Evolutionary Intelligence (2021) 14:1565–1579
https://doi.org/10.1007/s12065-020-00425-5

RESEARCH PAPER

Self-adaptive learning for hybrid genetic algorithms

Tarek A. El-Mihoub[1] · Adrian A. Hopgood[2] · Lars Nolle[1]

Received: 5 November 2019 / Revised: 14 April 2020 / Accepted: 7 May 2020 / Published online: 25 May 2020

Abstract
Local search can be introduced into genetic algorithms to create a hybrid, but any improvement in performance is dependent on the learning mechanism. In the Lamarckian model, a candidate solution is replaced by a fitter neighbour if one is found by local search. In the Baldwinian model, the original solution is retained but with an upgraded fitness if a fitter solution is found in the local search space. The effectiveness of using either model or a variable proportion of the two within a hybrid genetic algorithm is affected by the topology of the fitness function and the details of the hybrid algorithm. This paper investigates an intelligent adaptive approach to decide on the learning mechanism to be used by an individual over the course of the search. Evolution is used to self-adapt both the frequency of a steepest-descent local search and the relative proportions of Lamarckian and Baldwinian inheritance. Experiments have shown that this form of adaptive learning can improve the ability to find high-quality solutions and can accelerate the hybrid search without the need to find optimal control parameters for the learning process.

Keywords Hybrid genetic algorithms · Evolution strategies · Learning strategies · Self-adaptive learning · Reinforcement learning · Memetic algorithms · Metaheuristics · Baldwinism · Lamarckism

1 Introduction

The exploitation and exploration abilities of genetic algorithms [1] can be further improved by incorporating local search methods such as steepest-descent search [2]. The capability of local search methods to integrate with genetic algorithms is widely used to solve real-world problems using minimum resources [3, 4]. Hybridization of a genetic algorithm with a local search method can enhance the global search with local knowledge that can guide and accelerate the search towards the global optimum [5]. Such hybrids are often described as memetic algorithms [2].

The way in which the information gained through local search within a hybrid genetic algorithm is used has a strong influence on the performance of the search process [6]. As the local search within the genetic search algorithm plays a similar role to learning within the evolution process, the local search method is often viewed as a learning process.

Local search information is utilized within a genetic search according to strategies that are inspired by biological learning. The Lamarckian and the Baldwinian learning strategies [7] have been widely used [8]. In addition, partial Lamarckism [9–14], which is a hybrid of those basic models [15], has also been used.

The effectiveness of a learning approach depends on the control parameter settings of the hybrid genetic algorithm and the optimization problem to be addressed. The fitness landscape [11], the genetic algorithm setup [12, 16], the percentage of population performing local search [17], the duration of local search [18], and the local search method used [19] are among the factors that impact the effectiveness of a learning model. Applying different mixtures of the Lamarckian and the Baldwinian approaches over the course of the genetic search can be more beneficial than applying a single basic learning model or a fixed mixture of learning models during the entire run [11]. With no strategy available for choosing the learning model that best matches a given black-box problem in a hybrid search, the question

✉ Adrian A. Hopgood
adrian.hopgood@port.ac.uk

Tarek A. El-Mihoub
tarek.el-mihoub@jade-hs.de

Lars Nolle
lars.nolle@jade-hs.de

[1] Jade University of Applied Science, 26389 Wilhelmshaven, Germany

[2] University of Portsmouth, Portsmouth, UK

remains whether the effects of this choice might be mitigated via some form of intelligent adjustment while the search is progressing [20].

The aim of this paper is to investigate the use of an adaptive approach to decide on the learning mechanism. Assigning different learning strategies to the population's individuals over the course of the run using some intelligent means is investigated through applying evolution to self-adapt the learning mechanism within a hybrid genetic algorithm. The effect of this form of adaptation on the hybrid's performance is examined to get some insight into its advantages and disadvantages. This paper also investigates the interactions between this form of adaptive learning and two different adaptive hybrid genetic algorithms.

The paper starts with a very brief review of the learning approaches used in hybrid genetic algorithms. Then it proceeds to describe the proposed adaptation mechanism and the way it works. The paper concludes by presenting and discussing the results of the experiments that have been conducted using two different adaptive hybrid genetic algorithms on a selected set of test functions.

2 Local knowledge and learning models

Local search methods are incorporated into genetic algorithms to improve the algorithm's performance through learning [21]. The utilization of local knowledge of a sampled solution through learning can improve the chances of good building-blocks, i.e. schemata [1], propagating into the next generation even when solutions of below-average fitness are sampled. Learning can also refine sampled solutions to build better building-blocks. Learning can introduce diversity and help to overcome the genetic drift problem [22]. It also enables the fair representation of different search areas and helps to avoid premature convergence. The way in which gained information is used within a hybrid genetic algorithm can influence the performance of the search process. Using an appropriate learning mechanism can accelerate the search towards the global optimum [23, 24]. On the other hand, the use of an inappropriate mechanism can either cause premature convergence or decelerate the search towards the global optimum [25].

The two basic learning models that have been used to exploit local information are the Lamarckian and the Baldwinian approaches [26]. A mixture of the basic models has also been used and its effectiveness has been proven in solving real-world problems [2].

2.1 Lamarckian learning

The Lamarckian approach is based on the inheritance of characteristics that have been acquired through experience and learning. Although this model was widely refuted in biology in the past, more recent advances in epigenetics and their mechanisms have again raised the concept of transgenerational inheritance of acquired traits [27]. Lamarckism is often associated with memes, i.e. conceptual mental units, rather than genes, to avoid contradicting Darwinism [28]. The genetic structure and the fitness are changed to match the improved solution found by a local search method. In the Lamarckian approach, the local search method is used as a refinement genetic operator. Through its action, the genetic structure of an individual is modified and placed back in the genetic population.

Lamarckian evolution can accelerate the search process of genetic algorithms [29–31]. On the other hand, it can interrupt schema processing by changing the genetic structure of individuals, which can badly affect the exploration abilities of genetic algorithms and may lead to premature convergence [6]. The Lamarckian approach requires inverse mapping from the phenotype, on which learning works, to genotype, on which evolution works. The inverse mapping can be difficult to achieve for real-world problem solving [32].

2.2 Baldwinian learning

Baldwinian learning is similar to Lamarckian learning in its use of local search, but with a different mechanism. As with Lamarckian learning, it allows an individual's fitness to be improved by applying a local search and hence raising its chances to propagate its genes to subsequent generations. However, unlike Lamarckian learning, the learning process does not change the individual's genetic structure. As in natural evolution, the learning process increases an individual's chances of survival without any genetic changes. So Baldwinian learning does not allow parents to pass their acquired characteristics to their offspring but, instead, the potentially enhanced fitness after learning is retained. The local search method uses local knowledge to produce a new fitness score that can be used by the genetic algorithm during selection for reproduction. The increase in the fitness through local search is equivalent to the individual's ability to improve.

The Baldwin effect can accelerate the genetic acquisition of learned traits that are acquired during an individual's lifetime, without any direct Lamarckian transfer of traits from phenotype to genotype. With a strong correlation between genotype and phenotype space, the Baldwinian learning can speed up the genetic acquisition of learned traits [33] indirectly through the genetic assimilation mechanism [34].

The Baldwin effect can transform a challenging fitness landscape, associated with a difficult optimization problem, into an easier one [7] and enable fair representation of different search areas. The hindering effect [8] associated with the Baldwinian search can hinder the evolution process by obscuring genetic differences [33]. Compared with the Lamarckian approach, the Baldwinian strategy can follow a protracted path towards the global optimum but can nevertheless be more effective [6]. It can aggravate the problem of multiple genotype to phenotype mappings [11, 18], unlike the Lamarckian approach, which has been shown to help alleviate this problem [35]. While the Baldwinian strategy can be more suitable for dynamic environments, the Lamarckian strategy can suit static environments [36].

2.3 Hybrid Lamarckian–Baldwinian models

Hybrid models of Lamarckian and Baldwinian learning are designed to combine the advantages of both forms of learning models [14]. The hybridization can be done in two different ways. Firstly, some individuals evolve using the Lamarckian approach, while the other individuals evolve using the Baldwinian approach [11, 17]. Secondly, some genes evolve using the Lamarckian method and the remaining genes evolve using the Baldwinian method [15].

In a selected set of test problems, the individual-level approach outperformed both the pure Lamarckian and the pure Baldwinian approaches [9–11, 17]. On the other hand, the gene-level approach has been shown to reduce the search space and produce an efficient search when tackling the sorting-network problem [15].

The selected form of learning in a hybrid genetic algorithm has been shown to affect its performance [37]. Several researchers have compared the performance of pure genetic algorithms with alternative learning strategies in hybrid genetic algorithms [18]. The effectiveness of the Lamarckian approach, the Baldwinian approach, or a mixture of the two, depends on the details of the hybrid genetic algorithm and the problem to be solved.

