

The Royal Family

Applying a taximetric cluster analysis as an investigation into the role of
plasticity in cyclic evolution

Master Thesis in Artificial Intelligence
Radboud University Nijmegen

Author:

R. Janssen¹

t.: +31 (0)628806191

e.: r.janssen@student.ru.nl

Supervisors:

dr. W.F.G. Haselager¹²

dr. S. Nolfi³

dr. I.G. Sprinkhuizen-Kuyper¹²

October 30, 2012

¹Department of Artificial Intelligence, Radboud University Nijmegen

²Donders Institute for Brain, Cognition and Behaviour, Nijmegen

³Institute of Cognitive Sciences and Technologies, National Research Council, Rome

Abstract

Co-evolutionary algorithms are able to simulate multiple species evolving in a shared environment. Previously, master tournaments have been employed to establish more accurate fitness measurements, in response to the Red Queen Effect. This study proposes to apply a taximetric cluster analysis to master tournament data. This allows to build a hierarchical ‘family’ tree, based on the phenotypes (i.e. behaviours) displayed during the master tournament. Using this approach, the study explores the following issues:

First, co-evolution often shows cycling dynamics, which might be related to phenotypic plasticity, i.e. cycling might promote plasticity while plasticity might suppress cycling. Using the cluster analysis, this study shows that a cyclic phase in evolution might indeed be superseded by a plastic phase. Furthermore, it was demonstrated that the cluster analysis can be used in further formalizing previously established results, such as that plastic individuals are able to cope with multiple rigid individuals.

Secondly, a state of pseudo-plasticity might be realized on a genetic level. This study proposes the existence of ‘switching genes’, which control the expression of dormant phenotypes. It is plausible such genes might play a role in cyclic phases of evolution as well, as they could enable a species to adapt quickly, without resorting to costly ontogenics. The study shows that cyclic phases are expectedly devoid of large genetic change. When randomly mutating families from these phases, it is possible this could trigger switching genes, causing a switch in opponent specialization. However, no such effect was clearly seen, possibly due to clouded data resulting from an indiscriminate mutation technique employed.

Keywords: Co-evolution, cluster analysis, phenotypic plasticity, genetic pre-adaptability.

1	Introduction	3
1.1	Co-evolutionary robotics	3
1.2	Research questions	4
1.2.1	Phenotypic plasticity	4
1.2.2	Genetic pre-adaptability	4
1.2.3	Cluster meta-analysis	5
1.3	Hypotheses	5
1.4	Structure	6
2	Background	7
2.1	Why evolutionary robotics?	7
2.2	Progress in co-evolution	8
2.2.1	Arms races, cycling and the Red Queen	8
2.2.2	The role of ontogenics on progress	9
2.2.3	Conclusion	12
2.3	Making sense of co-evolution	12
2.3.1	Difficulties in measurements	12
2.3.2	Visualizing progress	13
3	Methods	16
3.1	Overview	16
3.2	Experiment	16
3.2.1	Computing environment	16
3.2.2	Simulated environment, sensors and actuators	18
3.2.3	Neural network	20
3.2.4	Genetic algorithm	21
3.3	Analysis	24
3.3.1	Computing environment	24
3.3.2	Hierarchical cluster analysis	24
3.3.3	Mutation tournament	26

4	Results	28
4.1	Procedure	28
4.2	Classical measures	28
4.3	Cluster analysis	28
4.3.1	Results	28
4.3.2	Conclusion	32
4.4	Mutation tournament	33
4.4.1	Results	33
4.4.2	Conclusion	36
5	Discussion	37
5.1	Interpretation of results	37
5.1.1	Cluster analysis as a tool in ER	37
5.1.2	Plasticity and cycling	38
5.1.3	Hints for switching genes	39
5.2	Extended discussion	40
5.2.1	A needle in the haystack	40
5.2.2	A definition of behaviour	40
5.3	Conclusion	41
	References	43

1.1 Co-evolutionary robotics

Evolutionary robotics (ER) is the field in artificial intelligence concerned with creating adaptive robots by applying *evolutionary algorithms* (EAs) to obtain optimized software controllers (Section 2.1). An example of such an EA is a genetic algorithm (GA), which uses a static fitness function to measure performance (Section 2.1). However, in the natural world there is no such thing as a fixed fitness measure; animals are engaged in tight interactions with their environment and species are said to be *co-evolving*. Co-evolutionary algorithms (CEAs) can in that respect be considered a closer approximation to the natural world than simple GAs, simulating multiple species in a shared environment. However, CEAs are known to add an additional layer of complexity to the comprehensibility of EAs, since in a CEA a species' fitness is directly dependent on a co-evolving species' fitness. This fitness interdependency is known as the *Red Queen Effect* (Section 2.2.1).

From a biological perspective, co-evolution is often related to the emergence of *arms races* (Section 2.2.1). An anecdotal example of this can be demonstrated by asking the question why, for instance, some animals are so fast. The hypothetical existence of evolutionary arms races might explain this observation. For instance, both the leopard and gazelle might be so fast because they could have been pressuring each other on evolutionary timescales to either catch or outrun the other respectively. In more technical terms, one might call the evolution of the speed increase seen in the leopard or gazelle examples of *adaptations*. However, increasing speed is not the only option an animal might have. The gazelle for example might favour *protean behaviour* (i.e. quick, irregular movements) to evade the leopard over pure speed. Both running at speed and irregular movements can be seen as two different *capacities*, which of course both can be adapted further. An adaptation in that respect is thus a refinement of a capacity, while the development of a new capacity can be called an *innovation*. Both adaptation and innovation are examples of evolutionary *progress*.

Plausible as they might seem, the existence of arms races in nature have by no means been unambiguously empirically verified, and little is known on the specifics. A similar state of affairs can be found in computational simulations running co-evolutionary scenarios, where experiments have demonstrated that the emergence of an arms race does not a-priori follow from interspecific competition. Instead, species in simulation are often observed to evolve in a *cycling* pattern,

a ‘rock-scissors-paper’-like scheme that might lead to short term benefits but as far as current understanding goes, does not result in long-term progress (Section 2.2.1).¹

1.2 Research questions

1.2.1 Phenotypic plasticity

Research Question 1 *Can phenotypic plasticity suppress cycling?*

One significant factor in the emergence of arms races might be the ability of many animals to develop *phenotypic plasticity*, such as learning. Plasticity has since the dawn of evolutionary thought been a source of confusion as well as a catalyst in its theoretical development. In fact, the idea that learned experiences and skills could be a unit of hereditary transmission at one time was considered a valid alternative to Darwin’s natural selection, and while Lamarckism in its original form is now refuted, the influence that ontogenics might have on the evolution of species is still in need of further research.

An explanation of how species might cope with a rapidly alternating (e.g. cycling) environment, is by evolving to a state of ontogenic predisposition (Chapter 2). Consider for example an unspecified primate. When compared to less developed organisms (such as reptiles), primates are slower to mature and become self-sufficient. They are born as the proverbial ‘blank slate’, but are quicker to learn new skills and to adapt to changing circumstances. During its lifetime, learning allows the primate to explore its phenospace in ways that would take generations for natural selection to accomplish.

Interestingly, this might imply that evolving to a state of plasticity could act as a normalizing factor in the cycling/progress trade-off; when a rigid (i.e. non-plastic) species is being outperformed by a plastic opponent, this could force the rigid species to evolve plastic individuals as well. In fact, the presence of cycling might very well be one of the main catalysts for adaptable behaviour to emerge. Since a cyclic phase in evolution might apply a selective pressure on species to adapt quickly, whether genetically or ontogenically, it might eventually favour the emergence of the latter (Section 2.2.2).

1.2.2 Genetic pre-adaptability

Research Question 2 *Are there signs of genetic ‘pre-adaptability’? If so, what is its role with regard to cycling?*

Certain degrees of adaptability might not be exclusive to phenotypic plasticity. Plasticity might enable an organism to quickly ‘switch’ from one (set of) capacities to another, as circumstances require (Section 2.2.2). We speculate that this switching could possibly also be realized by what one might call *genetic pre-adaptability*.

Consider how it might seem intuitive that genetically determined capacities are controlled by a set of alleles, and that the expression of a different capacity would depend on an alternate set of alleles. Under this assumption, to evolve from one capacity to another many genes might have to be changed. However, this study proposes that a genetically *pre-adapted* organism would contain multiple sets of dormant genes that are each responsible for expressing different capacities. A small number of dedicated *switching genes* could possibly control which of these capacities gets expressed.

¹Note that this analogy illustrates a forced instance of cycling, unlike what is naturally occurring in evolution where there is a plethora of alternative strategies, including possible phenotypic modifications of those.

While such a mechanism would not allow phenotypic plasticity, genotypic pre-adaptability could enable a species to adapt to scenarios from a broad spectrum of environmental circumstances within only a couple of generations. If it exists, it is plausible that cycling phases in evolution have a basis in pre-adapted genetics. Consider that cycling challenges a species with different types or ‘families’ of opponents in an alternating fashion. To counter an opponent like this, species might carry the innate capacities required to cope with each of these types, such that they get expressed by switching genes when required. This then gives the impression of a species ‘re-discovering’ previously lost capacities.

1.2.3 Cluster meta-analysis

Research Question 3 *Can a cluster analysis be used to formally define phenotypic ‘families’ of individuals?*

Unfortunately, co-evolving systems are notoriously difficult to understand, especially when taking ontogenics into account. This difficulty can primarily be attributed due to the Red Queen Effect, suggesting that a direct measure of a species’ fitness is not sufficiently informed and that alternative measures have to be used (Section 2.3.1).

One well-known example of these alternatives is measuring an individual’s fitness against a set of opponent elites from previous and future generations. These *master tournaments* can then be visualized in the form of master fitness graphs or CIAO (Current Individual versus Ancestral Opponents) plots (Section 2.3.2). A master fitness has the benefit of being a more reliable and objective means to measure progress as compared to when using classical ‘online’ fitness plots, while CIAO plots can be used to investigate cycling/progress dynamics. However, CIAO plots have the drawback that they often appear convoluted (Section 2.3.2).

This study suggests that the master tournament can be regarded as a formal representation of the behaviours displayed from both species. Once formalized, the similarity between two (groups of) behaviours could constitute a distance measure between them. This formalization can be used to group individuals into behaviourally based ‘families’, by using a hierarchical cluster algorithm (Section 3.3.2). This allows to construct a phenetic tree which enables the inspection of local dynamics, in order to better investigate Research Questions 1 and 2.

1.3 Hypotheses

To answer the research questions outlined in Section 1.2, this study proposes to formalize the concept of behaviour in the context of a simulated *predator-prey* scenario. Using the cluster analysis mentioned in Section 1.2.3, species’ performances can be represented and ordered more intuitively, by evaluating only a small number of phenotypic families against a small number of opponent families in an automated fashion, instead of having to inspect more complex, convoluted measures by hand. To the author’s knowledge, this approach is novel.

It has been shown before that species are able to exploit phenotypic plasticity to counter rigid opponent types (Section 2.2.2). A state of plasticity seems thus a valid and possibly superior alternative to a cycling strategy. By testing species’ families against each other, it can be expected that plastic families are able to cope with a higher number of rigid opponent families, than when evaluating rigid families against each other. Moreover, one could expect that plastic families will originate from non-cycling phases in evolution, while rigid individuals will from cycling ones.

While not having been directly observed in nature or in simulation, it does not seem unreasonable that the evolution to a state of pseudo-adaptivity might be realized on a genetic level,

Family	cycling _a	cycling _b	non-cycling
A	0.8	0.2	0.7
B	0.2	0.8	0.7

(a) Idealized legacy performance.

Family	cycling _a	cycling _b	non-cycling
A	0.2	0.8	0.7
B	0.8	0.2	0.7

(b) Idealized performance after mutation.

Table 1.1: An example of the highly idealized results one might expect when switching genes exist. Shown are the success-rates of phenotypic families from different phases (cycling and non-cycling) against two opponent families from cycling phases (A and B). The cycling families are only effective against one opponent family. When mutated, the cycling family expresses a different phenotype, and reverses its specialization. The plastic individuals are not susceptible to genetic mutation and show an overall higher performance against both opponent families.

especially when a species is situated in an environment where historical challenges are continuously resurfacing (i.e. when confronted with a cycling opponent). The switching genes mentioned in Section 1.2.2, if they exist, could thus be expected to be found in the families originating of cycling phases of evolution. More interestingly, such genes would be susceptible to physical mutation. Therefore, if one would mutate an individual from a genetically pre-adapted family and re-evaluate its performance, the individual's phenotype might switch from expressing one set of capacities to the next, thereby also switching its specialization to cope with a certain opponent. Conversely, such mutations should have less effect when applied to individuals that originate from families that have not been genetically pre-adapted (e.g. from non-cycling phases). Table 1.1 summarizes these expectations.

1.4 Structure

To investigate the hypotheses just discussed, the rest of this thesis is arranged as follows. Chapter 2 provides the reader with a more elaborate theoretical background on the topics introduced in this chapter. Moving from abstract and general topics to more applied and specific ones, this includes a brief, general outline of ER (Section 2.1), a discussion of the interactions between progress, cycling and plasticity (Section 2.2) and an exposition on the details of the master tournament (Section 2.3). Chapter 3 describes the methods employed to investigate the raised research questions, whose results are illustrated in Chapter 4. Finally, the results are discussed in Chapter 5.

2.1 Why evolutionary robotics?

The strength of evolutionary robotics comes from the fact that in many ways it takes a step back in robot design; instead of a *distal* perspective (i.e. from the point of view of a designer), a more *proximal* one (i.e. from the robot's perspective) is taken (Sharkey & Heemskerk, 1997). For example, when a human designer might want a robot to pick up a ball and drop it in a specific area, he or she might intuitively formulate this expected behaviour as a collection of goals the robot has to accomplish. However, interactions between robot and environment can be difficult to fathom. Difficulties in understanding real-world dynamics might not only be a limiting factor in the accuracy of a distally designed model, a human designer might also inadvertently impose unnecessarily and possibly unwanted constraints on robot design (Nolfi & Floreano, 2000). ER on the other hand might be less constricted by distal descriptions. By reverting to principles that are found in natural selection, which are essentially blind and self-regulating, EAs used by ER could design robot controllers with comparably little external intervention.

Of course, EAs in general have their own difficulties concerning design and fine-tuning. In a GA for example (the technique used in this study; Section 3.2.4), a problem is defined in close relation to a *fitness function*, that measures (possibly heuristically) how well a solution solves that problem. When running a GA, one first initializes a pool (or *population*) of candidate solutions (or *individuals*). These individuals are numerically encoded by a *genotype*, that can be easily modified. To evaluate an individual, its genotype is transformed into a *phenotype*. These phenotypes are evaluated using a *fitness function*, and the best scoring ones are selected to seed a new *generation* of individuals. Generating the next generation (of *offspring*) is done on the genetic level, by means of the *mutation* and/or *recombination* operators. The process of selecting, altering and evaluating only the fittest individuals is repeated until a termination condition is reached. It goes almost without saying that all these steps require careful parametric fine-tuning (Eiben & Smith, 2008).

