



UvA-DARE (Digital Academic Repository)

Tractable embodied computation needs embeddedness

van Rooij, I.; Blokpoel, M.; de Haan, R.; Wareham, T.

DOI

[10.12832/94728](https://doi.org/10.12832/94728)

[10.12832/94728](https://doi.org/10.12832/94728)

Publication date

2019

Document Version

Submitted manuscript

Published in

Reti, Saperi, Linguaggi

[Link to publication](#)

Citation for published version (APA):

van Rooij, I., Blokpoel, M., de Haan, R., & Wareham, T. (2019). Tractable embodied computation needs embeddedness. *Reti, Saperi, Linguaggi*, 6(1), 25-38.

<https://doi.org/10.12832/94728>, <https://doi.org/10.12832/94728>

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (<https://dare.uva.nl>)

Tractable embodied computation needs embeddedness

Iris van Rooij

Radboud University, Donders Institute for Brain,
Cognition and Behaviour, The Netherlands

Mark Blokpoel

Radboud University, Donders Institute for Brain,
Cognition and Behaviour, The Netherlands

Ronald de Haan

Institute for Logic, Language and Information,
University of Amsterdam, The Netherlands

Todd Wareham

Department of Computer Science,
Memorial University of Newfoundland, Canada

Abstract

It seems that a move from discrete to analog computation demands new types of complexity theory to demarcate the tractability border for embodied computation. We argue, however, that application of complexity theory need not hinge on the analog versus discrete distinction as much as one may think. Instead, we will show that *embeddedness*, one of the core commitments of enactivism, proves more critical for understanding tractable embodied computation.

Keywords: Enactivism, Embodied Cognition, Computation, Tractability

1 Enactivist approaches to cognition have long had a difficult relationship with computation-
2 alism. The relationship seems to be improving as more and more theorists spell out ways in
3 which the embodied, embedded, enactive, and extended (4E) nature of cognition on the one
4 hand, and computationalism on the other, are not incompatible (Milkowski, 2018; van Rooij,
5 2012; Villalobos & Dewhurst, 2017, 2018; c.f. Abramova, 2019). Isaac (n.d.) contributes to
6 this development, as he proposes analog computation such as embodied in fluid logics as a
7 framework for thinking about enactivist computational explanations. In this commentary,
8 we zoom in on a question raised by Isaac: What, if any, are the implications of this proposal
9 for understanding the conditions that make enactive cognition tractable?

10 Isaac sees two possibilities. The first possibility is that tasks that are intractable
11 for digital computation are also intractable for analog computation, and vice versa. In
12 other words, in this case the tractable-intractable border is insensitive to whether or not
13 the computation is achieved by a discrete or analog computation and, consequently, the

14 same tractability constraints that have been argued to apply to cognitivist accounts (see,
 15 e.g., Frixione, 2001; van Rooij, 2008) would also naturally apply to embodied accounts of
 16 cognition (see, e.g., van Rooij, 2012). The second possibility is that there exist computational
 17 transformations that can be tractably computed by analog computation but not by classical
 18 (Turing machine) computation. In that case, not only would assessing the tractability of
 19 embodied computation require its own, non-classical form of complexity analysis, but it
 20 may also speak in favour of embodied accounts whenever cognitivist accounts run into
 21 intractability but the embodied (analog computation) account does not.

22 In this commentary, we argue that regardless of which of these two possibilities holds,
 23 enactivists can benefit greatly from adopting the tractability constraint, as it will provide
 24 a means to explicate the important role that one of its E's plays in understanding the
 25 tractability of cognition in the world. The reason is, as we explain, that tractability cannot
 26 be achieved without proper embeddedness. Moreover, this is true, regardless of whether
 27 the computation performed is discrete or analog, and whether or not those are ultimately
 28 equivalent as far as tractability is concerned.

29 We will first briefly explain the formal notions from computational complexity theory
 30 and intractability on which our argumentation builds. We next consider the conditions
 31 under which the computations performed by an enactive, embodied agent can be tractable.
 32 We will show that this requires either (1) that the environment is structured such that the
 33 situations that arise for the agent, and that it can sense, are a proper subset of all logically
 34 possible situations; or (2) that the agent is made tractable by design in which case its fit
 35 with the environment must have been determined by 'design' (e.g., evolution).

