International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 8 (2016) pp. 155–162

An Evolutionary Algorithm Based On The Aphid Life Cycle

Michael Cilliers[1] and Duncan Coulter[2]

[1]Academy of Computer Science and Software Engineering, University of Johannesburg,
Auckland Park, Johannesburg, South Africa
cilliersm@uj.ac.za

[2]Academy of Computer Science and Software Engineering, University of Johannesburg,
Auckland Park, Johannesburg, South Africa
dcoulter@uj.ac.za

Abstract: **This paper proposes an evolutionary algorithm based on the reproduction cycle of aphids. The proposed algorithm will alternate between multiple reproduction operators based on the fitness of the population. Through the alternation of reproduction strategies the balance between exploration and exploitation can be manipulated to achieve faster convergence. Two variations on the proposed algorithm are implemented and compared to the standard evolutionary algorithm and clonal expansion. The comparison of converging times of the algorithms show that both variations of the proposed algorithm can be effective.**

Keywords: Aphid lifecycle, genetic algorithm, optimization

I. Introduction

This work is an extension of work that was published in Advances in Nature and Biologically Inspired Computing [1].

When working with evolutionary algorithms the trade-off between exploration and exploitation has to be considered. If a specific algorithm focuses too much on exploration, it will keep on searching the whole problem space instead of converging on an optimal (sufficient) solution. On the other hand if the algorithm focuses too much on exploitation it is possible that it may get stuck on sub-optimal solutions. The correct choice for the specific balance between exploration and exploitation is problem and search space specific. [2]

The first step needed to balance the focus on exploration and exploitation of an algorithm is based on the choice of selection method used during the reproduction process. Different selection methods will have varying effects on the emergent properties of the implemented algorithm and poor quality choices can lead to the algorithm being unable to find a satisfactory solution. In addition the problem space characteristics are not always fixed and therefore even with the contribution of the selection method a static trade-off between exploration and exploitation is not always optimal.[4]

Varying the magnitude of changes made during the application of the mutation operation (mutation size) by the algorithm over the lifetime of a particular run, allows for a dynamic balance between exploration and exploitation [3]. A large mutation size will cause large jumps to be made by candidate solutions within in the search space which will benefit exploration. Small mutation sizes will however cause small jumps which will help exploit the current position in the search space.

Varying the mutation size provides more focus towards exploration during the initial section of the algorithm and moves towards exploitation during the latter portion of the algorithm. To try and develop a better solution, inspiration can be drawn from nature. Aphids are a very successful group of insects that has developed interesting methods to be able to quickly adapt to changes in their environment [5]. By trying to emulate these methods that have been developed over many years of evolution, it could be possible to achieve better results than what is possible with the methods mentioned above.

The magnitude of the mutation can also be lowered and combined with an increased mutation rate. The combination will cause frequent small changes to individuals within in the population. In most cases the small changes can have a positive effect on the exploitation of the algorithm by moving the candidate solution up the local fitness gradient, but a negative effect on the exploration by reducing the likelihood the solution will cross an adjacent valley in the fitness landscape. It should be acknowledged however that it is possible for small changes to have profoundly positive or negative effects (for example by converting a valid solution in to an invalid solution) in most cases the magnitude of the effect will correlate with the magnitude of the change. The small mutation size should not contribute to the exploration of the algorithm, so exploration will be largely dependent on the reproduction operation. If the mutation rate is too high it can have a negative effect on the degree of exploitation of the algorithm. Too much mutation will cause individuals to jump around their current positions within the fitness landscape, this is not an effective method of covering the local search space.

A number of attempts have been made to improve the balance between exploration and exploitation. One example of this is through the simulation of states of matter [6]. By imitating the movement of gas molecules the algorithm can focus on exploration. Then when switching to gas and solid, the algorithm will move focus towards exploitation. Another attempt is through the extension of the artificial bee colony algorithm [7]. The extension introduce bees that have different roles. Some roles contribute more towards exploration while others contribute more towards exploitation. By controlling the ratio between the roles in the population, the trade-off between exploration and exploitation can be controlled.

