Preprint of
B. Inden and J. Jost, "Effects of Several Bioinspired Methods on the Stability of Coevolutionary Complexification," ***2015 IEEE Symposium Series on Computational Intelligence*****, Cape Town, 2015, pp. 1094-1101.**

LEGAL NOTICE

© 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
http://dx.doi.org/10.1109/SSCI.2015.157

Effects of several bioinspired methods on the stability of coevolutionary complexification

Benjamin Inden
Department of Computer Science and Technology
University of Bedfordshire
Luton, United Kingdom
benjamin.inden@beds.ac.uk
&
Jürgen Jost
Max Planck Institute for Mathematics in the Sciences
Leipzig, Germany
jost@mis.mpg.de
Santa Fe Institute
Santa Fe, New Mexico, USA

Abstract

We study conditions for sustained growth of complexity in an abstract model of parasitic coevolution. Previous research has found that complexification is hard to achieve if the evolution of the symbiont population is constrained by the hosts but the evolution of the hosts is unconstrained, or, more generally, if the task difficulty is much higher for the symbionts than for the hosts. Here we study whether three bioinspired methods known from previous research on achieving stability in coevolution (balancing, niching, and reduced resistance) can restore complexification in such situations. We find that reduced resistance, and to a lesser degree niching, are successful if applied together with truncation selection, but not if applied together with fitness proportional selection.

1 Introduction

Evolutionary algorithms have been used successfully to solve various optimization problems including for scheduling, symbolic regression in astronomy, optimizing antenna designs and shapes of car parts, finding electronic circuits that perform a given function, and game playing (e.g., [1, 2]). In the field of evolutionary robotics, they are used to evolve topologies and connection weights of neural networks that in turn control robot behavior [3]. However, when evolution is used to create robot behavior, the resulting complexity is typically rather limited, especially if compared to the results of more traditional engineering methods. It is therefore desirable to better understand how evolutionary processes can lead to the emergence of complex adaptations, and what kinds of adaptations they can produce.

In theoretical biology (where these questions are also important), it is well understood that direct evolution towards a fixed target cannot produce some kinds of complexity [4]. Coevolutionary processes, on the other hand, can overcome some of these limitations [5]. Within the field of evolutionary computation, it has been found that the coevolution of solution candidates with solution quality tests can lead to

better results than direct evolution of solution candidates against a static fitness function [6]. Obviously, coevolution can provide a path of incremental learning for autonomous agents. However, coevolutionary dynamics can also lead to undesired outcomes. Well known problems are:

Disengagement: One population becomes vastly superior to the other such that no gradient for learning is available any more;

Overspecification: One population becomes very successful in interactions with the particular individuals of the other population, but lacks sufficient generality;

Strategy cycling: Populations cannot find a generally superior strategy, but only strategies that are successful against some types of interaction partners (cf. the „rock /paper / scissors“ game), which leads to the cyclic re-emergence of simple strategies that are successful only against the currently present interaction partners.

A number of techniques have been used to prevent or reduce such problems:

Reduced virulence: Selecting for individuals in the superior population that are less than optimal can prevent disengagement [7];

Balancing: Reducing the speed of evolution (i.e., increasing generation length) for one population can help the other not to loose track [8];

Niching: If the resources that can be provided by the interaction with one particular hosts are limited and have to be shared by all symbionts, this can enhance diversity in the symbiont population [9, 10], which makes the population more adaptable in the long run;

Spatial structure: Only allowing local interactions between symbionts and hosts and/or for selection within one population can also promote diversity [6, 11];

Hall of fame coevolution: If individuals have to compete against interaction partners from previous generations as well, this can prevent strategy cycling [8].

So far, coevolution has been typically used either for solving fixed problems like function approximation (e.g., [6]) or for finding desired behaviors in evolutionary robotics (e.g., [12]), and success has been defined mainly either subjectively by inspection of evolved behavior, or as reaching the fixed goal. While a certain amount of complexification will typically have to occur in coevolutionary robotics for the emergence of an interesting strategy (e.g., [13]), this complexification is typically neither the main focus, nor is it explicitly measured. In fact, complex strategies do not necessarily correspond to complex internal structures of agents. They can also arise by interaction with a sufficiently complex environment [14]. However, natural agents typically have much more complex internal structures (e.g., nervous systems) than have artificially evolved agents. Furthermore, their behavioral complexity far exceeds the complexity so far achieved in artificial evolution. Besides, within a given environment, there is certainly a correlation between the internal complexity and the complexity of the behavior. It is therefore desirable to study more explicitly which conditions can lead to coevolutionary complexification of autonomous agents. The type of complexification that we are interested in is not just an increase in the number of components. We are interested in the number of components that, taken together, solve a particular problem arising as a consequence of the need to survive and reproduce, or in other words, that can be seen as performing a particular function.