36 1 Computational Complexity & Intractability

37 Computational complexity theory provides mathematical tools and concepts for assessing
 38 the resources (e.g., time, space, randomness) consumed during computation. We focus here
 39 on the resource time, as it is known to be one of the most sensitive resources (e.g., more so
 40 than space—see, e.g., Arora and Barak, 2009). One important distinction in computational
 41 complexity theory relevant for our purposes is the notion of 'intractability'. Two distinct
 42 formalisations of this notion have been put forth, one based on classical complexity theory
 43 (see, e.g., Arora & Barak, 2009; Garey & Johnson, 1979) and one based on parameterized
 44 complexity theory (see, e.g., Downey & Fellows, 2013; Flum & Grohe, 2006).¹ We next
 45 consider each in turn.

46 1.1 Classical Complexity Theory

47 According to classical complexity theory (Arora & Barak, 2009; Garey & Johnson, 1979), a
 48 computation is intractable if the time that it consumes grows faster than any polynomial
 49 function (i.e., faster than n^c , where c is some constant and n is some measure of the problem
 50 size, e.g., number of variables or states). Such computations are called non-polynomial time
 51 computations. An example is a computation that takes exponential-time, on the order of c^n
 52 for some constant $c > 1$. Even for intermediate problem sizes n , exponential time becomes
 53 prohibitively large (e.g., for $c = 2$ and $n = 60$ the number c^n is more than the number of

¹For an accessible treatment of both types of complexity theory applied in cognitive science, we refer to (van Rooij, Blokpoel, Kwisthout, & Wareham, 2019)

seconds that have passed since the dinosaurs went extinct). The embodied computations of enactive agents surely should operate on a time scale relevant for survival. In practice, this will be on the order of seconds, minutes, and in rare occasions hours. An organism that instead takes, say, millions of years to come to a decision about whether or not to eat a given food option will be dead before it can take the first bite.

An example of a classically intractable computation is 3-SAT.² In 3-SAT, one is given a logical formula in conjunctive normal form: i.e., the formula is a conjunction of a set of clauses of disjunctions of at most 3 variables, e.g., $(v_1 \vee v_2 \vee \neg v_3) \wedge (v_5 \vee \neg v_1 \vee v_3) \wedge (v_6 \vee \neg v_4)$. The result of the computation is a truth assignment to the variables that makes the whole formula true (e.g., the assignment $v_2, v_4, v_6 = \text{true}$ and $v_1, v_3, v_5 = \text{false}$ would make the above formula true). There exists no polynomial time procedure for computing 3-SAT. However, the problem 2-SAT, in which each clause is a disjunction of at most 2 variables, is tractable, i.e., it can be computed in polynomial time.

1.2 Parameterized Complexity Theory

According to parameterized complexity theory (Cygan et al., 2015; Downey & Fellows, 2013; Flum & Grohe, 2006; Niedermeier, 2006), computations are tractable when they are so-called fixed-parameter tractable for *small* parameters. A computation is said to be *fixed-parameter tractable* when the time it consumes can be upper bounded by some function $f(k)n^c$, where f can be an arbitrary function of the parameter k . Since the n^c -part is polynomial-time, fixed-parameter tractable computations are tractable even for large n provided only that k is relatively small. Note that classically intractable computations can be fixed-parameter tractable. For instance, we saw that 3-SAT is classically intractable. Yet, 3-SAT is fixed-parameter tractable when parameterized by the number of variables that appear negated in the formula.³

1.3 Invariance thesis

Even though both classical and parameterized complexity theory were originally formulated for Turing machines, according to the Invariance thesis the distinction between polynomial versus non-polynomial (for classical complexity) and between fixed-parameter tractable and fixed-parameter intractable (for parameterized complexity) is insensitive to the exact machine model used.⁴ As Isaac notes, it is not known if any future analog computer model may show that the Invariance thesis is false. Yet, at present there is no reason to believe that it is. For instance, Siegelmann and Sontag (1994) have shown that analog neural networks, even with infinite precision real valued connection weights, cannot perform intractable (NP-hard) computations in polynomial time. Similarly, Aaronson (2005) discusses various forms of physical computation, ranging from protein folding to quantum computing, and

²Formally, 3-SAT is known to be NP-hard (Garey & Johnson, 1979, Problem L02). This means that it is not polynomial-time computable, unless $P = NP$. It is famously conjectured, however, that $P \neq NP$ (Fortnow, 2009; Gasarch, 2002).