The potential of using an EA as the one just described becomes apparent when comparing its solutions to the solutions from classical engineering approaches. For example, in (Nolfi, 1996) a robot was evolved to stay close to a cylindrical object. Other than the EA yielding desired robot behaviour, this behaviour did not lend itself well to distal descriptors. Robot behaviour appeared to be based on emergent sensori-motor equilibria, called *behavioural attractors*. It is not to say a

classical approach could have had no success in letting a robot approach a cylindrical object, but it would probably have no basis on behavioural attractors. Moreover, ER can not only be used to design efficient robots, but in doing so one is also able to study the mechanisms underlying evolution in the natural world.

Co-evolutionary algorithms display some additional points of interest that regular EAs lack. One notable advantage of a CEA is that in simple EAs, it is often difficult to formulate the fitness function adequately such that it provides an appropriate challenge throughout the different phases of evolution. If the fitness function is either too easy or too hard, the genetic search space could appear homogeneous to the otherwise uninformed algorithm. For instance, an EA is often unable to solve a hard problem by simply applying a ‘suitably’ hard fitness function. The EA will not be able to progress beyond the initial stages of evolution, since at that point all candidate solutions are evaluated as equally inadequate.

As a solution to this *bootstrap problem*, one might use a series of fitness functions of an increasingly demanding nature; the further the algorithm progresses, the more demanding the fitness function can (and should) be (these compound fitness functions are known as *incremental fitness functions* (Urzelai, Floreano, Dorigo, & Colombetti, 1998)). Co-evolution offers another, more implicit alternative: A CEA could potentially be able to exploit the self-scaling properties that arise between co-evolving elements (Angeline & Pollack, 1993). Not only could this kickstart evolution, but it might even lead to long-term progress beyond the bootstrap problem.

2.2 Progress in co-evolution

2.2.1 Arms races, cycling and the Red Queen

In a simple EA where a static fitness function is used, one can expect to see a gradual fitness progression. In contrast, co-evolution knows no fixed metric that evaluates an individual’s performance objectively. Instead, a species’ fitness is completely dependent on and relative to its environment, which is itself also (partly) subject to natural selection. This implies that the fitness of two species is often directly related (Kendeigh, 1961). For instance, in a mutualistic interaction between species, this relation is symmetric (e.g. the clownfish and the sea anemone), while in a competitive (e.g. lions versus hyenas) or antagonistic (e.g. cheetahs versus gazelles) interaction the relations would be inverted. In the latter two cases, observe that one species’ progress is another one’s setback. Thus, when both species are progressing at an equal rate, those improvements will *ceteris paribus* cancel each other out. This observation is known as the Red Queen Effect (Van Valen, 1973).¹

One might suspect that the Red Queen Effect can be prolonged indefinitely. This could eventually lead to the formation of evolutionary arms races. However, one must realize that an arms race need not always be a long-lasting, incremental one. In (Dawkins & Krebs, 1979) it was already recognized that there are many forms of arms races, and that they could terminate relatively quickly. For instance, it was postulated that while conspecific symmetrical arms races (i.e. a well-balanced arms race between members of the same species) could lead to incrementality, this would not be something to be expected when observing interspecific competition. In the latter case, it was speculated that an individual taking step to decrease an interspecific opponent’s survivability would simultaneously increase the survivability of a conspecific ‘opponent’. This view however assumes that natural selection acts on the level of the individual (or even on that of the gene (Dawkins, 1976)), while there has been a recent resurgence of interest in group selection (Wilson & Wilson, 2008).

¹The Red Queen effect is named after Lewis Carroll’s *Through the Looking Glass*, where the Red Queen character has to keep running to just remain stationary.

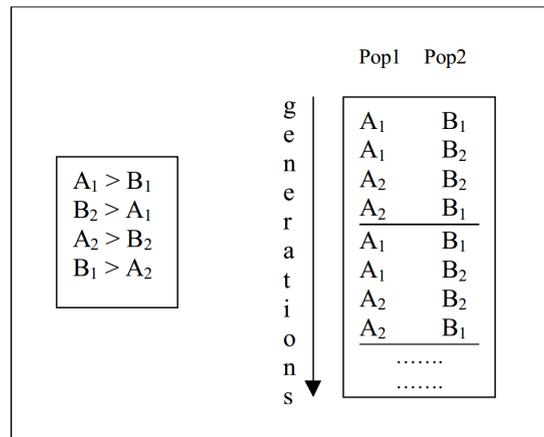


Figure 2.1: A more abstract representation of cyclic evolution. The right box shows two species (‘Pop1’ and ‘Pop2’) cycling through their two respective strategies in alternating fashion. The left box shows the prevalence of each interspecific combination of strategies (from (Nolfi & Floreano, 1998)).

Regardless of the exact form of the arms race at hand, it should eventually terminate in one of three possible outcomes (Dawkins & Krebs, 1979). First, one species may win the arms race, for example by causing the opponent to go extinct. Secondly, an arms race might end in an equilibrium (e.g. a virus being sufficiently virulent and a host being sufficiently resistant such that neither will perish, nor will prevail over the other). Finally, an arms race may end in *periodic cycling*, an example of which can be seen in conspecific parental investment (Parker, 1979).²

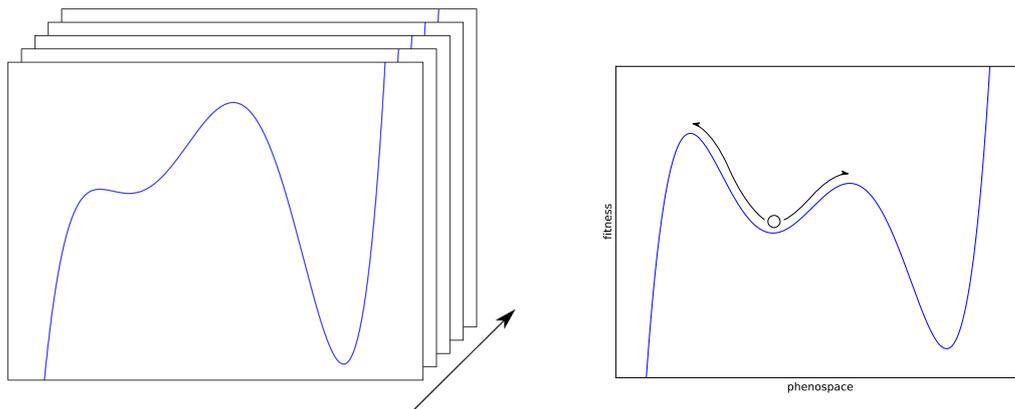
In simulation, the nature of arms races is often investigated in the context of predator-prey simulations (Section 2.3) (Cliff & Miller, 1995a; Miller & Cliff, 1994). Notably, Nolfi and Floreano (1998) illustrated a more abstract generalization of cyclic evolution (Figure 2.1).

2.2.2 The role of ontogenics on progress

In the context of co-evolving predator-prey robotics, Floreano, Nolfi, and Mondada (2001) have shown that phenotypic plasticity might enable an individual with enhanced performance. The experiment investigated three scenarios, all with neural network-based controllers with evolvable connection weights; a genetically determined one, one with evolvable noise applied to connections, and one with evolvable Hebbian rules (Hebb, 2002). It was demonstrated that both predator and prey benefited from plasticity; prey took advantage of the noisy controllers by demonstrating protean (i.e. unpredictable) behaviour, while predators were able to exploit Hebbian learning, resulting in superior performance against all three types of prey opponents. The authors conclude by speculating that ER should not focus on evolving to a state of optimality, but to adaptivity.

Evidently, phenotypic plasticity might enable an organism to deal better with a dynamical environment, but what is its relation to evolutionary progress? One well-known proposed mechanisms regarding this interaction is the *Baldwin Effect* (Baldwin, 1896). Generally, the effect postulates that 1) species might evolve plasticity, followed by 2) the genetic assimilation of those

²Offspring might evolve a ‘conflictor gene’ that exploits parental attention. Parents might evolve a ‘suppressor gene’ that allows them to invest in each offspring equally, regardless of the conflictor gene. Since both genes have associated costs, the conflictor gene will slowly diminish (since it is ignored), following by the extinction of the suppressor gene (since it is no longer needed) and the cycle repeats.



(a) The genospace is explored by the GA (in the direction of the arrow). Each slice represents a (snapshot of a) dynamic fitness landscape (i.e. a phenospace) that an evolved individual is confronted with during its life.

(b) A plastic organism might be able to dynamically move around the phenospace during its lifetime (the arrows) and explore high-fitness regions. Once at a (stable) local optimum, the acquired traits might be genetically assimilated.

Figure 2.2: The different ways (genetically and ontogenically) a genospace/phenospace might be explored. Note that these plots represent ever-changing fitness landscapes as simple snapshots; in an actual simulation they would continuously morph and warp as environmental pressure changes. Also note that the distinction between genospace and phenospace is not as clear as is suggested here, since the phenospace is indirectly accessible through the genospace.

traits ontogenically acquired. Now, the question arises why species have to go through a plastic stage, to arrive again at a rigid, genetic stage. Why not simply go from a genetically determined trait to an alternative genetically determined trait? There has to be some benefit and cost associated with both stages.

Emergence of plasticity

In (Turney, 1996) the first part of this question is elegantly summarized. To simplify, there exists a trade-off between phenotypic plasticity and rigidity that might be compared to the trade-off between flexibility and robustness. The phenospace neighbourhood of an organism would have various fitness associations, possibly with large differences. A plastic individual would be able to move to those positions that yield the highest fitness, contrary to rigid individuals. In effect, phenotypic plasticity thus ‘smooths’ the fitness landscape (Figure 2.2).

In (Hinton & Nowlan, 1987) an experiment was conducted that convincingly demonstrated the smoothing influence of plasticity in a simulated environment. Suppose environmental circumstances would require one binary weighted neural network with 20 specific connections to be the right and only right one. The phenospace of this scenario would look a completely flat surface of size 2^{20} , with one position sticking out (the figurative needle in the haystack).

The study applied a GA to a population of 1000 randomly generated candidate networks. In a rigid scenario, the network would be encoded by 20 genes that could either disable or enable a connection. In the plastic scenario, genes had a 0.25 chance to either disable or enable a connection, and a 0.5 chance to let it be undetermined (i.e. subjected to plasticity). Attempting to use 1000 rigid networks would revert to random search, since there is no indication whether a candidate solution is close to the global optimum. However, when enhancing the individuals

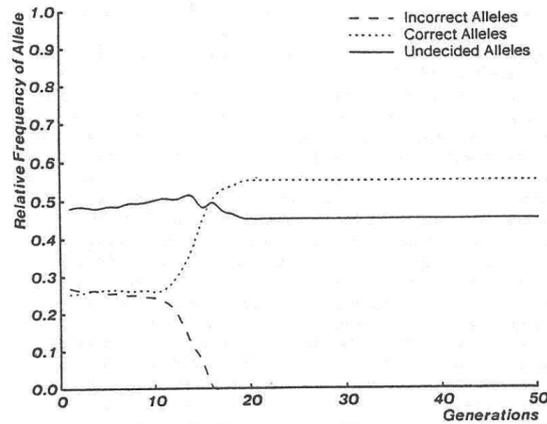


Figure 2.3: The results from (Hinton & Nowlan, 1987) show that plasticity enables evolution to internalize beneficial traits. Note that after 20 generation, more than 50% of the genes are correctly set, in a phenospace of size 2^{20} . Also observe that the number of fixed genes never rises over 60%, implying that in this experiment full rigidity is unneeded.

with plasticity (i.e. the undetermined connections would randomly flip 1000 times; if the needle would be found, the trial would end), the GA would produce individuals that had more than fifty percent of the genes correctly set after 20 generations. This indicates plasticity is able to guide evolution in certain cases (and that acquired traits can be genetically internalized).

In (Nolfi & Floreano, 2000) it was speculated how *cycling* might play a facilitating role in the emergence of plasticity. First, cycling could be expected to appear when there are no general strategies available. Instead, a species would resort to quickly switching from one partial strategy to the next (cycling). This would at the same time create an amplified conspecific selective pressure; those individuals who are able to adapt quickly would be favoured. As ontogenic adaptation is potentially much faster than genetic adaptation, individuals with a predisposition to learn would start to emerge. For instance, in linear evolution there might only be a need to adapt to the most recent opponent, while not preserving adaptations to historical opponents. In cycling, the perception of rapidly alternating types of opponents might create a selective pressure to exploit plasticity, or more speculatively; genetic pre-adaptability.

Genetic assimilation

So, why would, as shown in (Hinton & Nowlan, 1987), evolution favour rigidity after a plastic phase? Basically, the benefits of phenotypic rigidity would be that it requires no resource investments and corresponding risks associated with plasticity (Turney, 1996). Moreover, if environmental factors are relatively steady, they could be considered trustworthy and traits associated with them can be internalized. Still, the specifics on internalization are far from clear up until this point.

Godfrey-Smith (2003) gave three interpretations (that are not mutually exclusive) of the internalization phase of the Baldwin Effect. The *breathing space* interpretation, the one originally proposed by (Baldwin, 1896), supposes internalization arises because plasticity raises survivability. Thus, plastic individuals would have a higher chance to produce offspring that, through mutation and recombination, have those traits to survive internalized.

The *canalization* interpretation is derived from an idea proposed by Waddington (1942).

Canalization can be thought of as a predisposition to ontogenically develop some trait, with a lessened sensitivity to environmental circumstances. The hypothesis is that in some cases, the need for canalization of a trait can be so big that its development will happen almost without exception. In a later experiment, this hypothesis was backed up by empirical data (Waddington, 1953). Likewise, a predisposition to *learn* a certain trait might become so strong that the trait itself would be assimilated, as in (Hinton & Nowlan, 1987).

Finally, *niche construction* interprets internalization as a result from shifts in the social environment of a population when transitioning to a plastic phase (Deacon, 1997). The population that now occupies a new social niche are subject to additional conspecific social selective pressure, eventually favouring internalization of those traits that are beneficial to compete in the niche.

