Biologically-inspired computing for optimisation – Calibration Report

CSC3431 – Biodesign and Natural Computing

By Lucas Toner

Table of Contents

Algorithm Details

The two algorithms that will be considered in this report are Genetic Algorithms (GAs), and Particle Swarm Optimization (PSO).

Both algorithms are heuristic, biologically inspired, and attempt to find solutions to problems by iteratively searching the solution space for optimal solutions.

Genetic Algorithms

First dated in 1975 by John Holland [1]. GAs are inspired by the processes of natural selection seen in biology. They are population based, populations of candidate solutions are processed through a series of natural selection inspired operators. These operators include selection, crossover and mutation. [2]

Particle Swarm Optimisation

First dated in 1995 by James Kennedy and Russel Eberhart [3]. PSO is also population based, with each individual referred to as a particle. Each particle independently moves around the solution space, based on it’s own velocity and previous position, and can be influenced by the wider population to move towards optimal solutions. [4] This process can be easily visualised as particles moving around a physical space

Calibration process

The Technique

The calibration of parameters in a GA or PSO is, in itself, an optimisation problem, one with many dimensions when different operators and their respective parameters are taken into account.

Therefore, rather than calibrate these parameters manually, it may be more effective to use an optimisation algorithm to search for the optimal configuration. In this instance, calibration was performed using a Particle Swarm Optimisation algorithm, which shall be referred to as a meta-level PSO [6], while the algorithms they are optimising the parameters for will be referred to as base algorithms

This process is partially inspired by the work of John J. Grefemstette [6] similar experiments with genetic algorithms, also see Thomas Bäck's "An experiment in Meta-Evolution" [7].

Parameters will be calibrated with a constraint, the base algorithms may only perform 10,000 fitness calculations.

The meta-level PSO will search a solution space in which each dimension represents the value of a parameter. There will also be dimensions to represent variations on the base algorithms, for example the different mutation operators available to a genetic algorithm.

To frustrate the problem, the base algorithms are *stochastic optimisation algorithms*, meaning that they generate and use random variables, and inherently use randomness as part of their optimisation process.
This means, that in order to evaluate the effectiveness of a specific set of parameters, we must evaluate each set of parameters multiple times, and consider the mean of these multiple evaluations.

Fig.1 Contains Pseudocode to describe the calibration process

```
begin

population = Generate initial population
For iterations
        population.evaluate
   population.calculate_velocity

func evaluate:
        for each individual in population
        repeat n times
        fitnesses.add(optimise schaffer function using GA/PSO with individual's parameters)
      average fitness = mean(fitnesses)
   return average fitnesses
end func

end
```

Figure 1

Parameters

The parameters to be optimised for the base algorithms, and their bounds are briefly described below.

Population and iterations:

Due to the constraint on the permitted number of fitness calculations, the values of population and iterations will be limited to factor pairs x, y $\in \mathbb{Z}$ where xy =10,000

Parameters for Particle Swarm Optimisation

3 general parameters for PSO will be calibrated. These parameters are described in the Pyswarm documentation[8] inspired by the work of J. Kennedy and R.C. Eberhart[3]

The cognitive parameter (c1):

The extent to which a particle follows it's personal best position[8]. This will be bounded between 0.0 and 2.0

The social parameter (c2)

The extent to which a particle follows the global best position[8]. This will be bounded between 0.0 and 2.0

The inertial parameter (w)

The extent to which a particles previous velocity impacts it's next velocity[8]. This is bounded between 0 and 1

Topologies in Particle Swarm Optimisation

The topology of a swarm defines how particles communicate their personal best position with one another [9]. The below topologies will be included in the calibration, along with their respective parameters

Ring Topology

This topology only permits particles to share with their nearest neighbours [9][8]
The ring topology also has two further parameters that require calibration:

- Minowski p-norm (p)
- Number of neighbours to consider (k)

Star Topology

Every individual is connected to every other individual[8]

Parameters for Genetic Algorithms

Classical Genetic algorithms have three operators that are commonly adjusted depending on the problem, these are (discounting altering the encoding scheme): crossover, mutation and selection. [2]

Elitism

How many of the population with the best fitness will automatically survive a iteration of the selection algorithm. Typically elitism is only performed on the individual with the best fitness [2], however this can be extended to multiple individuals

Probability of Individual Mutation

The probability that an individual will mutate

Probability of Gene Mutation

The probability that a gene will mutate

Selection

The selection algorithm is a powerful factor in balancing the trade-off between exploration and exploitation in a GA [5]. The selection algorithms that will be considered are:

Roulette Wheel

Individuals are chosen with a probability proportional to their fitness value compared to every other individual[2]

Tournament

A pre-determined number of individuals are compared, from which the best is chosen [10]. The size of the tournament requires calibration

Crossover

Crossover is how individuals can share solution information with one another, with the goal that child individuals may contain their parents advantages.