Abstract models of coevolution allow measurements of complexity, and make identification of the factors leading to complexity more easy. Previous research on coevolutionary number games (e.g., [15]) may provide a starting point. However, this research was targeted at other questions about coevolutionary dynamics, and the used encodings (single numbers or vectors of a few component numbers) do not provide the potential for complexification. Previous biological research on “Gene for Gene” coevolution models

and various related extensions [16, 17] has focused on the dynamics of one gene or a fixed number of genes with several (typically 2) alleles each that can make a host resistant to infection, or a pathogen virulent. In these models, the more powerful alleles typically incur fitness costs, and the distributions of alleles over time in the two populations are studied. These models are not concerned with questions of complexification over time or the difficulties of finding powerful alleles in the first place, and "arms races" [18] arising in these models typically consist of cycles where the frequencies of a finite number of different strategies in the populations increase and decrease.

Further theoretical biology research on coevolution has been done using the Webworld family of models [19]. In the basic Webworld model, a species is characterized by a fixed size list of features. It interacts with species that have other features as defined by a randomly initialized feature interaction matrix. The model consists of phases were species sizes are computed iteratively by means of differential equations until the food web becomes stable. In between these phases, new species are created by randomly changing the feature lists of existing species. This model has been extended such that the number of features of a species can vary within bounds, and a growth in complexity has been observed in the extended model, although the reasons for that growth were not entirely clear [20].

Only very few models have been created to study complexification with abstract individual based models so far. The Foodchain model [21] models symmetric competitive coevolution between individuals that are strings of letters with a fixed length. Some letters can be used for attack, others for defense, and the rest have no function. Matching is done between sequences of attack and defense letters present in the genome of two individuals to determine which one gains from the interaction. Point mutations and duplications are used in an evolutionary process that leads to complexification, i.e., a growth of functional subsequences in the genomes. One of us has published more general work on number sequence games [22]. It was shown in this work that complexification can occur in models of mutualism, parasitism, and commensalism between two species. In some cases, the growth of complexity was apparently unbounded. It has also been argued that these models therefore fulfill a previous formal definition of open-ended evolution [23]. Those few systems that were designed to fulfill this definition before were either very complex, making it hard to understand why they exhibited this open-ended evolution, [24], or relied on diversity rather than complexification to produce open-ended evolution [25]. In the new model, the influence of different mutation and selection methods, as well as that of relative task difficulties for the two species, were also investigated. It turned out that significant and sustained complexification only occurs in these models if the task difficulty for the symbionts is not much more difficult for the hosts. Of course, when working with more realistic tasks (e.g., in evolutionary robotics), the task difficulty for the various populations is unknown in advance. Therefore, it is desirable to have a method of coevolution that can achieve complexification over a wide range of task difficulty ratios. We study here whether three of the above mentioned techniques can contribute to this goal. We focus on balancing, niching, and reduced virulence (the last method is called reduced resistance here because unlike in the work where the method was introduced, it is applied to the host population here, not to the symbiont population). These methods are applied on parasitic coevolution (the mutualistic case is not of interest here because as it has been defined in [22], the task difficulty is always the same for both populations). Furthermore, because previous research has indicated that complexification in these scenarios also depends on the used selection method (with truncation selection typically leading to faster and more stable complexification than fitness proportional selection), we conduct experiments using both truncation selection and fitness proportional selection here.

2 Methods

2.1 Number sequence games

There are two populations in the number sequence games studied here. The genotype — and the phe-

notype — of a member of either population is a sequence of numbers. These numbers can, for example, represent enzymes that need to be expressed in a particular temporal order, or behavioral primitives that can be composed to reach a certain goal. Typical parasitic scenarios include host defense enzymes, and corresponding neutralizing parasite enzymes, or behavioral primitives in a pursuit and evasion scenario.

In every generation, each organism in one population is tested against every organism in the other population. In the basic parasitic model, symbionts gain a fitness bonus for every host organism whose number sequence they completely match. This means that there must be pairwise matches between corresponding host and symbiont numbers. This pairwise matching is performed until either only the host sequence ends (the symbiont wins), only the symbiont sequence ends (the host wins), both sequences end (the symbiont wins), or a wrong number event occurs in the host sequence (the symbiont wins). A wrong number event means that the defense produced by the host is ineffective because it violates externally provided constraints. To use such constraints in the parasitic model has proved to be beneficial in previous research on the model because otherwise the problem to be solved by the symbionts (which have to exactly match the host numbers) is much more difficult than the problem to be solved by the hosts (which can choose any number), which normally leads to the hosts escaping from the symbionts early in evolution. We will report experiments below where we remove these constraints. A match between two numbers is defined as equality here, but other relations (like complement) would be equivalent as long as they do not change the number of possible solutions.

Each fitness bonus has a value of 1.0. To arrive at the final fitness of an organism, these bonuses are added, and then a fitness contribution due to genome length $f_l(o) = c_{lp} \exp(-0.1 \cdot [genome\, length])$ is added, where c_{lp} is a constant that determines the costs of adding and maintaining further genes.