2.2.3 Conclusion

Co-evolution does not always lead to incremental arms-races, but can also lead to cycling dynamics. Furthermore, the Baldwin Effect plays an intricate role regarding the interaction between ontogenics and genetics, which is of importance when one considers this study's hypothesis that plasticity might suppress cycling. First, the exploitation of plasticity can be expected, as it can a) accelerate evolution and b) enhance a species' coping effectiveness, especially when evolution is going through a cyclic phase. Secondly, when ontogenic traits are being internalized, plastic phases might eventually consolidate into rigid phases. It goes without saying that internalization might in turn influence evolutionary progress; while plastic phases might suppress cycling, rigid phases may again allow for them. This could lead to the formation of higher order cycling harmonics, where not only fundamental cycling is present, but cycling/non-cycling phases are themselves cycling.

2.3 Making sense of co-evolution

2.3.1 Difficulties in measurements

Consider the Red Queen Effect: Imagine a species A and B evolving in a shared environment, where A's fitness is (partly) dependent on B's, and vice versa. How could one measure a species' fitness? Say there would be an increase in A's fitness at some point. Would this be due to A having made progress, or due to loss of competence of B? Suppose one would observe a stagnation of both species' fitness. Does this simply imply progression has stagnated as well, or is one unable to measure symmetrical development? How would one know which species performs better?

An intuitively insightful approach to answer these questions would be to observe behaviour directly (in simulation). There are however some caveats when choosing to this approach.

- There is an inherent stochasticity in evolution and the resultant behaviour.
- The sheer number of combinations between all individuals between species is overwhelmingly large.
- Even when just looking at elites or top-x individuals in order to prevent a combinatorial explosions, these truncations might not be representative of a species as a whole. In the case of the presence of *sub-populations*, different types of elites might be promoted to the top layers by little more than chance.

- Behaviour, especially in the case of anything displaying more than purely reactive intelligence, can be very context-dependent. Some types of behaviour that might emerge against one opponent might be completely absent against another.

Because of these difficulties, the prevalence of whose are often hard to predict in advance, distal observations can in certain cases be supplemented or replaced with more advanced metrics.

An influential paper by Cliff and Miller (1995b) introduced some alternatives to measuring a species' fitness in situ, of which the so-called master tournament is of particular interest. The master tournament can be regarded as an offline metric, i.e. one that is obtained after the EA has been terminated. The master tournament is based on evaluating all a species' elites against each other. Thus, if an experiment is based on s species and i individuals per species, the master tournament would represent i^s evaluations; every elite is evaluated against all its opponent's elites. The advantage here is that, since an individual is tested against past, present and future opponents, the master tournament's fitness can be interpreted as a measure of long-term fitness, or robustness. For example, if an individual would be able to defeat all past, or perhaps even future opponents, it would now be justly measured as fitter than one who would only be able to defeat, say, a fraction of its past opponents.

2.3.2 Visualizing progress

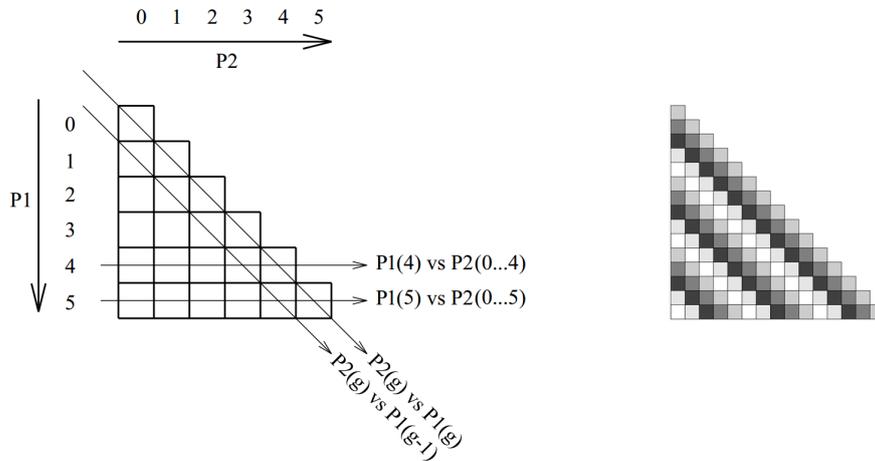
The master tournament is often visualized as a total fitness score, but perhaps more appropriately, a CIAO plot; an s -dimensional (in practice always 2-dimensional) graphic (Figure 2.4a).³ In the CIAO plot, each interspecific evaluation is presented by a voxel; the lightness or colouration of the voxel indicates the fitness score that was obtained during the evaluation (Figure 2.5).

In an ideal case, the CIAO plot would be diagonally bisected (Figure 2.5d), showing that all individuals are able to beat all opponents from previous generations, but none from future ones, implying that every new generation shows significant progress to the point where they are superior to all ancestors. In contrast, cyclic evolution would show up as diagonal banding (Figure 2.4b). In real-world applications however, these patterns are rarely to never seen (Figures 2.5a, 2.5b and 2.5c).

In Nolfi and Floreano (1998) a predator-prey scenario was evolved which provides some typical examples of what a practically obtained CIAO plot might look like. Here, two robot populations were evolved by applying a GA that modified a neural network's connection weights (similarly to this study; see Chapter 3). Figure 2.5a shows a scenario that seemed to show the predator and prey species being engaged in cycling, as illustrated by the perpendicular banding in the CIAO plot. In a second scenario, *Hall-of-Fame Selection* (Rosin & Belew, 1997) was used (i.e. individuals are evaluated against elites from previous generations *during* the GA), resulting in sudden and abruptly emerging innovation that showed up in the CIAO plot as a distinctive 'staircase' pattern (Figure 2.5b). Finally, in a third scenario, the prey robot's sensors were reconfigured to provide it with a 'richer' sensory input. Both species' evolution now showed a gradual progression, observable in the CIAO plot as a noisy gradient (Figure 2.5c) that is vaguely reminiscent of the ideal case plot (Figure 2.5d).

It has been suggested that the CIAO plot is less than clear when the cyclic nature of evolution is irregular (Cartlidge & Bullock, 2004). This is evident from the fact that most scenarios where cycling is suspected do not show the diagonal banding pattern (Figure 2.4b) but the perpendicular

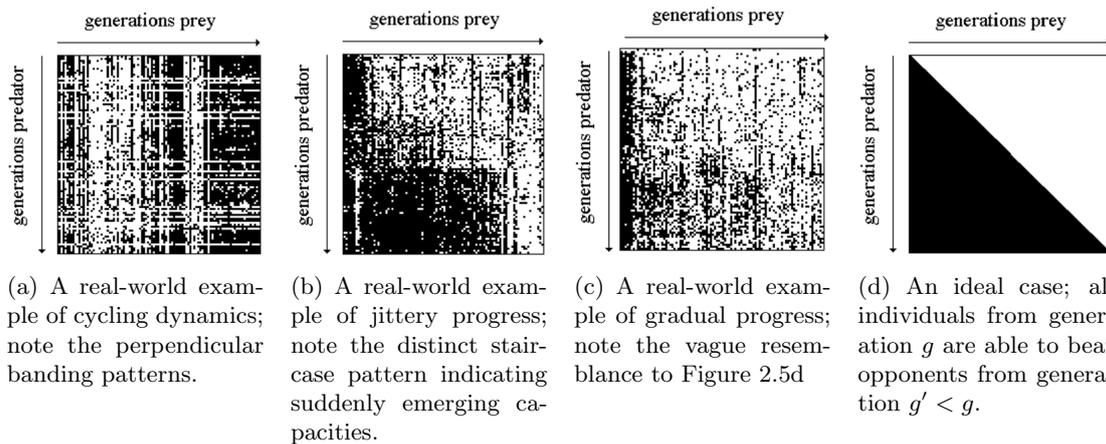
³In the original incarnation of the CIAO plot by (Cliff & Miller, 1995b), the plots were true to their name (Current Individual versus *Ancestral* Opponents) and were visualized as two triangular plots; one for each species. More recent versions of the plot often combine both halves for both species in one rectangular graphic that in effect also shows the performance of an individual against *future* opponents. For sake of simplicity, this study refers to both as simply 'CIAO plot'.



(a) A CIAO plot displayed schematically. Each axis denotes a species. The perpendicular and diagonal slices from the plot relate to each other in terms of the generations from which species' elites were obtained.

(b) An ideal case of cyclic evolution. Individuals from generation g are alternately successful against previous generations from the opponent species.

Figure 2.4: Two stylized examples of a 2-dimensional CIAO plot (adapted from (Cliff & Miller, 1995b)).



(a) A real-world example of cycling dynamics; note the perpendicular banding patterns.

(b) A real-world example of jittery progress; note the distinct staircase pattern indicating suddenly emerging capacities.

(c) A real-world example of gradual progress; note the vague resemblance to Figure 2.5d

(d) An ideal case; all individuals from generation g are able to beat opponents from generation $g' < g$.

Figure 2.5: Examples of CIAO plots. Pixel lightness indicates the fitness obtained during evaluation (from (Nolfi & Floreano, 1998)).

one (Figure 2.5a). An explanation is that the perpendicular CIAO plot is indeed a result of cycling, but due to complex ‘layers’ of cycling, this could result in plots that show interference and are far from intuitive to understand. In Cartlidge and Bullock (2004), it has been shown that a meta-analysis on the CIAO plot might prove to be fruitful; applying image processing algorithms on the perpendicular CIAO plots could reveal the obfuscated diagonal patterns to some degree again.

Another potential for a CIAO meta-analysis can be found in the bioinformatics community, where cluster algorithms have found a wide application with regard to various domains (Xu & Wunsch, 2005). Surprisingly, ER makes little use of these proven techniques, while the fields can have significant overlap. Notably, the use of taximetrics (i.e. the classification of organisms based on their phenotype (Sneath & Sokal, 1973)) could be just as suitable to categorize artificial life as it is to natural life. In fact, some hierarchical cluster algorithms have been specifically designed for this purpose (Sokal & Michener, 1958).

3.1 Overview

This study investigated the research questions outlined in Chapter 1 utilizing two e-puck robots (Mondada et al., 2009) in simulation, one fulfilling the role of ‘predator’, the other one of ‘prey’. The robots’ controllers were neural network-based (Section 3.2.3), and its weights were evolved using a GA (Section 3.2.4). Each scenario was replicated 10 times (as ‘Seeds’) with pseudo-random initial conditions.

A series of pilot studies were conducted to determine the most suitable algorithmic parameters and to refine the more conceptual design choices. Once finalized, the actual scenarios were simulated (Section 3.2.4). A cluster analysis on the master tournament data was applied by implementing a variant of the UPGMA/WPGMA ((Un)weighted Pair Group Method with Arithmetic mean) algorithm (Sokal & Michener, 1958) (Section 3.3.2). This analysis was designed to summarize and visualize cycling/progress dynamics for each species on a local level (Figure 3.1). The analysis could then be used to select interesting seeds and families for further investigation (Figure 3.2). For example, a transition from cycling to non-cycling dynamics could indicate the emergence of plasticity (Section 2.2.2, Figure 3.1).

To test whether it might be genetic pre-adaptability that enables cycling families to quickly switch from one capacity to the next, the cycling families’ performance was aligned and compared to ‘genetic bitmaps’ (a visualization that shows genetic change) (Cliff & Miller, 1995b). This served as a first check if phenotypic change is accompanied by large or little genetic change (the latter being a likely case if switching genes exist). A ‘mutation tournament’ (i.e. a master tournament variant where the tested individuals are subjected to random genetic mutations (Section 3.3.3)) was used in order to more precisely investigate the existence of switching genes.

3.2 Experiment

3.2.1 Computing environment

The simulation software was derived from the Evorobot* codebase (Nolfi & Gigliotta, 2010), modified to support multiple populations and provide more elaborate data recording (Figure 3.3). The code was compiled for testing using the Microsoft Visual C++ compiler v4.0.30319 under

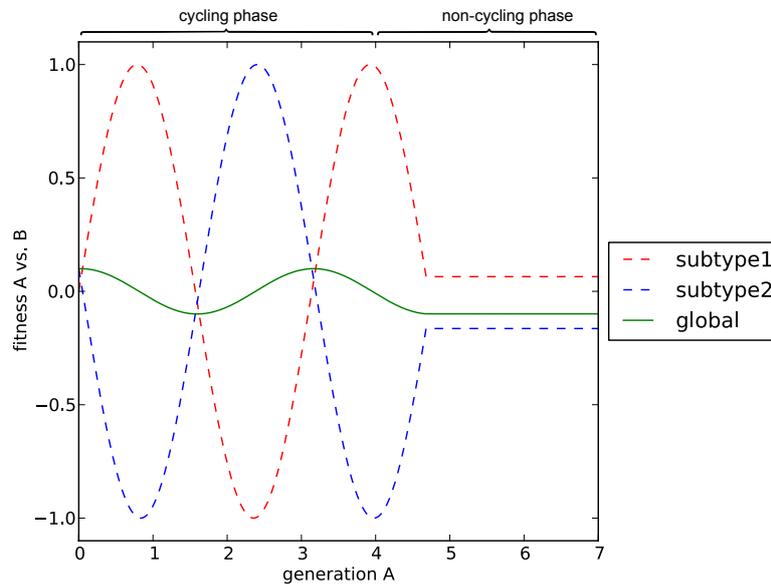


Figure 3.1: An illustration of how local dynamics might be hidden on a global level. The plots show the idealized master fitness scores of species A against species B. Species B ('global') has been bisected into two sub-families, identified by a cluster algorithm ('subtype1' and 'subtype2'). The y-axis denotes the fitness scores of species A against B. The x-axis denotes the generation from which A's elite was derived. Note that the cycling of A is hidden when plotted against the all of B, but becomes visible when plotted against B's sub-families. Also note that in the former case, A's transition from cycling to less-cycling is not visible.

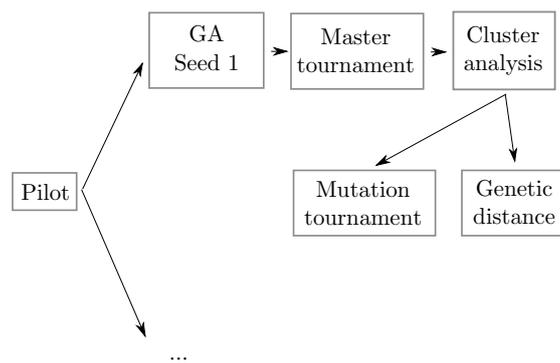


Figure 3.2: The experimental pipeline.

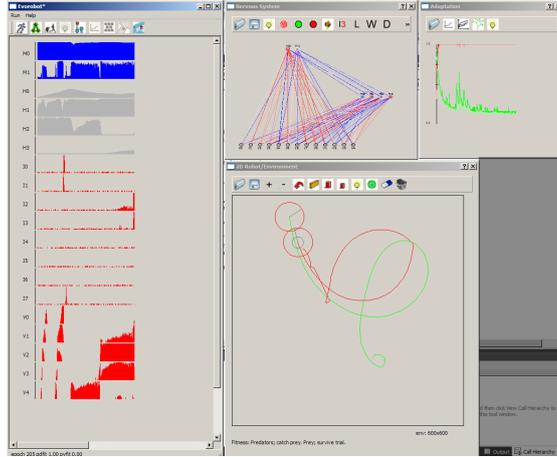


Figure 3.3: The Evorobot* GUI.