³This is a folklore result, that can be proved by combining a simple pre-processing rule with results mentioned by Niedermeier (2006, Chapter 1).

⁴As Parberry (1986) wrote, the Invariance (or Extended Church-Turing) Thesis “states [...] that time on all ‘reasonable’ machine models is related by a polynomial.”

89 concludes that the bounds of tractable computation seems to be a universal limit and an
 90 inherent property of physical computation generally, not just classical (Turing) machines.

91 Does this mean that the bounds of intractability are irrelevant for enactivist versus
 92 cognitivist debates about the nature of cognition? We think not. But we think that more
 93 relevant for understanding how enactivist approaches may be one step ahead of cognitivist
 94 approaches is that they view embeddedness of cognitive agents in their life world as a
 95 fundamental part of understanding cognitive behaviour. As we will argue, it is also vital for
 96 understanding the tractability of cognition.

97 2 An Enactive Agent Schema

98 Let's consider an embodied agent $A = (S, E, M)$ with a set of sensors $S = \{s_1, \dots, s_k\}$, a set
 99 of effectors $E = \{e_1, \dots, e_m\}$, and an embodied control mechanism M . The agent is embedded
 100 in an environment, its *life world* \mathcal{E} (see Figure 1; cf., Di Paolo, Buhrmann, and Barandiaran
 101 (2017)). In general, the agent's sensors and effectors may take on continuous values in a
 102 certain range (e.g., $[-1, +1]$) or binary values "on" or "off", or some combination of these.
 103 Even for medium sized sensor and effector sets, the sensory and action repertoire of our
 104 agent can already be quite substantial. Consider for instance, that for the agent which has
 105 six sensors with an accuracy such that each sensor can distinguish about 10 different states
 106 in the $[-1, +1]$ range (e.g., light intensity, colours, temperature, pitch), there are already a
 107 million sensory states that this agent can distinguish (not necessarily phenomenologically,
 108 but in principle effectively). Similarly, if the agent has eight effectors, each with four degrees
 109 of freedom, then it can perform, in principle, more than 65,000 actions, where each action
 110 corresponds to an effector state. This can pose considerable control problems for an embodied
 111 agent: How can it (learn to) select those actions that are adaptive for its life world?

112 We consider the process of selecting actions for an agent A given sensory information
 113 as a state transition function f , denoted as follows:

$$A, \overline{s}_t, \overline{m}_{t-1}, \overline{e}_{t-1} \xrightarrow{f} A, \overline{s}_{t+1}, \overline{m}_t, \overline{e}_t \quad (1)$$

114 Here, \overline{s}_t , \overline{m}_t and \overline{e}_t are the value assignments at time t to the sensors in S , internal states of
 115 M and effectors in E respectively.

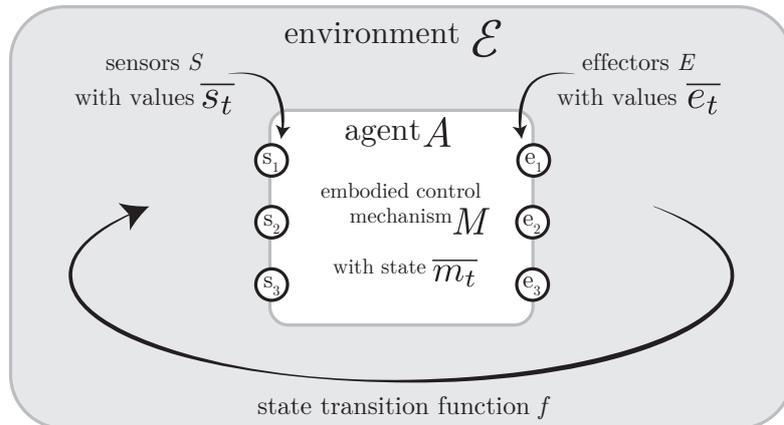


Figure 1. An agent A embedded in its life world \mathcal{E} .

