Second Annual Conference on Advances in Cognitive Systems Poster Collection (2013) 19-28

Formal Limits to Heuristics in Cognitive Systems

Tarek R. Besold TBESOLD@UOS.DE
Institute of Cognitive Science, University of Osnabrück, Osnabrück, Germany

Abstract

The recognition that human minds/brains are finite systems with limited resources for computation has led researchers in cognitive science to advance the Tractable Cognition thesis: Human cognitive capacities are constrained by computational tractability. As also human-level AI in its attempt to recreate intelligence and capacities inspired by the human mind is dealing with finite systems, transferring this thesis and adapting it accordingly may give rise to insights that can help in progressing towards meeting the final goals of AI. We therefore develop formal models usable for guiding the development of cognitive systems and models by applying notions from parameterized complexity theory and approximation theory to a general AI framework, putting special emphasis on connections and correspondences to the heuristics framework as recent development within cognitive science and cognitive psychology.

1. Introduction: The Importance of Formal Analysis for Cognitive Systems Research

After a certain abandonment of the original dream(s) of artificial intelligence (AI) towards the end of the last century, research in cognitive systems, artificial human-level intelligence, complex cognition and integrated intelligent systems over the last decade has witnessed a revival and now seems to enter its second spring with several specifically dedicated conference series, symposia, workshops, journals and a growing number of books and high-profile publicly visible and recognized research projects. Still, quite some fundamental questions remain to be answered before a unified approach to solving the big riddles underlying the (re)creation of human-level intelligence and cognition may arise. Currently, there are many different paradigms competing with each other: Symbolism versus connectionism, high-level modeling of specific cognitive capacities versus low-level models with emergent behavior, holistic versus modular approaches.

Each of these paradigms brings along its own terminology, conceptual perspective, and engineering methods, resulting in a wide variety of approaches to solving the intelligence puzzle. This, in turn, makes it hard to establish standards and insights in cognitive models and cognitive systems valid on a general level independent of the chosen perspective and methodology. Still, there are a few common elements to most (if not all) of the mentioned approaches (in that, for instance, they are applied in attempts to model one or several human cognitive capacities), making the wish for general principles and results more urgent. Here, formal methods and analysis can provide a solution: Due to their general nature they can often be applied without prior commitment to a particular formalism or architecture, allowing to establish high-level insights and generally applicable findings — and thus can also provide guidelines and hints at how to unify approaches and progress towards the overall goals of the respective research programs.

In what follows, we give an example for the type of analysis we are considering, by this providing both, evidence supporting the just made claim and concrete insights concerning heuristics and their use in cognitive systems as important topic. Sect. 2 introduces the mindset underlying our work before Sect. 3 reproduces important results from previous work. Sect. 4 constitutes the main body of our contribution, connecting the notion of cognitive heuristics (and their models) to recent results from parameterized complexity and approximation theory. Sect. 6 concludes the paper.

2. Complexity and Cognition

Two famous ideas conceptually lie at the heart of many endeavors in computational cognitive modeling, cognitive systems research and artificial intelligence: The "computer metaphor" of the mind, i.e. the concept of a computational theory of mind as described in (Pylyshyn, 1980), and the Church-Turing thesis (Turing, 1969). The former bridges the gap between humans and computers by advocating the claim that the human mind and brain can be seen as an information processing system and that reasoning and thinking corresponds to processes that meet the technical definition of computation as formal symbol manipulation, the latter gives an account of the nature and limitations of the computational power of such a system: Every function for which there is an algorithm (i.e., which is intuitively computable) is computable by a Turing machine — and functions that are not computable by such a machine are to be considered not computable in principle by any machine.

But computer metaphor and Church-Turing thesis also had significant impact on cognitive science and cognitive psychology. As stated in (Cummins, 2000), one of the primary aims of cognitive psychology is to explain human cognitive capacities which are often modeled in terms of computational-level theories of cognitive processes (i.e., as precise characterizations of the hypothesized inputs and outputs of the respective capacities together with the functional mappings between them; c.f. (Marr, 1982) for details). Unfortunately, computational-level theories are often underconstrained by the available empirical data, allowing for several different input-output mappings and corresponding theories. A first attempt at mitigating this problem can now be based on the aforegiven Church-Turing thesis: If the thesis were true, the set of functions computable by a cognitive system would be a subset of the Turing-computable functions. Now, if a computational-level theory would assume that the cognitive system under study computes an uncomputable function, the model could already be rejected for theoretical reasons.