Microsoft Windows XP. The final source code was compiled with cmake 2.8.3 under openSUSE 11.4. The experiments were run on a computer cluster at the ICTS¹, consisting of 10 nodes and one master node, each containing two quad-core AMD Opteron 2374HE processors. The Evorobot* software stored all relevant data on the cluster’s network storage device, either in a CSV or custom-built format.

3.2.2 Simulated environment, sensors and actuators

Two 75mm diameter e-puck robots were situated in a 600x600mm simulated environment. Each robot was positioned and oriented randomly. The robots were placed at minimally 175mm (2.5 times the e-puck diameter) from the walls to allow them some opportunity to avoid them. Moreover, if the spacing between two robots was less than a certain distance (Equation 3.1), one of these robots was assigned a new random position until both robots satisfied both distance constraint.

$$\text{minDistance} = 0.4\sqrt{\text{arenaX} \cdot \text{arenaY}} + \varnothing_{\text{robot}} \quad (3.1)$$

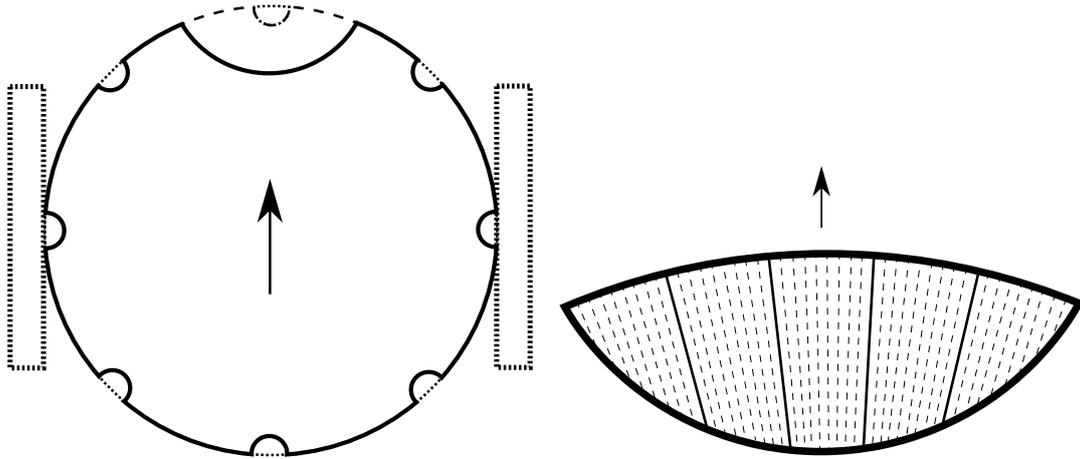
The robots were equipped with a VGA camera, eight infrared sensors and two wheels (Figure 3.4a).

The *linear VGA camera* was positioned at 0° on the robot’s body, with a predator field of view of 45° and a prey field of view of 360°. ² Furthermore, it was divided into five 9° or 72° *segments* respectively, which could each yield an activation value $0 \leq \alpha_{\text{seg}} \leq 1$. Each of these segments was connected to an input neuron in the neural network (Section 3.2.3). Moreover, a segment was divided into nine or 72 1° *photoreceptors* (Figure 3.4b).

For each timestep, the geometric projection of each visible object onto any photoreceptor was calculated. If the photoreceptor detected any object, it was activated with a value of $\alpha_{\text{photo}} = 1$. The activation of all photoreceptors was averaged to determine a segments’s total activation level. Dividing a segment into nine photoreceptors allowed a robot not only to perceive the direction of a stimulus (depending on which segment got activated), but also to get a better sense of distance

¹Institute of Cognitive Sciences and Technologies, National Research Council, Rome

²This distinction was inspired by the observation that prey animals often have eyes in the sides of their heads, while predator animals’ eyes are pointed forward.



(a) Sensors and actuator layout. Shown is the e-puck from a dorsal point of view. Eight infrared sensors (dotted lines) are positioned at 45° intervals. The linear camera (dashed line) is positioned at 0° . The wheels are attached at 90° and 180° positions.

(b) The 45° (predator) VGA camera configuration. Shown are the five 9° photoreceptors, each constituted out of nine 1° segments. Each photoreceptor is connected to an input layer neuron. The prey camera (not shown) had a 360° field of view; five 72° segments each composed out of 72 1° photoreceptors.

Figure 3.4: The e-puck sensors and actuators. The arrows show the robot's forward direction.

to it (a segment's activation is proportional to how many photoreceptors were activated; objects that were far away would activate it less than objects that were closer by). The camera was configured to detect only the opponent robot.

The eight *infrared sensors* were spaced 45° apart, with the first sensor positioned at 315° . Sensor activation could yield a value $0 \leq \alpha_{ifr} < 1024$ and was determined by calculating the distance and angle for a sensor to the closest object or wall present in the arena. The position of such an object relative to the sensor was then matched to a sample table that contained the actual activation measurements from real infrared sensors used with the e-puck. Each infrared sensor was connected to a neuron in the input layer of the neural network. Since the neural network internally used neuron activation values of $0 \leq O_j(\tau) \leq 1$ (Section 3.2.3), the raw infrared sensor values were normalized into the network's native activation range.

The two *wheels* were positioned at 90° and 180° on the robot's body. The wheels could move independently from each other. The Evorobot* software was designed to be computationally efficient. Therefore, the wheels were not simulated as actual objects with physical dimensions, but instead the activation from the two motor neurons $0 \leq O_\nu(\tau - 1) \leq 1$ and $0 \leq O_\phi(\tau - 1) \leq 1$ in the neural network's output layer were directly translated into wheel speed $s_{side \in \{left, right\}}(\tau)$, where $-s_{max} \leq s_{side} \leq s_{max}$. Neuron O_ν encoded the baseline robot velocity, while neuron O_ϕ encoded the robot's turning rate (Equations 3.2 and 3.3). The speed limit for the predator was set to $s_{max} = 8$, while for the prey is was set to $s_{max} = 10$

$$s_{left}(\tau) = \begin{cases} s_{max} O_\nu(\tau - 1) \phi & \text{if } O_\phi(\tau - 1) < 0.5 \\ s_{max} O_\nu(\tau - 1) & \text{else} \end{cases} \quad (3.2)$$

$$s_{right}(\tau) = \begin{cases} s_{max} O_\nu(\tau - 1) \phi & \text{if } O_\phi(\tau - 1) > 0.5 \\ s_{max} O_\nu(\tau - 1) & \text{else} \end{cases} \quad (3.3)$$

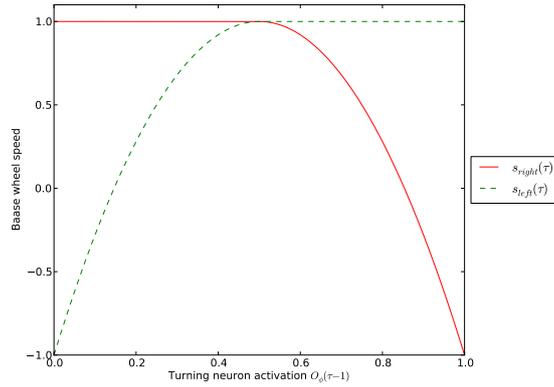


Figure 3.5: The turning neuron activation $O_\phi(\tau-1)$ decoded into wheel speed $s_{side}(\tau)$. The green line shows the left wheel speed, while the red line shows the right one. When O_ϕ 's activation is at ~ 0.5 , both wheel speeds remain unchanged and the robot moves forward. When O_ϕ 's activation starts to deviate from ~ 0.5 , one of the wheels decreases speed causing the robot to turn. Note that $s_{max} = 1$, $O_\nu = 1$ and $\psi = 1$ for this illustration.

Here, ϕ refers to Equation 3.4, where $\psi = 1$ refers to the turning ‘fall-off rate’ that determined how abruptly a robot’s velocity decreased when turning (this parameter was varied during the pilot studies).

$$\phi = -2^{2\psi+1}(O_\phi(\tau-1) - 0.5)^{2\psi} + 1 \quad (3.4)$$

Once wheel speed was calculated, it was matched to a sample table that contained real e-puck Cartesian displacement vectors for a large number of wheel speed combinations to calculate the final robot position.

The specialization between one neuron controlling baseline velocity and one controlling the turning rate was designed to reduce neuron interdependency when engaged in complex manoeuvres. If a robot would want to switch from being engaged in a left turn to initiating a right turn, this would only require an adjustment of the turning neuron while keeping the velocity neuron’s activity steady. If the output neurons were controlling the speed of one wheel each, both neurons’ output would have to change.

The turning neuron’s hyperbolic signature (Equation 3.4) ensured that robots had enough leeway to maintain a stable forward direction (Figure 3.5). For example, with a linear wheel speed decrease instead of a hyperbolic one, robots tended to wobble when moving forward, since their turning neuron O_ϕ was never able to maintain a precise activation level of 0.5.

3.2.3 Neural network

All the robots shared the same neural network architecture (Figure 3.6). The input layer of the network contained 13 neurons, of which the first eight received direct input from the infrared sensors, while the last five received input from the linear camera.

The hidden layer contained four neurons receiving connections from the input layer. Furthermore, these neurons projected onto themselves, in addition to retaining a fraction of their activation levels from previous timesteps. Both ‘leakiness’ and recurrent connections could allow the robots to utilize forms of memory to enable phenotypic plasticity.

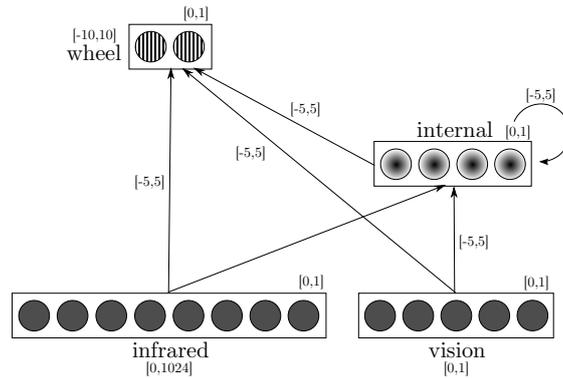


Figure 3.6: The neural network architecture. Arrows indicate neurons are fully connected between layers. Grated neurons denote an evolvable responsiveness. Graded neurons indicate both a evolvable responsiveness and activation retention (i.e. leakiness). The numbers indicate the internal ranges which the sensors, wheels and the network’s neurons and weights worked with.

Finally, the output layer contained two neurons, indirectly controlling robot velocity and bearing. These neurons received input from both the input and hidden layer.

Thus in total, the network contained 19 neurons. When connected, layers were fully connected, resulting 102 connections. The network’s activation state was computed for every timestep.

The activation $O_j(\tau)$ of (hidden/output) neuron O_j at timestep τ was computed using a logistic weighted sum function (Equation 3.5). A neuron’s activations was calculated from two sets of input connections; $-5 \leq w_{ij} \leq 5$ denotes the set of feed-forward connections from ‘upstream’ neurons while $-5 \leq w_{hj} \leq 5$ denotes the set of recurrent connections from lateral ones (which is an empty set in the case of the output layer). Furthermore, β_j denotes O_j ’s evolvable responsiveness parameter.

$$O_j(\tau) = \sigma \left(\beta_j \left(\sum_i w_{ij} O_i(\tau) + \sum_h w_{hj} O_h(\tau - 1) \right) \right) \quad (3.5)$$

Here, σ refers to the sigmoid function in Equation 3.6.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.6)$$

The activation of the leaky neurons (in the hidden layer) required an extra computation step, shown in Equation 3.7, where $0 \leq \delta_j \leq 1$ denotes O_j ’s evolvable decay-rate.

$$O_j(\tau) = (1 - \delta_j)O_j(\tau) + \delta_j O_j(\tau - 1) \quad (3.7)$$

3.2.4 Genetic algorithm

Representation

A steady state algorithm (Algorithm 1 and Table 3.1) was used to evolve two populations (predator and prey), each containing $N = 20$ individuals (this number formed a trade-off between computational resources available and population diversity). Subject to the GA operators were the connection weights and neurons’ responsiveness decay-rate parameters. Individuals were

Class	Steady state
Representation	Integer valued vector
Recombination	None
Mutation	Bitwise
Parent selection	Exhaustive
Survivor selection	$\mu + \lambda$
Replacement	Replace worst

Table 3.1: A brief summary of GA settings.

each genetically represented by a *genotype*; an integer vector \vec{v} where $0 \leq v \leq 255$. When decoded to a phenotype, values were normalized to the neural network’s connection weight range of $-5 \leq w_{ij} \leq 5$. The evolvable decay-rate required normalization into the range $0 \leq \delta_j \leq 1$ (Section 3.2.3).

The size of the genome was equal to the number of free parameters that determined an individual’s phenotype. Each connection weight would have to be represented by a single allele, as well as the responsiveness for all the neurons not in the input-layer. Additionally, the leaky neurons were each associated with an additional allele determining the decay-rate, totaling to 112 genes. All the genes of each individual of each population were randomized at the start of an experiment.

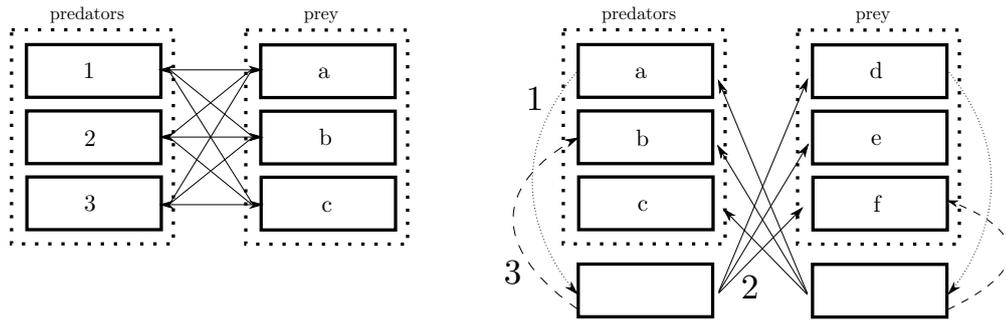
Evaluation

The GA iterated until the predetermined number of $g = 500$ generations was reached (this number allowed enough time for macro dynamics to emerge). For each generation, there were several computational steps taken. First, pairs of predators and prey were assigned fitness scores by evaluating them in a simulated environment. The individuals of both populations were exhaustively matched with each other (Figure 3.7a). So, the total number of *trials* played in this phase would be N^2 . For each trial, the two selected genomes were decoded into the neural network controllers. Each trial lasted a maximum of $t_{max} = 500$ discrete, 100ms *timesteps*. Thus, each trial lasted up until 50 seconds, unless it was prematurely terminated when the prey got caught.