II. The Aphid Reproduction Cycle

Aphids, also known as plant lice, are insects that live on and consume the sap of a host plant. Aphids tend to be found in large groups and such infestations can cause substantial damage to the host plant. One of the factors that contribute to the success of the aphid group is due to unique aspects of their lifecycle. The species group has developed a lifecycle that makes use of multiple reproduction methods based on current environmental conditions. It should be noted however that not all aphids exhibit such flexibility in reproductive strategy selection. [10]

During spring and summer these aphids reproduce asexually. A single aphid parent produces clonal offspring through parthenogenesis. Parthenogenesis is a form of asexual reproduction where an embryo is formed without fertilization. During parthenogenesis there is no genetic material contributed by a male, so the female has to fill in the absent portions. The female fills in the missing portions by copying her own genes across to the child. The offspring will receive all genetic material from the single parent and will thus be a clone of the parent. It should be noted however that due to the distinction between an individual's genotype and its subsequent expression as a phenotype it is possible for the fitness of these clones to differ in some evolutionary models. For example the contents of a protein strand are determined based on genotype alone however the function of a protein is also determined based on its tertiary structure (how the protein folds in three dimensional space) which can be influenced by environmental factors such as temperature.

In evolution the success of a parent is measured by its ability pass on its genes. In this regard clonal reproduction is ideal. If sexual reproduction is used the child will receive genetic material from both parents. The genes of both parents will be diluted in the next generation. Clonal reproduction on the other hand allows the parent to pass on all its genes to the child. Asexual reproduction allows a large number of aphids to be produced to increase population numbers while the environment is favourable. The increased birthrate can be achieved due to the efficiency of asexual reproduction. Asexual reproduction eliminates the finding of a suitable partner which can be a resource intensive activity. A disadvantage of asexual reproduction is that it limits the adaptability of the population due to the lack of genetic transfer. A lack of adaptation is however not a problem during those seasons when the population is prospering due to favourable environmental conditions such as high food availability and moderate temperatures.

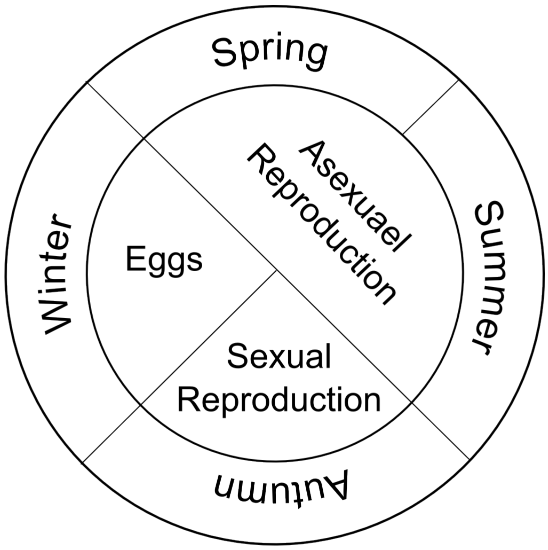

Figure. 1: Aphid Reproduction Cycle

During autumn the aphid population switches over to sexual reproduction as shown in Figure 1. At this point the conditions deteriorate due to changing weather conditions. The changing environment and inherent increase in selection pressure requires the aphid population to adapt. The asexual reproduction that was effectively used during the spring and summer will not allow the aphids to adapt fast enough. Sexual reproduction on the other hand is ideal for allowing the population to adapt. Sexual reproduction allows gene exchange to occur which in turn allows desirable genes to be distributed throughout the population.[8]

The life cycle of aphids can be compared to the production of B cells in the human immune system. The immune system produces a large number of varying B cells. The B cells are circulated throughout the body to identify antigens that are present. If a B cell has a high affinity to a corresponding antigen, it starts cloning itself to increase the number of appropriate B cells. The increased number of B cells facilitates the production of the required antibodies to deal with the detected pathogen. [9]

The sexual reproduction portion of the aphid lifecycle is similar to the initial production of B cells. In both cases the focus is on exploring the environment (search space) and reaching an acceptable state. With regards to the aphids, an acceptable state is when the population is able to survive in the current conditions. In the immune system an acceptable state is when all non self proteins in the body have been elicit an immune response while self proteins do not.

