Ignorance is Bliss: A Complexity Perspective on Adapting Reactive Architectures

Todd Wareham
Department of Computer Science
Memorial University of Newfoundland
St. John's, NL Canada A1B 3X5
Email: harold@mun.ca

Johan Kwisthout
Institute for Computing and
Information Sciences,
Radboud University
Nijmegen, the Netherlands
Email: j.kwisthout@science.ru.nl

Pim Haselager and Iris van Rooij
Donders Institute for
Brain Cognition and Behaviour,
Radboud University
Nijmegen, the Netherlands
Email: w.haselager@donders.ru.nl,
i.vanrooij@donders.ru.nl

***Abstract*—We study the computational complexity of adapting a reactive architecture to meet task constraints. This computational problem has application in a wide variety of fields, including cognitive and evolutionary robotics and cognitive neuroscience. We show that—even for a rather simple world and a simple task—adapting a reactive architecture to perform a given task in the given world is NP-hard. This result implies that adapting reactive architectures is computationally intractable regardless the nature of the adaptation process (*e.g.*, engineering, development, evolution, learning, etc.) unless very special conditions apply. In order to find such special conditions for tractability, we have performed parameterized complexity analyses. One of our main findings is that architectures with limited sensory and perceptual abilities are efficiently adaptable.**

I. Introduction

A popular class of control architectures in behavior-based robotics are the hybrid deliberative/reactive architectures [1]–[3]. These architectures combine the flexibility of a high-level deliberative system incorporating planning, world-knowledge, and memory with the speed and robustness of a low-level reactive system [4], [5]. A key issue for such architectures is the linking of these two components. One approach to this is for the deliberative component to adapt the reactive component in response to changing conditions, either by reconfiguring the interactions of existing reactive behaviors or augmenting existing behaviors with newly-designed ones [4, p. 214].

There are a number of robotic implementations which show that reactive adaptation is possible, *e.g.*, Autonomous Robot Architecture (AuRA) (see [4, Section 6.6.1] and references), Planner-Reactor [6], SSS [7]. These implementations show that one can efficiently adapt reactive architectures to meet *certain* task constraints in *certain* situations. An open question is to what extent such implementations can generalize to, and scale for, other types of tasks and situations.

Computational complexity theory provides techniques for addressing this question, and some preliminary results are known. For instance, Selman [8] found that devising a reactive plan for solving a given planning task is NP-hard (see also [9]), and Dunne, Laurence, and Wooldridge [10] proved that designing reactive agents that can perform a given achievement or maintenance task is NP-hard (see also [11]–[13]). Yet, the formalisms adopted by these researchers to model the reactive architectures and their life worlds were of such a high degree of generality that the intractability results may be due more to the formalisms used than the complexity inherent in adaptation. As our interest here is primarily in the latter, we study specifically the complexity of adapting subsumption reactive architectures [14] relative to simple static worlds.

Using techniques from classical complexity theory [15], we show that adapting a subsumption-based reactive architecture so that it can navigate a given world is NP-hard. This holds true, regardless whether the adaptation occurs by reconfiguring reactive-behavior layers in the architecture or by designing reactive-behavior layers anew and adding them to the architecture. These results indicate that adapting reactive architectures is computationally intractable unless very special conditions apply. This raises the question of which conditions characterize those situations in which adapting reactive architectures is tractable. An answer to this question may be relevant for various approaches to robotics. For instance, it can inform roboticists about the conditions that make adaptation of reactive architectures—*e.g.*, as done by deliberative modules in hybrid architectures or by evolutionary computation approaches to adapting a reactive robot to a task environment—feasible.[1] Moreover, it is of interest to cognitive neuroscience as it can inspire hypotheses about evolutionary or developmental explanations of animal and human brain structure.

In order to find conditions for tractability, we performed parameterized complexity analyses [20] of the problem of adapting reactive architectures. Our analyses reveal that only certain restrictions on either the internal structure of the architecture or the perceptual complexity of its sensory inputs render adaptation tractable. Though these results are derived in the context of a specific navigation task, we also show that they apply to any task for which a candidate architecture can be verified efficiently in a given world.

[1]Regarding evolutionary computation, our work has the additional advantage of providing a more robust understanding of why specific approaches (*e.g.* genetic algorithms and programming [16], 'try-out and see' evolutionary simulations aimed at extending reactive architectures with additional control mechanisms (see, *e.g.*, [17]–[19])) work or fail under certain circumstances.