Fitness function

The fitness function for predator and prey was inversely related (Equation 3.8). The function rewarded predators for catching the prey as fast as possible. Here, $0 \leq fit(n) \leq 1$ denotes the fitness value for individual n , t_i denotes the i^{th} timestep with $t_0 \leq t_i \leq t_{max}$ while PD and PY denote the predator and prey populations respectively. Maximum predator fitness was yielded when the predator caught the prey at timestep t_0 (which was in practice impossible due to the starting distance constraint (Equation 3.1)). Maximum prey fitness was assigned when the prey did not get caught at the 500th timestep t_{max} . Since each individual participated in 20 trials and could survive an indeterminate number of generations, the fitness score of an individual was constantly summed at the end of a trial (this allowed for individuals that survived for multiple generations to build a ‘solid’ average, less sensitive to random fitness fluctuations). The total number of trials an individual participated in was recorded likewise to calculate averages.

$$fit(n) = \begin{cases} 1 - \frac{t_i}{t_{max}} & \text{if } n \in PD \\ \frac{t_i}{t_{max}} & \text{if } n \in PY \end{cases} \quad (3.8)$$



(a) The first step to produce a new generation; establish the fitness of both parent populations. Alphanumeric indicate the sequence of evaluation (i.e. 1 vs. a, 1 vs. b, 1 vs. c, 2 vs. a, etc. Solid arrows indicate evaluations.

(b) The second step involves evaluating the offspring. The dotted arrows (1) shows the creation of offspring (only the first individual is shown here), in the order of alphabetic lettering. The solid arrows (2) show how that offspring is evaluated against all opponent parents. The dashed arrows (3) shows how the newly generated offspring might replace a conspecific parent individual.

Figure 3.7: The GA in a schematic visualization; shown are the steps to progress from one generation to the next. The dashed boxes represent the predator and prey populations, while the solid ones represent individuals. The solid arrows indicate the pairs of opponent species playing in the trials.

Mutation

When a generation was completed, each parent individual generated an offspring by means of mutation (Figure 3.7b, pointer 1). This offspring was then evaluated against the entire opponent parent population (Figure 3.7b, pointer 2). The opponent parent's fitness was not updated in this phase. This was done so to not influence the opponent population with untested offspring, preventing possible fitness inflation. So, the total number of offspring evaluated in this step is $2N^2$. For the first generation, no offspring was yet produced, in order to determine the fitness of the initial populations with a larger degree of certainty.

The mutation operator was a bitwise one. Each gene in the genome got base-converted from an integer to a vector of eight bits. Every bit has a 0.02 chance of being flipped, so the chance of a single gene mutating somewhere is $1 - 0.98^8 \approx 0.15$, resulting in an average of ~ 16.8 mutations in an entire genome.

Selection

After a single offspring's fitness was established, it was immediately compared to its own parent population. If the offspring had a higher fitness than the worst parent, it replaced it (Figure 3.7b, pointer 3).³ The new offspring, together with all the individuals that did not get replaced, formed 'the next generation'. Fitness averages were recorded, and the algorithm looped back to perform evaluations on the surviving individuals.

³Note that this implies that a newly generated offspring could be immediately replaced by another offspring from the same generation, if the latter had a higher fitness than the former and was generated at a later point in time. Also note that the steady state algorithm does not necessarily replace any parent individuals, if all those are of a higher fitness than any offspring.

```

for generation  $\leftarrow 0$  to  $x$  do
  // Evaluate parents
  foreach Predator do
    foreach Prey do
      pdFitness,pyFitness  $\leftarrow$  trial(Predator,Prey);
      totalPdFitness  $\leftarrow$  totalPdFitness + pdFitness;
      totalPyFitness  $\leftarrow$  totalPyFitness + pyFitness;
    if generation  $> 0$  then
      // Evaluate offspring (either predator or prey)
      foreach Individual do
        Child  $\leftarrow$  generateChild(Individual);
        foreach Opponent do
          childFitness  $\leftarrow$  trial(Child,Opponent);
          totalChildFitness  $\leftarrow$  totalChildFitness + childFitness;
        // From own species
        WorstParent  $\leftarrow$  getWorstParent();
        if avgChildFitness  $\geq$  WorstParent.avgFitness then
          replace(WorstParent, Child);
      writeFitnessToFile();

```

Algorithm 1: The genetic algorithm.

3.3 Analysis

3.3.1 Computing environment

The data obtained from the simulator was processed and visualized with custom-built Python 2.7.2 scripts, developed in (*Python(x,y)*, 2012), in conjunction with various open-source modules. Exploratory clustering was done using the Scikit-learn

3.3.2 Hierarchical cluster analysis

The master tournament allows to add a degree of formalization to the concept of ‘behaviour’, as individuals are likely to show specific rates of success against different types of opponents; an individual can be phenotypically defined based on its fitness-score against all its master tournament opponents. In effect, we can define a behaviour as in Definition 1.

Definition 1 *Behaviour*: A series of fitness scores against all an individual n_i ’s master-tournament opponents $0 \leq m_j \leq |M|$, represented by a vector \vec{n}_i such that $n_{ij} = \text{fit}(n_i, m_j)$. Here, $n_i \in N$ and $m_j \in M$ denote the i^{th} and j^{th} individual from species N and M respectively. Furthermore, $\text{fit}(n_i, m_j)$ forms a function that yields n_i ’s fitness value obtained from testing it against m_j .

The data of a single master tournament can be represented by an $|N| \times |M|$ matrix A , where $\vec{a}_{i,*}$ denotes the fitness score of the i^{th} individual from species N against all of M , and $\vec{a}_{*,j}$ denotes the fitness-score of the j^{th} individual from species M (Equation 3.9).

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,|N|} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,|N|} \\ \vdots & \vdots & \ddots & \vdots \\ a_{|M|,1} & a_{|M|,2} & \cdots & a_{|M|,|N|} \end{pmatrix} \quad (3.9)$$

It follows from Definition 1 and Equation 3.9, that the master tournament data contains a description of all the behaviours that were displayed during the tournament. This allowed to apply a cluster analysis to the master tournament data to build a well-informed ‘family tree’ of phenotypic families and associated individuals.

The cluster analysis applied in this study was based on the UPGMA/WPGMA algorithm, an agglomerative (‘bottom-up’) hierarchical cluster algorithm (Sokal & Michener, 1958). Agglomerative cluster algorithms construct their hierarchies by iteratively grouping the two most similar clusters in a collection together until all clusters are grouped under one big ‘root’ cluster, thereby forming a binary tree (note that initially unclustered elements form singleton clusters). Applied to the master tournament data, each node in this tree represents an abstract ‘family’ of individuals and each leaf represents a concrete individual.

```

MasterData ← loadMasterData();
foreach Species do
  Tree ← newEmptyTree();
  DistanceMatrix ← initDistanceMatrix(MasterData,Species);
  while sizeof(DistanceMatrix > 1) do
    // Euclidian metric used
    ClusterA,ClusterB ← removeClosestClusters(DistanceMatrix);
    // averaging measure used
    NewCluster ← mergeClusters(ClusterA,ClusterB);
    addCluster(NewCluster,DistanceMatrix);
    addCluster(NewCluster,Tree);
  writeToFile()

```

Algorithm 2: The UPGMA/WPGMA based cluster algorithm.

The cluster algorithm used a distance matrix to keep track of newly formed clusters and their distance to the other known clusters. The algorithm calculated this matrix at its initialization. Next, the two closest clusters were merged into a new cluster. The newly formed cluster’s (un)weighted average was calculated and used to update the distance matrix. Cluster representations were simultaneously stored in a binary tree datastructure to ease later visualization and data access. This process continued until every cluster was merged into one final cluster (Algorithm 2).

The metric used for calculating the distance between two singleton clusters q_a and q_b was simply the Euclidian distance between \vec{q}_a and \vec{q}_b (Equation 3.10).

$$\Delta(\vec{q}_a, \vec{q}_b) = \sqrt{\sum_{n=1}^{|N|} (q_{a,n} - q_{b,n})^2} \quad (3.10)$$

Distance computation between compound clusters necessitated the formulation of an appropriate averaging measure that specified how nested clusters could be represented by a single

vector to be used by Equation 3.10. Two measures were used in different instances; an unweighted (Equation 3.11) and weighted average (Equation 3.12).

Suppose there exists a cluster $Q = Q_a \cup Q_b$, where Q_i can either represent a fitness vector \vec{q}_i or another cluster Q . The averaging measure for such a cluster can be formulated by Equations 3.11 (unweighted) or 3.12 (weighted).

$$\mu_v(Q) = \frac{1}{2} \left(\frac{1}{|Q_a|} \sum_{q_a \in Q_a} Q_a + \frac{1}{|Q_b|} \sum_{q_b \in Q_b} Q_b \right) \quad (3.11)$$

$$\mu_w(Q) = \frac{1}{|Q_a| + |Q_b|} \left(\sum_{q_a \in Q_a} Q_a + \sum_{q_b \in Q_b} Q_b \right) \quad (3.12)$$

Conceptually, the unweighted average seems to be the more appropriate one, since this reflects the idea that sub-families are equally important when combined to form a new super-family, no matter out of how many individuals those sub-families are composed. However, ultimately which measure was used was simply based on which one yielded the more usable cluster tree.

To draw the phenogram, the binary tree was converted to a Newick formatted string (Felsenstein et al., 1986) which could then be visualized using (Huerta-Cepas, Dopazo, & Gabaldón, 2010). With some modifications, this allowed for additional graphics to be drawn into the tree visualizations. This was used to enhance the visualization by displaying small average-fitness histograms at the tree's nodes. This allows to quickly observe the overall performance of the family corresponding to that node against the opponent during the entire evolutionary run. Note that for this visualization one would not be interested in the unweighted but weighted performance (based on the intuition that a family's fitness performance should be an average over all the individuals it is composed of, not solely on the binary average of its two sub-families) (Equation 3.12). Thus, the visualization is always based on a weighted average, while the average used in building the tree varies.

3.3.3 Mutation tournament

The mutation tournament (Algorithm 3) was designed to investigate the existence of 'switching genes', who might be responsible for genetic pre-adaptability and play a role in cycling (Chapter 1). The mutation tournament was similar to a regular master tournament, except that individuals are subjected to random mutations before being evaluated in a trial. These mutations (as used in the GA; see Section 3.2.4) might result in a hypothetical switching gene being activated, causing a switch in opponent specialization. To save computing time, only selected families/seeds (i.e. representative of cycling and non-cycling phases) were selected to participate in the mutation tournament.

```

foreach IndividualA ∈ FamilyA do
  foreach IndividualB ∈ FamilyB do
    for n ← 0 to g do
      mutate (IndividualB);
      fitness ← trial (IndividualA,IndividualB);
    writeToFile (fitness);

```

Algorithm 3: The mutation tournament.

For example, suppose one wanted to test the existence of switching genes in species A in the idealized example shown in Figure 3.1. Now, 20 individual from each of A's selected families were randomly chosen and tested 112 times (i.e. the genome length) against both opponent B's families B_1 and B_2 ('subtype1' and 'subtype2' in the graphic). The random mutations would then result in a number of individuals starting being effective against B_1 at the expense of doing so against B_2 (or vice versa). An idealized outcome is shown in Table 1.1.

4.1 Procedure

This study selected one particular seed (Seed 9) for further investigation on the basis that it seemed to provide a case where a cycling phase was followed by a non-cycling one, aligning with this study's research questions (Chapter 1). Thus, this seed was explored in more detail using the cluster analysis and mutation tournament described in Sections 3.3.2 and 3.3.3 respectively.

4.2 Classical measures

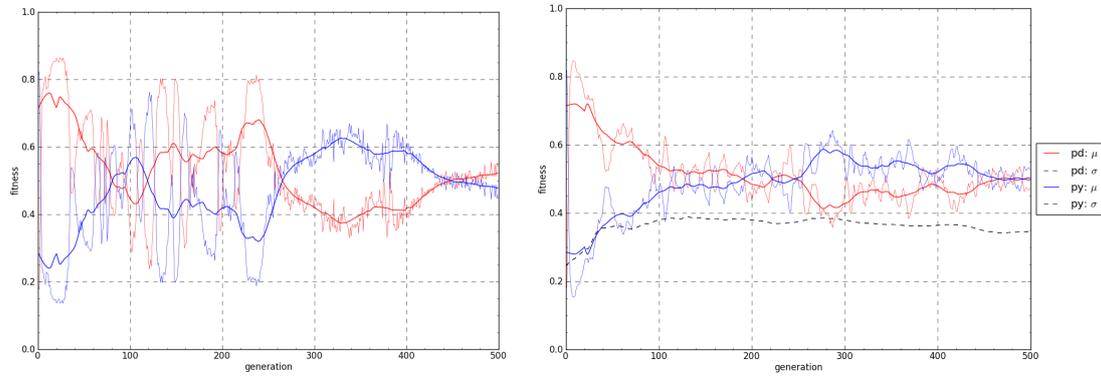
Figures 4.1 and 4.2 show the 'classical' online fitness and master fitness charts, on both the averaged data and the selected seed (Seed 9) in particular. Overall, one can observe a fairly large difference between the seed-specific data (Figures 4.1a and Figure 4.2a) and the seed-averaged one (Figures 4.1b and 4.2b). This gives testament to the notion that there can be a large degree of variability between different conditions. Figure 4.1a also illustrates the influence of the Red Queen Effect; fitness seems to be in constant flux due to a direct fitness interdependency between species. Both the master fitness charts (Figure 4.2) show a slow but steady progress for both species, which is particularly visible in the seed-average graph (Figure 4.2b).

The degree in variability between seeds and averages is also visible in the CIAO plots (Figure 4.3). The seed-specific plot (Figure 4.3a) shows a partial checkerboard pattern that is often interpreted as typical of cyclic evolution. Also visible is the 'smoother' half of the plot, indicating the transition from a cycling phase to a non-cycling one. The seed-average plot (Figure 4.3b) on the other hand shows a subtle but discernible diagonal bisection that is reminiscent of the ideal case shown in Figure 2.5d.