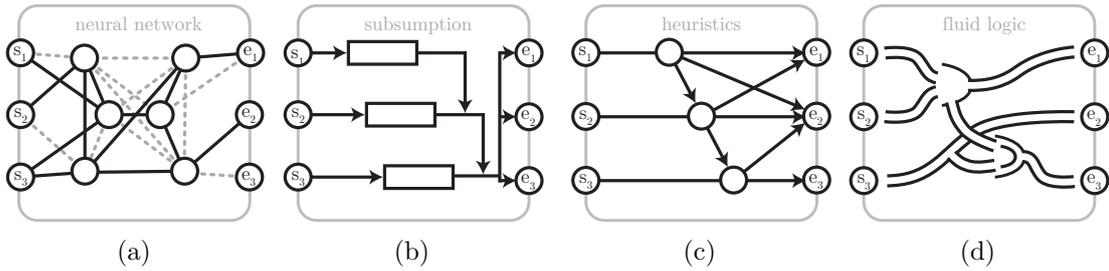


Figure 2. Example types of control architectures for embodied computation. a) Hopfield neural network. b) Brooks-type subsumption. c) Gigerenzer-type adaptive toolbox of heuristics. d) MONIAC fluid logic.

116 An embodied control architecture will need to control how the sensory states affect
 117 the effector states in such a way that agent can behave adaptively in its environment (e.g.,
 118 minimally move around, find food and try not to die.) Enactivists may envision different
 119 types of control architectures M . For instance, a Hopfield neural network that selects
 120 those effectors that are most coherent (energy minimum) given the sensory inputs and the
 121 internal network structure (Fig. 2a) (Bruck & Goodman, 1990; Thagard & Verbeurgt, 1998;
 122 Xu, Hu, & Kwong, 1996), M can be a Brooks-type subsumption architecture that given
 123 sensory states trigger the highest effector in the hierarchy that is not inhibited by any of
 124 the others (Fig. 2b) (Brooks, 1986), a Gigerenzer-type adaptive toolbox of heuristics that
 125 given sensory inputs decides based on cheap and dirty decision rules which effectors to
 126 activate (Fig. 2c) (Otworowska, Blokpoel, Sweers, Wareham, & van Rooij, 2018; Schmitt
 127 & Martignon, 2006), a MONIAC fluid logic mechanism that determines effector activation
 128 from sensor activation based on logic gates and fluid dynamics (Fig. 2d) (Isaac, n.d.) or a so
 129 far unknown architecture to be specified in the future. Importantly, though, for the agent to
 130 count as an enactive, and not cognitivist, we want the architecture to not involve any explicit
 131 planning and reasoning over representations of the world (and none of the abovementioned
 132 architectures do).⁵

133 3 The argument for embeddedness

134 To claim tractability of embodied computation, enactivists have access to two possible
 135 avenues: Either the control structure M is not *tractable-by-design* or it is. We consider each
 136 of these options and show that in each case *embeddedness* is central to understanding how
 137 an agent can tractably navigate its life world. We start, in Section 3.1, by considering the
 138 first option and consider the second option in Section 3.2.

139 3.1 Embedded tractable control

140 We say a control mechanism M is *not tractable-by-design* if there exist a logically
 141 possible world \mathcal{E} (which need not be A 's actual world) such that A could encounter control
 142 problems in \mathcal{E} that are NP-hard. This is the case when there can exist value assignments to

⁵Traditionally, the adaptive toolbox of heuristics may be considered cognitivist. Nevertheless, its architecture as formalised in Otworowska et al. (2018) has no internal reasoning over representations other than mapping sensory conditions to actions, not unlike a Brooks-type subsumption architecture.