Still, directly applying the notion of Turing computability as equivalent to the power of cognitive computation has to be considered overly simplistic. Whilst it seems plausible that the thesis holds in general, its practical relevance in cognitive agents and systems is at least questionable. As already recognized in (Simon, 1957), actual cognitive systems (due to their nature as physical systems) need to perform their tasks in limited time and with a limited amount of space at their disposal. Therefore, the Church-Turing thesis (which does not take into account any of these dimensions) by itself is not strict enough for being used as a constraint on cognitive theories. To mitigate this problem, different researchers over the last decades have proposed the use of mathematical complexity theory, and namely the concept of NP-completeness, as an assisting tool (see, e.g., (Levesque, 1988; Frixione, 2001)), bringing forth the so called "*P-Cognition thesis*": Human cognitive capacities are hypothesized to be of the polynomial-time computable type.

However, using the "polynomial-time computable" as synonymous with "efficient" may already be overly restrictive: In fact, it is often the case that we as humans are able to solve problems which may be hard in general but suddenly become feasible if certain parameters of the problem are restricted. This idea has been formalized in the field of *parameterized complexity theory*, in which "tractability" is captured by the class of fixed-parameter tractable problems FPT (for an introduction to parameterized complexity theory see, e.g., (Flum & Grohe, 2006; Downey & Fellows, 1999)):

Definition 1. FPT
A problem P is in FPT if P admits an $O(f(\quad)n^c)$ algorithm, where n is the input size, is a parameter of the input constrained to be "small", c is an independent constant, and f is some computable function.

Originating from this line of thought, (van Rooij, 2008) introduces a specific version of the claim that cognition and cognitive capacities are constrained by the fact that humans basically are finite systems with only limited resources for computation: Applying the just presented techniques from parameterized complexity theory, this basic notion of resource-bounded computation for cognition is formalized in terms of the so called "FPT-*Cognition thesis*", demanding for human cognitive capacities to be fixed-parameter tractable for one or more input parameters that are small in practice (i.e., stating that the computational-level theories have to be in FPT).

3. Previous Work: The Tractable AGI Thesis

But whilst the aforementioned P-Cognition thesis also found its way into AI (cf., e.g., (Cooper, 1990; Nebel, 1996)), the FPT-Cognition thesis this far has widely been ignored. Recognizing this as a serious deficit, for example in (Robere & Besold, 2012) and (Besold & Robere, 2013a), we proposed a way of (re)introducing the idea of tractable computability for cognition into AI and cognitive systems research by rephrasing and accordingly adapting the FPT-form of the Tractable Cognition thesis. As all of the currently available computing systems used for implementing cognitive models and cognitive systems are ultimately finite systems with limited resources (and thus in this respect are not different from other cognitive agents and human minds and/or brains), in close analogy we developed the "Tractable AGI thesis" (Tractable Artificial and General Intelligence thesis).

Tractable AGI thesis
Models of cognitive capacities in artificial intelligence and computational cognitive systems have to be fixed-parameter tractable for one or more input parameters that are small in practice (i.e., have to be in FPT).

Concerning the interpretation of this thesis, suppose a cognitive modeler or AI system designer is able to prove that his model at hand is — although in its most general form possibly NP-hard — fixed-parameter tractable for some set of parameters . This implies that if the parameters in are fixed small constants for problem instances realized in practice, then it is possible to efficiently compute a solution. However, there is also a non-trivial corollary of this: any instance of the problem can be reduced to a *problem kernel*.

Definition 2. Kernelization
Let P be a parameterized problem. A kernelization of P is an algorithm which takes an instance x of P with parameter and maps it in polynomial time to an instance y such that $x \in P$ if and only if $y \in P$, and the size of y is bounded by $f(\quad)$ (f a computable function).

Theorem 1. Kernelizability (Downey, Fellows, & Stege, 1997)
A problem P is in FPT if and only if it is kernelizable.