The optimal fraction of the population that should evolve using the Lamarckian approach depends on the problem at hand. Orvosh and Davis [14] have shown that a 5% Lamarckian strategy is optimal for solving the survival-network design problem and the graph-colouring problem. Michalewicz and Nazhiyath [16] used the equivalent of a 20% Lamarckian strategy when replacing 20% of the repaired solutions in a hybrid algorithm for the solution of numerical optimization problems with nonlinear constraints. In a selection of problems, Houck et al. [11] found that 20% and 40% partial Lamarckian strategies produced the best balance of solution quality and computational efficiency. Joines et al. [35] found that a pure Lamarckian approach (i.e. 100%) produced the fastest convergence to the best known solution of the cell-formation problem. Espinoza et al. [9, 10] used a 75% partial Lamarckian learning strategy, which they considered optimal in their algorithm for optimizing two simple continuous functions. Ishibushi et al. [12] showed that the 5% partial Lamarckian was effective for a single-population model when tackling the multi-objective 0/1 knapsack problem, yet the 50% partial Lamarckian was the best choice for the island model.

3 Evolutionary self-adaptation of the learning strategy

It is almost impossible to know which learning strategy is most suitable for a problem when there is only limited available knowledge of the topology of the fitness landscape. Given the restricted amount of theory currently available for selecting the learning strategy for a hybrid search in these circumstances, the use of an effective adaptive technique to decide on the learning strategy while the search is processing would clearly be beneficial.

The idea behind the adaptive strategies is that, as the search progresses, the effectiveness of each learning strategy in dealing with the current problem can be learned. Knowledge about the current population of solutions and each learning strategy can be built dynamically online, thereby identifying the strengths and weaknesses of the learning approach for the problem currently being worked on, given its current state.

The use of evolution to self-adapt the learning mechanism in this way can help to discover the effectiveness of each learning approach in relation to a given problem. This kind of adaptation can improve the hybrid's chances of finding good solutions by enabling different learning approaches to compete and cooperate with each other. By encoding the learning strategy used by an individual into its chromosome, the global genetic algorithm can promote competition among the different learning strategies based on its ability to improve the fitness of its associated solution. A good learning strategy will lead to good individuals and these will probably have more chance to survive and to propagate the encoded learning approach. Applying the evolutionary self-adaptation metaphor to decide on the learning strategy can also promote cooperation between the two basic learning models in order to improve the search's performance. The use of a suitable learning approach depending on the genetic structure of an individual, and the current search state, may lead to a search algorithm that makes use of the available learning strategies to improve the whole population's performance. By ensuring the participation of the two basic learning models in the problem search, the strategy promotes joint operation and hence cooperation between learning models.

4 The algorithms

The evolutionary self-adaptive learning mechanism was incorporated into two hybrids with different mechanisms for deciding between global and local search, in order to investigate its effect on their performance. It was incorporated into the novel Self-Adaptive local-search-Duration Hybrid (SADH) algorithm, based on three previously presented hybrids [8], where the number of local iterations that should be performed by an individual is encoded into its chromosome. It was also used with Espinoza's [9, 10] self-adaptive hybrid genetic algorithm (SAHGA) for comparison. This latter algorithm is referred to here and elsewhere [17] as the Adaptive Staged Hybrid (ASH) as it is a staged hybrid algorithm according to the definition by Mathias et al. [38] and it adapts by controlling the probability of the local search using a deterministic rule.

In the SADH algorithm, the representation of an individual incorporates the number of local search iterations. This parameter defines the number of local iterations that should be performed by the associated individual, thereby controlling the duration of the local search. At the global level, this parameter is evolved by the genetic algorithm and used as a control parameter to optimize the fitness-function variables. The global genetic algorithm decides dynamically, by evolutionary self-adaptation, which individuals will perform a local search. It also cooperates with the local search to determine the duration of the local search method by modifying the number of local iterations. In this way, the control parameter that determines the number of local search iterations can be adapted without external control.

The ASH algorithm decides between global and local search using feedback from the current state of the search process. It restricts the performance of local search to situations in which local knowledge can help to guide the search and new regions of search space are being discovered. A deterministic rule controls the probability of the local search and keeps it below a specific value. When the improvement in the average fitness using local search is no greater than the most recent global search iteration, the model reverts to global search.

The evolutionary self-adaptive learning mechanism was implemented by appending an additional bit that represents the learning strategy to the end of an individual's chromosome. In the SADH algorithm, the learning strategy bit is located after the genes that represent the number of local search iterations. For the ASH algorithm, it is located after the genes that represent the function variables. After performing a local search operation and before returning to the global genetic algorithm, the algorithm reads the value of the learning strategy bit. It does so in order to decide whether to change the genetic structure and the fitness score of the initial solution to match that of the improved solution or to keep its genetic structure unchanged and to modify the fitness score only. Depending on the value of the learning strategy gene, the hybrid decides on the learning strategy to use in order to utilize local search information of a given solution.

The association of the learning strategy with a solution through binding them into the same chromosome can help to associate the success or the failure of a learning technique to a specific solution or solutions of similar genetic structures. However, in cases where the learning strategy changes significantly during evolution, the encoded strategy will only represent a snapshot of the strategy at one point during the evolution process. Table 1 shows the pseudocode of the evolutionary learning algorithm with the SADH.

5 Experiments

In order to evaluate the proposed learning adaptation mechanism, it was incorporated into the two adaptive hybrid algorithms mentioned above. The performance of the resulting algorithms was compared with the performance of the two hybrids using fixed percentages of partial Lamarckian. The quality of the solutions produced by each algorithm and the speed of convergence were used to assess the algorithm's performance. The percentage of experiments that successfully converged on the global

Table 1: Pseudocode of evolutionary learning algorithm with SADH

```
Start
set(CrossoverRate)
set(MutationRate)
t=1
initialize(Population(t))
evaluate(Population(t))
While termination criteria not satisfied
  MatePool(t)=select(Population(t))
  MatePool(t)=crossover(MatePool(t))
  evaluate(MatePool(t))
  For each Chromosome(i) of the MatePool(t)
    LocalIterations=getNumberOfLocalIterations(chromosome(i))
    If LocalIterations > 0 then
      Solution(i)=mapToPhenetype(chromosome(i))
      For k=1 to LocalIterations
        Solution(i)=performLocalSearch(Solution(i))
      End (For k=…)
      LearningStrategy=getLearningStrategy(Chromosome(i))
      Fitness(Chromosome(i))=Fitness(Solution(i));
      If LearningStrategy = "Lamarckian"
        Chromosome(i)=mapToChromsome(Solution(i));
      End (If Learning…)
    End (If LocalIter…)
  End (For each)
  t=t+1
  Population(t)=MatePool(t-1);
End (While)
print(Results)
Stop
```

optimum was used as an indicator of the ability of the proposed adaptive learning mechanism to produce high-quality solutions. The performance was compared using different population sizes in order to evaluate the ability of the proposed mechanism to adapt to different search environments. The population sizes used were 50,100, 150, 200 and 250. The speed of finding the global optimum was also used to evaluate the performance of the self-adaptive learning strategy. The learning strategy was also evaluated in terms of its ability to adapt to different fitness landscapes.

A set of test functions has been chosen to evaluate the use of evolution to self-adapt the learning strategy. Four test functions have been used to evaluate the ability of this form of adaptation to improve the search performance. The test functions suite includes the 10-dimensional ellipsoidal, the 10-dimensional Rastrigin, the 10-dimensional Schwefel, and the 10-dimensional Griewank test functions [39]. A string of 16 bits was used to represent each function variable.

The results of optimizing the test functions using the self-adaptive learning mechanism were evaluated against the results obtained by using fixed learning strategies. The fixed learning strategies tested were the pure Baldwinian (0% partial Lamarckian), the 25% partial Lamarckian, the 50% partial Lamarckian, the 75% partial Lamarckian and the pure Lamarckian (100% partial Lamarckian).

The hybrids used an elitist genetic algorithm with binary tournament selection, two-point crossover, and simple mutation with independent identically distributed mutation at each gene. The crossover probability was 0.7 and the mutation probability was $1/N$ where N was the population size. For the ASH algorithm, the number of local iterations was limited to three, e was set to 0.2 (i.e. the probability of local search reduced by 20% after each local iteration), and the local threshold value was 0.6 [9, 10]. The ASH algorithm was tested using three different values of initial local search probability: 0.1, 0.2, and 0.99. For the SADH algorithm, the maximum number of local search iterations was set to 3.

