considered in this decentralized definition of perfect rationality. It means that this condition must be met by the agents to be perfectly rational in the decentralized case, but since f_i^* is independent of M_i, it may be that f_i^* is not feasible, i.e. $f_i^* \notin Feasible(M_i, E)$. For this reason we impose optimality constraints better on programs rather than agent functions. Agent i with architecture M_i is *bounded-optimal* for a set $\mathbf{E}$ of environments with other f_{-i} agents, if it has an agent program $\ell_i^* \in \mathcal{P}_{M_i}$ such that

$$\ell_i^* = argmax_{\ell_i \in \mathcal{P}_{M_i}}\left(V_i(\ell_i, M_i, f_{-i}, \mathbf{E})\right) \quad (12)$$

We can notice that ℓ_i^* is a best-response to M_i, f_{-i}, and $\mathbf{E}$, but this does not mean that f_{-i} (or any part of it) should be also a best response like in Nash-equilibrium [Nash, 1951], so in this sense we are more general than Nash-equilibrium.

[15] It is an expected utility since the chance agent is also part of f_{-i}.

[16] All $E \in \mathbf{E}$ should have same *agents*, *states*, *time* and *actions*. Only the *initial state* and the *state transition relation* may differ.

[17] Observe that because all the elements of $\mathbf{E}$ should have the same agents, i is present in every $E \in \mathbf{E}$. This is a necessary condition.

Nonetheless it can be hard to realize such a program. For this sake we propose the following relaxation: agent i is *time- or space-wise average-case asymptotically bounded optimal (ABO)* in $\mathbf{E}$ on M_i with other f_{-i} agents, if it has a program $\ell_i \in \mathcal{P}_{M_i}$ such that $\exists k$ for which for $\forall \ell_i'$ $V_i(\ell_i, kM_i, f_{-i}, \mathbf{E}) \geq V_i(\ell_i', M_i, f_{-i}, \mathbf{E})$ holds. kM_i denotes a variant of M_i, which is k times faster (or has k times more memory), i.e. the program is on the right lines, it only needs a better architecture. An initial idea for the realization of such programs in general was given in [Kovacs, 2005].

According to Def. 13 in [Russell & Subramanian, 1995] worst-case ABO could also be defined, but due to limited paper extent we omit it now. Still, based on their Def. 15 a decentralized notion of *universal asymptotic bounded optimality* (*UABO*) can be given: agent i is UABO if it has a program $\ell_i \in \mathcal{P}_{M_i}$ running on M_i in $\mathbf{E}$ with other f_{-i} agents for a family of value functions $\mathcal{V}_i$ iff ℓ_i is ABO in $\mathbf{E}$ on M_i with f_{-i} in case of every value function $V_i \in \mathcal{V}_i$. That means the program is flexible to (e.g. temporal) variation of the utility function, which is important for real-time systems.

We can extend all of the above decentralized rationality concepts (perfect rationality, BO, ABO, UABO) to a non-empty group of agents $\mathbf{G} \subseteq \mathbf{N}$ just by looking at them as one collective agent in $\mathbf{E}$.[18] Such a *collective agent* $\mathbf{G}$ has a collective agent function $f_{\mathbf{G}}: \Omega^* \to \times_{i \in \mathbf{G}} \mathbf{A}_i$, where $f_{\mathbf{G}}(\Omega^t) = \left(f_i(\Omega^t)\right)_{i \in \mathbf{G}}$ for $\forall t \in \mathbf{T}$ so that $f_i = Agent(\ell_i, M_i, E)$ holds for $\forall i \in \mathbf{G}$, $\forall E \in \mathbf{E}$. The utility $U_{\mathbf{G}}$ of $\mathbf{G}$ is an arbitrary function of individual utilities, $U_{\mathbf{G}}(\cdot) = h(\langle U_i(\cdot)\rangle_{i \in \mathbf{G}})$, where $h: \mathbb{R}^{|\mathbf{G}|} \to \mathbb{R}$ denotes an arbitrary real-valued function. The program, architecture or percept and action at time t of the group is a combination of individual programs, architectures or percepts and actions at time t of the members respectively. This way a group $\mathbf{G} \subseteq \mathbf{N}$ can be handled just like a single agent, and so the application of our above rationality concepts becomes straightforward.