We will be testing three well known crossover techniques, that cover a range of popular methodologies

Two Point Crossover

Genetic information is swapped as per three segments that are produced randomly [2]

Blended Crossover

Blends each attribute with one another [10]. This operator uses a parameter that requires calibration. Alpha, the extent to which attributes are blended

Uniform Crossover

This technique modifies the two individuals, swapping attributes between the two [10]. The probability of an attribute swap requires calibration

Mutation

Mutation is necessary for exploration in a GA, as it is the source of new genetic material [2]
The algorithms that will be included in the calibration are of two very different techniques:

Index Shuffle – also known as displacement

The attributes of the individual are shuffled with one another to some extent[10]

Gaussian Shift

Each attribute is mutated by probability according to a Gaussian distribution. This Operator has two further parameters that require calibration:

- Mu - The mean of the Gaussian distribution[10]
- Sigma – The standard deviation of the distribution[10]

In total, the solution space for Particle Swarm Optimisation has 7 dimensions, and the solution space for Genetic algorithms has 13 dimensions.

A flaw in the meta-PSO is due to a property inherent to simple PSO variants, such as described in [3]. An individual has a fixed length. So for example, a parameter that is specific to a certain mutation operator will not be relevant to an individual that does not use this mutation operator.

This parameter will continue to exist within the individual however, and could misinform other particles about it's optimal value.

Results

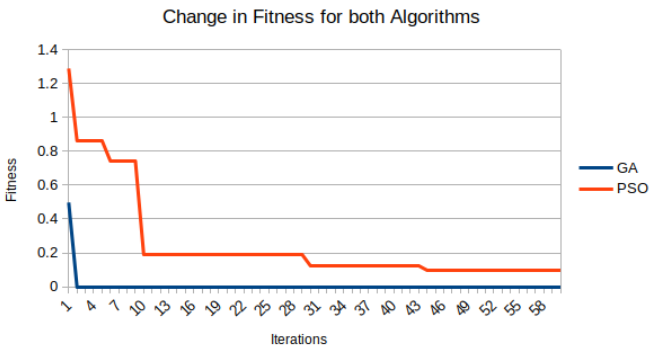

Figure 2

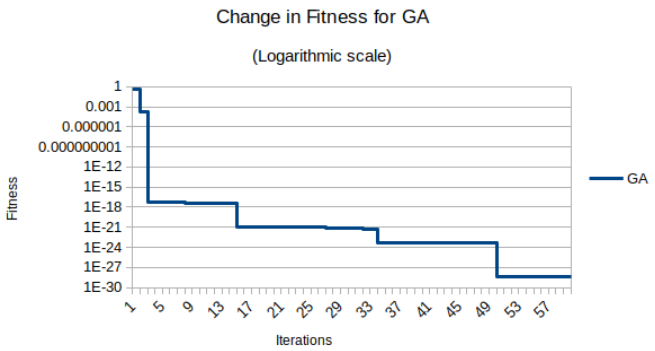

Figure 3

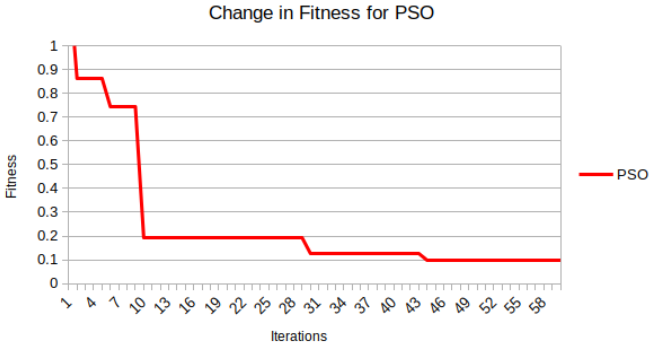

Figure 4

The best fitness individuals of both algorithms

Parameter	Value
Population/Iterations	8/1250
Selection	Tournament
Tournament Size	7
Elite	2
Mutation	Index Shuffle
Mutation Probability	0.88
Gene Mutation Probability	0.50
Crossover	Blend
Blend alpha	0.63
Crossover Probability	0.96

Figure 5

Parameter	Value
Population/Iterations	40/250
The Social Parameter (c1)	1.65
The Cognitive Parameter(c2)	1.66
The Inertial Parameter (w)	0.64
Topology	Star

Figure 6

Fitness

Fig. 2,3,4 display the best fitnesses found throughout the meta-PSO experiment, over 60 iterations. The best fitness that was found for a GA individual was 4.77108842198727E-29, while the best fitness found for a PSO individual was 0.0959891553652757.