For all experiments, the stopping criterion was the number of function evaluations. This parameter was set to 2000 times the population size for the Rastrigin, the Schwefel, and the Griewank test functions, and to 500 for the ellipsoidal test function due to its simplicity. Each experiment was repeated 50 times.

Steepest descent was used as the local search method. It was combined with Brent's method [40] to estimate the best step size towards the local optimum from the current position. Whereas the standard steepest-descent algorithm uses the derivatives of the fitness function to estimate the best step size, Brent's method fits a parabola to three initial solutions and uses the maximum of the parabola as the next potential solution of the overall function [40].

6 Results

6.1 Search effectiveness

The percentage of times a hybrid algorithm found a global optimum using the proposed adaptation mechanism is compared with that found by using fixed learning strategies. These percentages were used to evaluate the effectiveness of the proposed adaptive learning mechanism.

Figure 1 compares the different hybrid algorithms using the proposed adaptive learning mechanism with those using fixed learning strategies. The comparison is based on the percentage of times that the algorithm found the global optimum of the ellipsoidal function. The graphs show that combining the adaptive learning technique with the ASH algorithm produced a better performance than that produced by combining it with the pure Baldwinian approaches for different initial probabilities of local search and different population sizes. The combination produced a performance that is similar to that produced using the 25% partial Lamarckian. The graphs of the SADH algorithm illustrate that the adaptive learning mechanism produced the best performance for population sizes of 150, 200 and 250 compared with that produced by using different fixed learning strategies.

The results of the experiments that evaluate the effect of the self-adaptive learning mechanism on the search effectiveness on the 10-dimensional Griewank test function are shown in Fig. 2. The plots illustrate that, for most of the tested population sizes and probabilities of local search, the performance of the ASH algorithm was improved when combined with the self-adaptive learning compared with when combined with fixed learning techniques. The results of applying the adaptive mechanism to the self-adaptive hybrid algorithm show that this mechanism outperformed the pure Baldwinian and the 25% Lamarckian approaches for all the tested population sizes.

The self-adaptive learning mechanism when applied to optimize the 10-dimensional Rastrigin function using the two hybrids produced a similar performance in terms of the number of experiments that found the function's global optimum as shown in Fig. 3. The two hybrids when combined with self-adaptive learning produced a performance similar to that produced by the partial 25%, 50%, 75% and 100% Lamarckian approaches, which produced the best performance for most of the tested population sizes.

Since the self-adaptive learning mechanism encodes the learning strategy into the chromosomes, the original problem is transformed to a new problem with an additional dimension. Introducing an additional variable to optimize in the problem can raise the minimum population size for a genetic algorithm. Using small populations with the

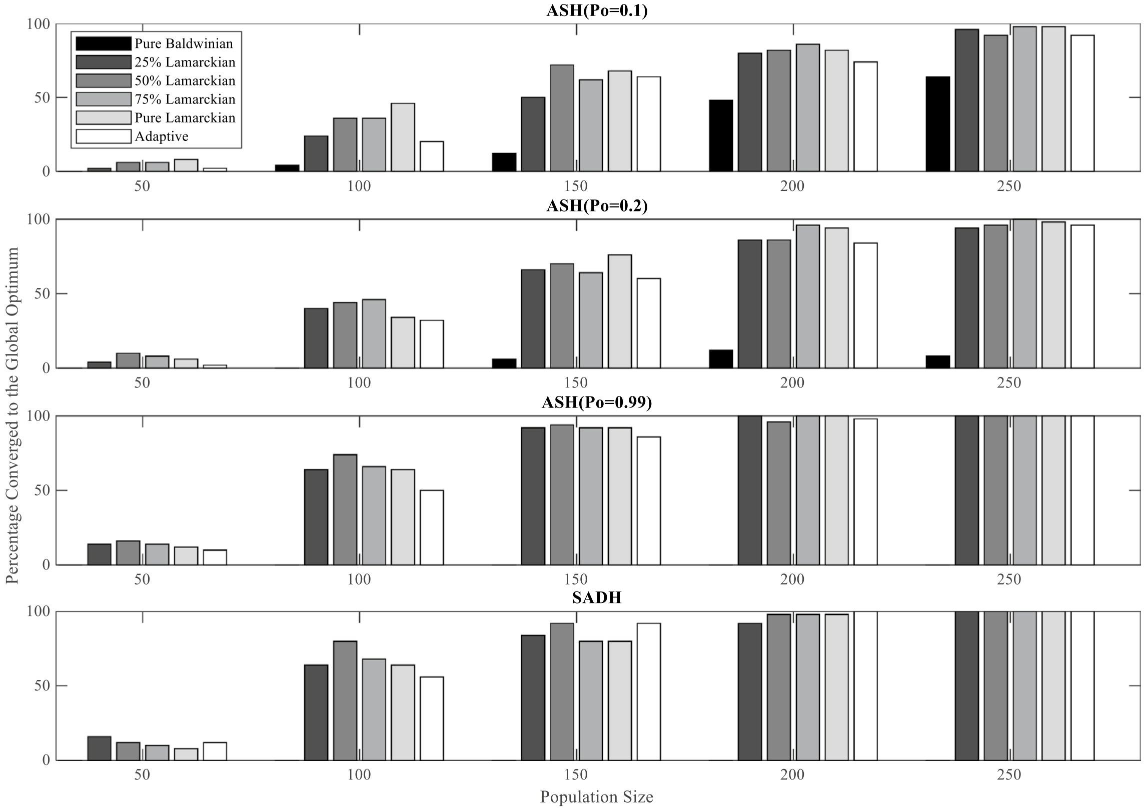

Fig. 1 Percentages converged to the global optimum of the ellipsoidal function

adaptive learning approach might not enable evolution to the global optimum within a given allocation of function evaluations, resulting in some small differences that can be seen in Fig. 3. As the combination of the self-adaptive strategy with the SADH introduces an additional variable to the search space, larger population sizes are required for good performance.

The self-adaptive learning technique outperformed the pure Baldwinian approach in terms of solution quality when applied to solve the 10-dimensional Schwefel problem using the different adaptive hybrid algorithms as depicted in Fig. 4. The self-adaptive learning mechanism, when combined with the ASH algorithm, outperformed the partial 25% Lamarckian approach in all experiments except the one that combined a population size of 200 with a local search probability of 0.1. No results are shown for 0.99 probability of local search with the Schwefel function as such a high probability prevented achievement of the global optimum. The graphs for the ASH algorithm show that there is no significant difference between the performance of this adaptation mechanism and the best found fixed learning strategy in about half of the tested combinations of population sizes and local search probabilities.

The plots of the SADH algorithm show that the adaptive learning strategy technique outperformed pure Baldwinian learning, which produced the worst performance in these experiments. This observation can be explained by the fact that allowing a small fraction of the population to evolve according to Lamarckian learning can help to alleviate the hindering effect, which in turn improves the possibility of finding the global optimum.

6.2 Search efficiency

The number of function evaluations needed by a hybrid algorithm to find the global optimum of a specific function was used to measure the effect of the self-adaptive learning mechanism on the search efficiency. This number is the sum of function evaluations used by the genetic algorithm and that used by the local search. The convergence speed of the hybrids that use the adaptive learning technique was compared with those using fixed learning strategies.

The results of these experiments are shown in Figs. 5, 6, 7 and 8. The graphs compare the speed of the self-adaptive learning technique with the selected set of fixed learning strategies. The graphs of the pure Baldwinian learning strategy were excluded from these figures since in most of the cases it failed to find the global optimum.