4.5 Connection to the original concept

The original idea of bounded-optimality is a special case of our model if **(1)** $n = 1$ or **(2)** $\mathbf{G} = \mathbf{N}$. In the *first case* only 1 agent is modeled explicitly in a deterministic environment either because it is the only one or since other deterministic agents' functionality is integrated into the transition function f_e which corresponds directly to the state transition relation $\mathbf{R}$. In the *second case* there may be multiple agents, but all of them are considered as a single collective agent $\mathbf{G} = \mathbf{N}$ in accordance with the end of Section 4.4.

In both cases chance agent *0* is either not present or deterministic and $f_p = f_{p\mathbf{G}}$ holds, so any (even implemented) agent function f in the original concept corresponds to f' in $f_{\mathbf{G}} = f' \circ f_{p\mathbf{G}}$ in our model[19], i.e. $f = f'$.

[18] $\mathbf{N}\backslash\mathbf{G}$ may include other groups of agents and agent $i \in \mathbf{N}$ may belong to more groups.

[19] Since in our model agent functions map from state trajectories.

5. Examples

Now we present a few examples of our model in case of a simplified Wireless Sensor Network (WSN) network-layer routing scenario [Tanenbaum & Wetherall 2011] (c.f. Fig. 1).

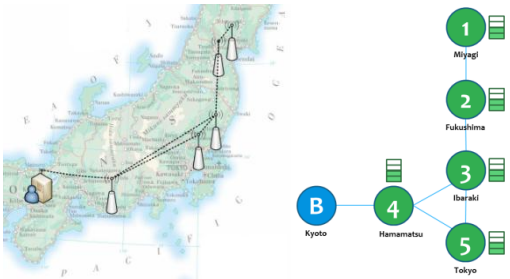

Fig.1 Example: geographical (*left*) and schematic (*right*) topology of an earth tremor measuring wireless sensor network (WSN)

Let's assume we have 5 static nodes for earth tremor measurement with limited amount of energy and 1 base station in 6 separate locations connected as in Fig. 1. Our goal is that the base receives measured data from every location, i.e. we need to design a protocol for nodes to realize this. A node can measure, send or aggregate data. Measurement takes 0.1 units of time and 1 unit of energy; sending takes 1 unit of time and energy. The cost of aggregation is negligible. Measurement produces data about the location of a node, which can then be either sent to another node or aggregated with received data. In case of aggregation the accuracy of aggregated data is reduced proportionally. Our question is: *how should the nodes behave to accomplish our goal in the shortest time, with minimal overall energy usage and data inaccuracy?*

To answer this question let's model the situation. Let $\mathbf{N} = \{1,2,3,4,5,\mathrm{B}\}$ be the set of nodes (agents), and let $\mathbf{A}_i = \{do\_nothing, measure\} \cup \left(\{\varepsilon, aggregate(I,J)_{I \subset \mathbf{N}, I \neq \emptyset, J \subseteq \mathbf{N}\backslash I}\} \times \{send(i,J,k)\}_{k \in \mathbf{N}:(i,k) \in \mathbf{L}, J \subseteq \mathbf{N}}\right)$ be the set of possible actions of agent $i = 1..5$, where $\mathbf{L} \subseteq \mathbf{N} \times \mathbf{N}$ is the set of directed links between nodes as in Fig. 1. $\mathbf{A}_{\mathrm{B}} = \{do\_nothing\}$, and set $\mathbf{\Omega}$ of states is a special case of Section 4, where $\mathbf{\Omega} = 2^{\mathbf{P}}$, and $\mathbf{P} = \{data\_at(i,j)\}_{i,j \in \mathbf{N}} \cup \{energy\_of(i,j)\}_{i \in \mathbf{N}\backslash\{\mathrm{B}\}, j \in \{0..4\}}$ is the set of propositions, i.e. states have a logical description. The state transition relation $\mathbf{R}$ should be according to the above informal description, with goal states being $\mathbf{\Omega}_g = \{\omega \in \mathbf{\Omega} | \{data\_at(i,\mathrm{B})\}_{i=1..5} \subseteq \omega\} \subseteq \mathbf{\Omega}$. Let the utility of agents be the same, $U_i = U$ for $\forall i \in \mathbf{N}$, as follows.

