Predictive Heuristics for Decision-Making in Real-World Environments

Helgi Páll Helgason[1], Kristinn R. Thórisson[1,2], Eric Nivel[1] & Pei Wang[3]

[1]Center for Analysis and Design of Intelligent Agents / School of Computer Science, Reykjavik University, Menntavegur 1, 101 Reykjavik, Iceland
[2]Icelandic Institute for Intelligent Machines, 2.h. Uranus, Menntavegur 1, 101 Reykjavik
[3]Department of Computer and Information Sciences, Temple University, Philadelphia, USA
helgih09@ru.is, eric@ru.is, thorisson@ru.is, pei.wang@temple.edu

Abstract. In this paper we consider the issue of endowing an AGI system with decision-making capabilities for operation in real-world environments or those of comparable complexity. While action-selection is a critical function of any AGI system operating in the real-world, very few applicable theories or methodologies exist to support such functionality, when all necessary factors are taken into account. Decision theory and standard search techniques require several debilitating simplifications, including determinism, discrete state spaces, exhaustive evaluation of all possible future actions and a coarse grained representation of time. Due to the stochastic and continuous nature of real-world environments and inherent time-constraints, direct application of decision-making methodologies from traditional decision theory and search is not a viable option. We present *predictive heuristics* as a way to bridge the gap between the simplifications of decision theory and search, and the complexity of real-world environments.

Keywords: artificial intelligence, heuristics, action-selection, resource management

1 Introduction

While real-world environments are the ultimate target domains of most AGI architectures, few solutions exist in the literature for rational decision-making under the constraints imposed by such environments. Most methods from decision theory rely on assumptions that preclude their application in this context; namely deterministic environments, discrete state spaces, coarse-grained representations of time and unlimited resources. For example, Russell (1989) presents a resource-bounded decision theoretic framework which accounts for the cost of decision-making, but fails to address the stochastic nature of the environment.

For an overview of how many present AGI architectures fail to address operating features common to all real-world environments, see Thórisson (2012a), such as uncertainty and incomplete knowledge.

adfa, p. 1, 2011.

In this paper, we propose *predictive heuristics* as a viable solution to the decision-making problem in the context of AGI and real-world environments. As opposed to exhaustive evaluation of all possible future states, its functionality is based on relaxing some of the constraints inherent in traditional search and employing rationally-directed, selective evaluation of possible and probable future states.

2 Traditional heuristic search

In traditional search (as presented in any entry-level AI textbook), action-selection in a particular state begins by enumerating and generating all possible next states - or nodes, on the next level of the search tree – in what is called the expansion phase. All of these possible future states are then evaluated using a utility function and the action leading to the state with the highest utility value is chosen as the next action. Some applications of search focus on terminal states and do not require a utility function. These include game-playing, where terminal states are states that end the current game either in a draw, in favor of the system as a player or in favor of the opponent. However, a terminal state is not a very intuitive concept to guide decisions of AGI systems operating in an open-ended fashion in real-world environments.

The expansion and evaluation phases are frequently repeated more than one step into the future in order to evaluate what lies beyond a particular single action. Time is represented in a coarse-grained manner where each decision step and following possible states are both atomic units of time; conceptually all possible next states are thus assumed to occur at a fixed step length in time while their actual time of occurrence is unspecified.

Heuristics may be defined as being “strategies using readily accessible, though loosely applicable, information to control problem solving in human beings and machines” (Pearl, 1983, p. 7) and are usually domain-dependent in some way, for example representing “rules-of-thumb” from the particular problem domain. They have commonly been used in search problems to increase the efficiency of search algorithms as approximation methods to identify future states that are likely to be more rewarding than others. As the concept of heuristics has a loose definition, implementations vary. Heuristics are part of the utility function for future states in A* search (Hart 1968). A more general type of heuristics, hyper-heuristics, has been proposed (Burke 2003). Hyper-heuristics are domain-independent in nature, described as methods for selecting lower-level heuristics at run-time from a predefined set of low-level heuristics as appropriate to the present step of the problem solving process (Özcan 2008). Hyper-heuristics may be understood as a method for optimizing the application of manually-generated domain-dependent heuristics at run-time. Realtime operation in search and heuristics has been addressed to a degree; most notably by the *Real-Time A** algorithm proposed by Korf (1990).