The asexual reproduction of aphids and the cloning of B cells have a similar role. An acceptable state has been reached and needs to be exploited. Aphids exploit the conditions by increasing the population numbers through asexual reproduction. The immune system on the other hand takes advantage by producing the appropriate antibodies. The antibody production is achieved through the increase of B cells due to the cloning.

Aphids are not the only species that makes use of multiple reproduction methods [11]. Rotifers are very small aquatic creatures that can be found in a variety of water bodies. As with aphids, some rotifer species make use of both sexual and asexual reproduction. During spring the rotifers reproduces asexually to boost population size. When the conditions deteriorate and the water body can't sustain the population, the rotifers switch to sexual reproduction. Both aphids and rotifers make use of similar reproductive strategies, but for the purpose of this paper aphids will be used as the prototypical example.

III. Proposed Algorithm

The presented algorithm draws its inspiration from the aphid lifecycle described previously. The algorithm will take the standard genetic algorithm and adapt it to make use of multiple reproduction operators based on a measure of the harshness of the selection pressure experienced by candidate solutions. Sexual and asexual reproduction will be used to conform to the reproduction strategies used by aphids. By making use of both sexual and asexual reproduction, advantages can be drawn from both.

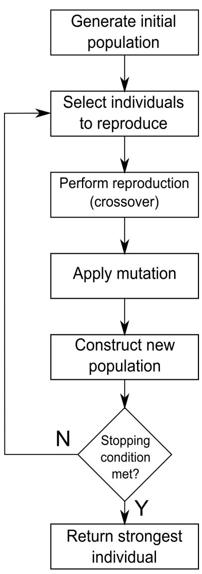

Figure. 2: Standard Genetic Algorithm

Sexual reproduction will allow the population to cover large portions of the search space to try and find the optimal areas. Once the optimal areas of the search space have been identified, asexual reproduction will allow the algorithm to find the optimal solution within these areas.

As with the standard genetic algorithm shown in Figure 2, the proposed algorithm will start with a randomly generated population of potential solutions. From the population a number of individuals will be selected to reproduce. The standard genetic algorithm will then make use of a single reproduction operator to perform the reproduction. The presented algorithm will replace the reproduction portion of the standard genetic algorithm with multiple reproduction operators and a method to switch between them. Once the new solutions have been created, mutation is applied to them. These new solutions are then used to form the next generation. At this point the stopping condition of the algorithm is checked. If the stopping condition is met the algorithm terminates, otherwise it loops back to the selection of parent solutions. Figures 3 and 4 illustrate two variations and the changes made to the standard genetic algorithm.

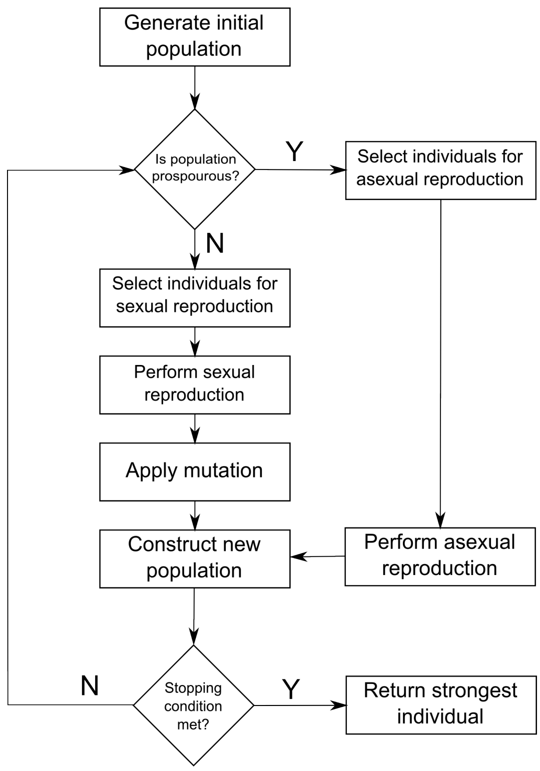

Figure. 3: Proposed Algorithms (Generational)