Figure 5 compares the performance of the self-adaptive learning mechanism with different fixed learning strategies

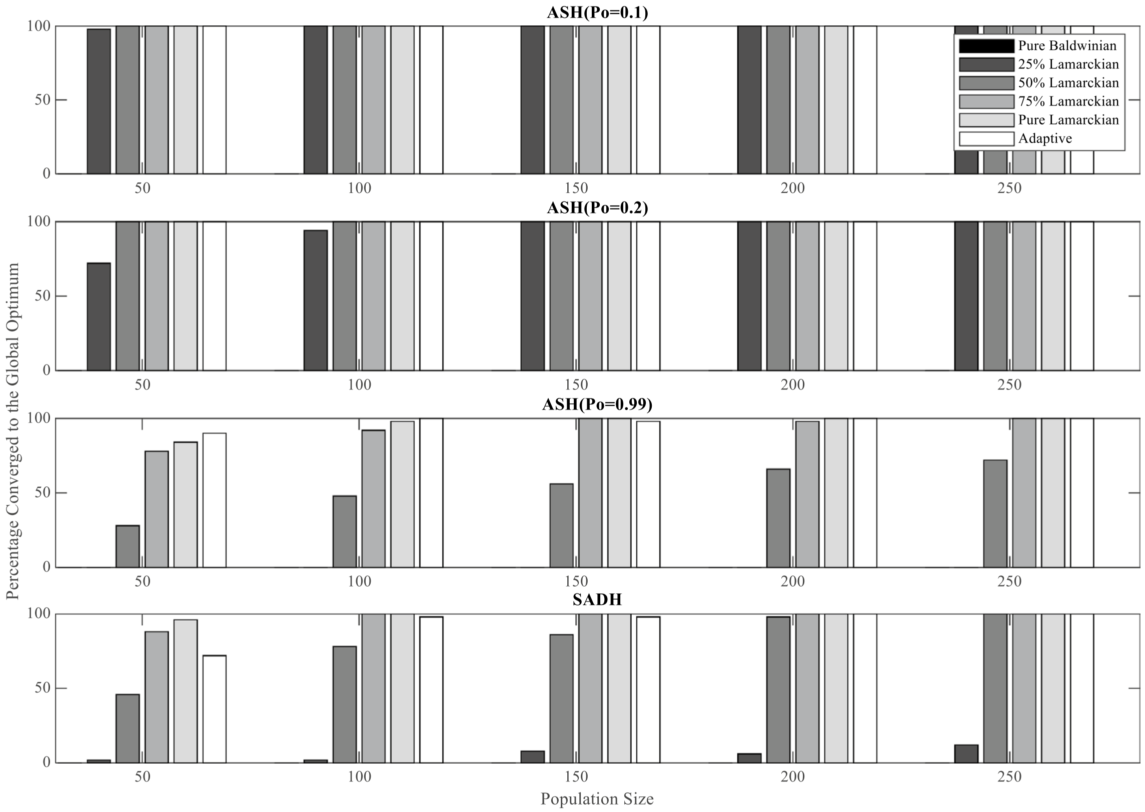

Fig. 2 Percentages converged to the global optimum of the Griewank function

on the ellipsoidal function in terms of convergence speed. The graphs for the ASH algorithm show that the adaptive learning technique found the global optimum of the ellipsoidal function faster than those using fixed learning strategies. The difference in the convergence speed becomes apparent as the population size increases. However, the self-adaptive learning technique, when combined with the SADH algorithm, produced the worst performance compared with the fixed learning techniques excluding the pure Baldwinian approach.

The results of comparing the convergence speed of the adaptive learning with the fixed learning approaches on the Griewank function are shown in Fig. 6. These graphs show no significant difference between the adaptive and the fixed learning strategies when combined with the ASH algorithm regardless of the local search probability. However, the adaptive learning technique produced the worst performance when combined with the SADH algorithm in comparison with the fixed learning strategies, excluding the pure Baldwinian approach.

The graphs in Fig. 7 show the speed of finding the global optimum of the Rastrigin function. It can be seen that the adaptive learning strategy is the second fastest learning mechanism when combined with the ASH algorithm using a local search probability of 0.2. However, it is the second slowest when used with a local search probability of 0.99. The curves also illustrate that the self-adaptive learning mechanism produced the worst performance when combined with the SADH algorithm.

The adaptive learning mechanism was almost the fastest in finding the global optimum of the Schwefel function when combined with the two adaptive hybrid algorithms and using different population sizes, as illustrated in Fig. 8. These graphs also show that the proposed learning mechanism was the fastest when combined with the ASH algorithm regardless of the probability of local search used.

7 Discussion

The ability of the self-adaptive learning strategy to adapt to different fitness landscapes and population sizes was evaluated through monitoring the changes in the learning strategy over time. The graphs of the evolution of the learning strategy and the best fitness for two test functions and two population sizes are presented and discussed in this section. The changes in the percentages of the population that used the different local search iterations over generations are also presented in the case of the SADH algorithm. Each graph shows the results of 50 experiments.

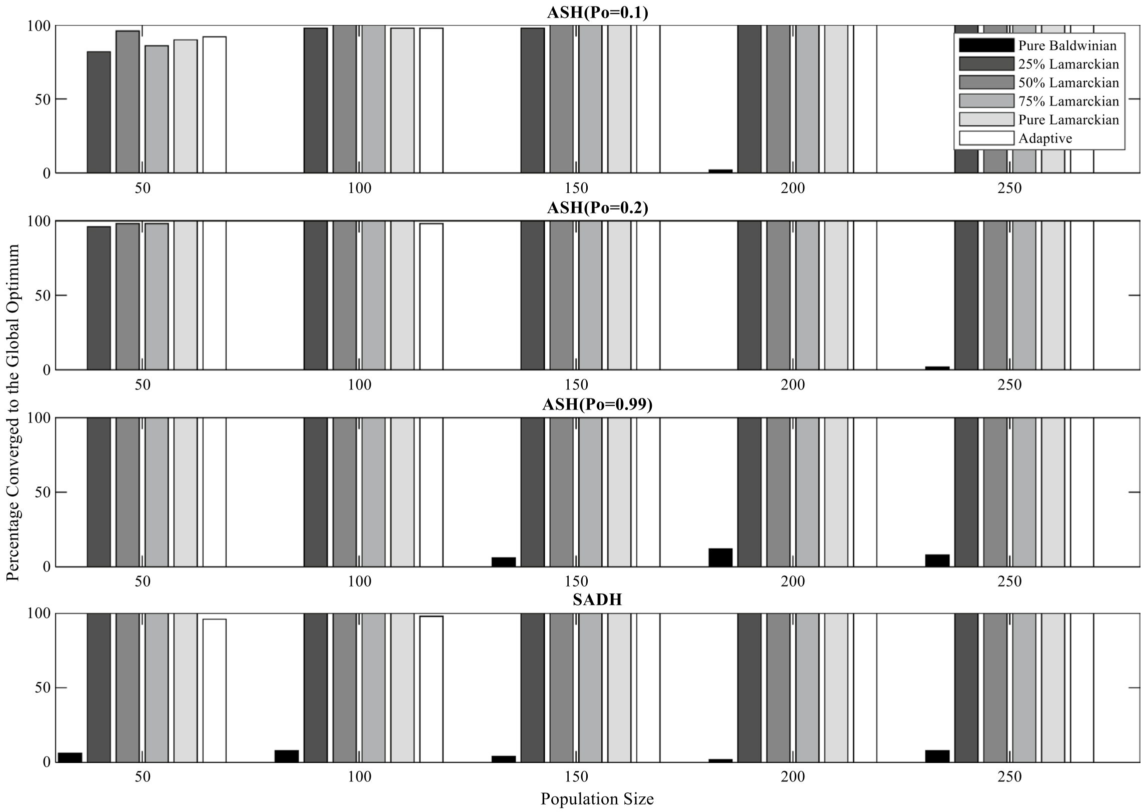

Fig. 3 Percentages converged to the global optimum of the Rastrigin function

Figure 9 shows the evolution of the best fitness, the learning strategy and the duration of local search of the SADH algorithm when used to solve the ellipsoidal problem. The graphs illustrate that, at the early stages of search, the fraction of the population that evolved using the Baldwinian learning increased slightly. After that, the percentage of the population that used Lamarckian learning increased as the search progressed until the whole population was using the Lamarckian approach. The figure clearly shows that the increase in the number of individuals using the Lamarckian approach is accompanied by an increase in the use of long durations of local search.

The graphs of the Rastrigin function in Fig. 10 show a similar trend to that found in Fig. 9. They show that the fraction of the population that evolved using the Lamarckian approach increased as the search progressed. Figure 10 also shows that there is a trend to use short durations of local search accompanied with the use of the Baldwinian search. This trend is apparent at the final stages of the search using a population size of 100, where individuals tended to use short durations of local search to overcome the hindering effect problem associated with the Baldwinian strategy. This observation clearly shows the ability of the self-adaptation mechanism to discover the relations between different control parameters such as the relations between the learning strategy and the duration of local search. The difference in the number of function evaluations consumed by each local search process caused the algorithm to evolve to different numbers of genetic generations. For example, the graphs for the population size of 100 in Fig. 10 show that most of the experiments consumed their allocation of function evaluations by the 11th generation. The graphs also illustrate that the evolution trajectory of the learning strategy in the Rastrigin function is more complicated than that of the ellipsoidal function.