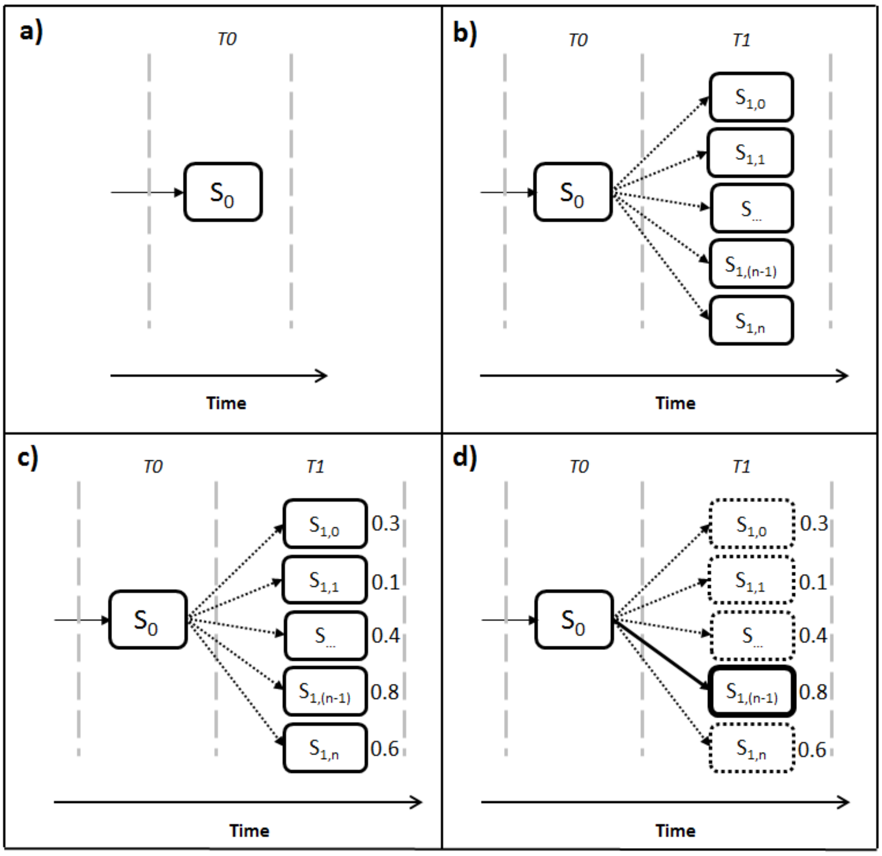

Fig. 1. State-spaces in typical search problems and the application of heuristics. a) The state-space is represented in atomic temporal steps with a tree structure where each level of the tree corresponds to an atomic moment of time. The initial state S0 occurs at time T0. b) All possible states in the next moment of time (T1) after S0 are enumerated resulting in the generation of possible future states $S_{1,0}$ to $S_{1,n}$. c) All states generated in the previous step are evaluated using a heuristic utility function. The resulting utility value for each state is noted in the figure. d) Comparison of utility values finds the state with maximum utility value. This results in either the selection of an action producing that state or an expansion of that state where following states are evaluated. In the latter case, heuristics control how the search tree is expanded.

3 Challenges of real-world environments

Determinism, discrete state-spaces and coarse-grained temporal representations all present significant problems for AGIs intended to operate in the real-world in environments of real-world complexity. In such environments, determinism is a problem since what has reliably worked in the past is not guaranteed to work in the future; the environment may change or some external entity may unexpectedly influence how events unfold. Discrete state-spaces are a problem as the state of real-world environments must be represented largely by continuous values, eliminating the possibility of enumerating all possible future states, let alone the resource requirements for evaluating all of them. While fine-grained discretization can approximate continuous values, each approximated value may still take anywhere from 2^{32} to 2^{64} different values. In operating situations involving multiple approximated values, the state-space quickly grows out of control from the resulting combinatorial explosion *if all possible future states must be considered*. A more coarsely grained approximation can reduce the state-space, but is also likely to negatively impact performance at some point. Coarse-grained representations of time are a problem as changes in real-world environments do not occur simultaneously at relatively wide, fixed, synchronized intervals. For these reasons, exhaustive evaluation of all possible future actions – and thus optimality in decision-making that guarantees the best outcome – is impossible in real-world environments in resource-bounded AGI systems.