This theorem on the one hand entails that any positive FPT result obtainable for the model in question essentially implies that there is a "downward reduction" for the underlying problem to some sort of smaller or less-complex instance of the same problem, which can then be solved — whilst on the other hand (assuming $\mathsf{W}[1]^1 \neq \mathsf{FPT}$) any negative result implies that there is no such downward reduction. As shown in the following section, this equivalence forms a connecting point to recent, much-noticed developments in cognitive science and cognitive psychology — and can also have important ramifications for the modeling of many cognitive capacities in computational cognitive systems.

4. Setting Limits to Heuristics in Cognitive Systems

A still growing number of researchers in cognitive science and cognitive psychology, already starting out in the 1970s with the "heuristics and biases" program (Kahneman, Slovic, & Tversky, 1982), and today prominently heralded, for instance, in the work of Gigerenzer and colleagues (Gigerenzer, Hertwig, & Pachur, 2011), argues that humans in their common sense reasoning do not apply any full-fledged form of logical or probabilistic reasoning to possibly highly complex problems, but instead rely on (mostly automatic and unconscious) mechanisms that allow them to circumvent the impending complexity explosion and nonetheless reach acceptable solutions to the original problems. Amongst the plethora of proclaimed automatisms are, for example, the representativeness heuristic (Kahneman, Slovic, & Tversky, 1982) or the take-the-best heuristic (Czerlinski, Goldstein, & Gigerenzer, 1999).

All of these mechanisms are commonly subsumed under the all-encompassing general term "heuristics". Still, on theoretical grounds, at least two quite different general types of approach have to be distinguished within this category: Either the complexity of solving a problem can be reduced by reducing the problem instance under consideration to a simpler (but solution equivalent) one, or the problem instance stays untouched but — instead of being perfectly (i.e., precisely) solved — is dealt with in a good enough (i.e., approximate) way.

Now, taking the perspective of an architect of a cognitive system considering to include human-inspired heuristics in his reasoning model for solving certain tasks, a crucial question quite straightforwardly arises: Which problems can actually be solved applying heuristics — and how can the notion of heuristics theoretically be modeled on a sufficiently high level as to allow for a general description? Having a look at recent work in parameterized complexity theory and in approximation theory, we find that the two distinct types of heuristics naturally correspond to two well-known concepts from the respective fields. As shown in the remainder of this section, this opens the way for establishing a solid theoretical basis for models of heuristics in cognitive systems.

4.1 The Reduction Perspective

In Sect. 3, with Theorem 1 and its interpretation, we introduced the notion of kernelizability of a problem. On the one hand, from a constructive perspective, by actively considering the kernelizability of problems (or rather problem classes), we can provide inspiration and first hints at hypothesizing a specialized cognitive structure capable of computing the reduced instance of a problem, which then might allow for an efficient solving procedure. On the other hand, taking a more theoretical stance, by categorizing problems according to kernelizability (or their lack thereof) we also can

1. $\mathsf{W}[1]$ is the class of problems solvable by constant depth combinatorial circuits with at most 1 gate with unbounded fan-in on any path from an input gate to an output gate. In parameterized complexity, the assumption $\mathsf{W}[1] \neq \mathsf{FPT}$ can be seen as analogous to $\mathsf{P} \neq \mathsf{NP}$.

establish a distinction between problem classes which are solvable by reduction-based heuristics and those which are not — and can thus already a priori decide whether a system implementing a reduction-based heuristics might generally be unable to solve a certain problem class.

Clearly, from a theoretical perspective the strict equivalence between FPT-membership and kernelizability of a problem does not come by surprise. But also on practical and applied grounds, the correspondence should seem natural and, moreover, should fairly directly explicate the connection to the notion of reduction-based heuristics: If cognitive heuristics are as fast and frugal as commonly claimed, considering them anything but (at worst) polynomial-time bounded processes seems questionable. But now, if the reduced problem shall be solvable under resource-critical conditions, using the line of argument from Sect. 2 and 3, we can just hope for it to be in FPT. Now, combining the FPT-membership of the reduced problem with the polynomial-time complexity of the heuristics, already the original problem had to be fixed-parameter tractable. Still, reduction-based heuristics are not trivialized by this: Although original and reduced problem are in FPT, the respective size of the parameters may still differ between instances (which possibly can make an important difference in application scenarios for implemented cognitive systems).