The externally imposed constraints on the host number sequence are designed to ensure that only a fraction $1/n_p$ of the newly added genes will be effective, so the space of host solutions is constrained exactly as much as the space of symbiont solutions (where also only one number will match at a given position). In principle, a random number could be drawn for each sequence position at various stages during coevolution when the constraint information is needed for the first time, but it is equivalent (and easier to implement) to choose the whole constraint sequence once at the beginning of evolution. Because this sequence is of potentially infinite length, we take the following rule-base approach: The first number in the sequence must be 1, and a number at position $n+1$ must have the value $v(n+1) = (v(n)+1)\%n_p$, where $0..n_p-1$ is the range of possible gene values and '%' denotes the modulo operation. Thus, the only effective gene sequence of length $n < n_p$ takes the form $[1, .., n]$. Now the resulting sequence is obviously one of high regularity and therefore low algorithmic (Kolmogorov) complexity, but it should be noted that neither the genetic system nor the fitness function (apart from this constraint) use any notion of neighborhood of numbers, and there is no way of predicting future constraints based on previously seen constraints with the simple genetic representation and mutation operators that we use here, so we could equivalently use any other constraint sequence, including a completely random one.

Simple fitness proportional selection (without any elite mechanism) and truncation selection (where 10% of the population is used as parents) is used. All experiments use a well-mixed population without any spatial structure.

2.2 Genetic representation and operators

As already mentioned, the genome is basically a sequence of numbers. Three mutation operators are used on these sequences: add a number (with probability 0.2), delete a number (with probability 0.1), and change a number (with probability 0.2). Values are always randomly drawn with uniform probabilities over the whole range. The operations are only applied at the end of the sequence. Previous research has shown that if mutations are applied with equal probabilities over the whole sequence length, complexification becomes slower and much less stable, and typically needs very strong selection to be

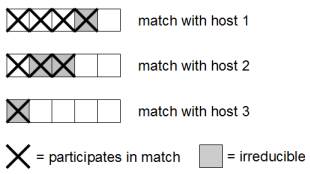

Figure 1: Measuring irreducible functional unit size (IFUS) in symbionts. In this example, the host population size is 3, and the symbiont sequence matches all host sequences. IFUS is defined as the maximum of the number of irreducible sites in a matching sequence, where irreducible means that there is no intermediate reward provided by other matches.

maintained at all [22]. We have argued elsewhere that applying mutations only to the most recently evolved elements can not only be considered as a useful simple "extreme case " model of certain biological scenarios, but can also lead to sustained complexification of solutions in evolutionary robotics [26].

All populations are seeded with a common ancestor that has a random sequence of length 1. Recombination is not used in the experiments reported here.

2.3 Measuring outcomes of coevolution

As in previous research [22], we calculate *irreducible functional unit size* (IFUS) for organisms in the symbiont population. This is done by iterating over all cases where an organism from the symbiont population matches an organism from the host population in a given generation (see Fig. 1). The sites in the symbiont genome that participated in that match are marked. All sites that also participated in shorter matches in the given generation are unmarked again. IFUS is then defined as the maximum of the number of marked sites over all matches. The highest such values are recorded every generation. Because IFUS takes into account only sites that participated in achieving a match, it does not just measure sequence length, but complexity of function. Because it ignores sites participating in other matches, it may actually underestimate complexity. Nevertheless, as explained in [22], it allows to study some interesting questions about the capabilities of evolutionary processes, and we kept it here for comparability to the earlier research. A variant of the measure, functional unit size (FUS), is calculated without unmarking sites that participated in shorter matches. Its value, which is not reported here, is typically higher by a small constant than that of IFUS in coevolutionary simulations such as those reported here.

2.4 Reduced resistance

Starting from the observation that pathogens that kill their host often do less well in the long run, a method called reduced virulence was introduced in [7], where symbionts that only win a fraction λ of the contests that the best adapted symbiont wins get optimal fitness. Here, we deal with the problem of escaping hosts, so we modify the fitness of hosts correspondingly: $f_{adj} = f_{best}\left(\frac{2\frac{f_{raw}}{f_{best}}}{\lambda} - \frac{\left(\frac{f_{raw}}{f_{best}}\right)^2}{\lambda^2}\right)$. This equation describes a parabola that has its maximum at λ. The fitness adjustment could be viewed as modeling in a very simple way the phenomenon that hosts that invest a lot into defense can spend less energy for other functions, e.g. reproduction, and are therefore disadvantaged as compared to hosts that invest less in defense. By default, $\lambda = 0.75$ as in previous work.

2.5 Balancing

In nature, hosts (e.g., mammals or birds) often have longer generation times than symbionts (e.g., viruses or bacteria). This provides biological motivation for reducing the speed of evolution for the hosts in order to help the symbionts not to loose track. Preliminary experiments where fixed generation ratios between host and symbiont populations between 1:2 and 1:20 were used have not shown much promise. Therefore, we here focus on a method similar to balancing as introduced in [8]: A new generation of hosts is only created if, in the previous generation, no host won

over all symbionts. Otherwise, only a new generation of the symbionts is created.