Changing the assumption of environmental determinism into a probabilistic environment leaves the nature of the issue unchanged. For example, in a Markov decision process (MDP) the next state after an action is random, with a probabilistic distribution. While closer to the real-world environment by capturing the uncertainty about the consequences of actions, a stationary probabilistic distribution for the states following an action are nevertheless unavoidable, and consequently truly novel situations and unanticipated situations are precluded. Furthermore, probabilistic models usually have even higher resource demands than deterministic models, given the large number of possible consequences of each action.

This implies that if we want to properly address this issue, the only feasible approach left is the *selective evaluation* of all possible future states. However, accepting this challenge gives us another problem; that of mapping out selected *future points of interest*: We must invent a process of *selective generation of possible future states of value to the AGI system.*

4 Adapting search to real-world environments

The traditional setting for search can be altered to accommodate decision-making in real-world environments. First, a fine-grained representation of time must be accommodated in the decision-making process. The distinction between fine- and coarse-grained representations should be viewed relative to the frequency of changes in the operating environment where finer grained representations encode the actual sequence of events with greater accuracy. This distinction may also be viewed as the difference between an order-based versus a measure-based representation, the latter being desired here. While this only applies to events relevant to the operation of the system, these events are unknown at design time due to the domain-independent nature of AGI systems; consequently, the finest possible or practical granularity should be targeted. This is possible if the requirement of considering only simultaneous possible actions (at the next coarse-grained time step) is simply dropped. The focus of the decision-making process is still one step of action into the future. However the size of such a step is allowed to vary in length along the future part of the temporal dimension for each possible action. This length is determined by the timing of selected states that end up being evaluated. The result is that meaning is given to the length of the links in Figure 1, representing when in time the possible future states occur. As already discussed the enumeration of all possible future states – even at a fixed point in time – is intractable in real-world environments. For this reason, the requirement of generating all possible future states must be dropped in favor of selectively generating only a small subset of these. This addresses the enumeration problem. Finally, the stochastic nature of the environment must be acknowledged by estimating the likelihood of generated future states as opposed to taking their occurrence for granted, given some action leading up to them. The evaluation of likelihood does not have to assume a stationary probability distribution. Even so, the likelihood of a future state should influence its evaluation; it seems reasonable to discount the value of a highly favorable future state (in terms of the utility function of the AGI system) if its actual occurrence is not likely. Conversely, it is rational to amplify the value of a possible future state of average value (relative to other possible future states) if its actual occurrence is virtually guaranteed. This addresses the issue of deterministic environments.

Conceptually, the search tree structure is still valid for representing the decision problem, but evenly distributed levels of the tree disappear as the length of links between nodes now represents the duration of time elapsing between states. This implies that the depth of a node becomes its distance in the temporal dimension from the root node, as opposed to the number of intermediate actions.

5 Predictive heuristics

While AGI systems require some type of heuristics-like functionality in order to detect future states of potential interest, these cannot be directly unleashed on an existing set of possible future states as that information is not available. One possible solution is to generate "imaginary" future situations that are likely to occur in the future, where each situation is not fully specified (as a "state" in the traditional sense). The application of search to such partial states, which only deal with changes in the operating environment that have goal-relevance and leave other changes unaddressed, coupled with the modified search methodology presented in the previous section, which allows simultaneous evaluation of actions at arbitrary distances into the future, and the formulas presented below, that incorporate uncertainty, incomplete knowledge and temporal context represent the core of the idea presented in this paper.

Such predictions could be made based on the present operating situation, the operational experience of the system and some suggested actions on part of the system (which should include inaction). By continuously generating future predictions that collectively represent a set of events that have more probability of occurring than others, the AGI system can attempt to stay some steps ahead of the environment and thus increase its chances of being prepared, by pre-computing – mentally preparing – some aspects of the potential actions that might achieve its active goals at those future steps. It seems rational to direct the resources of the system towards events that have a greater probability of occurring rather than towards the much greater (infinite?) number of improbable ones. An implication of this approach is that the system will be unable to anticipate, prepare for or actively avoid events that cannot be rationally predicted in some way by its operational experience. But no known intelligence has this ability either.