$$U(\Omega^{\mathbf{T}}) = \begin{cases} 0 & \text{if } \Omega^{\mathbf{T}}(t_{max}) \notin \mathbf{\Omega}_g \\ 1/TOI(\Omega^{\mathbf{T}}) & otherwise \end{cases} \quad (\forall \Omega^{\mathbf{T}} \in \mathbf{\Omega}^{\mathbf{T}}) \quad (13)$$

Here $TOI(\cdot) = \alpha_t t(\cdot) + \alpha_e e(\cdot) + \alpha_d d(\cdot)$ denotes the *Trade-off Index* [Li et al., 2010] which we seek to minimize, and $\alpha_t, \alpha_e, \alpha_d \in [0,1]$ are coefficients of timespan $t(\cdot)$, overall energy usage $e(\cdot)$ and data inaccuracy $d(\cdot)$ of a state trajectory. $d(\cdot) = D_{gen}(\cdot)/D_{base}(\cdot)$ is the proportion of the number of data measured in the network, $D_{gen}(\cdot)$, and the

number of separate messages received by the base, $D_{base}(\cdot)$. $t_{max} = max(\mathbf{T})$ denotes the latest time instant. We could extend this model to a probabilistic setting (e.g. packet loss, data generation rate, failure) by introducing a chance agent *0*, but for now we decided not to complicate things further.

Example 1: first let's assume that initially node 1-5 has 2 units of energy as in Fig.1, i.e. $\omega_0 = \{energy\_of(i, 2)\}_{i \in \mathbf{N} \setminus \{B\}}$, and say $\alpha_t = \alpha_e = \alpha_d = 1$. Since every agent has the same utility, they should aim for the same goal, i.e. it is rational to cooperate. It is easy to see what actions bounded-optimal programs $\{\ell_i\}_{i \in \mathbf{N}}$ should choose by identifying optimal collective behaviors, which now are *fully aggregating* (every node should do a measurement at the very beginning simultaneously, then wait for farther nodes' data to arrive, aggregate it with its own data, and then send the package toward the base along the shortest path as soon as possible). That is so because initially every node has only 2 units of energy, while measurement and sending both take 1-1 unit. So after a measurement a node can send only 1 message.

Example 2: now if node 4 would have 3 units of energy initially, then it would be worth for it to send not only 1, but 2 messages (e.g. its own data separately), i.e. to do *partial aggregation* to reduce data inaccuracy (from 5 to 2.5). I.e. a change of node 4's resources modifies its bounded-optimal program while other BO agents remain fully aggregating.

Example 3: suppose that $\alpha_t = \alpha_d = 0$, but $\alpha_e = 1$ in Example 1. In this case any collective behavior reaching the goal would have the same constant $TOI(\cdot) = 5 \cdot 2 = 10$, which makes it impossible to distinguish among them. All of them would be bounded-optimal even if they wait for arbitrarily long with measurement or sending data. This is a limitation of our model: a strong dependence on the definition of utility (beside being discrete and allowing only finite sets of environments, agents, actions and percepts).