4.3 Cluster analysis

4.3.1 Results

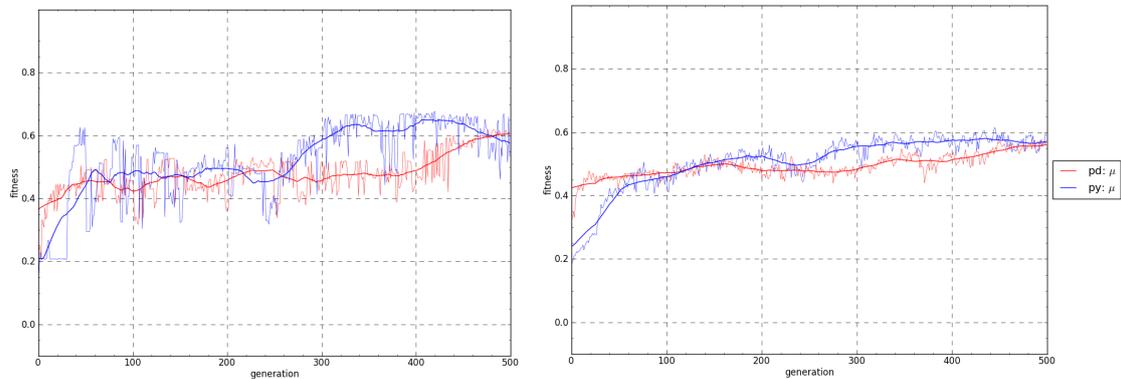
Figures 4.4 and 4.5 show the output of the weighted cluster algorithm on the master tournament data from Seed 9 (Figure 4.3a). Figure 4.4 shows the tree structure truncated at a depth of



(a) Seed 9. Values are averaged over 400 trials per generation (20 individuals, each against 20 opponents). Note the large fluctuations due to the Red Queen Effect.

(b) Average over 10 seeds (4000 trials per generation). The dashed line indicates the fitness standard deviation (over all 4000 trials; which is equivalent for both species, since species' fitnesses were related inversely proportional).

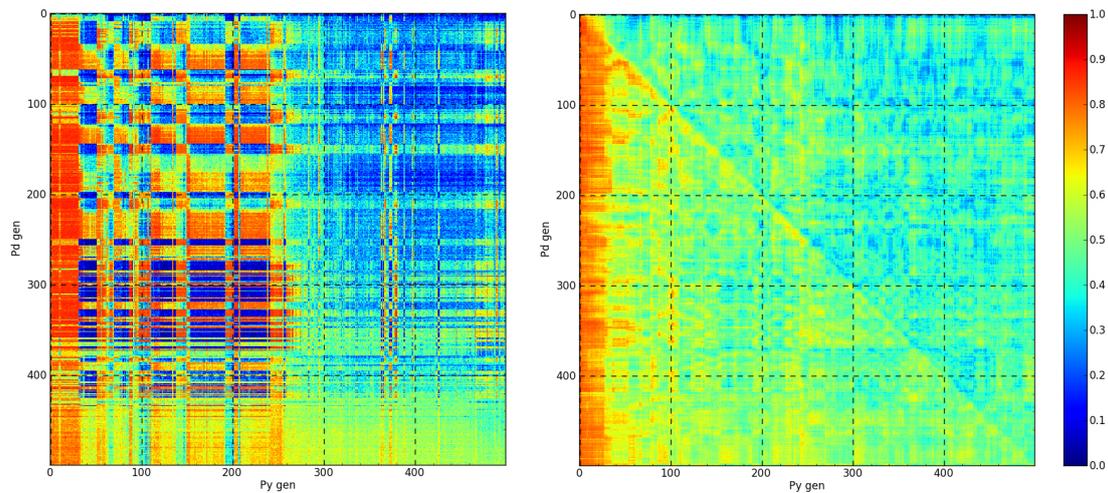
Figure 4.1: Online fitness progression. Red lines indicate predator fitness, while blue ones indicate prey fitness. Thick lines indicate a moving average over 25 generation. Thin lines indicate the actual recorded fitness.



(a) Seed 9. Values are based on 12500 trials (500 opponents, 25 replications).

(b) Average over 10 seeds. Values are based on 125000 trials.

Figure 4.2: Master fitness progression. See Figure 4.1 for legend.



(a) Seed 9. Note the possible transition from cycling to non-cycling, especially along the horizontal (prey) axis. Values are averaged over 25 trials.

(b) Average over 10 seeds. Note the resemblance to the ideal case scenario (Figure 2.5d). Values are averaged over 250 trials.

Figure 4.3: CIAO plots. Each colored dot represents the performance in terms of predator fitness (i.e. red indicates high predator performance and low prey performance, while blue indicates low predator performance and high prey performance). The y-axis denotes the predator elite’s originating generation, while the x-axis denotes the prey elite’s one.

level four. Shown is a hierarchy of phenotypic families. The *small histograms* indicating the performance of that family against all 500 opponent elites. Figure 4.5 shows the same histograms, but stacked vertically for easier visual comparison. One can observe big families with alternating performance dynamics for both species (predator families 1 and 2 and prey families 12 and 14). The smooth CIAO half in Figure 4.3a is visible here as the plateau that starts to emerge around generation 250 in the prey families.

Effectively reading master plots and their derivatives might not be a trivial task for the uninitiated reader. First and foremost, it is important to remember that master tournament data is acquired *after* the GA has been terminated. It therefore does not indicate any of ‘online’ fitness progression, but constitutes a means to estimate a progression of that which already has been previously evolved.

Secondly, the master tournament performance of an individual (and thus, according to our definitions, its behaviour) is always defined in relation to all its opponent’s *elites* (i.e. the best individual that each generation produced). When applying a cluster analysis to one of the species represented in the master tournament data, one is effectively clustering on a series of performances against all opponents elites. In that case, it is best to position oneself in the perspective of the clustered species. In doing so, it becomes clear that the cluster visualizations are showing hierarchical, behaviour based families, each represented by a performance sequence (i.e. a histogram showing fitness averages) (Figures 4.4 and 4.5).

Figure 4.6 shows how the cluster families are distributed over the course of the generic GA

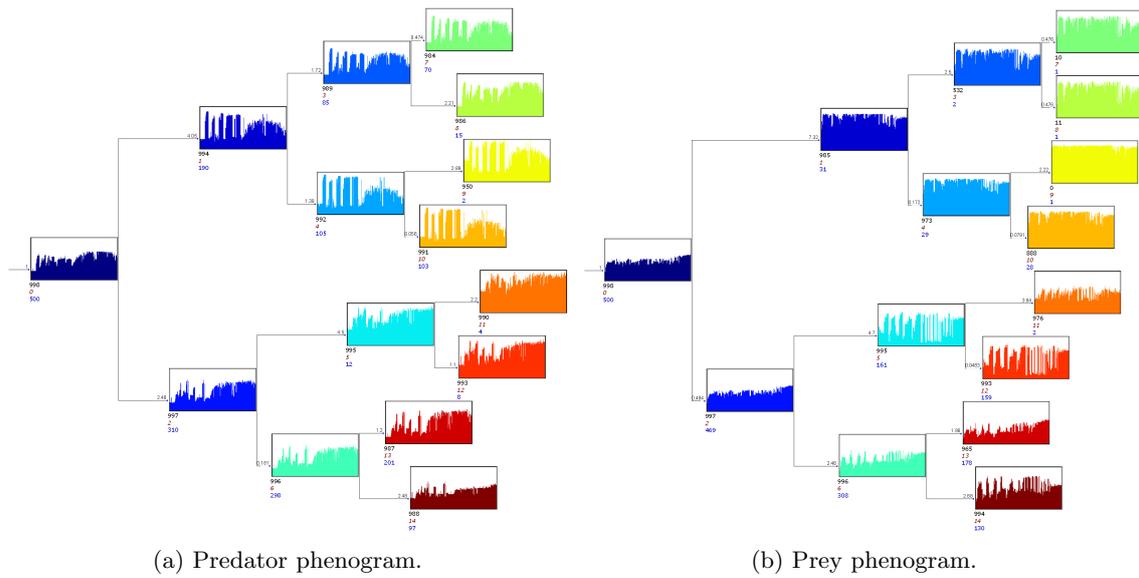
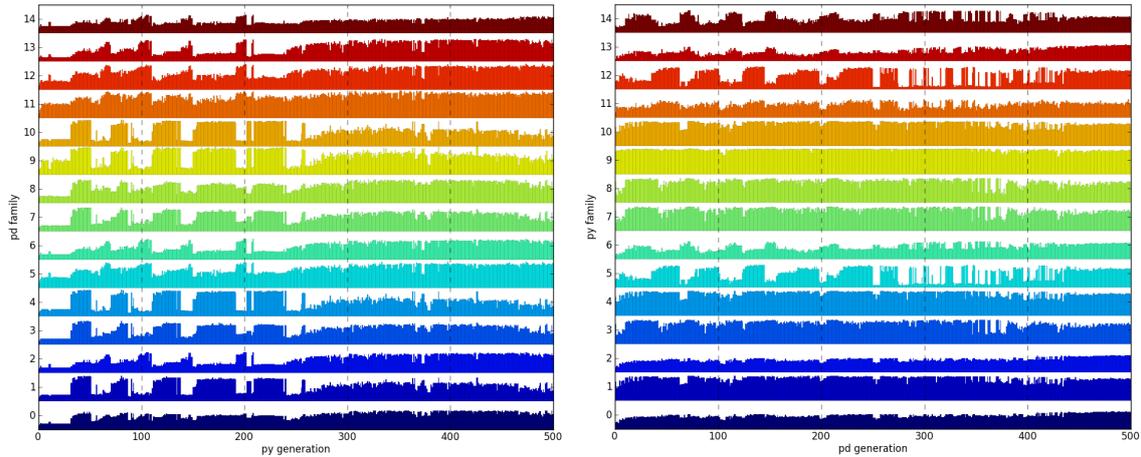


Figure 4.4: The phenograms as generated by the cluster analyses on Seed 9. Each node represents a family of individuals. The histogram shows the average fitness score of that family against all 500 opponents (note here that the root node corresponds to the graphs in Figure 4.2a; these nodes represent families encompassing the *whole* master pool). The numbers near the branches indicate the Euclidian distance from parent to child node. Each node shows three additional numbers. The first one identifies the cluster by a unique ID. The second one corresponds to the enumeration seen in Figure 4.5 and histogram colouring. The third one denotes the size of the node's subtree. The phenogram was truncated at a depth of four levels.



(a) Predator family performance. Note the presence of two phases; the first one having a distinct alternating pattern between the two large families 1 and 2, and the second one showing a more gradual, plateauing pattern. Also note the resemblance to Figure 3.1.

(b) Prey family performance. Note the alternating pattern between families 12 and 14.

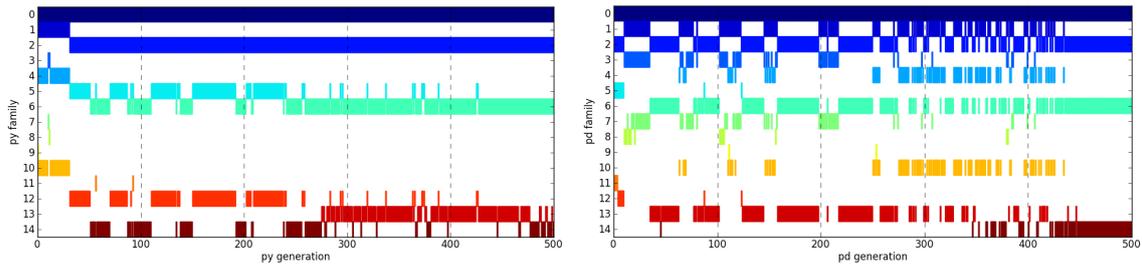
Figure 4.5: The nodes from Figure 4.4 stacked for easy comparison. The y-axis corresponds to the red numbering in Figure 4.4

run. Observe how the large alternating families align with peaks and drops in the opponent species' performance in Figure 4.5. In particular, the predator performance of families 1 and 2 (Figure 4.5a) seems to align with the distribution of prey families 12 and 14 (Figure 4.6a), while predator performance of families 12 and 14 (Figure 4.5b) seems to do so with the distribution of predator families 1 and 2 (Figure 4.6b). This shows the mutual interleaving pattern that is typical of cycling phases in evolution. Also of interest is the alignment between the distribution of prey family 13 (Figure 4.6a) and the general plateauing phase in Figure 4.5a.

4.3.2 Conclusion

It is tempting to propose that the cluster algorithm has (partially) captured two phases in the evolution of the prey species; a cycling one followed by a plateauing one. The cycling phase is constituted of two families; 12 and 14. These families are superseded by another family (13) around generation 300, marking the plateauing phase (Figure 4.6a).

Secondly, if cycling is the result of genetic pre-adaptability one should expect some switching genes in families 12 and 14. Moreover, this study's hypotheses suggest that the plateauing phase represented by family 13 should be the result of phenotypic plasticity (Section 1). Both these issues will be investigated in Section 4.4.



(a) Prey family distribution. Note how the presence of prey families 12 and 14 correspond with performance in predator families 1 and 2 in Figure 4.5a. Also note how the coupling with prey family 13 seems to be related to the more general second (plateauing) phase among predator families.

(b) Predator family distribution. Note the alignment between the distribution of predator families 1 and 2 and prey families 12 and 14's performance (Figure 4.5b).

Figure 4.6: The classification of elites over the course of the GA. Solid bars indicate that the elite from the x^{th} generation has been classified as belonging to the family denoted on the y-axis. Note that individuals are always members of multiple families, with more specific classifications among the higher-numbered families. Also note that with respect to Figure 4.5, the plots have been swapped horizontally, to emphasize interspecific alignment.

4.4 Mutation tournament

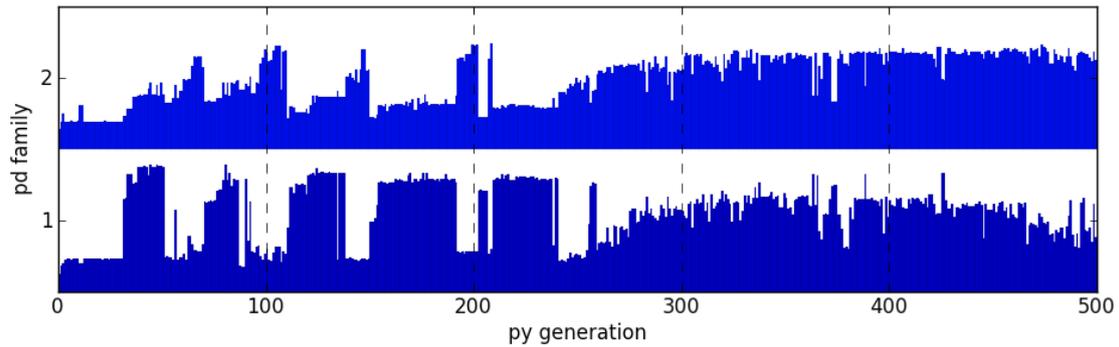
4.4.1 Results

If switching genes exist, they would suggest that large behavioural changes does not need to require large genetic change (Section 1.2.2). Figure 4.7 shows the prey (elites) over the course of the evolutionary run against two large predator families which are showing cycling dynamics (Figure 4.7a). Furthermore, the prey's performance has been aligned with a visualization of genetic change (Figure 4.7b).