Similar results can be obtained from techniques based on simulated annealing. Initially the simulated annealing algorithm will focus on exploration to try and cover the majority of the search space. As time passes the focus of the algorithm will shift towards exploitation to find the optimal solution in the areas identified in the first phase. Unlike the proposed algorithm, simulated annealing is time based. Simulated annealing will thus not be effective for problems with dynamic search spaces. The problem with time based algorithm is that it requires knowledge regarding the duration of the algorithm. This information is required to select an appropriate decay function/rate. The proposed algorithm will however update its behaviour based on the prosperity (fitness) of the population as a measure of the harshness of the environment. A change in the search space will affect the prosperity of the population which in turn could cause the population to adjust its strategy.

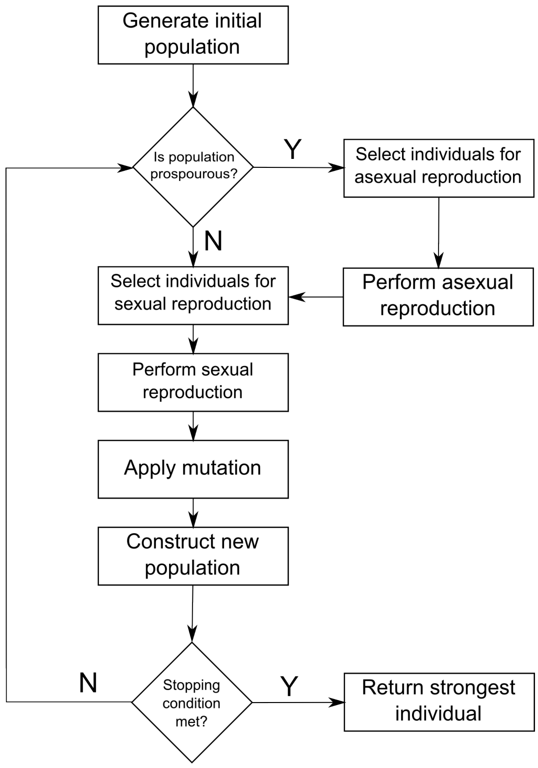

Figure. 4: Proposed Algorithms (gradual)

As mentioned in the previous section the algorithm is based on the standard genetic algorithm. The only change that was made is in the crossover (reproduction) step. During the crossover step a decision is added to determine whether sexual or asexual reproduction should be applied or a combination of the two. Two separate methods of applying the reproduction were implemented. The first method, henceforth referred to as the generational method, evaluates the average fitness of the population and then performs the same reproduction operator on the whole population. This approach will cause all the individuals in the population to perform sexual reproduction until the majority of the population starts to converge on possible optimal solutions. At this point the whole population will switch to asexual reproduction and start exploiting the current positions. The optimal point to switch reproduction operators will differ depending on the problem and will have to be adjusted accordingly. The check will also differ depending on whether the problem is a minimization or maximization problem.

The second method, henceforth referred to as the gradual/adaptive method, that was implemented does not use an absolute point where reproduction switches from sexual to asexual. This approach allows one portion of the population to reproduce asexually and the other to reproduce sexually. By increasing the number of individuals that reproduce asexually, as the population improves, a gradual switch from sexual to asexual reproduction can be achieved. As with the first implementation the average fitness is used to determine the fitness of the population as a whole. The average fitness is then converted to a probability that will help determine whether each individual will form part of the asexually or sexually reproducing sections of the population. As the average fitness get closer to the switch point the asexual group size increases and the sexual reproduction group size decreases. During the reproduction step of the algorithm each individual is randomly placed into one of the two reproduction groups. Once the appropriate reproduction operator has been applied to each of these groups the children are combined to form the next generation. As with the generational method, the optimal point to switch reproductive operator will be problem dependent.

IV. Comparison

The prototype system will be applied to a function fitting problem. The chromosome of each solution stores the coefficients of the function being represented. The implemented algorithm will fit a cubed function and will thus store four separate values. To evaluate the fitness of each function it will be compared to the points being matched. The distance between the specific function and each point is added together to determine the quality of the function. A function with a smaller fitness will match the points closer than a function with a higher fitness.