Having adapted the search problem to real-world environments, some challenges remain. One of the key ones is the issue of how possible future states are selectively generated and the estimation of their likelihood. Clearly, possible future states constitute states that are likely to occur in case of a) inaction and b) selected actions on part of the AGI system. Predictions made on the basis of possible future actions of the AGI system can be viewed as a form of *goal-directed simulation*, not to be confused with simulation-based search methods such as *Monte-Carlo Tree Search* (Chaslot 2008). A complete enumeration of all possible actions on part of the system is intractable for the same reason as exhaustive enumeration of all possible future states is; most actions can be assumed to include parameters with continuous values making the set of all possible actions potentially infinite. For this reason, the system must suggest a set of goal-relevant actions. While the functionality required for this is outside the scope of this paper, our experience indicates that attentional functionality is of key importance for this purpose (Helgason et al. 2012). In general, any slight or major improvement in predicting relevancy will increase the value of the work proposed here.

If we denote the set of suggested actions as $\mathbf{\Psi}$ and a set containing inaction is denoted as $\mathbf{I}$, the complete set of actions for consideration is the union of $\mathbf{\Psi}$ and $\mathbf{I}$, denoted as $\mathbf{\Omega}$. Given the set $\mathbf{\Omega}$, the selective generation of possible future states of interest can be approached as a prediction problem where the hypothetical states resulting from each action contained in $\mathbf{\Omega}$ are predicted. The set containing these possible future states is denoted $\mathbf{\Theta}$. In this case, the decision-making problem boils down to computing an expected value (where the likelihood of the occurrence of the state is given consideration) for each possible future state in $\mathbf{\Theta}$ and finding the state with maximum value. A set of predictors, denoted $\mathbf{P}$, is used to generate $\mathbf{\Theta}$ where each predictor ($\mathbf{p_i}$) is a process that takes a subset of the present state of the environment and the system itself in addition to a single action from $\mathbf{\Omega}$ as inputs and outputs a prediction of a new state occurring at an explicit point in time (in the future). Each predictor uses (or is built from) the operational history of the AGI system, which is a necessary basis of all predictions. Furthermore, the performpance of each predictor is measured over time with two parameters: *success rate* and *confidence*. These parameters and the way in which they are calculated are based on Wang's (2006: 59-62) method for evaluating truth of logical statements in NARS, which is motivated in the cited text.

No further selection is necessary after the set $\mathbf{\Theta}$ has been selectively populated; the very fact that a state was generated (predicted) indicates that it is worthy of the resources necessary for evaluation. Not only does this allow the AGI system to choose rational actions likely to advance its goals, it may also allow the system to detect that undesirable events are likely to occur in the near future, which the system can then generate explicit goals to avoid.

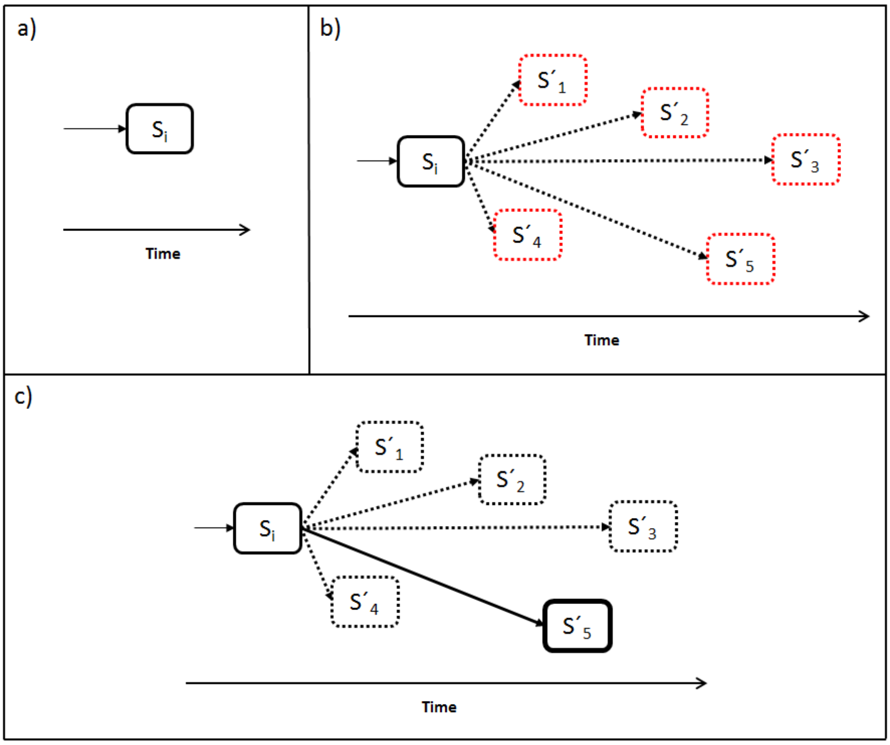

Fig. 2. Predictive heuristics. a) The initial state of S_i occurs at a specific point on a continuous (or fine-grained) axis of time. b) Based on the state S_i and the set of suggested actions (Ω), a finite set of possible future states (each denoted S') is generated that may be distributed on the future part of the temporal axis. c) Each S' state is evaluated and S'_5 found most desirable (having the highest expected value), causing the selection of the action leading to that state or the examination of states following S'_5 where the steps depicted here are repeated with S'_5 as the initial state.