The remainder of this paper is organized as follows. In Section II, we formalize reactive adaptation in terms of a simplified subsumption architecture for a basic navigation task, distinguishing between two forms of adaptation: reconfiguration and design. Section III demonstrates the general intractability of both of these problems. Section IV describes a methodology for identifying conditions for tractability, which is then applied in Section V to identify such conditions for the reconfiguration and design variants of adapting reactive architectures. Due to space limitations, all proofs of results are given in an online supplement.[2] Finally, our conclusions and directions for future work are given in Section VI.

II. Formalizing Reactive Adaptation

A. Adaptation for a Task

An adaptation mechanism for a particular robot architecture A that will enable A to perform some task T relative to a world W can be construed as a mechanism that adapts A in some limited fashion so that it can perform T in W. The computation performed by this mechanism is modeled by the following informal computational problem:

T-Adaptation by M (TA-M)
Input: World W, an architecture that can only partially perform task T in W, and an integer d.
Output: An architecture A' derived from A by at most d modifications of type M that can fully perform T in W, if such an A' exists, and special symbol $\perp$ otherwise.

Analyses of the computational complexity of this problem can show both (1) whether any adaptation mechanism suffices, *i.e.*, can operate in a reasonable amount of time and space, relative to the particular choices of world, task, architecture, and architecture-modifications, and if not, (2) under which restrictions on these choices such adaptation might be possible.

B. Adaptation for Basic Navigation

In this section, we will consider a particular formalization of problem TA-M relative to the task of basic navigation for a simplified subsumption-based reactive architecture.

Our worlds will be finite square-based maps in which compass movement is possible between adjacent squares, *i.e.*, north, south, east, and west, and each square is either a freespace (which a robot can occupy or travel through) or an obstacle. Each square has an associated type; let this set of types be denoted by E.

Within such a world, the navigation task will be to, starting from an arbitrary initial freespace, move to eventually occupy another arbitrary final freespace denoted by a specially-marked square-type. A robot that can do this relative to any two initial and final freespaces in W is said to be fully navigable for W; otherwise, the robot is partially navigable for W. Note there are no optimality restrictions on navigation, *e.g.*, the paths travelled need not be the shortest possible between the initial and final freespaces.

[2] http://www.cs.mun.ca/~harold/Papers/ICDL11supp.pdf

Our robot will be a simplified subsumption-based reactive architecture consisting of sensors, a set of layers, a total ordering on these layers, and a set of subsumption connections between layers. The sensors can see outwards in a radius r around the robot in every direction up to the closest obstacle in that direction, and can only verify, for each square-type $e \in E$, the presence of e within that perceptual radius, *i.e.*, $exists(e)$. Each layer has a trigger-condition that is a Boolean formula of length f over the available sensory $exists$-predicates and an action $a \in \{N, S, E, W\}$. If a layer's formula evaluates to $True$, the layer produces output a; otherwise, it produces the special output null. Given a set of layers L, we will assume that the formula in each layer contains at least one $exists$-predicate and no two layers encode formulas that both compute the same Boolean function and produce the same output. Relative to the total order on the layers, a layer i can have subsumption-links to any layer j that is lower than i in the ordering; between any two layers, there can exist an output-inhibition or output-override link (but not both). The output of any layer that subsumes at least one lower-level layer is not available directly for output; otherwise, that layer's output is available. The output of a set of ordered layers with subsumption links will be that of the highest layer relative to the order that is both available and non-null.

Relative to such a reactive architecture, we will consider two types of architecture modifications for adaptation:

1) Adding a selection of layers from a specified layer-library, along with some number of subsumption-link additions and deletions; and
2) Adding a selection of possible layers, along with some number of subsumption-link additions and deletions.

These modifications correspond to those considered in [4], [6], [7]. The above yields the following formalizations of problem TA-M relative to the navigation task and subsumption-based reactive architectures:

Navigation Adaptation by Reconfiguration
Input: A world W, a subsumption architecture A that is only partially navigable for W, a library M of layers, and integers s and l.
Output: A subsumption architecture A' derived from A by the addition of at most l layers from M and the addition or deletion of at most s subsumption-links that is fully navigable for W, if such an A' exists, and special symbol $\perp$ otherwise.

Navigation Adaptation by Design
Input: A world W, a subsumption architecture A that is only partially navigable for W, and integers s and l.
Output: A subsumption architecture A' derived from A by the addition of at most l layers and the addition or deletion of at most s subsumption-links that is fully navigable for W, if such an A' exists, and special symbol $\perp$ otherwise.

