How the “Baldwin Effect” Can Guide Evolution in Dynamic Environments

Nam Le(✉), Anthony Brabazon, and Michael O’Neill

Natural Computing Research and Applications,
University College Dublin, Dublin, Ireland
namlehai90@gmail.com

Abstract. Evolution and learning are two different ways in which the organism can adapt their behaviour to cope with problems posed by the environment. The second type of adaptation occurs when individuals exhibit plasticity in response to environmental conditions that may strengthen their survival. Individuals seek a behaviour that increases fitness. Therefore, it is plausible and rational for the individual to have some learning capabilities to prepare for the uncertain future, some sort of prediction or plastic abilities in different environments. Learning has been shown to benefit the evolutionary process through the Baldwin Effect, enhancing the adaptivity of the evolving population. In nature, when the environment changes too quickly that the slower evolutionary process cannot equip enough information for the population to survive, having the ability to learn during the lifetime is necessary to keep pace with the changing environment. This paper investigates the effect of learning on evolution in evolutionary optimisation. An instance of dynamic optimisation problems is proposed to test the theory. Experimental results show that learning has a significant impact on guiding evolutionary search in the dynamic landscapes. Indications for future work on dynamic optimisation are also presented.

Keywords: Baldwin effect · Dynamic environments
Phenotypic plasticity

1 Introduction

For many biological organisms, adaptation is necessary for survival and reproduction in an uncertain environment. There are two important kinds of adaptation that should be distinguished. The first is a change at the genetic level of a population, in which organisms reproduce selectively subject to mechanisms, like mutation or sexual recombination, which maintain inter-individual variability. This is usually modeled in terms of biological evolution, which causes changes in the population from one generation to the next. The second adaptation mechanism, on the other hand, is the phenotypic change at the individual level. This can be called *lifetime-adaptation* which changes the phenotypic behaviour of the

D. Fagan et al. (Eds.): TPNC 2018, LNCS 11324, pp. 164–175, 2018.
https://doi.org/10.1007/978-3-030-04070-3_13

organism during its lifetime. Plausibly, lifetime adaptation happens at a quicker pace than the evolutionary process which takes place through the generational timescale, preparing the organism for rapid uncertain environments.

There exists an intriguing idea, called the **Baldwin Effect**, saying how learning interacts with and influences the evolutionary process, enhancing the adaptivity and performance of the population. Hinton and Nowlan (henceforth H&N) presented a classic paper around 1987 [6] to demonstrate an instance of the Baldwin Effect in a computer simulation. Their initial success motivated a number of further studies [1,9,11], to name but a few.

The initial success of H&N motivated several further studies in evolving neural networks, Artificial Life and Evolutionary Robotics, such as [1,9,11], to show how learning can enhance the evolutionary search. On the other side, it is pretty surprising that the interaction between learning and evolution has been rarely studied and employed in the field of Evolutionary Computation (EC), despite the fact that there exists not a small number of dynamic problems in the literature [4]. This might be due to the reason that some people in EC tend to go too far to regard evolution as learning based on the belief that evolutionary algorithms can be used to solve machine learning problems [3].

In the scope of this paper we are not going to take part in the debate on whether evolution is learning or not. We treat learning and evolution separately as they are in nature, in order to make use of the idea that learning can enhance the evolutionary process to cope with dynamic environments. The main aim of this paper is to investigate the effect learning might have on the evolutionary process when dealing with rapid changing environments. We combine evolution and learning, and propose a dynamic optimisation problem to see how they behave. In the remainder of this paper, we briefly present research on learning and evolution. We in turn describe the experiments we use in this paper. Results are analysed and discussed, then the conclusion and some future directions are proposed.

2 The Baldwin Effect, Learning, and Evolution

The orthodox view of evolution is that changes due to learning during life are not inherited and, more generally, that learning does not influence evolution. The basis for such a view is the physical separation between the germ cell line and the somatic cell line. Changes due to learning concern somatic cells whereas evolution is restricted to the germ cells. Since the two types of cells are physically separated, it is not possible for changes in the somatic cells to have a direct influence on evolution. On the other hand, Baldwin [2], Waddington [12], and several others [6,9] have claimed that there is an interaction between learning and evolution and, more specifically, that learning can have an influence on evolution. This is called **The Baldwin effect**.