$$Success\ rate(p_i) = \frac{|Sp_{i+}|}{|Sp_i|}$$

$$Confidence(p_i) = \frac{|Sp_i|}{|Sp_i| + 1}$$

where:
$\boldsymbol{Sp_{i+}}$ is the set of prior successful predictions made by p_i
$\boldsymbol{Sp_i}$ is the set of all prior predictions made by p_i

The success rate is the ratio of successful predictions that the predictor has made in the past. The confidence represents the reliability of the success rate value based on the amount of evidence supporting it. Using these two values, a *likelihood* value can be computed that indicates the likelihood of a particular future state occurring. This value should be interpreted as relative to the likelihood of other future states under consideration as opposed to a strict probabilistic interpretation. The likelihood of a prediction **S'** made by predictor $\mathbf{p_i}$ is computed using Wang's (2006: 75-76) formula for expectation as:

$$Likelihood(S') = Confidence(p_i) * (Success\ rate(p_i) - 0.5) + 0.5$$

Unlike in probability theory, this likelihood measurement is not based on a stationary distribution function, since we do not assume the prediction results are random numbers governed by a fixed distribution. The formula above incorporates two critical issues for decision theory: Uncertainty and incomplete knowledge. It addresses non-determinism in the operating environment without using probability distributions for action outcomes, which has inherent limitations.

As the system is expected to be goal-driven, the evaluation function for future states should be based on goal-achievement. Each goal of the system is assumed to have an associated priority and deadline values; however in the absence of these, default initial values may be used. Time is represented as a single numeric value. The *Achieved* function evaluates to 1.0 if goal $\mathbf{g}_i$ is achieved in the specified state and -1.0 otherwise. Each state occurs at a specific time **t**, which is encoded in the state itself. The *Utility* function evaluates the value of achieving a goal given the priority of the goal and temporal context of a specified state. For the sake of clear notation, two helper functions are used and a special set, **H**, is introduced which contains all *time horizons* (quantified intervals of time) between the time of all states currently under consideration and the deadline of the goal spawning the state. The *Urgency* function returns 0 if either value is equal or less than 0.

$$Horizon(g, S') = Deadline(g) - TimeOf(S')$$

$$Urgency(g, S') = \frac{Horizon(g, S')}{MAX(H)}$$

$$Utility(g, S') = Priority(g) * Urgency(g, S')$$

With key functions in place, we can compute the expected value of a future state **S’** using the formula below where **m** is the total number of states that must be considered, as the occurrence of S’ may be dependent on (m-1) intermediate states occurring, forming a sequential chain of predictions each having their own likelihood value. If S’ is not dependent on intermediate states, then **m** is 1.

$$Expected\ value(S'_m) =$$

$$\prod_{j=0}^{m} Likelihood(S'_j) * \sum_{i=0}^{n} Achieved(g_i, S') * Utility(g_i, S'_m)$$

In systems based on fine-grained architectures the decomposition of a top-level into several sub-goals may be expected. Increasing the number of active goals involved with regular operation of the system results in finer granularity of the evaluation process. For this reason, this evaluation method may be particularly useful for AGI architectures designed under a constructivist methodology (Thórisson 2012).

Resource availability can be expected to affect the number of predictions made by the AGI system at each point in time. Predictive functionality has strong links to learning, as learning can result from discovering solutions by way of generating predictions with desirable outcomes in terms of active goals. This indicates that during periods where the system lacks knowledge and actively seeks to learn, a greater share of resources should be devoted to the generation and evaluation of predictions than under normal circumstances; this causes the system to explore a greater number of future states. This represents a resource-bounded, interruptible and directed fashion of discovery and learning as part of the decision-making process.