143 its sensor, internal and effector states such that the transition function is NP-hard. This
 144 would be the case, for instance, if the control mechanism M were a Hopfield neural network
 145 and the control problem is conceived as minimizing⁶ the energy of the activation pattern
 146 (i.e., the value assignments over the internal and effector nodes in the network, given a value
 147 assignment for its sensory nodes).⁷ Such an agent would have the computational capacity
 148 for solving NP-hard problems, because it is possible to so-called polynomial-time reduce
 149 known NP-hard problems (like 3-SAT) to the operation of the control system of such an
 150 agent.⁸ This is not to say that an embodied agent A would explicitly solve such problems in
 151 the sense that the agent intentionally or mentally represents such problems explicitly or
 152 even at all. It is merely to say that the transition function has sufficient degrees of freedom
 153 or expressive power such that if we were to have access to such an M then we, scientists or
 154 external observers, would be able to read off the states computed by the transition function
 155 and use this information for solving NP-hard problems. This proves formally that an agent
 156 with such an M is not tractable-by-design, as it can encounter control situations for which
 157 no polynomial-time algorithm (of any kind; see Section 1.3) can possibly exist.

158 How could such an agent manage to nevertheless navigate its life world tractably?
 159 There is only one way possible: the agent's life world \mathcal{E} must be structured in such a way
 160 that the agent only encounters a proper subset of all the possible value assignments for its
 161 sensors *and* the control problem that M computes is tractable *for that subset*. In other words,
 162 tractability in this case will have to arise, in one way or another, from the *embeddedness* of
 163 the embodied agent.

164 This means that an enactivist program that wants to explain how enactive computation
 165 can be tractable, without enactive architecture being tractable-by-design, will need to
 166 characterize the specifics of the embeddedness that make tractable computation possible.
 167 This could be achieved if f is either polynomial-time computable or fixed-parameter tractable
 168 for the set of sensory inputs that can be induced by A acting in \mathcal{E} (Haselager, van Dijk, &
 169 van Rooij, 2008; Wareham et al., 2011).

170 3.2 Embedded tractable adaptation

171 We say a control mechanism M is *tractable-by-design* if for all its possible sensory states and
 172 environments \mathcal{E} the state transition function f is tractably (polynomial-time) computable.

⁶Nothing in our argument hinges on the assumption of optimality. First of all, non-optimality—be it by satisficing, heuristics, approximation or appeals to ‘as if’—does not generally buy one tractability (see van Rooij, Wright, and Wareham (2012), van Rooij, Wright, Kwisthout, and Wareham (2018)). Second, even when it does, that renders the architecture tractable-by-design and then our argument as laid out in Section 3.2 applies.

⁷Another example would be a hybrid deliberative-reactive architecture, as analyzed by Wareham, Kwisthout, Haselager, and van Rooij (2011). Enactivists may object that the deliberative component disqualifies this architecture as an enactive one. However, given that in the analyses of Wareham et al. the deliberative component itself could operate in non-classical non-representational manner, e.g., like a Hopfield network, we think such judgment would be premature.

⁸Computing a minimum energy activation pattern for a Hopfield neural network is NP-hard (Šima, 2001). This means that we can polynomial-time reduce 3-SAT to this problem. By carrying out this reduction from appropriate 3-SAT inputs (e.g., 3-CNF formulas used in the proof of Theorem 2.10 in Cadoli, Donini, Liberatore, & Schaerf, 2002), one can show that even for a fixed network, the problem of computing a minimum energy activation pattern corresponds to solving different (arbitrary) 3-SAT instances depending on the value assignment to the sensory nodes of the network.

173 This would be the case, for instance, for a Brooks-type reactive architecture, Gigerenzer-type
 174 adaptive toolbox of heuristics, and the MONIAC fluid logic mechanism proposed in Isaac
 175 (n.d.).⁹ All three are feedforward mechanisms and, accordingly, the transition function
 176 can be computed in polynomial time (in these specific cases, the time required is directly
 177 proportional to the size of the mechanism, and hence the transition function would be
 178 linear-time computable).