Rank based selection is used to select the individuals who will participate in sexual reproduction. Two parents are selected and one-point crossover is applied to two child solutions. Once the new individuals have been created mutation is applied to them. Individuals who reproduce asexually are only mutated. To generate the test points for the function to match, Bezier curves are used. The Bezier curve is taken and converted into a number of points. Each of the points is taken a set distance along the curve. These points are then used as the points that the algorithm attempts to fit. To compare the two implemented algorithms to the standard genetic algorithm, all three will be applied to match the same curve multiple times. For each generation the maximum fitness in the population will be averaged with the other executions of the same algorithm. These values can then be compared to determine whether one of the algorithms converges faster than the others.

The above mentioned process will be repeated multiple times for each implemented algorithm. For each run the parameters of the algorithm being evaluated, will be adjusted.

A second variation of the above mentioned procedure will also be applied. The variation will replace the Bezier curve being fitted once an acceptable state has been reached in order to introduce change into the environment. The algorithm will have to adapt to the new environment. The dynamic substitution of the fitness function poses a different problem to the initial adaptation of the population. The state of the population at the moment of substitution embed information from the previous problem space. Some of this embedded information will be correct and therefore beneficial while other parts will be incorrect and must therefore be unlearned. These considerations are referred to as transfer learning [5]. The time that it takes each algorithm to converge on the new solution will be compared to determine the adaptability of each algorithm.

The prototype algorithms will also be compared to a clonal selection algorithm. As with the genetic algorithm, the clonal selection algorithm will be applied multiple times to match the same curve. The maximum fitness at each generation will be averaged out across all of the runs. The values produced by the runs of the prototype algorithms can then be compared to the values produced by the clonal selection.

V. Results

The Aphid Inspired Evolutionary algorithms and the standard genetic algorithm were implemented in a prototype system using C# as the implementation language [12]. The implementation allows the problem-space defining curve to be manipulated during execution.

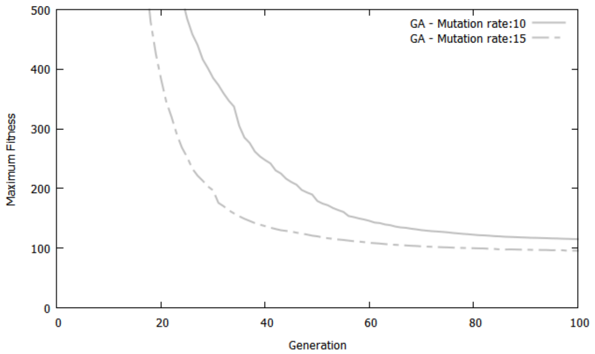

Figure. 5: Comparison of parameters (Genetic algorithms)

The first comparison that was made is between two standard genetic algorithms with differing mutation rates. In Figure 5 it can be seen that the genetic algorithm with the higher mutation rate converged faster than the genetic algorithm with the lower mutation rate. This illiterates the effect that a small change in the focus of the algorithm can have. The high mutation rate that is used for the problem is due to the small size of the chromosome. A lower mutation rate would result in majority of the population not experiencing mutation, even when it is required.

For the generational variation of the developed algorithm multiple versions were tested, differing in mutation rate and

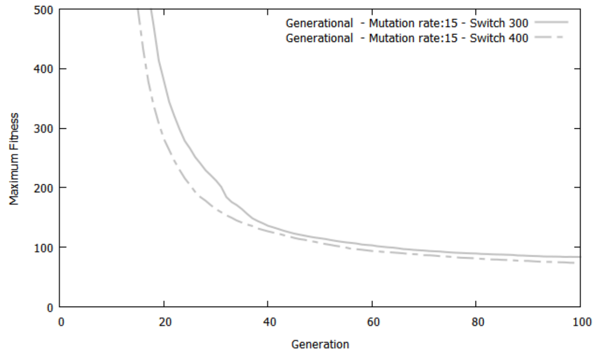

Figure. 6: Comparison of parameters (Generational)

reproductive strategy switching point. As with the standard genetic algorithm, configuration that made use of a higher mutation rate performed better. This result is not unexpected as the method used to control the focus of the algorithm is through the introduction of clonal reproduction. The clonal reproduction is reliant on mutation to introduce change into the population and the higher mutation rate should improve this process. It should be noted that if the mutation rate is too high it could have a negative effect on the convergence of the algorithm.