In 1987, the British Cognitive Scientist Geoffrey Hinton and his colleague Kevin Nowlan at CMU presented a classic paper [6] to demonstrate an instance of the Baldwin effect in a computer simulation. Hinton and Nowlan used a Genetic

Algorithm to evolve a population in an extreme landscape called *Needle-in-a-haystack*, showing that learning can help evolution to search for the solution when evolution alone fails to do so. An interesting idea can be extracted from their work is that in stead of genetically fixing the genotype, it is wiser to let just a portion of the genotype be genetically fixed, and the other be *plastic* that allows for changes through learning. It is these plastic individuals that promote the evolutionary process to search for the optimal solution, although the H&N landscape is static.

The model developed by Hinton and Nowlan, though simple, is interesting, opening up the trend followed by a number of studies investigating the interaction between learning and evolution. There has been several papers studying the Baldwin effect in the NK-fitness landscape [7]. Some notable studies include the work by Giles Mayley [10], and some others like [5]. Their results, again, demonstrated that the Baldwin Effect does occur, and learning helps evolutionary search overcome the difficulty of a rugged fitness landscape. (Please refer to [9] for more literature review of the Baldwin Effect in the computer).

Evolutionary Algorithms (EAs) have been claimed a potential technique to solve dynamic optimisation problems [4]. EAs are a family of algorithms based on biological evolution metaphor. A number of parameter tweaking and operators have been proposed in the literature to enhance evolutionary search when dealing with dynamic optimisation. Though these engineering techniques show some good results in some way, relying on the evolutionary metaphor to cope with dynamic environments is not the optimal way if the optimal solution changes so fast that it takes many more generations for evolution alone to encode the environmental information back to the gene-like pool. Here we propose another view of dealing with dynamic environment, combining the metaphor of evolution and learning to cope with hard environmental dynamics.

3 Experimental Design

In this section we present the problem domain and the experimental settings we use to investigate our hypothesis.

3.1 The Dynamic String Match Problem

The chosen problem domain is the String Match problem, in which we have to match the target string. The String Match domain is not a rare problem, and it can be seen in a wide range of contexts, such as in Immune Systems (both natural and artificial), in Antivirus or Intrusion Detection systems. All these systems need some sort of string matching mechanism to match their dictionaries to an incoming signal to see if that signal is abnormal or not. In the scope of this paper, we propose a simple instance of the String Match problem, in which the target string contains only binary characters (0 or 1). The target string changes over time forming a dynamic problem. For simplicity, we restrict the length of the target string to 20.

Without loss of generality, suppose we have to match the original string ***111...11*** (20 ones). The target will change based on two parameters: the *frequency* and *magnitude* of change. The first parameter tells us after many generations the target will move to another point in the landscape, while the latter helps determine the likelihood of change for each element of the target. Assume that at a generation g the target is all-one (20 bits of one), *frequency* = 10 and *magnitude* = 0.1 (10%). This informs us that after 10 generations or at generation $g + 10$ the target t = 111...1 (20 bits of 1) is likely to be changed. The magnitude of 0.1 tells us that there are, on average, $20 \times 10\% = 2$ bits in the target that are likely to be modified. For each bit in the target sequence, a random number is generated and then compared with the *magnitude*: if the random value is less than 0.1, the current bit is mutated to its subtraction from 1 (1 becomes 0, and vice versa). Suppose the new target at generation $g + 10$ is $t_1 = 001...1$ (two first bits are changed).

3.2 Experimental Setup

In this section, we present two experimental setups used in our paper as follows:

Experimental Setup I: Evolution Only

The first setup evolves a population of individuals without learning capabilities. This is the canonical Genetic Algorithms. Every individual has the genome of 20 bits randomly initialised. The genotype-phenotype mapping is one-to-one. The fitness of an individual is calculated as the proportion of matched characters between the individual and the target over the number of characters (the length of the target string). Mathematically, the fitness of an individual x_i (with phenotype p_i) is computed as one of the following two ways:

$$f(x_i) = \frac{Number\ of\ matched\ characters}{Length\ of\ the\ target\ string} = 1 - \frac{dist(p_i, target)}{length(target)} \tag{1}$$

with $dist(p_i, target)$ is the hamming distance between the phenotype p_i and the *target*. Based on this fitness function, an individual with higher proportion of matching has a higher fitness; and the higher the fitness value, the better the individual. The optimal individual has the fitness of 1, whereas the worst possible fitness value is 0.