2.6 Niching

In nature, a single host has only a fixed amount of resources that can be exploited by the symbionts. The well known evolutionary method of niching, which has sometimes been applied to coevolution as well [9, 10], models this by dividing the bonus obtainable from winning over a given host equally among those symbionts in the same niche — in this case, this is just those symbionts that won against that particular host. This simple niching method does not have any parameters, but we can easily generalize it by stating that $f_{final} = \frac{f_{orig}}{(niche\ count)^x}$. Then $x = 1$ is the standard case and $x = 0$ corresponds to not using niching at all. If we set $0 < x < 1$, this corresponds to a situation in which several symbionts deplete host resources subadditively, whereas for $x > 1$, they have synergistic (superadditive) effects on host resources. In principle, it is also possible to set $x < 0$, although one might wonder to what kind of biological scenario this corresponds. What comes to mind are situations where the host's immune system can be overpowered more easily if more parasites are present, and therefore the gain for individual parasites is greater. Scenarios of roughly this kind have been reported in the context of investigating quorum sensing in bacteria [27].

3 Experiments and results

A first set of experiment examines complexification when the host population is constrained and fitness proportional selection is used (Fig. 2; as for all following results, 20 runs with different random seeds have been performed per configuration). All configurations achieve sustained linear growth of complexity in this case. Compared to plain selection, niching increases the final complexity significantly when applied on it own or together with balancing, whereas balancing on its own, as well as reduced resistance, significantly decrease the final complexity.

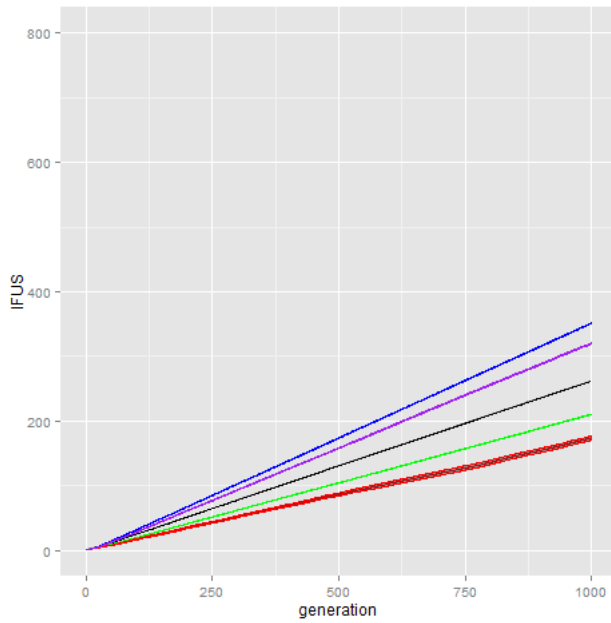

Figure 2: Complexification when constraints are present and fitness proportional selection is used. Black: default; blue: niching; green: reduced resistance; red: balancing; purple: balancing + niching. The central line indicates the mean of 20 runs, whereas the surrounding ribbon indicates the uncertainty of the mean (standard error).

When the host population is constrained and truncation selection is used (Fig. 3), all configurations achieve sustained linear growth of complexity, which is even faster than when using fitness proportional selection. Compared to plain selection, niching alone does not change anything. Balancing with or without niching leads to a significant decrease in final complexity, and reduced resistance even more so.

If the host population is not constrained, it becomes much more difficult for the symbionts to win over the hosts. As a result, much less complexity evolves. For fitness proportional selection (Fig. 4), complexity converges at a low level for plain selection, niching and reduced resistance. However, balancing applied alone or with niching leads to a moderate growth of complexity.

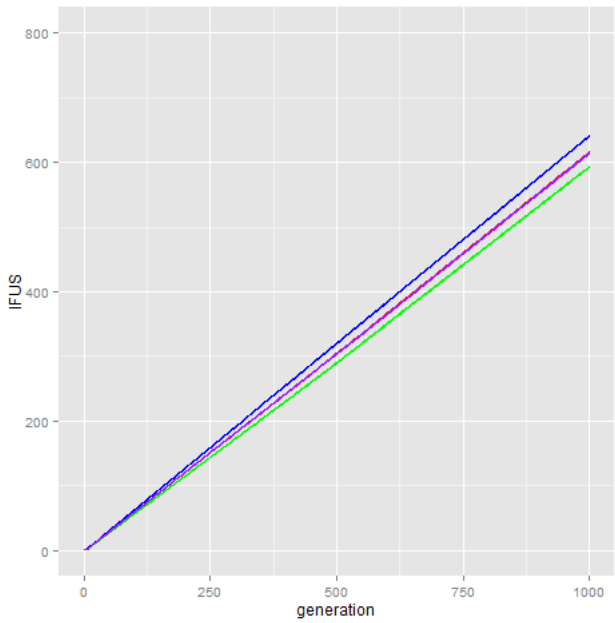

Figure 3: Complexification when constraints are present and truncation selection is used. Black (hidden behind blue): default; blue: niching; green: reduced resistance; red (hidden behind purple): balancing; purple: balancing + niching. The central line indicates the mean of 20 runs, whereas the surrounding ribbon indicates the uncertainty of the mean (standard error).