4.2 The Approximation Perspective

But there also is a complementary perspective offering an alternate possibility of (re)interpreting heuristics, namely approximation theory: Instead of precisely solving a kernel as proposed by reduction-based heuristics, compute an approximate solution to the original problem (i.e., the solution to a relaxed problem). The idea is not any more to perfectly solve the problem (or an equivalent instance of the same class), but to solve the problem to a satisfactory degree.

Here, a candidate lending itself for being considered a standard analogous to FPT in the Tractable AGI thesis is APX, the class of problems allowing polynomial-time approximation algorithms:

Definition 3. APX
An optimization problem P is in APX if P admits a constant-factor approximation algorithm, i.e., there is a constant factor $\quad > 0$ and an algorithm which takes an instance of P of size n and, in time polynomial in n, produces a solution that is within a factor $1 + \quad$ of being optimal (or $1 - \quad$ for maximization problems).

Clearly, here the meaningfulness and usefulness of the theoretical notion in practice crucially depends on the choice of the bounding constant for the approximation ratio: If the former is meaningfully chosen with respect to the problem at hand, constant-factor approximation allows for quantifying the "good enough" aspect of the problem solution and, thus, offers a straightforward way of modeling the notion of "satisficing" (Simon, 1956) (which in turn is central to many heuristics considered in cognitive science and psychology).

As in the case of the reduction-based heuristics, one of the main advantages of formally identifying approximation-based heuristics with APX lies in its limiting power: If a problem shall be solved at least within a certain range from the optimal solution, but it turns out that the problem does not admit constant-factor approximation for the corresponding approximation parameter, the problem can a priori be discarded as unsolvable with approximation-based heuristics (unless one wants to also admit exponential-time mechanisms, which might be useful in selected cases but for simple complexity considerations does not seem to be feasible as a general approach).[2]

2. On the other hand, considering more restrictive notions than APX as, for instance, PTAS (the class of problems for which there exists a polynomial-time approximation scheme, i.e., an algorithm which takes an instance of a optimization problem and a parameter $\quad > 0$ and, in polynomial time, solves the problem within a factor $1 + \quad$ of

4.3 Joining Perspectives

Having introduced the distinction between reduction-based and approximation-based heuristics, together with proposals for a formal model of the mechanisms behind the respective class, we now want to return to a more high-level view and look at heuristics in their entirety. This is also meaningful from the perspective of the initially mentioned system architect: Instead of deciding whether he wants to solve a certain type of task applying one of the two types of heuristics and then conducting the corresponding analysis, he might just want to directly check whether the problem at hand might be solvable by any of the two paradigms. Luckily, a fairly recently introduced theoretical concept allows for the integration of the two different views — FPT and APX can both be combined via the concept of fixed-parameter approximabiltiy and the corresponding problem class FPA:

Definition 4. FPA
The fixed-parameter version P of a minimization problem is in FPA if — for a recursive function f, a constant k, and some fixed recursive function g — there exists an algorithm such that for any given problem instance I with parameter k, and question $OPT(I) \leq k$, the algorithm which runs in $O(f(k)n^c)$ (where $n = |I|$) either outputs "no" or produces a solution of cost at most $g(k)$.

As shown in (Cai & Huang, 2006), both polynomial-time approximability and fixed-parameter tractability with witness (Cai & Chen, 1997) independently imply the more general fixed-parameter approximability. And also on interpretation level FPA artlessly combines both views of heuristics, at a time in its approximability character accommodating for the notion of satisficing and in its fixed-parameter character accounting for the possibility of complexity reduction by kernelizing whilst keeping key parameters of the problem fixed.

Clearly, the notion of fixed-parameter approximability is significantly weaker than either FPT and kernelization, or APX. Nonetheless, its two main advantages are the all-encompassing generality (independent of the type of heuristics) and yet again the introduction of a categorization over problem types: If problems of a certain kind are not in FPA, this also excludes membership in any of the two stricter classes — and thus (in accordance with the lines of argument given above) in consequence hinders solvability by either type of heuristics.

Recalling the Tractable AGI thesis introduced in Sect. 3, we can use the just outlined conception of FPA for not only considering classical strict processing and reasoning in cognitive systems, but for also accounting for models of cognitive heuristics. This allows us to adapt the original thesis into a (significantly weaker but possibly more "cognitively adequate") second form:

Fixed-Parameter Approximable AGI thesis
Models of cognitive capacities in artificial intelligence and computational cognitive systems have to be fixed-parameter approximable for one or more input parameters that are small in practice (i.e., have to be in FPA).