At each generation, two individuals are selected from the population as parents to produce one child. The newly born child is mutated based on a small probability, then being added into a new population. This process repeats until the new population is filled up and replaces the old population of parents, i.e. generational replacement without elitism.

Experimental Setup II: Evolving Learning Individuals

In this simulation, we allow lifetime learning, in addition to evolutionary algorithm, to update the phenotype of the individual. To allow for lifetime learning we used the same encoding scheme as in [6,8,9]: Instead of being fully specified, the genotype now is composed of three alleles '0', '1', and '?'. The allele '?' allows for lifetime learning (or plasticity). Each agent will have 1000 rounds of learning during its lifetime. On each round, an individual agent is allowed to do individual learning by changing its allele '?' to either '0' or '1' as the expressed value. Thus, the behaviour of an individual agent is partly specified by its genetic composition, and partly by what it learns in the course of its lifetime.

When an individual learns, it updates its phenotypic behaviour, and hence its fitness. However, instead of being implemented as a blind random search as in previous work [6,8,9], we devise a new learning algorithm as a *hill-climbing* process. The learning algorithm adopted by every individual is presented as Algorithm 1 below.

Algorithm 1. Learning

```
function LEARNING($ind$)
    $best\_fitness = ind.fitness$
    $best\_phenotype = ind.phenotype$
    for $i \in range(1000)$ do
        Flip all question marks to get a new phenotype
        $ind.fitness = compute\_fitness(ind.phenotype)$
        if $ind.fitness > best\_fitness$ then
            $best\_fitness = ind.fitness$
            $best\_phenotype = ind.phenotype$
        end if
    end for
    $ind.fitness = best\_fitness$
    $ind.phenotype = best\_phenotype$
end function
```

The above algorithm is relatively self-explanatory. When an individual expresses a new phenotypic behaviour, it checks if the new behaviour is more adaptive than the current before deciding to replace the current phenotype by the new one. This process helps each agent keeps its best behaviour as the current phenotype.

Please note that, unlike the so-called memetic algorithm and Lamarckian Evolution, learning in our experiments only happens at the phenotypic level, what an individual learns does not change its genotype. The recombination operators work on the genotypic level, so children may inherit question marks from their parents.

After lifetime learning, the population goes through the evolutionary process in the same way as the previous experimental setup. We use the same parameter setting for two experimental setups for a fair comparison. The parameter setting is summarised in Table 1 below.

We run our experiments through 16 different combinations of *frequency* and *magnitude*. It can be understood that the lower the *frequency* value, the faster the target will change; the bigger the value of *magnitude*, the bigger the change of the target. The environment becomes more dynamic or harder to cope with by faster changing and bigger magnitude of change, and vice versa. We also compare the two populations when the environment is static, the target is kept stable over generations. It is interesting that in this sense our problem becomes the canonical ***one-max*** problem – the trivial problem solvable by Genetic Algorithms.

Table 1. Parameter setting

Parameter	Value
Original target	111...1 (20 bits of 1s)
Genome length	20
Replacement	Generational
Generations	51
Elitism	No
Population size	100
Selection	Fitness-Proportionate selection
Reproduction	Sexual reproduction
Mutation rate	0.05
Fitness function	Equation 1
Maximal learning trials	1000
Frequency	2, 5, 10, 20
Magnitude	0.05, 0.1, 0.2, 0.5

4 Results and Analysis

In this section we present the comparison between the two experimental setups, evolving populations with and without learning, in terms of both best fitness and average fitness of the population. All results are averaged over 30 independent runs. The learning population is plotted in green, while the population without learning in red.