These problems will be denoted below by NA-REC and NA-DES, respectively. Note that layers may be added in any order relative to the layers in A in both problems, and that all layers have trigger-formulas of length $\leq f$.

III. Reactive Adaptation is Intractable

In this section, we address whether or not reactive adaption for the navigation task can be done efficiently relative to the subsumption architecture and architecture modifications described in Section II. Following general practice in Computer [15] and Cognitive (see [21] and references) Science, we define efficient solvability as being solvable in the worst case in time polynomially bounded in the input size, and show that a problem is not polynomial-time solvable by proving it to be at least as difficult as the hardest problems in problem-class NP, *i.e.*, NP-hard (see [15] for details).

Result 1: NA-REC and NA-DES are NP-hard.

Modulo the conjecture $P \neq NP$ which is widely believed to be true [22], the above shows that neither NA-REC nor NA-DES are polynomial-time solvable. Moreover, as $|M| = l = 0$ in the proofs of these results, this holds even if *no* new layers are added to the architecture, *i.e.*, the only modifications are to the subsumption-links between existing layers.

IV. A Method for Identifying Tractability Conditions

A computational problem that is intractable for unrestricted inputs may yet be tractable for non-trivial restrictions on the input. This insight is based on the observation that some NP-hard problems can be solved by algorithms whose running time is polynomial in the overall input size and non-polynomial only in some aspects of the input called *parameters*. In other words, the main part of the input contributes to the overall complexity in a "good" way, whereas only the parameters contribute to the overall complexity in a "bad" way. In such cases, the problem Π is said to be **fixed-parameter tractable** for that respective set of parameters. The following definition states this idea more formally.

Definition 1: Let Π be a problem with parameters $k_1, k_2, \ldots$. Then Π is said to be *fixed-parameter (fp-) tractable* for parameter-set $K = \{k_1, k_2, ...\}$ if there exists at least one algorithm that solves Π for any input of size n in time $f(k_1, k_2, ...)n^c$, where $f(\cdot)$ is an arbitrary function and c is a constant. If no such algorithm exists then Π is said to be *fixed-parameter (fp-) intractable* for parameter-set K.

In other words, a problem Π is fp-tractable for a parameter-set K if all superpolynomial-time complexity inherent in solving Π can be confined to the parameters in K. In this sense the "unbounded" nature of the parameters in K can be seen as a reason for the intractability of the unconstrained version of Π.

There are many techniques for designing fp-tractable algorithms [23], [24], and fp-intractability is established in a manner analogous to classical polynomial-time intractability by proving a parameterized problem is at least as difficult as the hardest problems in one of the problem-classes in the W-hierarchy $\{W[1], W[2], ...\}$ (see [20] for details). Additional results are typically implied by any given result courtesy of the following lemmas:

Lemma 1: [25, Lemma 2.1.30] If problem Π is fp-tractable relative to parameter-set K then Π is fp-tractable for any parameter-set K' such that $K \subset K'$.

Lemma 2: [25, Lemma 2.1.31] If problem Π is fp-intractable relative to parameter-set K then Π is fp-intractable for any parameter-set K' such that $K' \subset K$.

Observe that it follows from the definition of fp-tractability that if an intractable problem Π is fp-tractable for parameter-set K, then Π can be efficiently solved even for large inputs, provided only that all the parameters in K are relatively small. This strategy for rendering (otherwise intractable) problems tractable has been successfully applied in a variety of areas (see [20], [26] and references). In the next section we report on our investigation of whether or not the same strategy may be used to render the problems NA-REC and NA-DES tractable.

V. What Makes Reactive Adaptation Tractable?

The problems NA-REC and NA-DES have several parameters whose restriction could conceivably render reactive adaptation tractable. An overview of the parameters that we considered in our fp-tractability analyses is given in Table I. These parameters can be divided into three groups:

1) Restrictions on the (perceived) world ($|E|$);
2) Restrictions on subsumption architectures ($|L|, f$); and
3) Restrictions on architecture-modification ($s, |M|, l$).