6. Conclusions and future work

In this paper we introduced a decentralized notion of bounded-optimality (DBO), compared it with other rationality concepts, showed that it is an implicit multi-agent extension of AIXI, because AIXI converges to BO, and connected DBO to the original notion of BO. A few examples were given to show the use and limitations of our approach, which is globally optimal if other agents' f_{-i} functionality is given (c.f. Eq. 12), otherwise it is difficult to achieve global optimality. There is no central mechanism posed upon the collective of agents, i.e. DBO agents can be analyzed in pre-existing environments with other users' non-controllable agents (e.g. for assessing the performance of our Internet poker agent in a room filled with arbitrary competitors). We have also shown that DBO allows direct connection of feasible individual and group-level rationality in a straightforward way, which wasn't possible until now.

In the future we wish to give a formal analysis of the asymptotic realization of DBO programs in general; extend our model to the continuous case, infinite sets of states, agents, actions and percepts; and to conduct a deeper investigation of individual and group-level rationality.

Acknowledgements

We wish to thank SUZUKI Foundation for sponsoring this research, to Dr. Stuart Russell for invaluable insights and to Dr. Tadeusz Dobrowiecki for his helpful critical suggestions.

◊ References ◊

[Aumann 1976] Aumann, R. J.: *Agreeing to Disagree*. The Annals of Statistics, Vol. 4(6), pp. 1236–1239 (1976)

[Bowling et al. 2002] Bowling, M., Jensen, R., Veloso, M.: *A formalization of equilibria for multiagent planning*. AAAI Workshop on Planning with and for Multiagent Systems, pp. 1236–1239 (2002)

[Carlin & Zilberstein 2011] Carlin, A., Zilberstein, S.: *Decentralized Monitoring of Anytime Decision Making*, Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS), pp. 157-164 (2011)

[Cox & Raja 2011] Cox, M. T., Raja, A.: *Metareasoning: Thinking about thinking*, MIT Press (2011)

[Hutter 2005] Hutter, M.: *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*, Springer (2005)

[Kovacs 2005] Kovacs, D. L.: *Virtual Games: A New Approach to Implementation of Social Choice Rules*, In Proc. of 4th International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS), Lecture Notes in Computer Science (LNCS), Vol. 3690, pp. 266-275, Springer (2005)

[Li et al. 2010] Li, W., Bandai, M., Watanabe, T.: *Tradeoffs among Delay, Energy and Accuracy of Partial Data Aggregation in Wireless Sensor Networks*, In Proc. of IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 917-924 (2010)

[Nash 1951] Nash, J. F.: *Non-cooperative Games*, Annals of Mathematics, Vol. 54, pp. 286-95 (1951)

[Neumann & Morgenstern 1944] Neumann, J., Morgenstern, O.: *Theory of games and economic behavior*, Princeton (1944)

[Ong et al. 2010] Ong, S. C. W., Png, S. W., Hsu, D., Lee, W. S.: *Planning under Uncertainty for Robotic Tasks with Mixed Observability*, International Journal of Robotics Research, Vol. 29:8, pp. 1053-1068 (2010)

[Pattanaik 2008] Pattanaik, P. K.: *Social welfare function*, in S. Durlauf and L. Blume (eds.), The New Palgrave Dictionary of Economics, 2nd edition, Palgrave Macmillan Ltd. (2008)

[Russell & Subramanian 1995] Russell, S. J., Subramanian, D.: *Provably bounded-optimal agents*, Journal of Artificial Intelligence Research, Vol. 2, pp. 575-609 (1995)

[Russell & Norvig 2010] Russell, S. J., Norvig, P.: *Artificial Intelligence: A Modern Approach. 3rd Ed.*, Prentice Hall (2010)

[Silver & Veness 2010] Silver, D., Veness, J.: *Monte-Carlo Planning in Large POMDPs*, In Proc. of Advances of Neural Information Processing Systems (NIPS), pp. 2164-2172 (2010)

[Tanenbaum & Wetherall 2011] Tanenbaum, A. S., Wetherall, D. J.: *Computer Networks*, 5th ed., Prentice Hall (2011)

[Veness et al. 2011] Veness, J., Ng, K. S., Hutter, M., Uther, W., Silver, D.: *A Monte-Carlo AIXI Approximation*, Journal of Artificial Intelligence Research, Vol. 40, pp. 95-142 (2011)