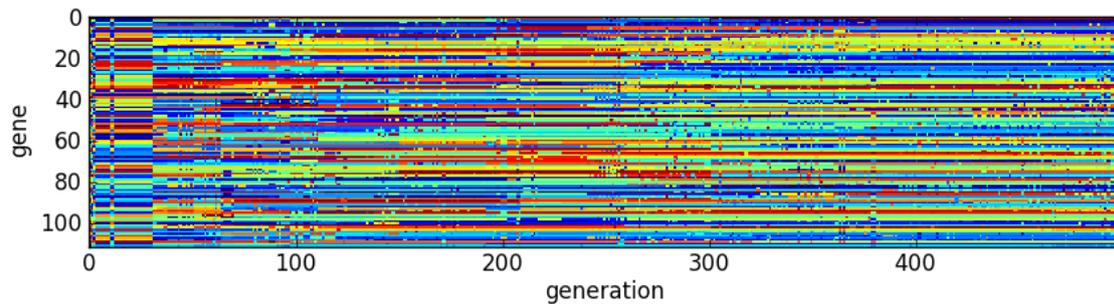
One can observe that overall there is little abrupt genetic change accompanying phenotypic change. The only clear transition point is located around generation 20, which might mark the departure from random behaviour. The genes that align with phenotypic cycling (up until around generation 200) however seem to behave more chaotically than those aligned with the plateauing phase, and some suggestions of genetic transitions are visible. This would suggest that a) robot behaviour was either a-priori dependent on only a small number of genes (i.e. there is a lot of unused 'junk DNA'), b) plasticity is a large factor in the cycling phase as well as it is hypothesized to be in the plateauing phase or c) the use of switching genes is indeed of importance in the cycling phase. Conclusively, it can be said that large phenotypic change does not require large genetic change.

The clusters obtained previously were used to select individuals for the mutation tournament. More specifically; 20 individual from prey families 12, 14 (both from the cycling phase) and 13 (from the plateauing phase) were randomly selected and played against *all* opponent predators. Figures 4.8, 4.9 and 4.10 show how those families' original performance compares to when having applied random mutations.

It appears there is some degree of patterned change in performance visible for all prey families, especially when having yielded a high success rate (blue color) in the pre-mutation scenarios.

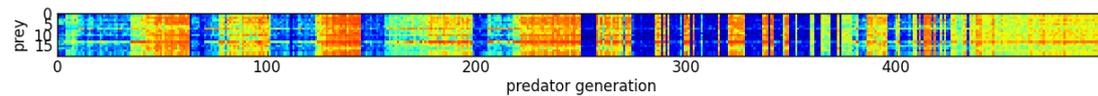


(a) The performance (i.e. phenotypic change) of *all* prey elites against (cycling) predator families 1 and 2 from Figure 4.5a.

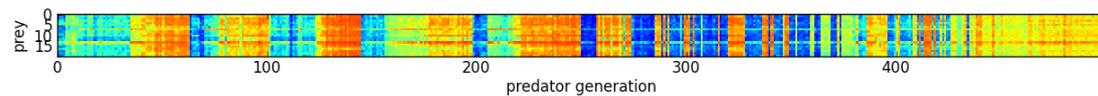


(b) The genetic change of *all* prey elites. The x-axis denotes the prey originating generation. The y-axis denote prey genes. Colours indicate allele values (higher values represented by red, lower ones by blue). Note the lack of (clear) vertical banding that could be expected in the alternating phase, when phenotypic change would largely depend on genetic change.

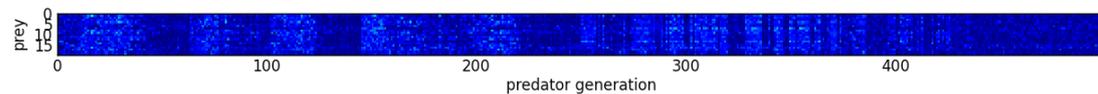
Figure 4.7



(a) Legacy performance.



(b) Mutation performance.



(c) Difference between legacy and mutation performance.

Figure 4.8: Prey family 12 (cycling phase) mutation sensitivity.

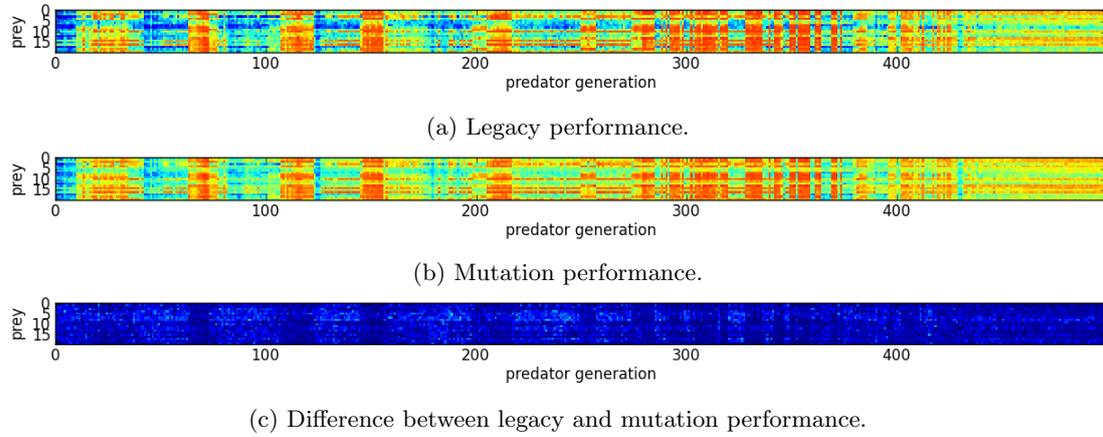


Figure 4.9: Prey family 14 (cycling phase) mutation sensitivity.

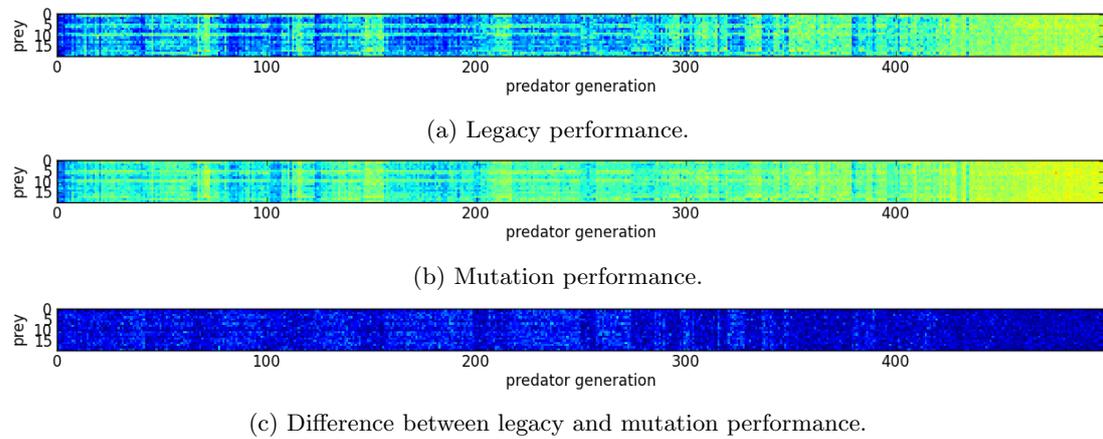


Figure 4.10: Prey family 13 (plateau phase) mutation sensitivity.

Family	py12 (cycling)	py14 (cycling)	py13 (non-cycling)
pd1	0.79	0.30	0.61
pd2	0.34	0.52	0.63

(a) Legacy performance.

Family	py12 (cycling)	py14 (cycling)	py13 (non-cycling)
pd1	0.68	0.30	0.55
pd2	0.32	0.45	0.53

(b) Performance after random mutation.

Family	py12 (cycling)	py14 (cycling)	py13 (non-cycling)
pd1	-0.11	0.00	-0.06
pd2	-0.02	-0.07	-0.10

(c) Differences between post-mutation and pre-mutation performance.

Table 4.1: The average performances of prey families 12, 14 and 13 against predator families 1 and 2 (see Table 1.1 for the expected performance).

4.4.2 Conclusion

Table 4.1 summarizes the performance of the pre-mutation (Table 4.1a) and post-mutation individuals (Table 4.1b) against predator families 1 and 2. First, it shows that the prey family representing the plateauing phase (family 13) is able to cope with both predator families, and only performs worse than prey family 12 against predator family 1, which in turn performs much worse against predator family 2. This strongly suggests the use of a form of plasticity by family 13 (see Section 5.1.2). It also shows that the prey families from the cycling phases each have specialized in coping with one type of predator.

Secondly, there appears to be a large drop in performance for prey family 13 when having been subjected to random mutations (Table 4.1c). This could be due to a) the existence of a switching gene having been flipped a number of times or (more likely) b) the suspicion that random mutations affect higher performing individuals more than lower performing ones (the cycling families show large drops in performance as well when evaluated against their ‘preferred’ opponent).

5.1 Interpretation of results

5.1.1 Cluster analysis as a tool in ER

The results obtained demonstrated the usability of phenotypic taxometrics in ER. In particular, it was shown that a cluster analysis could be applied to master tournament data to extract behavioural families of individuals. Notably, this could be used to construct a phenetic tree not unlike those used in the bioinformatics community.

The cluster analysis provides a level of formalization and automation that, to the author's knowledge, has not been seen before with respect to master tournament meta-analyses. First, the analyses could be used to 'filter' master tournament data, in that it is able to extract and visualize performances between selected families. This enables the experimenter to focus on their (interspecific) interactions, instead of having to deal with the possibly obfuscated totality of data present in the original master tournament.

As a more speculative suggestion concerning future developments of cluster analyses in ER, it would be interesting to broaden the taximetric approach to include genotypes. For example, future studies could record ancestral continuity between all generations/individuals and construct an unrooted genetic tree, in which nodes (individuals) can then be phenotypically identified using the clusters obtained from the master tournament. This should provide insight on behavioural sub-populations that constitutes the generations of the generic evolutionary run (Figure 5.1).

However, this would require a modified/alternative cluster algorithm, that allows for ad-hoc classification (i.e. classifying individuals using clusters that have already been finalized).¹ When using a hierarchical cluster algorithm, one could for example use the centroid of a cluster as a form of 'prototype' to compare the GA's individuals to, although this might be a crude solution. Alternatively, one could use more exotic algorithms, such as OPTICS (Ankerst, Breunig, Kriegel, & Sander, 1999) which combines ideas from hierarchical clustering with density-based clustering.²

¹It would also require a significant computational resource investment; every individual produced during the GA needs to be saved and tested against all 500 master tournament opponents, and this performance needs to be recorded.

²Density based clustering has the advantage of being more 'accurate' (i.e. produces clusters that correspond more with human intuition), can handle complex cluster shapes better, and offers performance benefits.

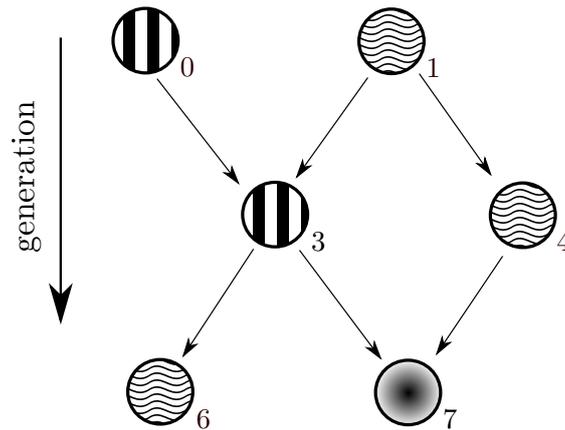


Figure 5.1: An unrooted genetic tree that has been classified on a phenotypic basis using a cluster algorithm. Nodes indicate the different *sub-populations* of a generation, where each level represents a generation from the GA. Arrows indicate genetic continuity while node patterning indicates phenotypic classification. Shown are possible transitions (not exhaustive), e.g.: $1 \rightarrow 3 \rightarrow 6$; cycling, $1 \rightarrow 4$; transference, $1 \rightarrow 3$; innovation, $3 \rightarrow 7 \leftarrow 4$; convergence, $3 \leftarrow 1 \rightarrow 4$; divergence, $3 \rightarrow 6$; extinction, $3 \rightarrow 6$; re-emergence, $0 \rightarrow 3 \leftarrow 1$; absorption.

5.1.2 Plasticity and cycling

The families extracted by the cluster analyses were used to confirm the suggestion that plastic individuals have an extended range of flexibility, compared to rigid individuals. First of all, this has been demonstrated by evaluating interspecific family performance in the context of a simulated predator-prey scenario. It was shown that selected non-cycling prey families performed noticeably better against cycling predator families than when cycling prey families were observed.

However, it should be noted that the experiments have by no means *unambiguously* confirmed the existence of the hypothesis that phenotypic plasticity can subdue cycling, although we think strong hints for this hypothesis have been provided. The problem in answering this hypothesis boils down to three issues; that of the conventional conceptualization of cyclic evolution, how to measure it, and how to identify plasticity.

Firstly, in this study the robots had the potential for phenotypic plasticity at any time. Now consider that the master tournament works on phenotypic data. This implies that the dynamics shown in the master tournament and its meta-analyses can have both a genetic *and* ontogenic basis. Thus, one needs to carefully consider the definition of cycling; does it include the phenotypic kind? In artificial life often the distinction between genetic and ontogenic adaptation is made. However, this position is difficult to justify biologically (Yamauchi & Beer, 1994). In this study for instance, while the GA modifies the neural network's interconnectiveness and the neural network modifies its own internal state, both mechanisms are essentially modifying neuron activation levels. Nevertheless, while realizing macro-strategies such as cycling using expensive and risky ontogenics instead of a genetic alternative seems implausible, it remains possible. Still, one should keep in mind that the cycling behaviour discussed in this study, could in fact be of the phenotypic kind. In order to narrow the scope and, for instance, use the cluster analysis to investigate purely genetic cycling, robot controllers would have to be adjusted to not allow for plasticity.