4.1 When the Environment is Static

As noted above, our problem becomes ***onemax***-like (as the initial target string is comprised of all ones) when the environment is static – a simple genetic algorithm has been shown to find the correct solution after generations. It can be seen in Fig. 1 that the learning population shows some advantage over initial generations. Over time, the difference between the two populations are smaller and smaller, and becomes zero at the end of the run.

A similar trend can be observed in Fig. 2 for average fitness. The population of learning individuals has some initial advantage over the other in terms of average fitness. Over generations, however, the two populational average fitness are relatively converged.

One plausible explanation for this behaviour is that when the environment is stable, the problem is easy enough (the one-max in this case) so that evolution is sufficient to encode the information of the environment for the population (can find the solution) over time. Therefore, adding learning does not bring much more advantage to the population in terms of both best fitness and average fitness.

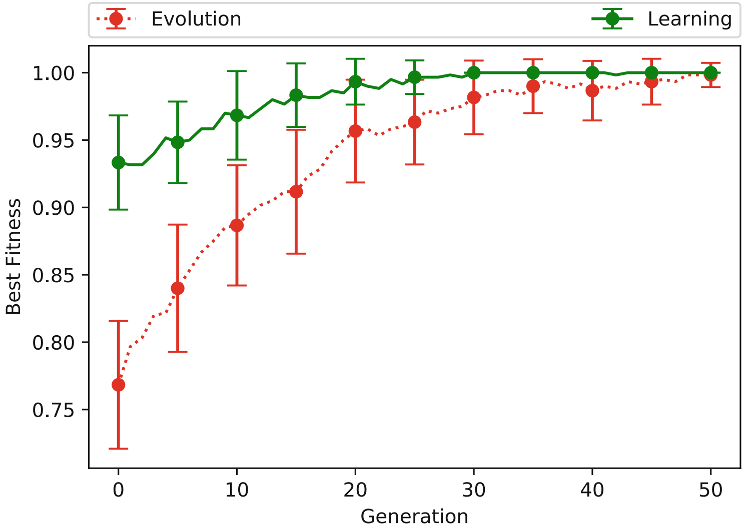

Fig. 1. Best fitness comparison when the environment is static

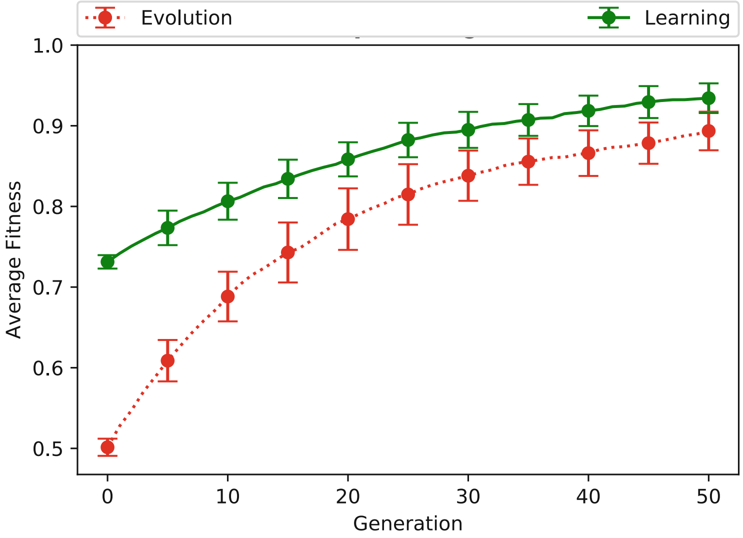

Fig. 2. Average fitness comparison when the environment is static

4.2 When the Environment Changes

The main point of our hypothesis is learning shows benefit over evolution alone when the environment changes. We investigate and demonstrate the effect of learning on evolution when the environment becomes harder to be tackled by the evolving populations. All results are grouped together, sharing the same label