To encapsulate these ideas, we propose the concept of *predictive heuristics* for the functionality just described; this concept represents an extended scope and altered functionality in contrast to traditional heuristics. To explicitly motivate this naming: *Predictive* refers to reliance on predictors to guide action selection and generation of resulting states as this has traditionally not been viewed as part of heuristic functionality since a separate expansion phase has been the norm. Compared to traditional methods, in our approach the heuristics for selective state generation are integrated at deeper levels of the search mechanism.

6 Discussion

Predictive heuristics represent one possible way to relate work in state-space search and decision theory to the AGI problem. At a minimum, the proposed ideas highlight problems faced by traditional search methods in real-world environments and provide a potential bridge from which techniques from traditional search and decision theory could possibly be brought to bear on AGI-level problems, although most probably in some slightly altered form.

With prediction-based generation of future states, the evaluation of possible future events is restricted to states that have a non-astronomical probability of occurring. Rather than working backwards from all possible future states - the number of which approaches infinity in real-world environments – it seems greatly more feasible to work forward from the current state to the states that are likely to follow while using the goals to guide the process (so a form of backward inference is absorbed into it); the resulting decrease in complexity of the decision problem can hardly be overstated as the number of states to be considered can drop by several orders of magnitude (or even from infinity to finite number). Furthermore, it is no longer a fixed number, but is adapted to the system's available resources. When the system is idle, it can afford the time to consider some unusual possibilities; when it is busy, it will focus on the most promising paths.

A variation of the functionality presented in the present paper has been successfully implemented in the AERA architecture (Nivel et al. 2012a & 2012b, Thórisson 2012b) with further publications to follow.

7 Acknowledgements

This work has been supported in part by the EU-funded project HUMANOBS: Humanoids That Learn Socio-Communicative Skills Through Observation, contract no. FP7-STREP-231453 (www.humanobs.org), and by grants from Rannis, Iceland. We thank Hilmar Finnsson for his thoughtful comments.

8 References

1. Burke, E. K., Kendall, G., Newall, J., Hart, E., Ross, P., Schulenburg, S. (2003). Hyper-heursitics: an emerging direction in modern search technology. In F. Glover and G. Kochenberger (eds.) Handbook of Metaheuristics, pages 457-474. Kluwer Academic Publishers.
2. Chaslot, G., Bakkes, S., Szita, I., & Spronck, P. (2008, October). Monte-carlo tree search: A new framework for game ai. In Proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference (pp. 216-217).
3. Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. Systems Science and Cybernetics, IEEE Transactions on, 4(2), 100-107.
4. Helgason, H. P., Nivel, E., & Thórisson, K. R. (2012, December). On attention mechanisms for AGI architectures: a design proposal. In Proceedings of the 5th international conference on Artificial General Intelligence (pp. 89-98). Springer-Verlag.
5. Korf, R. E. (1990). Real-time heuristic search. Artificial intelligence, 42(2), 189-211.
6. Nivel, E., et al. (2012a). HUMANOBS Architecture. Technical report available at http://wiki.humanobs.org/_media/public:humanobs-d9-r2-v2-architecturedesign.pdf
7. Nivel, E., Thurston, N., Bjornsson, Y. (2012b). Replicode Language Specification. Technical report available at http://wiki.humanobs.org/_media/public:publications:proj-docs:d3-1-specification-replicode.v1.0.pdf
8. Pearl, J. (1983). Heuristics: Intelligent Search Strategies for Computer Problem Solving. New York, Addison-Wesley.
9. Russell, S., Wefald, E. (1989). Principles of metareasoning. Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning, Toronto, 1989.
10. Thórisson, K. R. (2012a). A New Constructivist AI: From Manual Construction to Self-Constructive Systems. In P. Wang and B. Goertzel (eds.), Theoretical Foundations of Artificial General Intelligence. Atlantis Thinking Machines, 4:145-171.
11. Thórisson, K. R. (2012b). Final Project Evaluation Report for HUMANOBS. Technical report, available at http://wiki.humanobs.org/_media/public:d23_progress_report_v1.1.pdf
12. Thórisson, K. R., Helgason, H. P. (2012). Cognitive Architectures and Autonomy: A Comparative review. Journal of Artificial General Intelligence, vol. 3, pages 1-30.
13. Wang, P. (2006). Rigid Flexibility: The Logic of Intelligence. Volume 34 of Applied Logic series. New York: Springer.
14. Özcan, E., Bilgin, B., Korkmax, E. E. (2008). A comprehensive analysis of hyperheuristics. Intell. Data Anal., vol 12, no. 1, pages 3-23.