The meaning and relevance of parameters $|L|, s, |M|$, and l in the context of cognitive robotics is, we believe, rather straightforward. However, a more intuitive characterization of the parameters $|E|$ and $|f|$ may help in appreciating the import and relevance of our findings later on. Intuitively, one can think of $|E|$ as the number of distinct features of the world that are relevant for the agent to decide how to act. Besides the possibility of viewing $|E|$ as a property of the world, it can also be viewed as a property of the system, viz. its sensory sensitivity or discriminability. This is so because we are here working under the assumption that sensation is error-free.[3] Further, parameter f can be thought of as a characterization of the complexity of the perceptual patterns that the architecture distinguishes and uses to select its actions. Namely, the larger the value of f, the longer the possible logical formulas that can trigger action layers, and thus the larger the allowable complexity of the patterns of perceived features in the world that determine the selection of actions.

In the remainder of this section, we will assess the fp-tractability of NA-REC and NA-DES relative to all parameters in Table I (Section V-A), note how these results apply in more general settings (Section V-B), and discuss the implications of these results (Section V-C).

A. Results

Result 2: NA-REC is fp-intractable for $\{s, f, l, |M|\}$ and $\{s, f, l, |L|\}$.

[3]Admittedly this is an idealization, but its consequence is only that our computational complexity results lower bound the complexity of adaptation for reactive architectures with noisy sensory systems (see Section V-B).

TABLE I
PARAMETERS CONSIDERED IN ANALYSES OF NA-REC AND NA-DES.

Param.	Definition	Appl.
$\|E\|$	Number of distinguishable square-types in world	All
$\|L\|$	Number of layers in derived architecture A'	All
f	Maximum length of layer trigger-formula	All
s	Maximum number of subsumption-link changes	All
l	Number of layers added to A	All
$\|M\|$	Number of layers in provided library	NA-REC

Result 3: NA-REC is fp-tractable for $\{|L|, |M|\}$ and $\{|E|\}$.

Note that these results, combined with those implied by Lemmas 1 and 2, *completely* characterize the parameterized complexity of NA-REC relative to each subset of parameters in the set $\{|E|, |L|, |M|, f, s, l\}$.

Result 4: NA-DES is fp-intractable for $\{s, f, l, |L|\}$.

Result 5: NA-DES is fp-tractable for $\{|E|, f\}$.

Note that these results, combined with those implied by Lemmas 1 and 2, characterize the parameterized complexity of NA-DES relative to each subset of parameters in the set $\{|E|, |L|, f, s, l\}$ *except* those subsets consisting of the union of $\{|E|\}$ and a subset of $\{|L|, s, l\}$.

B. Generality of Results

Our intractability results, though defined relative to admittedly unrealistic types of worlds, tasks, and architectures, have remarkable generality. Observe that the worlds, tasks, and architectures for which these results hold are in fact restricted versions of more realistic alternatives, *e.g.*,

- static deterministic worlds are special cases of dynamic non-deterministic or probabilistic worlds (restrict motion and you have stasis; restrict choices of motion and you have determinism);
- fully-observable worlds are special cases of partially-observable worlds (restrict unobservability and you have full observability);
- the point-to-point navigation task (which is the actual task for which we show all of our intractability results above) is a special case of many other tasks; and
- the simplified reactive architecture defined in Section II-B is a special case of architectures that allow more complex types of layers or action-sets.

Intractability results for these more realistic alternatives then follow from the well-known observation in computational complexity theory that intractability results for a problem Π also hold for any problem Π' that has Π as a special case and can hence solve Π (suppose Π is intractable; if Π' is tractable, then it can be used to solve Π efficiently, which contradicts the intractability of Π – hence, Π' must also be intractable).

Our fp-tractability results are more fragile, as innocuous changes to worlds, tasks, or architectures may in fact violate assumptions critical to the operation of the algorithms underlying these results. For now, we can say that as our fp-tractability results depend only on the combinatorics of reconfiguring and designing reactive architectures and require only that a candidate architecture can be verified to perform a particular task in a given world in time polynomial in the size of that world, these results apply relative to these architectures for *all* choices of world and task that are polynomial-type verifiable relative to these architectures.

C. Discussion

We have found that adapting reactive architectures, whether by recruiting pre-existing layers or by designing layers anew, is NP-hard (Result 1). This NP-hardness holds even for a basic navigation task in a simple 2D static world. Moreover, adapting reactive architectures remains NP-hard even if the adaptation is restricted to rewiring the given subsumption architecture, *i.e.*, without adding any new layers to it. These intractability results underscore the computational difficulty of adapting reactive architecture, be it by a human designer, a deliberative component in a hybrid robot, or by evolution, development or learning in a (human) brain.