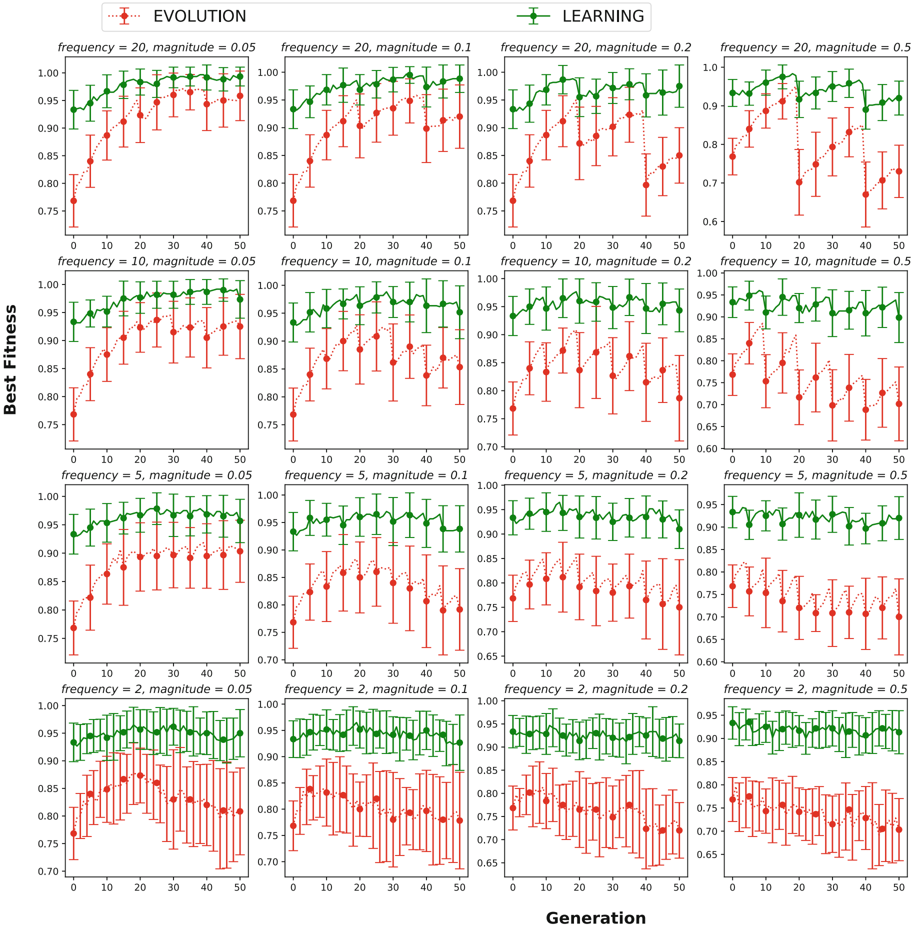

Fig. 3. Best fitness comparison when the environment changes

for x-axis and y-axis as well as the annotation. Each row and column shows the increasing level of difficulty of the problem from left to right and from top to bottom.

It is simply seen in Figs. 3 and 4 that there is a drop in both the best fitness and the average fitness of all settings at the generation when the environment begins to change. This is understandable because when the environment changes, a number of adaptive behaviors from previous generations are no longer fit in the current generation, reducing performance of the population.

A global trend can be observed is that the difference between the population with and without learning becomes bigger, with respect to both best and average fitness, when the environment becomes harder over time in the direction of both frequency and magnitude of change.