Whilst the original Tractable AGI thesis was aimed at AI in general (thus also including forms of high-level AI which may not be human-inspired, or which in their results shall not appear human-like), the just postulated thesis, due to its strong rooting in the (re)implementation of human-style processing, explicitly targets researchers in cognitive systems and cognitive AI.

---

the optimal solution) does not seem meaningful to us either, as also human satisficing does not approximate optimal solutions up to an arbitrary degree but in experiments normally yields rather clearly defined cut-off points at a certain approximation level.

5. The Importance of Formal Analysis for Cognitive Systems Research Revisited

An often heard fundamental criticism of trying to apply methods from complexity theory and formal computational analysis to cognitive systems and cognitive models are variations of the claim that there is no reason to characterize human behavior in terms of some "problem" (i.e., in terms of a well-defined class of computational tasks), from this drawing the conclusion that computational complexity is just irrelevant for the respective topics and fields.[3] In most cases, this judgement seems to be based on either a misconception of what a computational-level theory in the sense of (Marr, 1982) is (which we will refer to as "*description error*"), or a misunderstanding of what kind of claim complexity theory makes on this type of theory (in the following referred to as "*interpretation error*").

The *description error* basically questions the possibility of describing human behavior in terms of classes of computational tasks. The corresponding argument is mostly based on the perceived enormous differences between distinct manifestations of one and the same cognitive capacity already within a single subject (at the moment for descriptive simplicity's sake — without loss of generality — leaving aside the seemingly even more hopeless case of several subjects). Still, precisely here Marr's Tri-Level Hypothesis (i.e., the idea that information processing systems should be analyzed and understood at three different — though interlinked — levels) comes into play. Marr proposed three levels of description for a (biological) cognitive system, namely a computational, an algorithmic, and an implementation level. Whilst the latter two are concerned with how a system does what it does from a procedural and representational perspective (algorithmic level), and how a system is physically realized (implementation level), the computational level takes the most abstract point of view in asking for a description of what the system does in terms of giving a function mapping certain inputs on corresponding outputs. So what is needed to specify a computational-level theory of a cognitive capacity[4] is just a specified set of inputs, a set of corresponding outputs (each output corresponding to at least one element from the set of inputs) and a function establishing the connection between both (i.e., mapping each input onto an output). But now, due to the high degree of abstraction of the descriptive level, this allows us to characterize human cognitive capacities in general in terms of a computational-level theory by specifying the aforementioned three elements — where inputs and outputs are normally provided (and thus defined) by generalization from the real world environment, and the function has to be hypothesized by the respective researcher.[5] Once all three parts have been defined, formal computational analyses can directly be conducted on the obtained computational-level theory as the latter happens to coincide in form with the type of prob-

3. For reasons unclear to the author this perspective seems to be more widespread and far deeper rooted in AI and cognitive systems research than in (theoretical) cognitive science and cognitive modeling where complexity analysis and formal computational analysis in general by now have gained a solid foothold.
4. Here we presuppose that cognitive capacities can be seen as information processing systems. Still, this seems to be a fairly unproblematic claim, as it simply aligns cognitive processes with computations processing incoming information (e.g., from sensory input) and resulting in a certain output (e.g., a certain behavioral or mental reaction) dependent on the input.
5. Fortunately, this way of conceptualizing a cognitive capacity naturally links to research in artificial cognitive systems. When trying to build a system modeling one or several selected cognitive capacities, we consider a general set of inputs (namely all scenarios in which a manifestation of the cognitive capacity can occur) which we necessarily formally characterize — although maybe only implicitly — in order to make the input parsable for the system, hypothesize a function mapping inputs onto outputs (namely the computations we have the system apply to the inputs) and finally obtain a well-characterized set of outputs (namely all the outputs our system can produce given its programming and the set of inputs).

lem (i.e., formal definition of a class of computational tasks) studied in computational complexity and approximation theory. And also the existence of at least one computational-level theory for each cognitive capacity is guaranteed: Simply take the sets of possible inputs and corresponding outputs, and define the mapping function element-wise on pairs of elements from the input and output, basically creating a lookup table returning for each possible input the respective output.