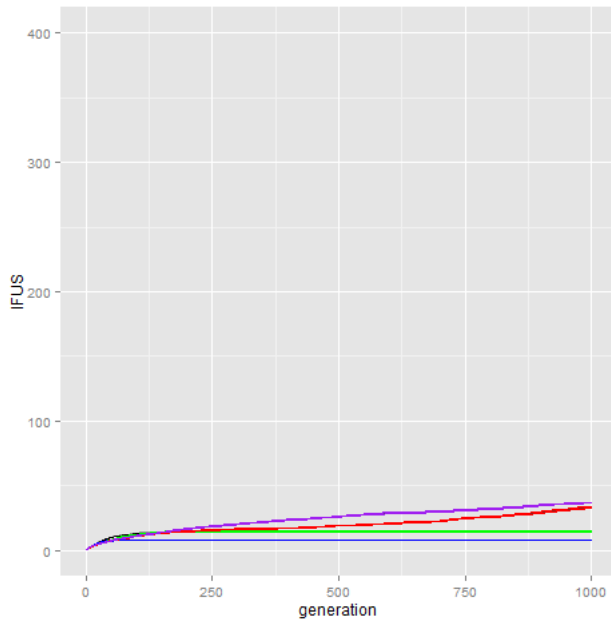

Figure 4: Complexification when constraints are not present and fitness selection is used. Black (hidden behind green): default; blue: niching; green: reduced resistance; red: balancing; purple: balancing + niching. The central line indicates the mean of 20 runs, whereas the surrounding ribbon indicates the uncertainty of the mean (standard error).

If the host population is not constrained and truncation selection is used (Fig. 5), a moderate growth of complexity is achieved with plain selection. Balancing leads to convergence (if applied with niching, the level is higher than if applied without niching). Niching significantly increases the final complexity. Reduced resistance leads to a much more dramatic increase of final complexity, reaching about $\frac{2}{3}$ of the final complexity in the constrained hosts case.

Taking a closer look at the reduced resistance method when applied with truncation selection in the unconstrained hosts case (Fig. 6) and in the constrained hosts case (Fig. 7), we find that significant complexification is achieved in a relatively wide parameter range, although the optima are at different parameter values, and that the parameter ranges where significant complexification is achieved are similar for very different task difficulty ratios.

Standard niching proves superior in terms of complexification when compared to "generalized niching" with other niching coefficients, as Fig. 8 reveals. Truncation selection was used with these experiments Because very strong selection pressure was found to be necessary for some of the experiments reported in [22], we also examined complexification when standard niching is used together with truncation selection with different selection strengths. As a result, it was found (see Fig. 9) that using a 5% selection threshold did not lead to better results than the 10% threshold that is used in all other experiments here, whereas using a 20% threshold lead to much worse

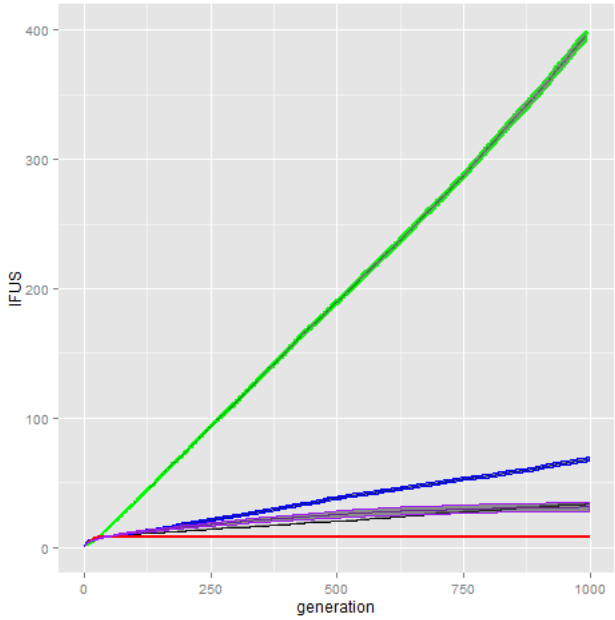

Figure 5: Complexification when constraints are not present and truncation selection is used. Black: default; blue: niching; green: reduced resistance; red: balancing; purple: balancing + niching. The central line indicates the mean of 20 runs, whereas the surrounding ribbon indicates the uncertainty of the mean (standard error).

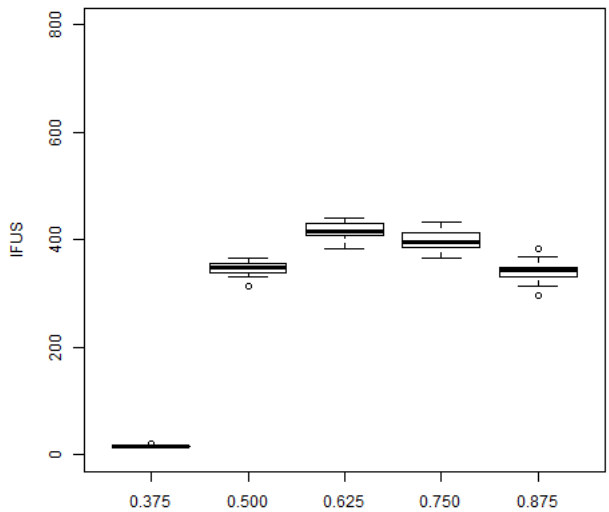

Figure 6: Complexity (IFUS) reached after 1000 generations without constraints when using reduced resistance with different coefficients.

results.