To our knowledge, no explicit conjectures about the sources of computational difficulty in reactive adaptation have been made in the literature, but on the basis of successful reactive system design done by humans (consisting of small (≤ 10) numbers of layers developed in an incremental add-and-test manner ([27], [28]; see also [4, pp. 74–77])), it seems reasonable to conjecture that restrictions on the subsumption architectures ($|L|, f$) and degree of allowed modification ($s, l, |M|$) should render reactive adaptation tractable. However, it does not (Results 2 and 4). What does result in tractability is when the total number of layers that can be used to configure a reactive architecture is small (*i.e.*, both $|L|$ and $|M|$ are small) (Result 3). Though useful to know, we can imagine this condition may be of limited interest or applicability for roboticists, as such reactive architectures will—by definition—have quite restricted behavioral repertoires.

Of greater interest, perhaps, is the second class of conditions for tractability that we have identified; viz., restrictions pertaining to the sensory and perceptual abilities of the architectures. For instance, we found that reconfiguring a reactive architecture to perform a task can be done efficiently provided only that the sensory sensitivity of the architecture (*i.e.*, the number of environmental features it can distinguish, $|E|$) is not too large (Result 3). In the more general case, where also newly designed layers can be added to the architecture, a simultaneous restriction of this sensory complexity *and* perceptual complexity—in the sense of the ability of layers to encode patterns in detected features ($|E|, f$)—renders adaptation of a reactive architecture tractable (Result 5).

Of course, these tractability results are modulo the assumption that for the task, world and architecture under consideration there *exists* a reactive architecture that can be constructed through adaptation and perform the task in the given world. One possible reading of our finding, then, is that adapting

reactive control is feasible in environments that are structured such that the features and patterns that are relevant for successful behavior can be succinctly represented by a small set of features and percepts of low complexity (cf. what [29] called "being ignorantly successful"). Although it is plausible that low perceptual complexity may characterize perception for humans [30], [31], probably only simpler organisms are characterized by low sensory sensitivity. It is expected then that if such simple organisms—or more generally, agents—enter an environment to which they can in principle adapt that they can do so quickly. Quick adaptation of more sensory-complex agents may still be possible, *e.g.*, by exploiting restriction on the classes of sensory information that can be detected (cf. the sensory modalities) or by exploiting a limited sensory radius (r), but if and how this could be done is an open question for future research.

VI. Conclusions

We have presented two formal characterizations of the problem of adapting a reactive architecture, one for reconfiguring such an architecture and one for designing parts of it anew. Our complexity analyses reveal that, while these problems are computationally intractable in general, there are conditions that render them tractable. Knowledge of these conditions can be exploited in both robotics and cognitive neuroscience to understand which properties a reactive architecture needs to have to be efficiently adaptable.

In future research, we will explore the precise extent to which our results hold for more complex worlds, tasks, and architectures. In particular, we will aim to identify more conditions for tractability by exploring alternative ways of characterizing an agent's sensory and perceptual complexity.

Acknowledgments

The authors would like to thank five anonymous reviewers for comments that improved the presentation of this paper. TW was supported by NSERC Discovery Grant 228104. JK was supported by the OCTOPUS project under the responsibility of the Embedded Systems Institute.

References

[1] J. Togelius, "Evolution of a subsumption architecture neurocontroller," *Journal of Intelligent and Fuzzy Systems*, vol. 15, no. 1, pp. 15–20, 2004.

[2] E. Yoshida, K. Yokoi, and P. Gergondet, "Online replanning for reactive robot motion: Practical aspects," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 5927–5933.

[3] S. Yue, R. Santer, Y. Yamawaki, and F. Rind, "Reactive direction control for a mobile robot: a locust-like control of escape direction emerges when a bilateral pair of model locust visual neurons are integrated," *Autonomous Robotics*, vol. 28, no. 2, pp. 151–167, 2010.

[4] R. C. Arkin, *Behavior-Based Robotics*. Cambridge, MA: The MIT Press, 1998.

[5] H. Hexmoor and D. Kortenkamp, "Issues on building software for hardware agents," *Knowledge Engineering Review*, vol. 10, no. 3, pp. 301–304, 1995.

[6] D. Lyons and A. Hendricks, "Planning as incremental adaptation of a reactive system," *Robotics and Autonomous Systems*, vol. 14, no. 4, pp. 255–288, 1995.