Leaving the description error behind us, we want to have a look at the *interpretation error* as further common misconception. Even when modifying the initial criticism by not questioning the overall possibility of characterizing human behavior in terms of classes of computational tasks, but rather by stating that even if there were these classes, it would not have to be the case that humans have to be able to solve all instances of a problem within a particular class, we believe the argument grasps at nothing. First and foremost, as further elaborated upon in the initial paragraph of the following section, we propose to use complexity and approximability results rather as a safeguard and guideline than as an absolute exclusion criterion: As long as a computational-level theory underlying a computational cognitive model is in FPT, APX, or FPA — where in each case the system architect has to decide which standard(s) to use — the modeler can be sure that his model will do well in terms of performance for whatever instance of the problem it will encounter. Furthermore, it is clear that in cognitive systems and cognitive models in general a worst-case complexity or approximability analysis for a certain problem class only rarely (if at all) can be taken as a disqualifier for the corresponding computational-level theory.[6] It might well be the case that the majority of problem instances within the respective class is found to be well behaving and easily solvable, whilst the number of worst-case instances is very limited (and thus possibly unlikely to be encountered on a basis frequent enough as to turn their occurrence into a problem). Still, this does not bar the way for a meaningful application of formal analysis techniques in general, but might just require a different selection of tools: Where worst-case analyses may on certain grounds be questionable as decisive criterion about the overall usefulness of a particular computational-level theory for a cognitive capacity, average-case analyses (which admittedly are significantly harder to perform) can change the picture dramatically.

6. Conclusion: Limiting the Limits

A second frequent criticism (besides the general objection discussed in the previous section) against the type of work presented in this paper is that demanding for cognitive systems and models to work within certain complexity limits might always be overly restrictive: Maybe each and every human mental activity actually is performed as an exponential-time procedure, but this is never noticed as the exponent for some reason always stays very small. Undoubtedly, using what we just presented, we cannot exclude this possibility — but this also is not our current aim. What we want to say is different: We do not claim that cognitive processes are without exception within FPT, APX, or FPA, but we maintain that as long as cognitive systems and models stay within these boundaries they can safely be assumed to be plausible candidates for application in a resource-bounded general-purpose cognitive agent (guaranteeing a high degree of generalizability, scalability, and reliability). Thus, if a system architect has good reasons for plausibly assuming that a particular

6. Of course this also explicitly includes the case in which the considered classes are conceptually not restricted to the rather coarse-grained hierarchy used in "traditional" complexity theory, but if also the significantly finer and more subtle possibilities of class definition and differentiation introduced by parametrized complexity theory and other recent developments are taken into account.

type of problem in all relevant cases only appears with a small exponent for a certain exponential-time solving algorithm, it may be reasonable to just use this particular algorithm in the system. But if the architect should wonder whether a problem class is likely to be solvable by a resource-bounded human-style system in general, or if it should better be addressed using reduction-based or approximation-based heuristics, then we highly recommend to consider the lines of argument presented in the previous sections.

Concerning related work, besides the conceptually and methodologically closely related, but in its focus different efforts by van Rooij (van Rooij, 2008) and colleagues in theoretical cognitive science (see, e.g., (Kwisthout & van Rooij, 2012; Blokpoel et al., 2011)), of course there also is work relating fixed-parameter complexity to AI. Still, except for very few examples as, e.g., (Wareham et al., 2011), the applications mostly are limited to more technical or theoretical subfields of artificial intelligence (see, e.g., (Gottlob & Szeider, 2008) for a partial survey) and — to the best of our knowledge — this far have not been converted into a more general guiding programmatic framework for research into human-level AI and cognitive systems.

We therefore in our future work hope to develop the overall framework further, also showing the usefulness and applicability of the proposed methods in different worked examples from several relevant fields: The range of eligible application scenarios spans from models of epistemic reasoning and interaction, over cognitive systems in general problem-solving scenarios, down to models for particular cognitive capacities as, for example, analogy-making (see, e.g., (Robere & Besold, 2012; Besold & Robere, 2013b) for an initial proof-of-concept analysis).

Acknowledgements

I would like to express my sincere gratitude to Robert Robere (University of Toronto) for introducing me to the field of parameterized complexity theory, reliably providing me with theoretical/technical backup and serving as a willing partner for feedback and discussion.

Also I want to thank the anonymous reviewers for their insightful comments and recommendations which helped to develop the present paper into its current state.