One might expect that niching increases the diversity in the population in the experiments reported here. Snapshots were taken at generation 200 for niching and no niching configurations using fitness proportional and truncation selection. It was measured for all generations back to the first how many of the individuals in that generation still had offspring in the final generation, in other words, the number of lineages that survived to generation 200 (Fig. 10). This shows that niching does not make a difference if the host population is constrained. If the host population is unconstrained, a higher lineage diversity is present transiently approximately between 20 and 60 generations back when niching is not used. When niching is used, there is no higher diversity than in the constrained hosts case.

4 Discussion

The goal of finding a method that leads to stable coevolutionary complexification even in the case of unequal task difficulties has been achieved. Reduced resistance achieves this if applied together with truncation selection. The results indicate that this method is relatively robust with respect to different task difficulty ratios (from 1:1 to 1:10) and resistance reduction factors. Niching together with truncation selection also achieves sustained complexification, although much slower. However, much remains unclear about why the examined stabilization techniques interact with the selection methods the way they do. For the simplest scenarios discussed here, it is possible that using analytical models could lead to further

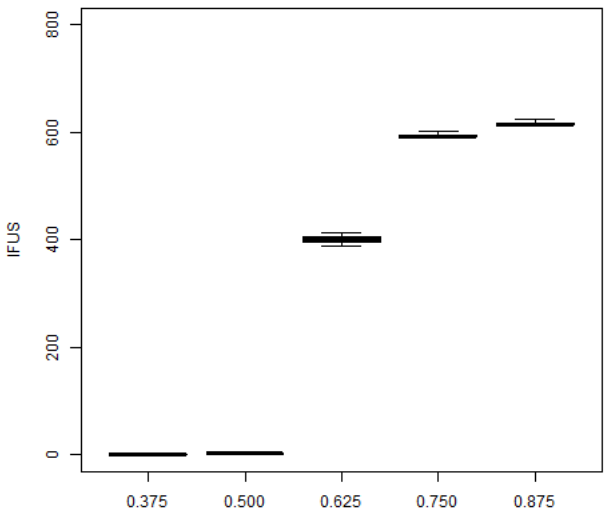

Figure 7: Complexity (IFUS) reached after 1000 generations with constraints when using reduced resistance with different coefficients.

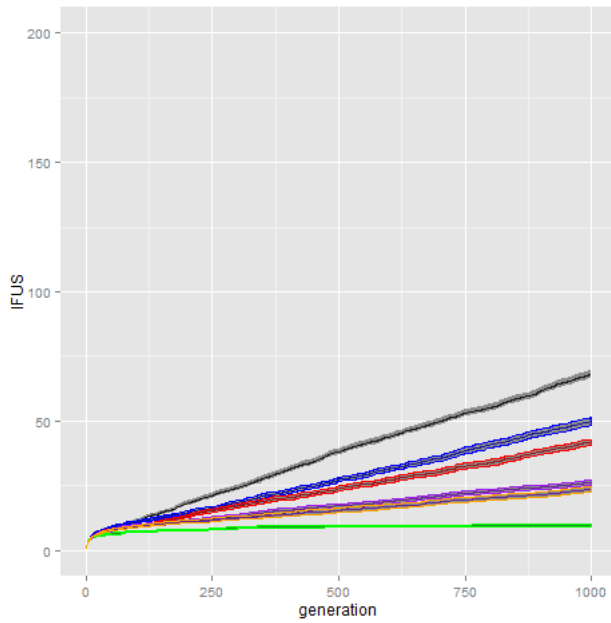

Figure 8: The influence of niching coefficients on complexification. Black: 1.0; blue: 0.5; green: 2.0; red: -0.5; purple: -1.0; orange: -2.0.

insights regarding that question.

While we have only looked at two extreme points in terms of task difficulty ratios here, previous research suggests that results for intermediate task difficulty ratios will also be intermediate between the results for the extreme cases presented here [22] in terms of both the final complexity achieved and the presence of linear growth of complexity.

It is also desirable to maintain a high diversity of solutions in the population both for practical (further adaptability) and theoretical (modeling niching and speciation) reasons. Standard niching does not achieve this goal with the studied task. In the future, further research on the influence of modified versions of niching on population diversity will be conducted. In this regard, one might expect that introducing spatial structure into the populations, either on its own or in combination with the methods studied here, will lead to much higher diversity.

The achieved rate of complexification can be compared to theories on the rate of evolution [28, 29, 30]. In the original configuration (constrained sequences, truncation selection, none of the methods for enhancing coevolutionary stability used), which is also one of the fastest complexifying, we have an IFUS of 641 after 1000 generations on average. Because each site can be one of 10 different numbers, the average increase of information content (algorithmic complexity) is $\mathrm{ld}10 \cdot \frac{641}{1000} \approx 2.1$ bit / generation (strictly speaking, it is slightly lower because the last few elements of the sequence are not converged in the population). Following Worden [29], for a selection strength of 10 (i.e., $\frac{1}{10}$ of the population is selected as parents and each has 10 offspring) we would expect a rate of less than $\mathrm{ld}10 \approx 3.3$ bit / generation. (Notice that the model assumes a fixed genome length. However, because we are applying mutations only at the end of the sequence per default, the part of the genome that is under active evolution is of constant size here, so we