Figure 6 illustrates the configurations that produced the best results. As mentioned both these configurations make use of the higher mutation rate. The switching points that were tested ranged from 200 to 500. The best configurations made use of 300 and 400 respectively. This indicates that if the focus of the algorithm is updated too soon or too late, it can have a negative effect on the convergence rate of the algorithm.

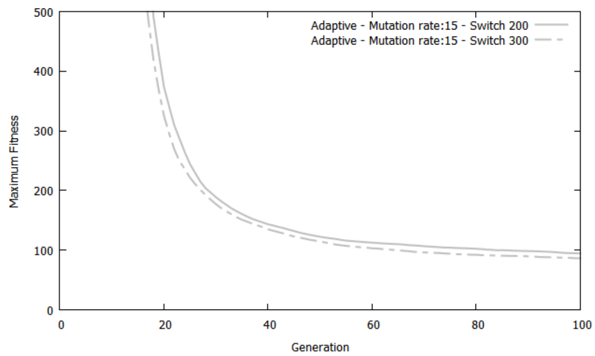

Figure. 7: Comparison of parameters (Adaptive)

As with the generational variation the adaptive variations configurations differed in mutation rate and switching point. Once again the configurations that made use of the higher mutation rate produced better results. The points evaluated for switching reproduction strategy again ranged from 200 to 500. The specified point is where the adaptive algorithm will switch over to full clonal reproduction without any sexual reproduction. This before the specified point is reached the algorithm will start to gradually switch from sexual reproduction to clonal reproduction. The configurations that produced the best results made use of 200 and 300 respectively as shown in Figure 7. Even though the points that produced the best results were the lowest values tested, they will have an effect on the reproduction strategy of the algorithm before the average fitness reaches that point.

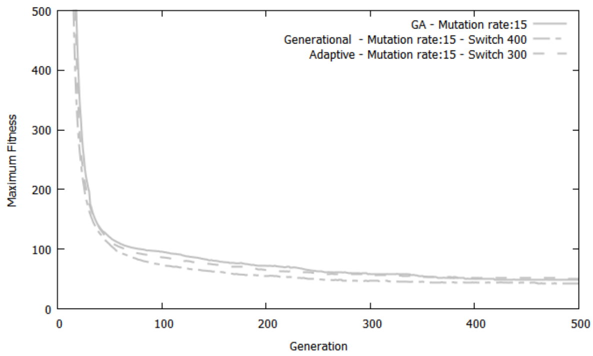

Figure. 8: Comparison of methods

In Figure 8 the best results for each of the algorithms are compared. From the graph it can be seen that the algorithms perform very similarly during the initial stages of the algorithms. Eventually both variations of the proposed algorithm start to outperform the standard genetic algorithm. The generational method performed the best out of the two evaluated algorithms. The heavy focus on exploitation allowed the algorithm to converge faster than the other algorithms. The adaptive variation of the implemented algorithm probably starts adjusting the focus of the algorithm while a larger focus on exploration is still required. So even though the algorithm provides the extra focus on exploitation during the end of the algorithm, the portion of the algorithm requiring exploration suffers.

The negative impact on the exploration of the algorithm could have been a contributing factor towards selection of the lowest point values. Smaller values will cause the algorithm to start affecting the population at a lower average fitness. This would allow a larger portion of the algorithm to focus on exploration during the initial stages of the algorithm.