There is clearly a stark difference in fitness between the two algorithms. However, on average it takes roughly half the time to perform the PSO than the GA, with PSO taking 0.15 seconds, and GA taking 0.37 seconds.
The assumption is, that in this particular implementation of GA, roughly half the time of execution is used by the selection, mutation and crossover operators, as these are significantly more intensive than the single velocity operation that PSO must compute [3].
PSO is often regarded as more computationally efficient in general than genetic algorithms[11-12], at least in regards to specific problems,so the constraint in number of fitness calculations may be working against the favour of PSO.

The worse PSO fitness can also be attributed to this implementation of the meta-PSO, it clearly needed improvements in it’s technique of calibrating PSO parameters.

Individual Fitness Consistency

When the best fitness individuals for each algorithm are run multiple times, both algorithms have a fairly wide range of fitnesses for the Schaffer function. Again this speaks to the stochastic nature of these algorithms.
When run 500 times, the GA individual showed a wide range of values, if we ignore the outliers (1.8% had a fitness above 0.001, but the next lowest was 4.35e-22), then we can say that the a GA ran with the same parameters 491 times, had a range 100 times greater than it's mean.
When the same calculations are performed on the greatest PSO individual, the range is only 16 times greater than the mean (ranging from 11.51 to 0.02).

Fig. 2 is drawn in a logarithmic scale, this is because it's fitness improved very rapidly, and at those points in the graph at which fitness improved, it improved by a large proportion (roughly a proportion of 0.001). Whereas, when a new most efficient individual for PSO was found, is is generally much less of a fitness jump.

Parameter/Operator Trends.