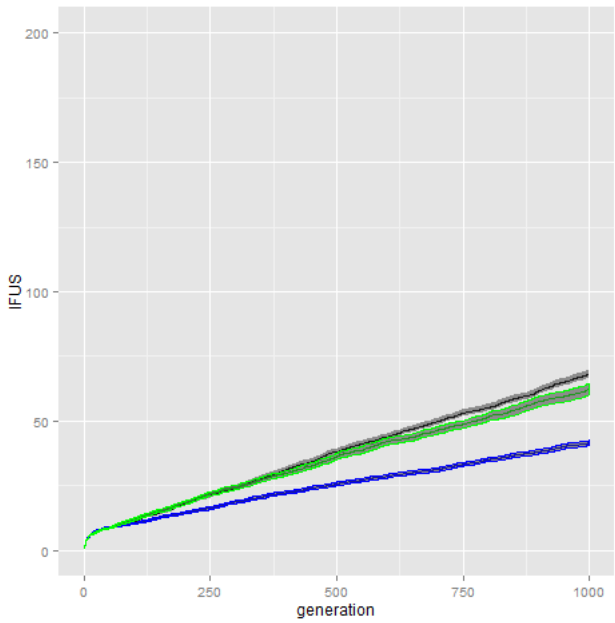

Figure 9: The influence on different selection thresholds on complexification when niching is used together with truncation selection. Black: 10%; blue: 20%; green: 5%.

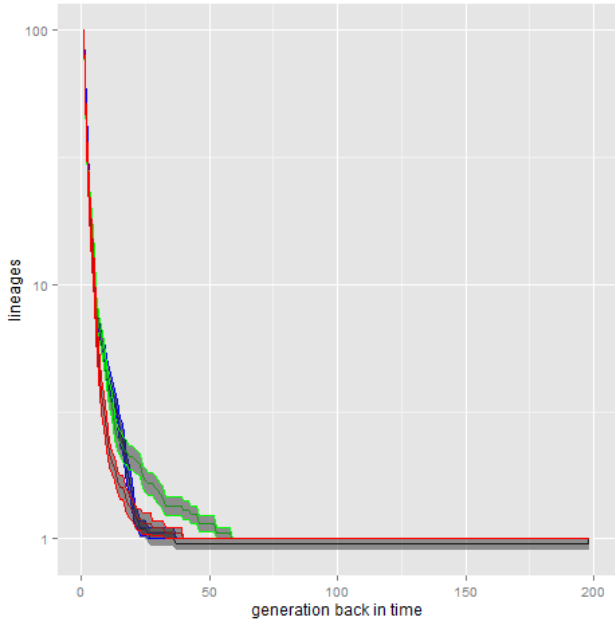

Figure 10: The influence of niching on lineage diversity. Black (hidden behind blue): constraints / no niching; blue: constraints / niching; green: no constraints / no niching; red: no constraints / niching.

expect the model to be valid for our case.) Furthermore, this theory predicts that the rate grows logarithmically with selection strength, but is not correlated to population size. As can be seen in figure 11, the rate grows less than logarithmically at higher selection strengths, and there is a weak correlation between the rate and population size for the examined range of parameters. As discussed in [29], there can be various factors in any but the simplest scenarios that prevent the theoretical speed limit from being reached. Nevertheless, the results indicate that Worden's theory may be useful to get a rough estimate of possible rates of evolution in this scenario and others, which may also allow investigating how various factors related to evolutionary operators and encodings influence the actual rates of evolution in future experiments.

The functional sequences in the genome are essentially equivalent to random sequences (see remarks in section 2.3), therefore we basically measure the algorithmic or Kolmogorov complexity of features involved in matches [31]. Of course, another fundamental question around the issue of complexification is whether the complexity concerned with the structural regularities can increase [32]. This questions is not addressed by our simulations. We regard the growth of algorithmic complexity of the genotype and phenotype as a necessary (but not sufficient) condition for the (co-)evolution of complex behaviors.

Obviously, sustained linear growth of functional complexity is a desirable goal for evolutionary robotics. Some previous research indicates that the results achieved here can indeed be transferred to evolutionary robotics provided that the representations and evolutionary operators for the controllers (in this case, neural networks) are adjusted accordingly [26].

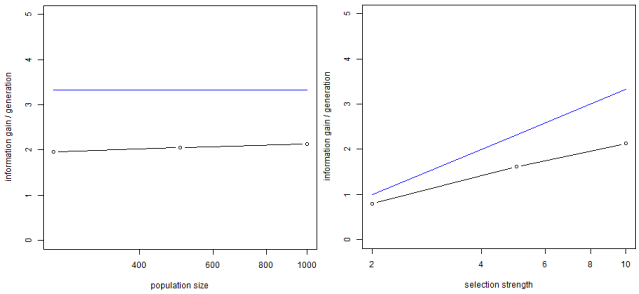

Figure 11: Information gain per generation (black) as compared to Worden's predicted speed limit for evolution (blue). Left: For different population sizes; Right: For different selection strengths.

Acknowledgment

This work was partially funded by the Bielefeld University programme "Bielefelder Nachwuchsfonds" (project "Coevolutionary complexification of autonomous agents").