Next the algorithms was applied in the dynamic environment. For this comparison the fitness values were only considered after the change was introduced into the environment. By using these values, the rate at which the algorithms adjusts to the change can be evaluated. Figure 9 illiterates that the are almost no difference between the three evaluated algorithms. This indicates that the standard genetic algorithm is able to adapt to some changes in the environment. The introduced changes probably ended up having sufficient solutions in the same area of the search space. If the changes that where introduced resulted in solutions being located in other parts of the search space the

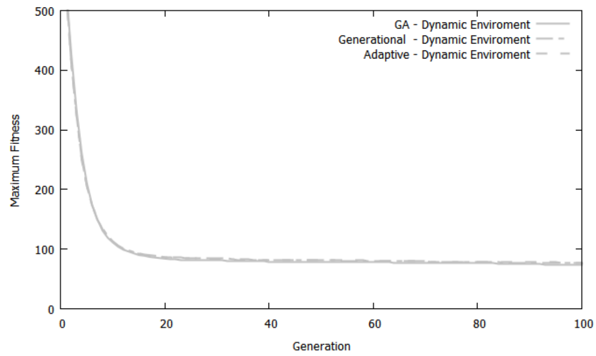

Figure. 9: Comparison of methods (Dynamic Environment)

aphid based algorithms should perform better.

Testing of the clonal selection algorithm revealed that the algorithm has a problem with getting stuck on local minima. The algorithm focuses too much on exploitation without providing enough exploration. A large portion of the runs got stuck on local minima which had a negative effect on the average convergence time of the clonal selection algorithm as can be seen in Figure 10.

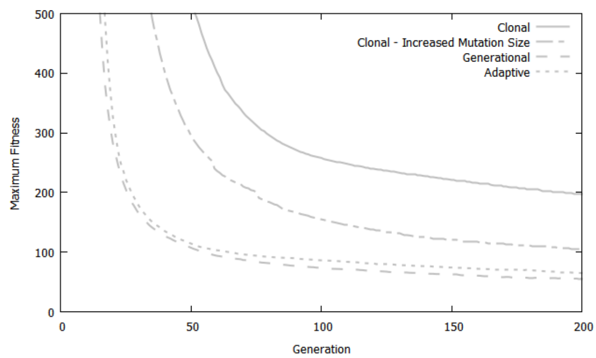

Figure. 10: Clonal selection

Due to the sub-optimal results produced by the clonal selection algorithm, a second variation was implemented. The second implementation has a larger mutation size to help avoid getting stuck on local minima. The second variation performed better than the first clonal selection algorithm as can be seen in Figure 11. Even with the improved results the prototype algorithms still performed better than the clonal selection algorithm. The clonal selection algorithm does not have enough focus on exploration. The good areas of the search space are thus not identified to be exploited. Even though the clonal selection algorithm excels at exploitation, it struggles to find the appropriate areas to exploit.

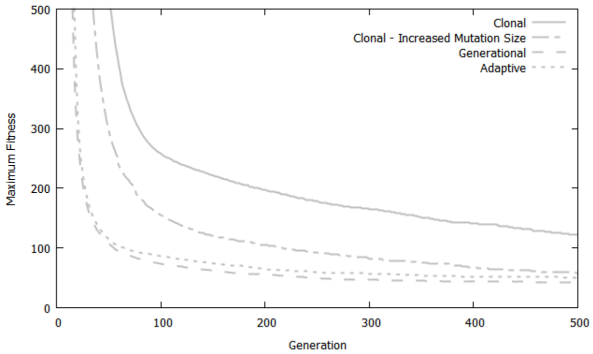

Figure. 11: Clonal selection - Full lifetime

VI. Conclusion

Both variations of the proposed algorithm were able to outperform the standard genetic algorithm when applied to the curve fitting problem. This indicated that the proposed algorithm can be effective for some problems. The effectiveness of the algorithm will be dependent on the specific problem and on the related search space. As the algorithm makes use of the fitness of the population to make decisions and adjust reproduction strategy, it should work for problems with dynamic search spaces. The evaluation performed on dynamic environments demonstrated no advantage over the standard genetic algorithm, but this could be the result of the changes not being severe enough. The proposed algorithms were also able to outperform clonal selection algorithms. Even after adjusting the parameters of the clonal selection algorithms, both variations of the proposed algorithm were able to outperform the clonal selection algorithm.