References

Besold, T. R., & Robere, R. (2013a). A Note on Tractability and Artificial Intelligence. *Artificial General Intelligence - 6th International Conference, AGI 2013, Proceedings* (pp. 170–173). Springer.

Besold, T. R., & Robere, R. (2013b). When Almost Is Not Even Close: Remarks on the Approximability of HDTP. *Artificial General Intelligence - 6th International Conference, AGI 2013, Proceedings* (pp. 11–20). Springer.

Blokpoel, M., Kwisthout, J., Wareham, T., Haselager, P., Toni, I., & van Rooij, I. (2011). The computational costs of recipient design and intention recognition in communication. *Proceedings of the 33rd Annual Meeting of the Cognitive Science Society* (pp. 465–470).

Cai, L., & Chen, J. (1997). On fixed-parameter tractability and approximability of {NP} optimization problems. *Journal of Computer and System Sciences*, *54*, 465–474.

Cai, L., & Huang, X. (2006). Fixed-parameter approximation: Conceptual framework and approximability results. In H. Bodlaender & M. Langston (Eds.), *Parameterized and exact computation*, Vol. 4169 of *Lecture Notes in Computer Science*, 96–108. Springer Berlin Heidelberg.

Cooper, G. (1990). The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, *42*, 393–405.

Cummins, R. (2000). "How does it work?" vs. "What are the laws?" Two conceptions of psychological explanation. *Explanation and cognition* (pp. 117–145). MIT Press.

Czerlinski, J., Goldstein, D., & Gigerenzer, G. (1999). How good are simple heuristics? *Simple Heuristics That Make Us Smart*. Oxford University Press.

Downey, R. G., & Fellows, M. R. (1999). *Parameterized complexity*. Springer.

Downey, R. G., Fellows, M. R., & Stege, U. (1997). Parameterized complexity: A framework for systematically confronting computational intractability. In *Contemporary Trends in Discrete Mathematics: From DIMACS and DIMATIA to the Future*. AMS.

Flum, J., & Grohe, M. (2006). *Parameterized complexity theory*. Springer.

Frixione, M. (2001). Tractable competence. *Minds and Machines*, *11*, 379–397.

Gigerenzer, G., Hertwig, R., & Pachur, T. (Eds.). (2011). *Heuristics: The Foundation of Adaptive Behavior*. Oxford University Press.

Gottlob, G., & Szeider, S. (2008). Fixed-parameter algorithms for artificial intelligence, constraint satisfaction and database problems. *Comput. J.*, *51*, 303–325.

Kahneman, D., Slovic, P., & Tversky, A. (1982). *Judgment under Uncertainty: Heuristics and Biases*. Cambridge University Press.

Kwisthout, J., & van Rooij, I. (2012). Bridging the gap between theory and practice of approximate bayesian inference. *Proceedings of the 11th International Conference on Cognitive Modeling* (pp. 199–204).

Levesque, H. (1988). Logic and the complexity of reasoning. *Journal of Philosophical Logic*, *17*, 355–389.

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing visual information*. Freeman.

Nebel, B. (1996). Artificial intelligence: A computational perspective. In G. Brewka (Ed.), *Principles of knowledge representation*, 237–266. CSLI Publications.

Pylyshyn, Z. (1980). Computation and cognition: Issues in the foundation of cognitive science. *The Behavioral and Brain Sciences*, *3*, 111–132.

Robere, R., & Besold, T. R. (2012). Complex Analogies: Remarks on the Complexity of HDTP. *Twentyfifth Australasian Joint Conference on Artificial Intelligence* (pp. 530–542). Springer.

Simon, H. A. (1956). Rational choice and the structure of the environment. *Psychological Review*, *63*, 129–138.

Simon, H. A. (1957). *Models of man: Social and rational*. New York: John Wiley & Sons, Ltd.

Turing, A. (1969). Intelligent machinery. *Machine Intelligence* (pp. 3–23). Edinburgh University Press.

van Rooij, I. (2008). The tractable cognition thesis. *Cognitive Science*, *32*, 939–984.

Wareham, T., Kwisthout, J., Haselager, P., & van Rooij, I. (2011). Ignorance is bliss: A complexity perspective on adapting reactive architectures. *Proceedings of the First IEEE Conference on Development and Learning and on Epigenetic Robotics* (pp. 465–470).