Figure 11 shows how the learning strategy evolved over time when combined with the ASH algorithm to solve the ellipsoidal function using a population size of 100. The figure shows the same trend as the figures of the SADH algorithm. As the search progressed, the number of solutions that evolved according to Lamarckian learning increased. The plots show that, for a local search probability of 0.1, the proportion of the population that evolved using the Lamarckian approach was in the range 70–100%. However, for a local search probability of 0.99, the proportion approached 100% as the search progressed. This difference can be explained by the fact that using a small probability of local search can help to fight the hindering effect, which in turn enables the algorithm to find the global optimum even when combined with the Baldwinian approach. On the other hand, the

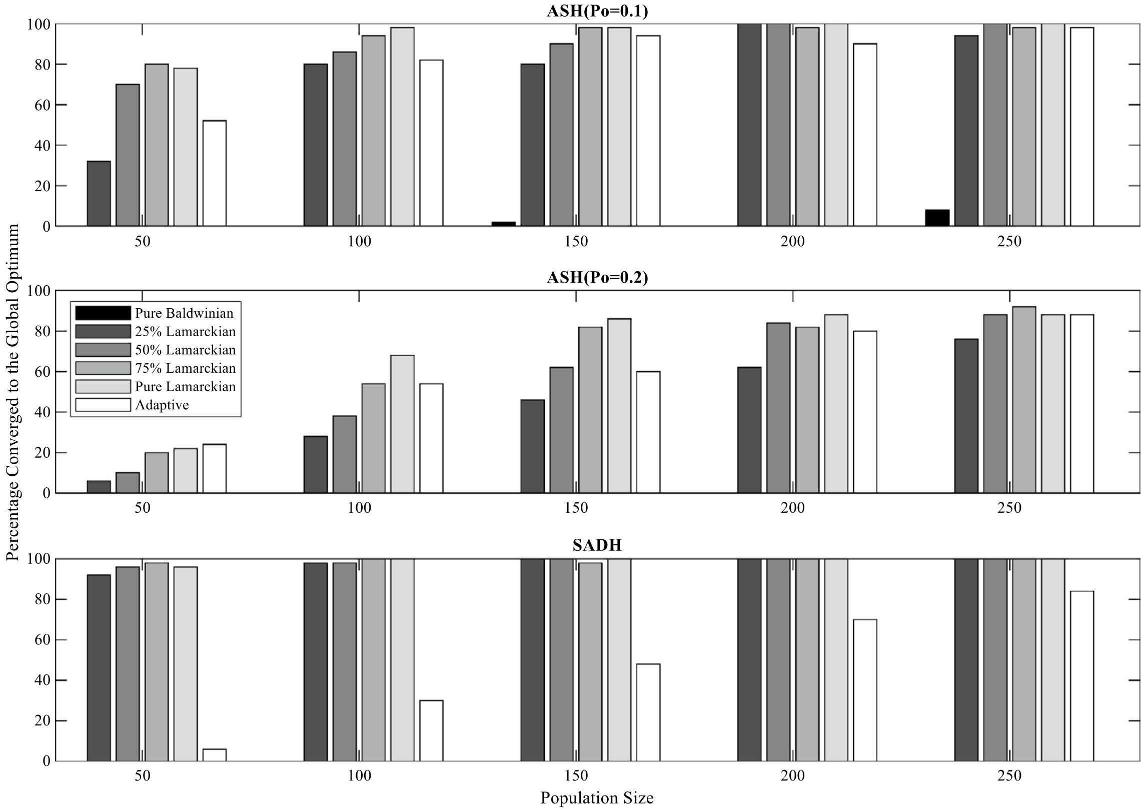

Fig. 4 Percentages converged to the global optimum of the Schwefel function

probability of finding the global optimum increases with the increase in the partial Lamarckian approach for high probabilities of local search once the algorithm has guided the search towards the global optimum's basin of attraction. These graphs also demonstrate that the hybrid that used a local search probability of 0.1 was faster than the one using a probability of 0.99 in optimizing the learning strategy control parameter. This improvement was because the algorithm has more chances to evolve in the first case than in the latter one, due to the differences in the number of function evaluations consumed at each local search process.

Figure 12 shows the results of optimizing the Rastrigin function using a population size of 200. The fitness landscape of this function is more complicated than that of the ellipsoidal function. This, in turn, makes the evolution path of the learning strategy more complicated. The graphs also show that as the local search probability increases, the trend towards using more Lamarckian learning increases.

A binary coding scheme has been used in this work. Changing the coding scheme would change the fitness landscape of the problem, the effects of which would depend on the details of the scheme and the operators used. Since each individual has only two options for learning, i.e. Lamarckian or Baldwinian, evolution of a learning strategy is expected to be effective with a small overhead cost. Encoding the learning strategy within the chromosome eliminates the need for tuning the learning strategy as a control parameter. However, it increases the influence of other control parameters such the crossover rate and mutation rate on the evolution path of the search algorithm.

8 Conclusion

The experiments conducted illustrate that the use of the self-adaptive learning strategy can be beneficial. It can improve the search's ability to find solutions of high quality and can accelerate the search. The experiments also show that this mechanism was able to adapt to different environments. That finding was illustrated by testing this mechanism on a set of different test functions using two different adaptive hybrid algorithms with different control parameters. A further advantage of self-adaptive learning is that there are fewer control parameters to set manually.

These experiments demonstrate that combining the self-adaptive mechanism with the ASH algorithm produced an algorithm that is faster than the tested fixed learning strategies on most of the tested functions. However, combining this mechanism with the SADH algorithm produced a slower search algorithm. The latter combination was nevertheless

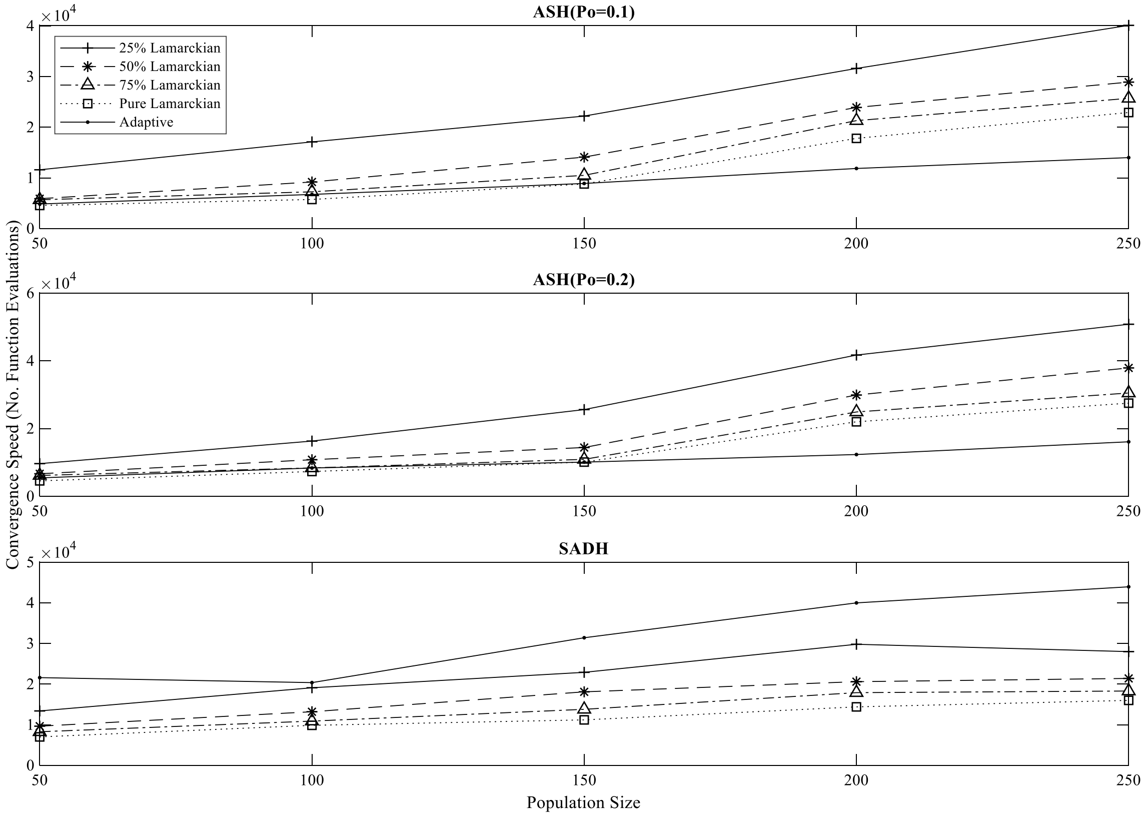

Fig. 5 Convergence speed of the ellipsoidal function

able to find the global optimum of the whole set of test functions more often and faster than the fixed pure Baldwinian approach.

As further work, investigations could be conducted into the benefits of applying the proposed self-adaptive learning approach to both static and dynamic real-world problems. More research is also needed into the balance between the benefits of hybrid algorithms with evolving control parameters and the computational cost of the additional complexity.