[7] J. Connell, "SSS: A hybrid architecture applied to robot navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Nice, France, 1992, pp. 2719–2724.

[8] B. Selman, "Near-optimal plans, tractability, and reactivity," in *Proceedings of the Fourth International Conference on Knowledge Representation and Reasoning*, Bonn, Germany, 1994, pp. 521–529.

[9] P. Jonsson, P. Haslum, and C. Bäckström, "Towards efficient universal planning: A randomized approach," *Artificial Intelligence*, vol. 117, pp. 1–29, 2000.

[10] P. E. Dunne, M. Laurence, and M. Wooldridge, "Complexity results for agent design," *Annals of Mathematics, Computing & Teleinformatics*, vol. 1, no. 1, pp. 19–36, 2003.

[11] M. Wooldridge, "The computational complexity of agent design problems," in *Proceedings of the Fourth International Conference on Multi-Agent Systems*, Boston, MA, 2000, pp. 341–348.

[12] I. A. Stewart, "The complexity of achievement and maintenance problems in agent-based systems," *Artificial Intelligence*, vol. 146, pp. 175–191, 2003.

[13] M. Wooldridge and P. E. Dunne, "The complexity of agent design problems; determinism and history dependence," *Annals of Mathematics and Artificial Intelligence*, vol. 45, pp. 343–371, 2005.

[14] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 1, pp. 14–23, 1986.

[15] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: W.H. Freeman, 1979.

[16] A. Eiben and J. Smith, *Introduction to Evolutionary Computing*. Berlin: Springer-Verlag, 2007.

[17] S. Lagarde, I. Sprinkhuizen-Kuyper, G. de Croon, and W. Haselager, "An intermediate form of behavioral control in reactive' robots," in *Proceedings of the 22nd Benelux Conference on Artificial Intelligence*, 2010. [Online]. Available: http://bnaic2010.uni.lu/Papers/Category%20A/Lagarde.pdf

[18] L. Bax, *Cognitive Control in Reactive Agents: Surviving Predators through the Evolution of a Circadian Rhythm*. Bachelor thesis, Department of Artificial Intelligence, Radboud University Nijmegen, 2009. [Online]. Available: http://www.nici.ru.nl/∼idak/teaching/batheses/bax_l_bathesis.pdf

[19] Y. Fang, *Evolving Minimalistic Control for Complex Behaviour*. Bachelor thesis, Department of Artificial Intelligence, Radboud University Nijmegen, 2009. [Online]. Available: http://www.nici.ru.nl/∼idak/teaching/batheses/fang_y_bathesis10.pdf

[20] R. Downey and M. Fellows, *Parameterized Complexity*. Berlin: Springer, 1999.

[21] I. van Rooij, "The Tractable Cognition Thesis," *Cognitive Science*, no. 32, pp. 939–984, 2008.

[22] L. Fortnow, "The Status of the P Versus NP Problem," *Communications of the ACM*, vol. 52, no. 9, pp. 78–86, 2009.

[23] R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[24] C. Sloper and J. A. Telle, "An overview of techniques for designing parameterized algorithms," *Computer Journal*, vol. 51, no. 1, pp. 122–136, 2008.

[25] T. Wareham, *Systematic Parameterized Complexity Analysis in Computational Phonology*. Ph.D. thesis, Department of Computer Science, University of Victoria, 1999.

[26] M. Cesati. (2006) Compendium of Parameterized Problems. [Online]. Available: http://bravo.ce.uniroma2.it/home/cesati/research/compendium/

[27] R. A. Brooks, "A robot that walks: Emergent behavior from a carefully evolved network," *Neural Computation*, vol. 1, no. 2, pp. 253–262, 1989.

[28] ——, "Intelligence without representation," *Artificial Intelligence Review*, vol. 47, pp. 139–160, 1991.

[29] W. Haselager, J. van Dijk, and I. van Rooij, "A lazy brain? Embodied embedded cognition and cognitive neuroscience," in *Handbook of Cognitive Science: An Embodied Approach*, P. Calvo and T. Gomila, Eds. Elsevier, 2008, pp. 273–290.

[30] K. Koffka, *Principles of Gestalt Psychology*. New York: Harcourt Brace, 1935.

[31] P. van der Helm, "Dynamics of Gestalt psychology (Invited review of *perceptual dynamics: theoretical foundations and philosophical implications of gestalt psychology* by F. Sundqvist)," *Philosophical Psychology*, vol. 19, no. 1, pp. 274–279, 2006.