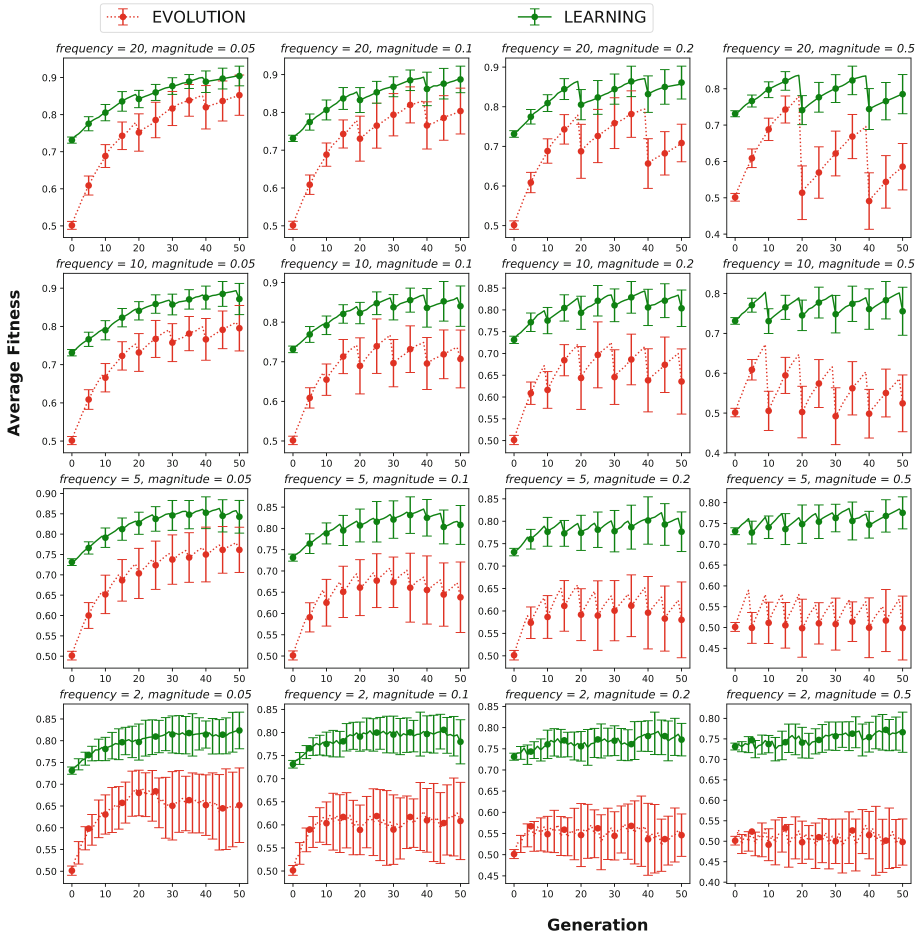

Fig. 4. Average fitness comparison when the environment changes

Specifically, first we look at the best fitness of the population as a measurement of how well each simulation performs. When the frequency of change is 20, the environment changes at a slow rate between generations. In this case, the two populations behave relatively similar in the first 2 instances of the magnitude, and show some divergence at the end of the run when the volumes of change are 0.2 and 0.5 – with %20 and 50% of the target being changed. Looking at the minimum magnitude of change at 0.05, there is little difference between the two populations, even when the environment changes after every 2 generations.

The magnitude of change shows more effects on the performance of the population. When the magnitude of change is bigger (0.2, and especially 0.5 in our experiment), the learning population shows a better performance than the population without learning. This is even clearer when the rate of change between generations is faster.

It can be explained here that when the environment slightly changes, the evolutionary process sill has time to encode new information to keep the population on track over generations (the same clue as the previous section when the environment is static). However, when the environment becomes harder to tackle, the target changes at a sufficiently big level, the evolutionary process looses information about the environment. Therefore, evolution alone cannot track the environment very well because it takes a few to many generations to find a sufficiently good region of the landscape (through genetic operators only and natural selection) to direct the population to. Conversely, in the learning population, individuals are equipped with some sort of learning capability during their lifetime. This means that when the environment changes, they can learn to change their behaviour towards the target solution, apart from some sort of initial behaviour created by evolution. Moreover, it is evolution to create learning capabilities for individuals (plastic allele '?'). Individuals that can learn better behaviour showing more learning capabilities, hence having better fitness values, are favoured by the selection process, leaving more offspring. These offspring inherit both innate behaviour and some sort of learning capabilities from their parent, having more chance to learn during their lifetime, moving their phenotype closer to the target. Learning equips the individual with the capability to track the target, even when the target changes a certain amount, keeping the population on track.

A similar trend can be observed when comparing the average fitness of the two populations and the explanation is the same as above. The difference between the two evolving populations is a bit clearer in terms of average fitness, even when the environment changes slightly, compared to that of the best fitness. It can be explained, again, by the effect of learning capabilities when the environment changes. Without learning, individuals born with the '*wrong*' setting will have low fitness until the end of the generation. With learning abilities, every individuals in the learning population have more chance to update their behaviour to match with the new target, increasing their fitness, hence the average fitness of the whole population. This applies for the whole population, thus the difference in average fitness is a bit clearer between the two evolving populations, with and without learning.