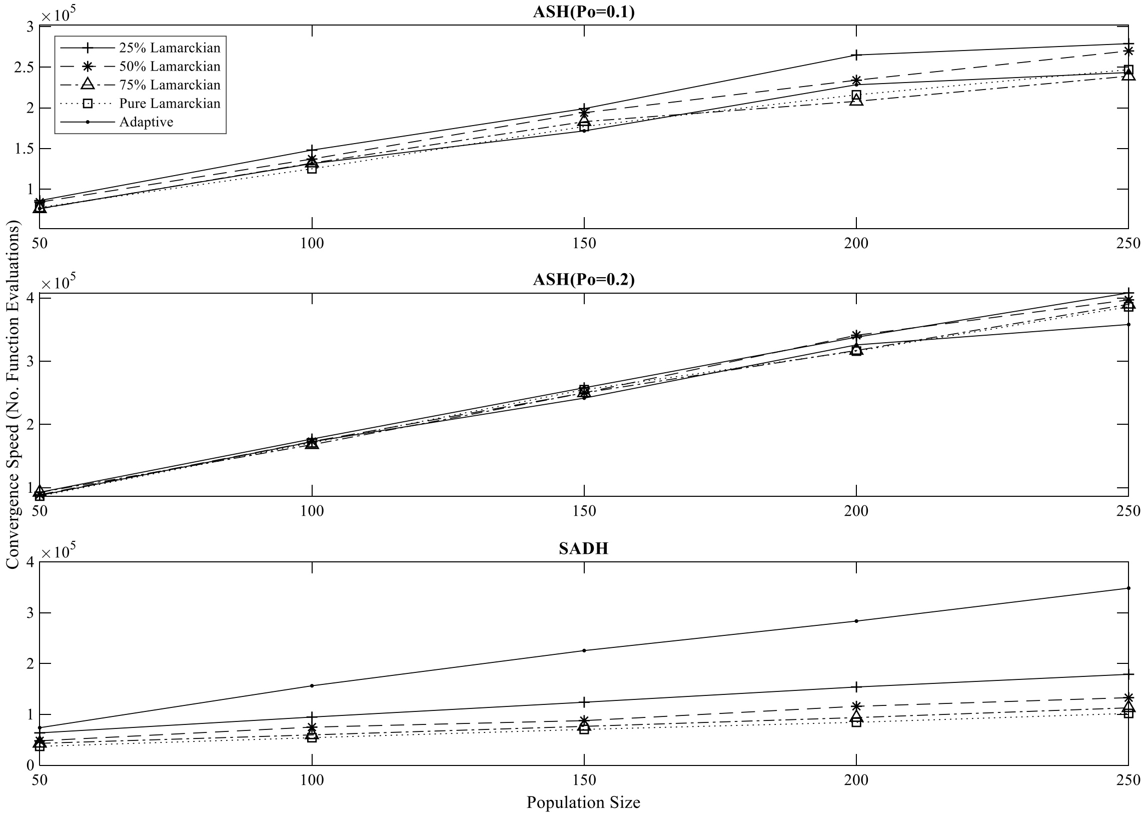

Fig. 6 Convergence speed of the Griewank function

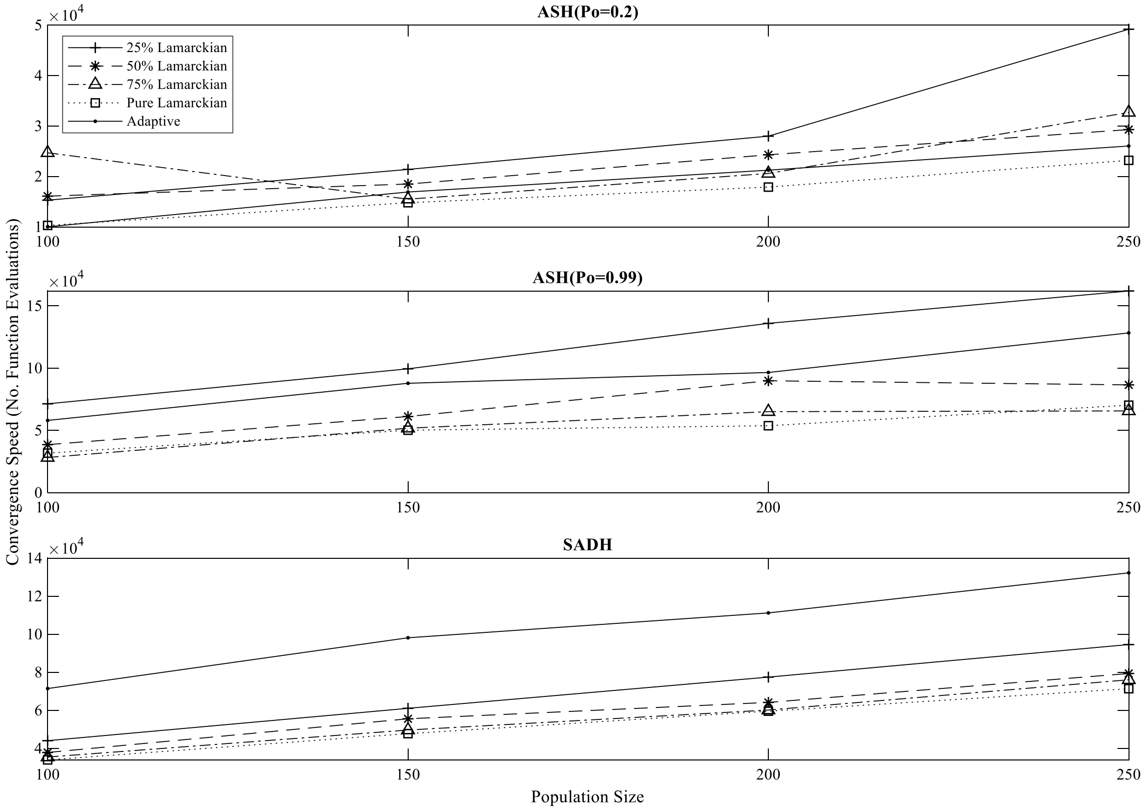

Fig. 7 Convergence speed of the Rastrigin function

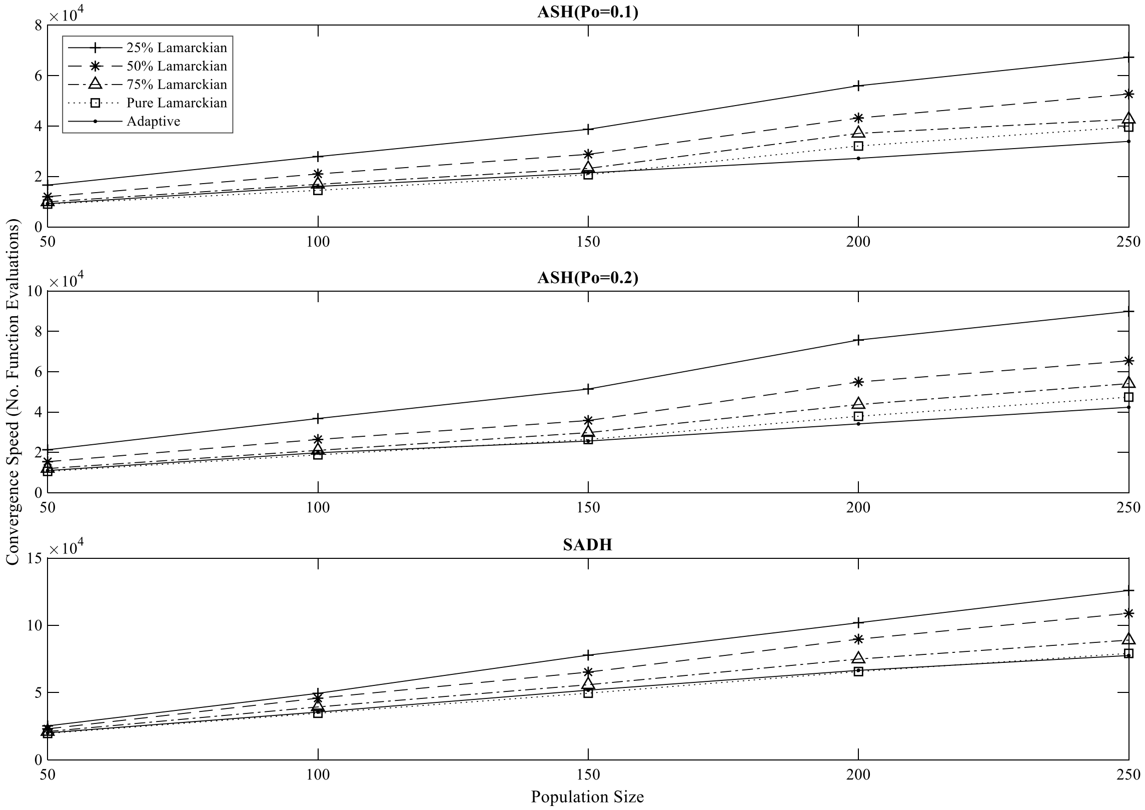

Fig. 8 Convergence speed of the Schwefel function

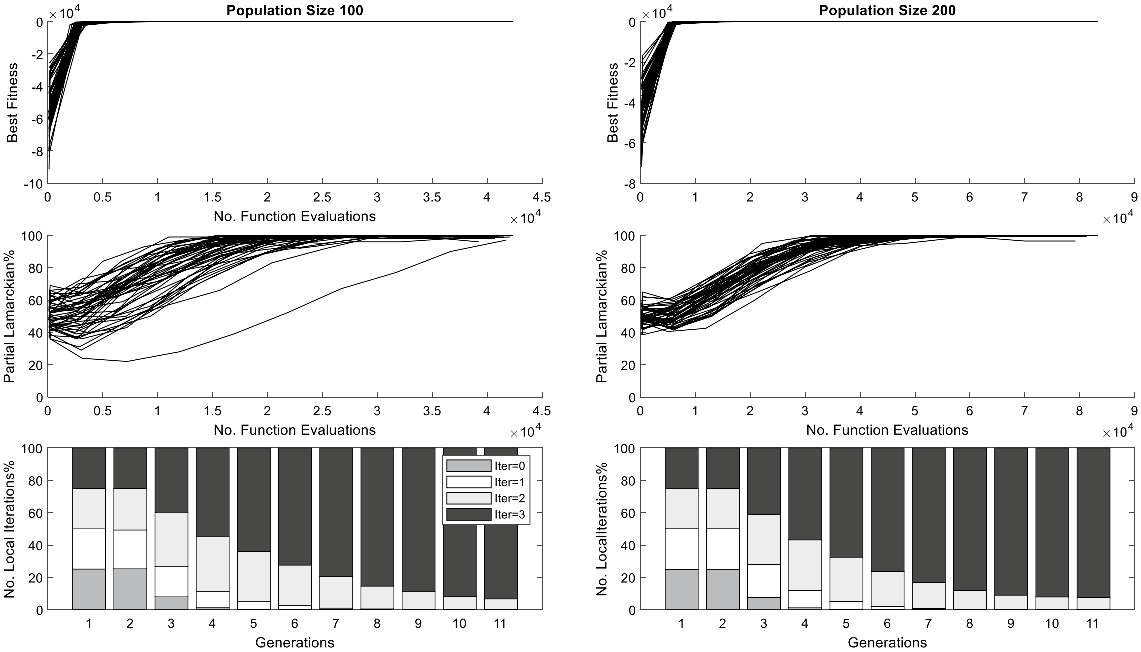