179 An agent with such a control mechanism purchases tractability at the cost of degrees
 180 of freedom or expressive power of its transition function. This means that it can only solve
 181 much simpler problems than an enactive agent that is not tractable-by-design (see Section
 182 3.1). Despite its simple control structure, such an agent can still display complex behaviour.
 183 We are reminded of Simon’s ant:

184 “An ant, viewed as a behaving system, is quite simple. The apparent complexity
 185 of its behavior over time is largely a reflection of the complexity of the environment
 186 in which it finds itself.” (Simon, 1996, p. 64)

187 How could such a simple agent ensure that its behaviour sufficiently fits its life world? Not
 188 being a cognitivist agent, it has no access to reasoning to weigh pros and cons of different
 189 actions nor can it internalise any non-cognitivist deliberation mechanism, as that would make
 190 it not tractable-by-design anymore (Wareham et al., 2011). Hence, its apparent fit must
 191 have been the result of an external, intergenerational adaptation process such as evolution
 192 by natural selection. However, tractability of that adaptation process cannot be taken for
 193 granted. For example, it has previously been shown that adapting Brooks-type reactive
 194 architectures and Gigerenzer-type toolboxes of heuristics is NP-hard in general (Otworowska
 195 et al., 2018; Wareham et al., 2011), and arguably the same holds for adapting the MONIAC
 196 fluid logic mechanism.¹⁰

197 Adaptation processes could be tractable if the environments in which adaptation is
 198 taking place are structured in such a way that the adaptation process only has to deal
 199 with a subset of all possible life worlds *and* adaptation is tractable *for that subset*. In
 200 other words, tractability will have to arise, in one way or another, from embeddedness
 201 of the adaptation process. Notably, this embeddedness goes beyond the nature of the
 202 embeddedness of the embodied agent required for sufficient fitness. This follows from
 203 previous complexity-theoretic results that establish that even if it is ‘promised’ that the
 204 environment \mathcal{E} is properly structured such that there exists a way to adapt a fit embodied
 205 agent, the process of adaptation can still be intractable.¹¹

⁹Computing transition functions using MONIAC fluid logic mechanism is computationally equivalent to solving the Circuit Value Problem for Boolean circuits—both are solvable in polynomial time (see, e.g., Arora & Barak, 2009, Theorem 6.30).

¹⁰The NP-hardness proof of Otworowska et al. (2018) of adapting Gigerenzer-type toolboxes of heuristics can straightforwardly be adapted to work also for the problem of adapting other mechanisms. To make this work, the most important properties are (i) that the mechanisms are required to be upper bounded in size (and that the upper bound is polynomially smaller than the size of the environment, but not trivially small), (ii) that the mechanism’s size limits the number of observations it can make about the environment, (iii) that the mechanism is sophisticated enough to compute if-then statements, and (iv) that carrying out the mechanism can be done in polynomial time. The MONIAC fluid logic mechanism satisfies these conditions (i)–(iv), so its adaptation problem is NP-hard.

¹¹One can prove, *reductio ad absurdum*, that so-called ‘promise’-adaptation (PA) is intractable (i.e., cannot

206 This means that an enactivist program that wants to appeal to evolutionary adaptation
 207 to not only explain how enactive computation can be tractable-by-design but also to ensure
 208 that embodied agents controlled by such architectures are fit in their life worlds will need
 209 to characterize the specifics of the embeddedness that make tractable adaptation possible.
 210 This could be achieved if the adaptation process is either polynomial-time computable (see
 211 Section 1.1) or fixed-parameter tractable (Section 1.2) for the set of environments \mathcal{E} for
 212 which the adaptation process yields an agent A that is tractable-by-design

213

4 Conclusion

214 Isaac (n.d.) proposed that consideration of computational complexity (in particular, tractabil-
 215 ity) can be informative for enactive approaches as it can help define the class of cognitive
 216 capacities that are tractable for embodied cognitive architectures. This holds regardless of
 217 whether or not we assume that enactive computation is analog or digital, and whether or not
 218 the tractability border is the same or different for enactivist and cognitive architectures. We
 219 wholeheartedly agree with Isaac on this point. The intent of our commentary is to highlight
 220 the vital role played by embeddedness in this endeavor.