5 Conclusion

In this paper, we have set out to understand the role of learning in evolving populations under different environmental dynamics. For the specific problem (the binary string match problem) and parameter settings, learning has been shown to be beneficial, and more beneficial when there is a bigger change in the environment which happens at a quicker pace. When the environment is stable and easy enough for evolution to cope with, adding learning shows little advantage. Learning is only more advantageous when the environment becomes harder, and approaches the level of difficulty that evolution alone cannot tackle. This observation is the same as shown in previous work [8,9].

In the scope of this paper, we have mainly discussed the adaptive advantage of learning in dynamic environments. The evolution of learning can also been extracted from our results and analyses on the frequency of learners. It is suggested to investigate the question as to how learning evolves more deeply in future work. An easy way to do this is to try different parameters to control the initial proportion of plastic allele (the question mark) in learning population to see how the frequency of plastic changes over different settings.

We admit ourselves that the problem instance used in this paper is quite simple. Future work will investigate the method and verify the findings in this paper on different types of problems and landscapes in dynamic optimisation domains, and compare the evolving learning method with some well-known working methods [4]. Even in this simple problem instance, we suggest more deeply theoretical proof on how much learning contributes to the evolving population.

More interestingly, learning can be classified into two types. The first is social learning, or learning from others, e.g. imitation learning. The second is asocial (or individual) learning – learning by directly interact with the environment, e.g. trial-and-error. The form of learning used in this paper is an instance of asocial learning when individuals sample the environment themselves, trying different combinations of bits by flipping question marks. A learning strategy (or learning rule) is a combination of social and asocial learning in a strategic (probabilistic) way. A learning rule has been shown to be more beneficial than both social and asocial learning alone in several work in the literature, such as [8]. Future work will investigate this.

Acknowledgments. This research is funded by the Science Foundation Ireland under Grant No. 13/IA/1850.

References

1. Ackley, D., Littman, M.: Interactions between learning and evolution. In: Langton, C.G., Taylor, C., Farmer, C.D., Rasmussen, S. (eds.) Artificial Life II, SFI Studies in the Sciences of Complexity, vol. X, pp. 487–509. Addison-Wesley, Reading (1992)
2. Baldwin, J.M.: A new factor in evolution. Am. Nat. **30**(354), 441–451 (1896)
3. Brabazon, A., O'Neill, M., McGarraghy, S.: Natural Computing Algorithms, 1st edn. Springer, Heidelberg (2015)
4. Branke, J.: Evolutionary Optimization in Dynamic Environments. Springer, US (2002)
5. Bull, L.: On the baldwin effect. Artif. Life **5**(3), 241–246 (1999). https://doi.org/10.1162/106454699568764
6. Hinton, G.E., Nowlan, S.J.: How learning can guide evolution. Complex Syst. **1**, 495–502 (1987)
7. Kauffman, S.A., Weinberger, E.D.: The NK model of rugged fitness landscapes and its application to maturation of the immune response. J. Theor. Biol. **141**(2), 211–245 (1989). https://doi.org/10.1016/s0022-5193(89)80019-0
8. Le, N.: Adaptive advantage of learning strategies: A study through dynamic landscape. In: Parallel Problem Solving from Nature – PPSN XV. Springer International Publishing (forthcoming)

9. Le, N.: The baldwin effect reconsidered through the prism of social learning. In: IEEE Congress on Evolutionary Computation, CEC 2018. IEEE Press (8–13 July forthcoming)
10. Mayley, G.: Guiding or hiding: explorations into the effects of learning on the rate of evolution. In: In Proceedings of the Fourth European Conference on Artificial Life, pp. 135–144. MIT Press (1997)
11. Nolfi, S., Parisi, D., Elman, J.L.: Learning and evolution in neural networks. Adapt. Behav. **3**(1), 5–28 (1994)
12. Waddington, C.H.: Canalization of development and the inheritance of acquired characters. Nature **150**(3811), 563–565 (1942)