Fig. 9 Evolution of the learning strategy when solving the ellipsoidal function using the SADH algorithm

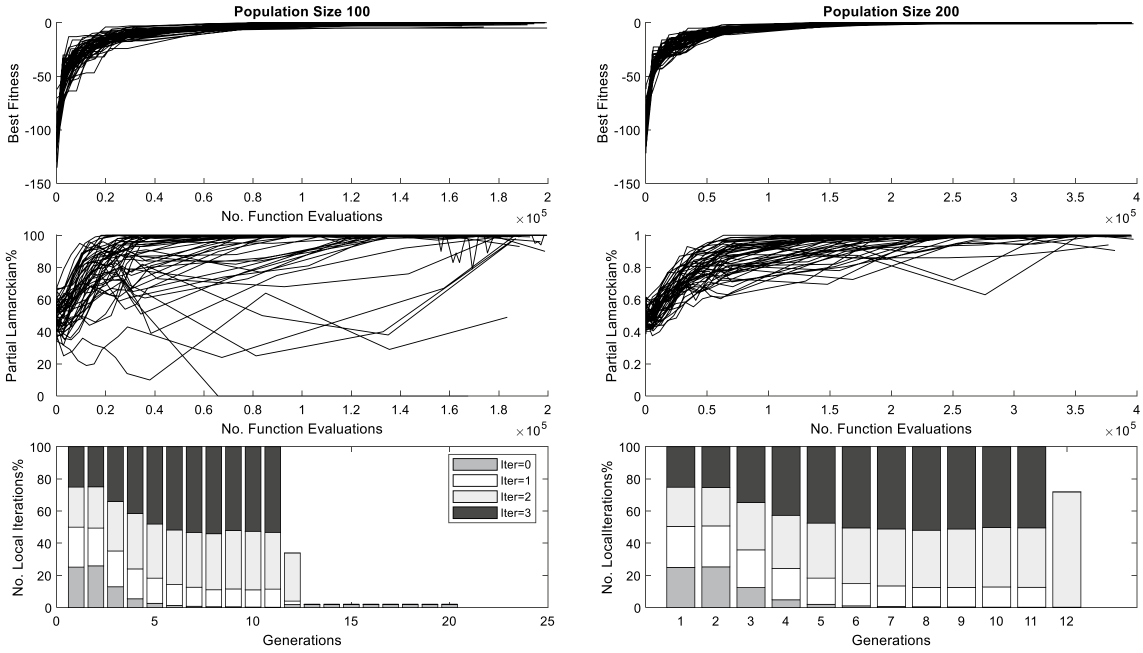

Fig. 10 Evolution of the learning strategy when solving the Rastrigin function using the SADH algorithm

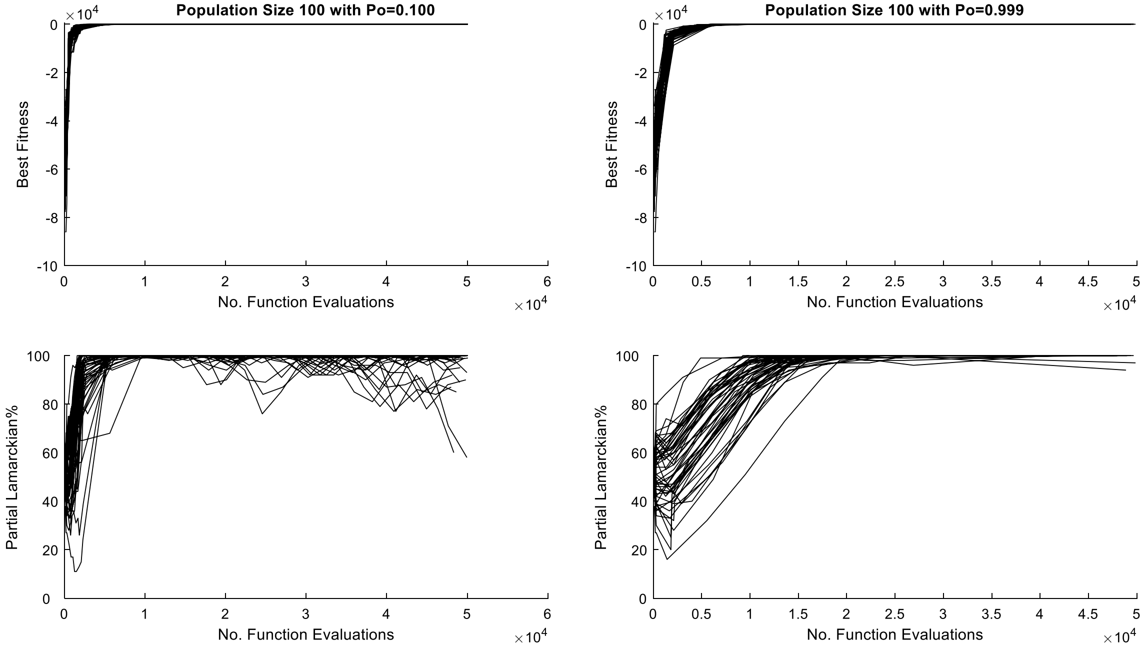

Fig. 11 Evolution of the learning strategy when solving the ellipsoidal function using the ASH algorithm