Secondly, consider the fact that the master tournament is based on only the *elites* that each generation produces. This implies that there is no guarantee of ancestral continuity throughout

the master tournament. Since it has been shown that large changes in behaviours, i.e. shifts from one family to another, show little genetic change, it is tempting to conclude this implies the existence of a few genes that are responsible for the expression between different phenotypes. However, these shifts observed can theoretically just as well be caused by the presence of multiple *sub-populations* during the generic evolutionary run. By random or environmental circumstances, it is possible that, at different phases in evolution, different subpopulations provide the elite that gets transferred to the master tournament pool. This might create the illusion of not only cyclic evolution, but also raises questions on the reliance on the master tournament to establish the presence of cycling in the first place. A logical first step for a follow-up study in this direction would therefore be to keep track of the ancestral lineage of at least a number of individuals. If those individuals show the same cycling pattern as observed in the master tournament and its derivatives, it can be safer to assume they are indeed representing cyclic or plastic phases.

Finally, the last remaining issue is that one might argue that this study has not conclusively demonstrated that it is indeed plasticity that is responsible for the termination of the cycling phase that was observed; one might suggest it could also be possible that a single, general strategy is responsible. However, it seems likely that general strategies are a-priori harder to obtain than plastic strategies (where one switches between simpler partial strategies). Furthermore, in Nolfi and Floreano (2000) it was reasoned that, if general strategies do not exist or can not be found, evolution might have to resort to cycling when confronted with a challenging opponent. Thus, since our case clearly shows cycling, one is forced to admit a general solution is at least hard to find. Moreover, as argued in Section 2.2.2, at this point the cycling phase has a certain bias to develop plasticity. Thus, one must conclude it is more likely that the plateauing phase following the cycling phase is based on plasticity, not generality.

However, to be certain, a thorough distal analysis is required. As a suggestion, future research might use the cluster analysis to filter distal data in the same way this study filtered master tournament data. For instance, during the master tournament robot coordinates for all trials could be recorded. The cluster analyses could then extract families based on master tournament fitness values, and average the coordinates to obtain family based density and vector plots, that might demonstrate plasticity more convincingly. However, one should keep in mind that a human interpretation of behaviour might lead to an unjust interpretation of it, and be based on false assumptions (Section 5.2.2).

5.1.3 Hints for switching genes

As a more explorative investigation, the issue of genetic pre-adaptability was questioned. If rapid phenotypic change as seen in cyclic evolution is present, it could possibly be realized by a few genes. As these ‘switching’ genes would have to be sensitive to mutation, mutating individuals from cyclic phases should at times result in a performance switch from being effective against one type of rigid opponent to another.

This study found no conclusive evidence that such a mechanism exists. When subjected to genetic mutations, individuals from cyclic prey families did not seem so switch predator specialization. However, it was shown that high performing prey seemed more sensitive to mutation, both originating from cycling as well as non-cycling families; in both cases performance dropped after mutation. In retrospect, this would be something to anticipate in follow-up studies. The number of effective phenotypes is evidently a small fraction of the total phenospace, thus indiscriminate mutation would logically result in a nett performance degradation.

Nevertheless, comparing prey genetic change with phenetic change showed a degree of correspondence. It seemed that cyclic changes in a prey family’s expressed phenotype was reflected by some degree of genetic change. This is expected if switching genes control the cyclic behaviour;

it refutes the idea that that substantial phenotypic change requires substantial genetic change as well. However, if the majority of genetic mutations does not explicitly target switching genes, chances are high such mutation would simply destroy the phenotype. A future study would therefore have to select those genes that would be suspected to be the switching ones (probably those who show the most change during cyclic phases), and subject *them* to random mutations.

5.2 Extended discussion

5.2.1 A needle in the haystack

One should stress that all results discussed here are only based on a fraction of the total amount of data that has been generated during the experiments. Moreover, all experimental seeds were based on random initial conditions, and these conditions often lead to large differences between scenarios and even seeds. This fact has two important implications.

First, it is quite likely there are many more cases in which interesting dynamics emerge, where this study only focused on one in specific (the cycling/plasticity transition). These interesting, yet unexplored dynamics, might already be present in the data that has been recorded, but more likely, remains dormant in an uninitialized random seed. Secondly, this observation also implies that one might get the impression there are interesting dynamics in every condition. Let it be stressed this is definitely not the case.

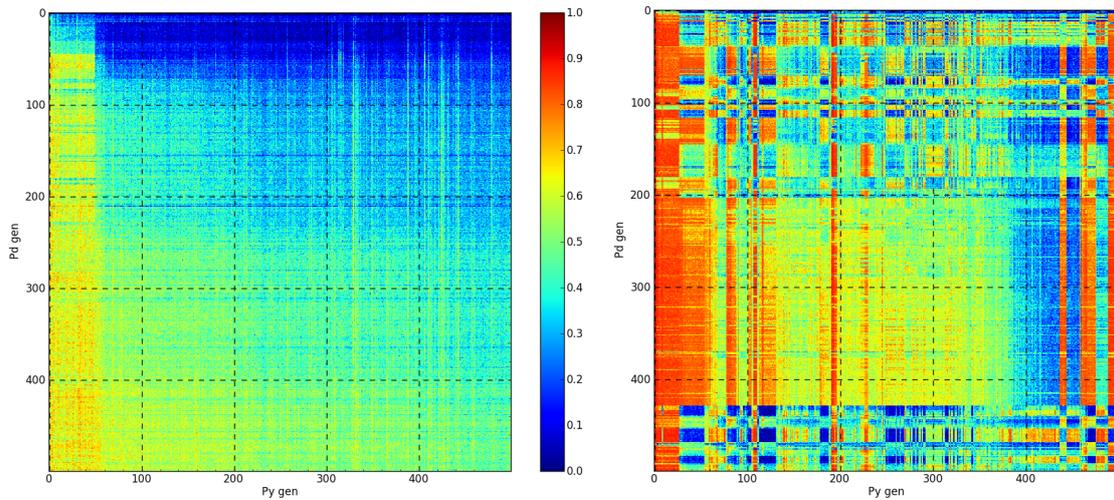
For example, while this study has shown the possibility that the emergence of phenotypic plasticity can be a valid alternative and beneficial alternative to cycling, most seeds in the same experiments did not show this transition, or not as clearly. Some experiments do not show cycling at all, but a smooth performance transition throughout evolution (Fig 5.2a). In the latter case, the lack of clear distinction between behavioral categories makes applying a cluster analysis seem counter-intuitive (Section 5.2.2).

Other seeds show only cycling, while some seem to show cycling, followed by plateauing, followed by cycling again (Figure 5.2b). This latter observation is particularly interesting from a theoretical perspective. A tempting explanation is that genetic internalization, discussed in Section 2.2.2, might play a role in this. For instance, it could happen that during a plateauing phase, a species might encounter one certain opponent family/strategy disproportionately often, thus the species will be pressured to switch to the appropriate coping strategy in the majority of cases. If sufficiently stable, this might allow for genetic assimilation. Now, if this process of assimilation has produced individuals of a certain degree of rigidity, this might in turn lead to the re-emergence of cycling. The cluster analysis proposed in this study can be used to investigate such issues further.

5.2.2 A definition of behaviour

When observing the actual robot behaviour, one can be struck by the overall subtleness and obfuscated character that differentiate the families as classified in Section 4.3, unlike previous studies, where behaviours could be relatively easily grouped in clear categories. It seems very minute adjustments like the angle of approach or speed can lead to big changes in performance against different types of opponents. Moreover, these observation are complicated by a large degree of context-dependency, as mentioned in Section 2.3.1, particularly when dealing with phenotypic plasticity.

The unavoidable discrepancy between a human definition of behaviour and the formal (master fitness based) one now becomes apparent. For example, a prey might evade a predator by moving at high speed against the walls or run fast, small circles around it. A human observer might be



(a) Seed 1 of a scenario that includes five random obstacles. Note a lack of banding that is seen in the other CIAO plots. Values are averaged over 50 trials.

(b) Seed 1 of the same scenario discussed in Section 2.2.3. Observe how the predator goes through three phases: cycling, non-cycling and cycling again. Values are averaged over 25 trials.

Figure 5.2: An illustration of how alternate seeds or scenarios can result in different evolutionary progressions.

tempted to classify these behaviours as belonging to different categories, while performance-wise they could show virtually identical performance characteristics. Of course, the formal definition of behaviour proposed in this study forms an abstraction that might not always correspond to a distal perspective (Section 2.1); what to a human observer might be separate forms of behaviour, might be one and the same according to definition.

However, from a blind evolutionary perspective, as well as from a proximal one, the distal viewpoint (Sharkey & Heemskerk, 1997) could perhaps be regarded as irrelevant. From a certain point of view, the definition of behaviour introduced in this study seems to resonate quite harmoniously with the philosophy of doing ER in the first place; i.e. to let the ‘algorithm’ of natural selection be in charge of neural controller design; to switch from the distal perspective to a more proximal one and to relieve the human designer from the burden of (possibly unnecessarily constraining) design (Nolfi & Floreano, 2000). However, one should emphasize that any definition is in effect an arbitrary one. This also is evident from the fact that, since this study applied a relatively unsophisticated cluster algorithm, multiple distance measures were used to obtain workable results (in alternative scenarios not discussed in detail in this study). In the end, a cluster analysis can be powerful tool to explore local dynamics in the master tournament, but exactly *what kind* of analysis is a thing that is subservient to the usability of the results. One might suspect that more informed algorithms could yield more impressive results.

5.3 Conclusion

Conclusively, this study has demonstrated the application of cluster analysis in the context of evolutionary robotics. In doing so, it has illuminated some of the performance characteristics

of cycling and non-cycling phases in simulated evolution. In particular, it was shown a) plastic individuals are able to counter multiple rigid opponents, which might in turn lead to the termination of the cycling phase b) there are hints that switching genes might exist and are related to cycling and c) the use of taximetric cluster analyses can be of value to the ER community. In the process, this study has raised a number practical, theoretical as well as philosophical questions which require further exploration.

REFERENCES

- Angeline, P., & Pollack, J. (1993). Competitive environments evolve better solutions for complex tasks. In *Proceedings of the fifth international conference on genetic algorithms* (pp. 264–270).
- Ankerst, M., Breunig, M., Kriegel, H., & Sander, J. (1999). OPTICS: Ordering points to identify the clustering structure. *ACM SIGMOD Record*, 28(2), 49–60.
- Baldwin, J. (1896). A new factor in evolution. *American naturalist*, 30(354), 441–451.
- Cartlidge, J., & Bullock, S. (2004). Unpicking tartan CIAO plots: Understanding irregular coevolutionary cycling. *Adaptive Behavior*, 12(2), 69–92.
- Cliff, D., & Miller, G. (1995a). *Co-evolution of pursuit and evasion II: Simulation methods and results*. Citeseer.
- Cliff, D., & Miller, G. (1995b). Tracking the red queen: Measurements of adaptive progress in co-evolutionary simulations. *Advances In Artificial Life*, 200–218.
- Dawkins, R. (1976). *The selfish gene*. Oxford University Press.
- Dawkins, R., & Krebs, J. (1979). Arms races between and within species. In (Vol. 205, p. 489). The Royal Society.
- Deacon, T. (1997). *The symbolic species: The co-evolution of language and the brain* (No. 202). WW Norton & Company.
- Eiben, A., & Smith, J. (2008). *Introduction to evolutionary computing*. springer.
- Felsenstein, J., James Archie, J., Day, W., Maddison, W., Meacham, C., Rohlf, F., & Swofford, D. (1986). *The newick tree format*.
- Floreano, D., Nolfi, S., & Mondada, F. (2001). Co-evolution and ontogenetic change in competing robots. *Advances in the evolutionary synthesis of intelligent agents*, 273–306.
- Godfrey-Smith, P. (2003). Between Baldwin skepticism and Baldwin boosterism. *Evolution and learning: The Baldwin effect reconsidered*, 53–67.
- Hebb, D. (2002). The organization of behavior. 1949. *New York Wiley*.
- Hinton, G., & Nowlan, S. (1987). How learning can guide evolution. *Complex systems*, 1(1), 495–502.
- Huerta-Cepas, J., Dopazo, J., & Gabaldón, T. (2010). ETE: a Python environment for tree exploration. *BMC bioinformatics*, 11(1), 24.
- Kendeigh, S. (1961). Animal ecology. *Animal ecology*.
- Miller, G., & Cliff, D. (1994). Co-evolution of pursuit and evasion I: Biological and game-theoretic foundations.

- Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., . . . Martinoli, A. (2009). The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions* (Vol. 1, pp. 59–65).
- Nolfi, S. (1996). Adaptation as a more powerful tool than decomposition and integration. In *Proceedings of the workshop on evolutionary computing and machine learning, 13th international conference on machine learning, bari*.
- Nolfi, S., & Floreano, D. (1998). Coevolving predator and prey robots: Do “arms races” arise in artificial evolution? *Artificial Life*, 4(4), 311–335.
- Nolfi, S., & Floreano, D. (2000). *Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines*.
- Nolfi, S., & Gigliotta, O. (2010). Evorobot*: A tool for running experiments on the evolutions of communication. *Evolution of Communication and Language in Embodied Agents*, 1, 297–301.
- Parker, G. (1979). Sexual selection and sexual conflict. *Sexual selection and reproductive competition in insects*, 123, 166.
- Python(x,y)*. (2012). <https://code.google.com/p/pythonxy/>. ([Online; accessed 25–10–2012])
- Rosin, C., & Belew, R. (1997). New methods for competitive coevolution. *Evolutionary Computation*, 5(1), 1–29.
- Sharkey, N., & Heemskerk, J. (1997). The neural mind and the robot. *Neural network perspectives on cognition and adaptive robotics*, 169–194.
- Sneath, P., & Sokal, R. (1973). *Numerical taxonomy. the principles and practice of numerical classification*.
- Sokal, R., & Michener, C. (1958). A statistical method for evaluating systematic relationships. *Univ. Kans. Sci. Bull.*, 38, 1409–1438.
- Turney, P. (1996). Myths and legends of the baldwin effect. In *In proceedings of the workshop on evolutionary computing and machine learning at the 13th international conference on machine learning* (pp. 135–142).
- Urzelai, J., Floreano, D., Dorigo, M., & Colombetti, M. (1998). Incremental robot shaping. *Connection Science*, 10(3), 341–360.
- Van Valen, L. (1973). A new evolutionary law. *Evolutionary theory*, 1(1), 1–30.
- Waddington, C. (1942). Canalization of development and the inheritance of acquired characters. *Nature*, 150(3811), 563–565.
- Waddington, C. (1953). Genetic assimilation of an acquired character. *Evolution*, 118–126.
- Wilson, D., & Wilson, E. (2008). Evolution for the good of the group.. *American Scientist*, 96(5), 380–389.
- Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3), 645–678.
- Yamauchi, B., & Beer, R. (1994). Sequential behavior and learning in evolved dynamical neural networks. *Adaptive Behavior*, 2(3), 219–246.