221 With our analysis we have demonstrated that, regardless of whether or not one
 222 assumes that embodied computation is tractable by design, embeddedness is central to
 223 understanding how embodied agents can successfully navigate their life worlds in a tractable
 224 manner. Of course, given that embeddedness is one of the E's in the 4E program, this
 225 observation is in harmony with this program. Yet, it is our impression that to date it has
 226 been underestimated how critical embeddedness actually is for understanding the *tractability*
 227 of embodied computation. While the role of embeddedness in successful behaviour has been
 228 generally acknowledged, it seems that the 4E program seeks the solution for intractability in
 229 the type of control mechanism, e.g., analog, parallel, dynamical and/or non-representational
 230 (van Gelder, 1995, 1998). However, 4E may have been looking for a solution for intractability
 231 in a place where it cannot be found, and has overlooked that the key solution lies in one
 232 of its E's. For understanding tractable embodied computation, embeddedness should be
 233 given its proper status. In order to achieve this, enactivists will have to take on the task
 234 of explicitly modeling the environment \mathcal{E} and studying those properties of \mathcal{E} that render
 235 embodied control and adaptation tractable.

236

5 Acknowledgements

237 The authors thank Anco Peeters, Joe Dewhurst and an anonymous reviewer for helpful
 238 comments on an earlier version of this commentary.

be solved in polynomial time, unless $P=NP$). Assume that PA can be solved in polynomial time. It is then possible to solve Satisfiability (SAT; Garey & Johnson, 1979) in polynomial time by first polynomial-time reducing SAT to PA and then solving PA in polynomial time. There are three possible outputs of PA in this situation: (1) the promise holds, PA returns an adapted agent, and hence the solution to SAT is 'satisfiable'; (2) the promise does not hold, PA returns a maladapted agent, and hence the solution to SAT is 'unsatisfiable'; or (3) the promise does not hold, PA returns an uninterpretable solution, and hence the solution to SAT is 'unsatisfiable'. This implies that SAT can be solved in polynomial time. However, this contradicts our assumption that $P \neq NP$, as SAT is known to be NP-hard. Hence our assumption that PA can be solved in polynomial time must be false.

References

239

- 240 Aaronson, S. (2005). Guest column: NP-complete problems and physical reality. *ACM*
241 *SIGACT News*, 36(1), 30–52.
- 242 Abramova, E. (2019). *Joint action: an enactive mechanistic account* (Doctoral dissertation,
243 Radboud University, Nijmegen, The Netherlands).
- 244 Arora, S. & Barak, B. (2009). *Computational complexity – a modern approach*. Cambridge
245 University Press.
- 246 Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of*
247 *Robotics and Automation*, RA-2(1), 14–23.
- 248 Bruck, J. & Goodman, J. (1990). On the power of neural networks. *Journal of Complexity*,
249 6, 129–135.
- 250 Cadoli, M., Donini, F. M., Liberatore, P., & Schaerf, M. (2002). Preprocessing of intractable
251 problems. *Information and Computation*, 176(2), 89–120.
- 252 Cygan, M., Fomin, F. V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., . . . Saurabh,
253 S. (2015). *Parameterized algorithms*. Springer.
- 254 Di Paolo, E., Buhrmann, T., & Barandiaran, X. (2017). *Sensorimotor life: an enactive*
255 *proposal*. Oxford University Press.
- 256 Downey, R. G. & Fellows, M. R. (2013). *Fundamentals of parameterized complexity*. Berlin:
257 Springer.
- 258 Flum, J. & Grohe, M. (2006). *Parameterized complexity theory*. Berlin: Springer.
- 259 Fortnow, L. (2009). The status of the P versus NP problem. *Communications of the ACM*,
260 52(9), 78–86.
- 261 Frixione, M. (2001). Tractable Competence. *Minds and Machines*, 11(3), 379–397.
- 262 Garey, M. R. & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory*
263 *of NP-completeness*. San Francisco, CA: W. H. Freeman.
- 264 Gasarch, W. I. (2002). The P =? NP poll. *ACM SIGACT News*, 33(2), 34–47.
- 265 Haselager, W., van Dijk, J., & van Rooij, I. (2008). A lazy brain? Embodied embedded
266 cognition and cognitive neuroscience. In P. Calvo & T. Gomila (Eds.), *Handbook of*
267 *ciognitive science: an embodied approach* (pp. 273–290). Elsevier.
- 268 Isaac, A. M. C. (n.d.). Embodied cognition as analog computation. *Italian Journal of*
269 *Cognitive Science*, (this issue).
- 270 Milkowski, M. (2018). Embodied cognition meets multiple realizability. *Reti, saperi, linguaggi,*
271 *Italian Journal of Cognitive Sciences*, (2/2018), 349–364
- 272 Niedermeier, R. (2006). *Invitation to fixed-parameter algorithms*. Oxford University Press.
- 273 Otworowska, M., Blokpoel, M., Sweers, M., Wareham, T., & van Rooij, I. (2018). Demons
274 of ecological rationality. *Cognitive Science*, 42(3), 1057–1066.
- 275 Parberry, I. (1986). Parallel speedup of sequential machines: a defense of the parallel
276 computation thesis. *ACM SIGACT News*, 18(1), 54–67.
- 277 Schmitt, M. & Martignon, L. (2006). On the complexity of learning lexicographic strategies.
278 *Journal of Machine Learning Research*, 7(Jan), 55–83.
- 279 Siegelmann, H. T. & Sontag, E. D. (1994). Analog computation via neural networks. *Theo-*
280 *retical Computer Science*, 131(2), 331–360.
- 281 Simon, H. A. (1996). *The sciences of the artificial* (3rd). The MIT Press.