VII. Future work

Based on the results of this work, the aphid inspired evolutionary algorithm can be effective for some problems. To get a better idea of the problems that are suited towards the aphid inspired evolutionary algorithm, the algorithm will have to be applied to more problem types. A variety of different problem instances should be used to do the evaluation. Problems with more complex chromosomes are one of the areas that will have to examined. Just because the algorithm was effective when working with a simple chromosome structure does not mean that it will be effective when working with more complex chromosome structures. Testing should also be extended to more real world problems. This will allow the developed algorithm to be evaluated against existing algorithms that have been optimised for the specific problem.

The portion of the work covering dynamic search spaces should be expanded. Currently only a single change was made to the environment and the result evaluated. By extending the types of changes that are made to the environment a more complete picture can be developed of how different algorithms react in dynamic environments. The changes should be selected to include optimal solutions moving small distances in the search space as well as changes that completely changes the search space.

Finally the work can be extended by applying the aphid life cycle to other evolutionary algorithms. The work covered in this paper extended the standard genetic algorithm to mimic the life cycle of the aphid. This same update to the reproduction operation can be applied to other evolutionary algorithms, for example genetic programming. This reproduction strategy could be applied to any evolutionary algorithm that have reproduction operations that mimic sexual reproduction as well as operations that mimic mutation.

References

[1] M. Cilliers and D. Coulter. An Aphid Inspired Evolutionary Algorithm. *Advances in Nature and Biologically Inspired Computing*, Springer International Publishing, pp. 293–303. 2016.

[2] A.E. Eiben, C.A. Schippers. On evolutionary exploration and exploitation. *Fundamenta Informaticae*, 35.1, pp. 35-50, 1998.

[3] A.E. Eiben, R. Hinterding, Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3.2, pp. 124-141, 1999.

[4] A.P. Engelbrecht. *Computational Intelligence: An Introduction*, 2 edition. Wiley, 2007.

[5] S.J. Pan, Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22.10, pp. 1345-1359, 2010.

[6] E. Cuevas, A. Echavarra, M.A. Ramrez-Ortegn. An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation. *Applied intelligence*, 40.2, pp. 256–272, 2014.

[7] S.S. Hasan, F. Ahmed. Balaning Explorations with Exploitations in the Artificial Bee Colony Algorithm for Numerical Function Optimization. *International Journal of Applied Information Systems*, 9.1, pp. 42–48, 2015.

[8] C. Rispe, J.S. Pierre, J.C. Simon, P.H. Gouyon. Models of sexual and asexual coexistence in aphids based on constraints. *Journal of Evolutionary Biology*, 11.6, pp. 685-701, 1998.

[9] F. Sablitzky, G. Wildner, K. Rajewsky. Somatic mutation and clonal expansion of B cells in an antigen-driven immune response. *The EMBO journal*, 4.2, pp. 345, 1985.

[10] J.C. Simon, C. Rispe, P. Sunnucks, Ecology and evolution of sex in aphids. *Trends in Ecology & Evolution*, 17.1, pp. 34-39, 2002.

[11] T.W. Snell. Effect of temperature, salinity and food level on sexual and asexual reproduction in Brachionus

plicatilis (Rotifera). *Marine Biology*, 92.2, pp. 157-162, 1986.

[12] C# Language Specification (C#). [Online]. Available: http://msdn.microsoft.com/en-us/library/aa645596(v=vs.71).aspx.

Author Biographies

Michael Cilliers Michael was born in South Africa in 1989. He completed his BSc Information Technology degree in 2010 at the University of Johannesburg in South Africa. In 2011 he completed his BSc Hons in information technology degrees also at the University of Johannesburg. Currently he is in the process of completing his MSc computer science degree at the University of Johannesburg. Research interests includes evolutionary algorithms and image processing.

Duncan Anthony Coulter Duncan was born in South Africa in 1983. He completed his BSc information technology in 2003, BSc Honours in computer science in 2004, MSc in computer science in 2008, and PhD in computer science in 2014. All of these were completed at the University of Johannesburg in South Africa. His research interests concern biologically inspired artificial intelligence.