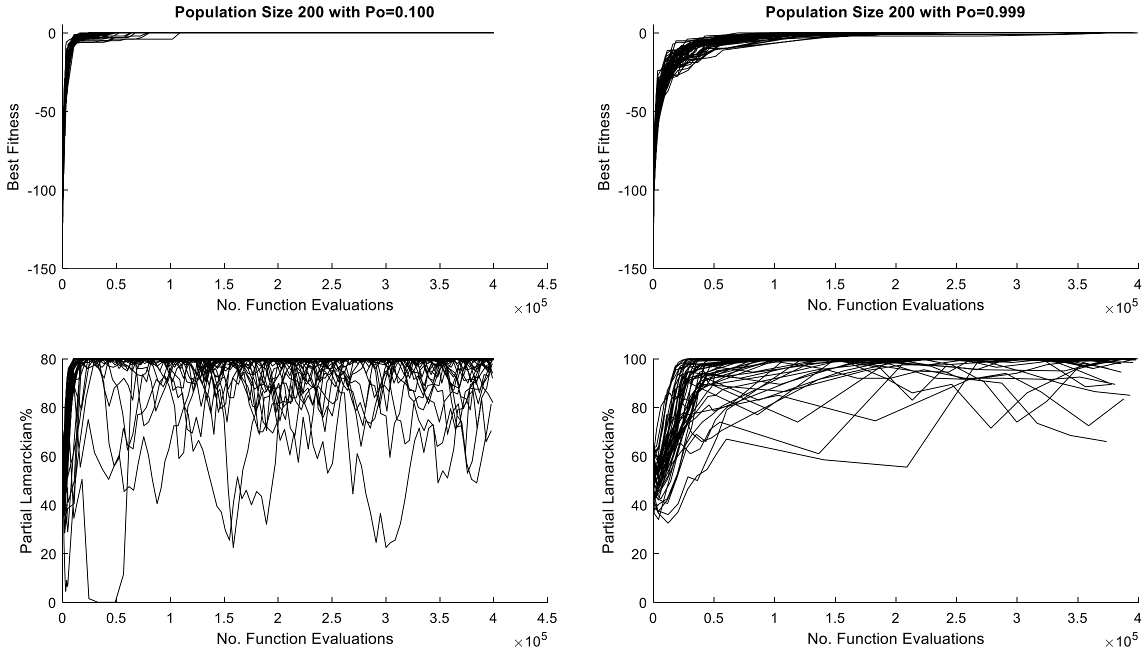

Fig. 12 Evolution of the learning strategy when solving the Rastrigin function using the ASH algorithm

Acknowledgements This work was supported in part by the Libya Institute for Advanced Studies.

References

1. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Boston
2. El-Mihoub TA, Hopgood AA, Nolle L, Battersby A (2006) Hybrid genetic algorithms: a review. Eng Lett 3(2):12–45
3. Nolle L, Krause R, Cant RJ (2016) On practical automated engineering design. In: Al-Begain K, Bargiela A (eds) Seminal contributions to modelling and simulation. Simulation foundations, methods and applications. Springer, Cham
4. Tang Q, Liang Y, Zhang L, Floudas CA, Cao X (2015) Balancing mixed-model assembly lines with sequence-dependent tasks via hybrid genetic algorithm. J Glob Optim 65(1):83–107
5. Hopgood AA (2012) Intelligent systems for engineers and scientists, 3rd edn. CRC Press, Boca Raton
6. Whitley D, Gordon VS, Mathias K (1994) Lamarckian evolution, the Baldwin effect and function optimization. In: Parallel problem solving from nature—PPSN III. Springer, Berlin, pp 5–15
7. Hinton G, Nowlan SJ (1987) How learning can guide evolution. Complex Syst 1:495–502
8. El-Mihoub TA, Hopgood AA, Nolle L, Battersby A (2006) Self-adaptive Baldwinian search in hybrid genetic algorithms. In: 9th fuzzy days international conference on computational intelligence, Dortmond
9. Espinoza FB, Minsker B, Goldberg D (2001) A self adaptive hybrid genetic algorithm. In: The genetic and evolutionary computation conference, San Francisco
10. Espinoza FP, Minsker BS, Goldberg DE (2003) Performance evaluation and population reduction for a self adaptive hybrid genetic algorithm (SAHGA). In: GECCO 2003, LNCS 2723
11. Houck CR, Joines JA, Kay MG, Wilson JR (1997) Empirical investigation of the benefits of partial Lamarckianism. Evol Comput 5(1):31–60
12. Ishibuchi H, Kaige S, Narukawa K (2005) Comparison between Lamarckian and Baldwinian repair on multiobjective 0/1 knapsack problems. In: Evolutionary multi-criterion optimization, Guanajuato
13. Joines JA, Kay MG, King R, Culbreth C (2000) A hybrid genetic algorithm for manufacturing cell design. J Chin Inst Ind Eng 17:549–564
14. Orvosh D, Davis L (1993) Shall we repair? Genetic algorithms, combinatorial optimization, and feasibility constraints. In: The fifth international conference on genetic algorithms, Urbana-Champaign
15. Sung-Soon C, Byung-Ro M (2005) A graph-based Lamarckian–Baldwinian hybrid for the sorting network problem. IEEE Trans Evol Comput 9:105–114
16. Michalewicz Z, Nazhiyath G (1995) Genocop III: a co-evolutionary algorithm for numerical optimization problems with nonlinear constraints. In: 2nd IEEE international conference on evolutionary computation, Perth
17. El-Mihoub TA, Hopgood AA, Nolle L, Battersby A (2004) Performance of hybrid genetic algorithms incorporating local search. In: 18th European simulation multiconference, Magdeburg
18. Julstrom B (1999) Comparing Darwinian, Baldwinian, and Lamarckian search in a genetic algorithm for the 4-cycle problem. In: Genetic and evolutionary computation conference, Orlando

19. Ku KW, Mak MW (1997) Exploring the effects of Lamarckian and Baldwinian learning in evolving neural networks. In: International conference on evolutionary computation, Indianapolis
20. El-Mihoub T, Hopgood AA, Aref I (2014) Self-adaptive hybrid genetic algorithm using an ant-based algorithm. In: 2014 IEEE international symposium on robotics and manufacturing automation, Kuala Lumpur
21. Whitsed R, Smallbone L (2017) A hybrid genetic algorithm with local optimiser improves calibration of a vegetation change cellular automata model. Int J Geogr Inf Sci 31(4):717–737
22. Gibbs MS, Dandy GC, Maier HR (2008) A genetic algorithm calibration method based on convergence due to genetic drift. Inf Sci 178(14):2857–2869
23. Ruud E-L, Samuelsen E, Glette K (2016) Memetic robot control evolution and adaption to reality. In: 2016 IEEE symposium series on computational intelligence (SSCI)
24. Khmeleva E, Hopgood AA, Tipi L, Shahidan M (2018) Fuzzy-logic controlled genetic algorithm for the rail-freight crew-scheduling problem. KI Künstliche Intell 32(1):61–75. https://doi.org/10.1007/s13218-017-0516-6
25. Gancarski P, Blansche A (2008) Darwinian, Lamarckian, and Baldwinian (co)evolutionary approaches for feature weighting in K-means-based algorithms. IEEE Trans Evol Comput 12(5):617–629. https://doi.org/10.1109/TEVC.2008.920670
26. Holzinger A, Blanchard D, Bloice M, Holzinger K, Palade V, Ramadan R (2014) Darwin, Lamarck, or Baldwin: applying evolutionary algorithms to machine learning techniques. In: The 2014 IEEE/WIC/ACM international joint conferences on web intelligence (WI) and intelligent agent technologies (IAT)
27. Heard E, Martienssen RA (2014) Transgenerational epigenetic inheritance: myths and mechanisms. Cell 157(1):95–105
28. Neri F, Cotta C (2012) Memetic algorithms and memetic computing optimization: a literature review. Swarm Evol Comput 2:1–14
29. Garg P (2009) A comparison between memetic algorithm and genetic algorithm for the cryptanalysis of simplified data encryption standard algorithm. Int J Netw Secur Appl 1(1):34–42
30. Boriskin AV, Sauleau R (2010) Synthesis of arbitrary-shaped lens antennas for beam-switching applications. In: The 40th European microwave conference, Paris
31. El-Mihoub TA, Hopgood AA, Aref IA (2013) Accelerating genetic schema processing through local search. In: Computer, control, informatics and its applications, Jakarta
32. Turney IA (1996) Myths and legends of the Baldwin effect. In: 13th international conference on machine learning, Bari
33. Mayley G (1996) Landscapes, learning costs and genetic assimilation. Evol Comput 4:213–234
34. Turney P, Whitley D, Anderson R (1996) Evolution, learning, and instinct: 100 years of the Baldwin effect. Evol Comput 4:iv–vii
35. Joines JA, Kay MG (2002) Hybrid genetic algorithms and random linkage. In: 2002 congress on evolutionary computation, Honolulu
36. Sasaki T, Tokoro M (1997) Adaptation toward changing environments: why Darwinian in nature?. In: Fourth European conference on artificial life, Brighton
37. Tersi L, Fantozzi S, Stagni R (2015) Characterization of the performance of memetic algorithms for the automation of bone tracking with fluoroscopy. IEEE Trans Evol Comput 19(1):19–30
38. Mathias KE, Whitley LD, Stork C, Kusuma T (1994) Staged hybrid genetic search for seismic data imaging. In: IEEE conference on evolutionary computation, Orlando, FL
39. Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y-P, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005. Nanyang Technological University, Singapore
40. Press W, Teukolsky S, Vetterling W, Flann B (1993) Numerical recipes in C. Cambridge University Press, Cambridge