- 282 Šima, J. (2001). Energy-based computation with symmetric hopfield nets. In S. Ablameyko, M.
283 Gori, L. Goras, & V. Piuri (Eds.), *Limitations and future trends in neural computation*
284 (pp. 45–69). IOS Press.
- 285 Thagard, P. & Verbeurgt, K. (1998). Coherence as constraint satisfaction. *Cognitive Science*,
286 *22*(1), 1–24.
- 287 van Gelder, T. (1995). What might cognition be, if not computation? *The Journal of*
288 *Philosophy*, *92*(7), 345–381.
- 289 van Gelder, T. (1998). The dynamical hypothesis in cognitive science. *Behavioral and brain*
290 *sciences*, *21*(5), 615–628.
- 291 van Rooij, I. (2008). The Tractable Cognition Thesis. *Cognitive Science*, *32*(6), 939–984.
- 292 van Rooij, I., Blokpoel, M., Kwisthout, J., & Wareham, T. (2019). *Cognition and intractability:*
293 *a guide to classical and parameterized complexity analysis*. Cambridge University Press.
- 294 van Rooij, I., Wright, C. D., & Wareham, T. (2012). Intractability and the use of heuristics
295 in psychological explanations. *Synthese*, *187*(2), 471–487.
- 296 van Rooij, I. (2012). Self-Organization Takes Time Too: Topics in Cognitive Science. *Topics*
297 *in Cognitive Science*, *4*(1), 63–71.
- 298 van Rooij, I., Wright, C. D., Kwisthout, J., & Wareham, T. (2018). Rational analysis,
299 intractability, and the prospects of ‘as if’-explanations. *Synthese*, *195*(2), 491–510.
- 300 Villalobos, M. & Dewhurst, J. (2017). Why post-cognitivism does not (necessarily) entail
301 anti-computationalism. *Adaptive Behavior*, *25*(3), 117–128.
- 302 Villalobos, M. & Dewhurst, J. (2018). Enactive autonomy in computational systems. *Synthese*,
303 *195*(5), 1891–1908.
- 304 Wareham, T., Kwisthout, J., Haselager, P., & van Rooij, I. (2011). Ignorance is bliss: a
305 complexity perspective on adapting reactive architectures. In *Proceedings of the first*
306 *joint ieee international conference on development and learning and on epigenetic*
307 *robotics* (Vol. 2, pp. 1–5).
- 308 Xu, Z. B., Hu, G. Q., & Kwong, C. P. (1996). Asymmetric hopfield-type networks: theory
309 and applications. *Neural Networks*, *9*(3), 483–501.