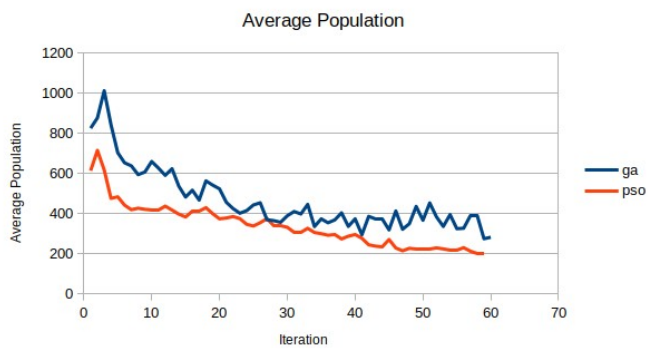

Figure 7

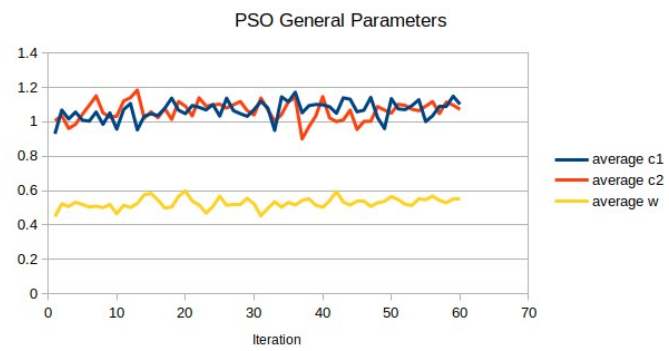

Figure 8

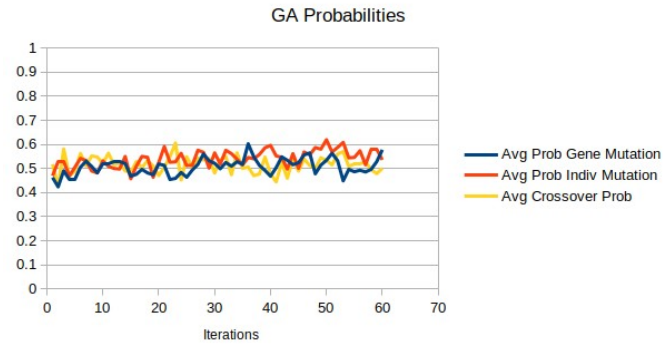

Figure 9

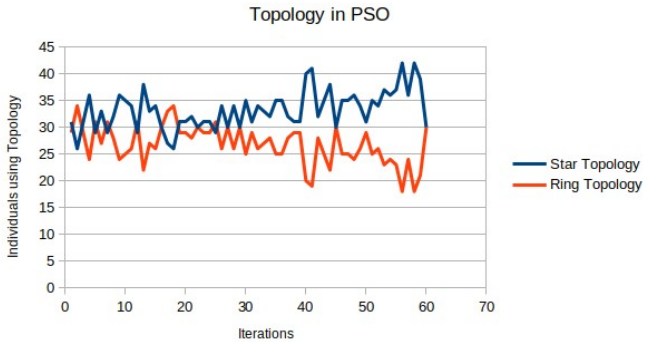

Figure 10

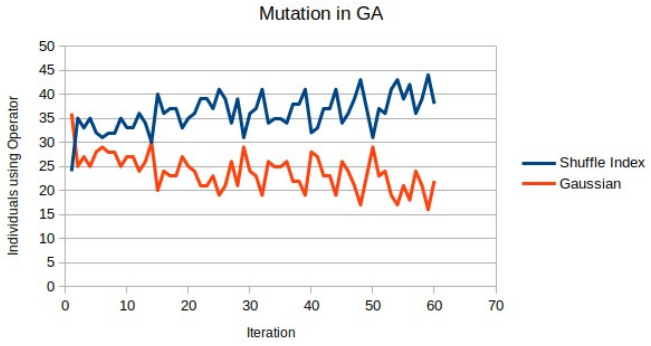

Figure 11

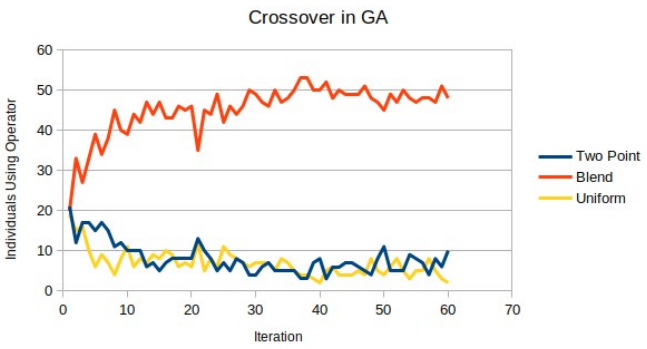

Figure 13

Figure 12

Interestingly, both algorithms tended towards lower populations, and higher iterations, this can be seen on Fig.5, 6, 7 *Note: In Fig.7 The value for average population is skewed upwards due to the proportional nature of the population options that were available for meta-individuals*
This shows that lower populations and higher iterations were optimal for both GA and PSO when solving the Schaffer function.

There is a clear trend in Fig. 11, 12, 13 defining the optimal available operators for GA, all three of the dominant operators exist in the GA parameters with the best fitness. The most notable can be seen in Fig. 13, the blend crossover operator is by far preferred by the meta-PSO algorithm to the other two.
The mutation operator, “shuffle index”’s dominance is a surprise, as it is typically described as having a risk of premature convergence [2]. As is the “star” topology in PSO [9]. Perhaps due to the constraints imposed upon these experiments, there are not enough iterations for convergence to prove a problem.

Consistently, PSO individuals of good fitness using the star topology have similar c1 and c2 parameters, regardless of the actual value, they tended towards one another.

References

[1] J. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, The MIT Press, 1992

[2] Katoch, S., Chauhan, S.S. & Kumar, V., A review on genetic algorithm: past, present, and future. Multimed Tools Appl 80, 8091–8126, 2021 doi: /10.1007/S11042-020-10139-6

[3] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.

[4] Maurice Clerc. Standard Particle Swarm Optimisation. 2012. ⟨hal-00764996⟩

[5] Jebari, Khalid. (2013). Selection Methods for Genetic Algorithms. International Journal of Emerging Sciences. 3. 333-344.

[6] J. J. Grefenstette, "Optimization of Control Parameters for Genetic Algorithms," in IEEE Transactions on Systems, Man, and Cybernetics, vol. 16, no. 1, pp. 122-128, Jan. 1986, doi: 10.1109/TSMC.1986.289288.

[7] Bäck, Thomas, 'An Experiment in Meta-Evolution', *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms* (New York, 1996; online edn, Oxford Academic, 12 Nov. 2020), https://doi.org/10.1093/oso/9780195099713.003.0013

[8] Pyswarm Documentation, [Online]. Available: https://pyswarms.readthedocs.io [Accessed: [[01-11-2023]]].

[9] T. Blackwell and J. Kennedy, "Impact of Communication Topology in Particle Swarm Optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 689-702, Aug. 2019, doi: 10.1109/TEVC.2018.2880894.

[10] Deap Documentation, [Online]. Available: https://deap.readthedocs.io [Accessed: [[01-11-2023]]].

[11] Y. Duan, R. G. Harley and T. G. Habetler, "Comparison of Particle Swarm Optimization and Genetic Algorithm in the design of permanent magnet motors," 2009 IEEE 6th International Power Electronics and Motion Control Conference, Wuhan, China, 2009, pp. 822-825, doi: 10.1109/IPEMC.2009.5157497.

[12] C. Ou and W. Lin, "Comparison between PSO and GA for Parameters Optimization of PID Controller," 2006 International Conference on Mechatronics and Automation, Luoyang, China, 2006, pp. 2471-2475, doi: 10.1109/ICMA.2006.257739.