References

[1] K. A. De Jong, *Evolutionary Computation — A Unified Approach*. MIT Press, 2006.

[2] R. Poli, W. B. Langdon, and N. F. McPhee, *A Field Guide to Genetic Programming*. Published via http://lulu.com, 2008. [Online]. Available: http://www.gp-field-guide.org.uk

[3] S. Doncieux, N. Bredeche, J.-B. Mouret, and A. E. Eigen, "Evolutionary robotics: what, why, and where to," *Frontiers in Robotics and AI*, 2015.

[4] N. Barton and L. Partridge, "Limits to natural selection," *BioEssays*, vol. 22, pp. 1075–1084, 2000.

[5] R. H. Thornhill and D. W. Ussery, "A classification of possible routes of darwinian evolution," *Journal of Theoretical Biology*, vol. 203, pp. 111–116, 2003.

[6] N. Williams and M. Mitchell, "Investigating the success of spatial coevolution," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2005.

[7] J. Cartlidge and S. Bullock, "Combating coevolutionary disengagement by reducing parasite virulence," *Evolutionary Computation*, vol. 12, pp. 193–222, 2004.

[8] J. Paredis, "Coevolution, memory and balance," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1999.

[9] C. D. Rosin and R. K. Belew, "New methods for competitive coevolution," *Evolutionary Computation*, vol. 5, pp. 1–29, 1997.

[10] S. Goings and C. Ofria, "Ecological approaches to diversity maintenance in evolutionary algorithms," in *Proceedings of IEEE Alife*, 2009.

[11] F. de Boer and P. Hogeweg, "The role of speciation in spatial coevolutionary function approximation," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2007.

[12] S. Nolfi and D. Floreano, *Evolutionary Robotics — The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, 2000.

[13] K. O. Stanley and R. Miikkulainen, "Competitive coevolution through coevolutionary complexification," *Journal of Artificial Intelligence Research*, vol. 21, pp. 63–100, 2004.

[14] R. Pfeifer and G. Gómez, "Interacting with the real world: design principles for intelligent systems," *Artificial Life and Robotics*, vol. 9, pp. 1–6, 2005.

[15] R. A. Watson and J. B. Pollack, "Coevolutionary dynamics in a minimal substrate," in *Proceedings of the Genetic an Evolutionary Computation Conference*, 2005.

[16] J. N. Thompson and J. J. Burdon, "Gene-for-gene coevolution between plants and parasites," *Nature*, vol. 360, pp. 121–125, 1992.

[17] J. Bergelson, G. Dywer, and J. J. Emerson, "Models and data on plant-enemy coevolution," *Annual Reviews of Genetics*, vol. 35, pp. 469–499, 2001.

[18] R. Dawkins and J. R. Krebs, "Arms races between and within species," *Proceedings of the Royal Society B*, vol. 205, pp. 489–511, 1979.

[19] B. Drossel, P. G. Higgs, and A. J. McKane, "The influence of predator-prey population dynamics on the long-term evolution of food web structure," *Journal of Theoretical Biology*, vol. 208, pp. 91–107, 2001.

[20] M. Maron and C. Fernando, "Food webs and the evolution of organism complexity," in *Artificial Life X Workshop Proceedings*, 2006.

[21] N. Guttenberg and N. Goldenfeld, "Cascade of complexity in evolving predator-prey dynamics," *Physical Review Letters*, vol. 100, p. 058102, 2008.

[22] B. Inden, "Open-ended coevolution and the emergence of complex irreducible functional units in iterated number sequence games," in *Proceedings of the 14th annual conference on genetic and evolutionary computation*, 2012.

[23] M. A. Bedau, E. Snyder, and N. H. Packard, "A classification of long-term evolutionary dynamics," in *Artificial Life VI*, C. A. et al., Ed., 1998.

[24] A. Channon, "Unbounded evolutionary dynamics in a system of agents that actively process and transform their environment," *Genetic Programming and Evolvable Machines*, vol. 7, pp. 253–281, 2006.

[25] C. C. Maley, "Four steps toward open-ended evolution," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 1999.

[26] B. Inden and J. Jost, "Neural agents can evolve to reproduce sequences of arbitrary length," in *Proceedings of the European Conference on Artificial Life*, 2013.

[27] B. L. Bassler, "How bacteria talk to each other: regulation of gene expression by quorum sensing," *Current Opinion in Microbiology*, vol. 2:6, pp. 582–587, 1999.

[28] M. Kimura, "Natural selection as the process of accumulating genetic information in adaptive evolution," *Genetical Research*, vol. 2(1), pp. 127–140, 1961.

[29] R. P. Worden, "A speed limit for evolution," *Journal of Theoretical Biology*, vol. 176, pp. 137–152, 1995.

[30] I. Harvey, "How fast can we evolve sometsome?" in *Proceedings of the European Conference on Artificial Life*, 2013.

[31] M. Li and P. Vitányi, *An introduction to Kolmogorov complexity and its applications.* Springer, 1997.

[32] N. Ay, E. Olbrich, N. Bertschinger, and J. Jost, "A geometric approach to complexity," *Chaos*, vol. 21, p. 037103, 2011.