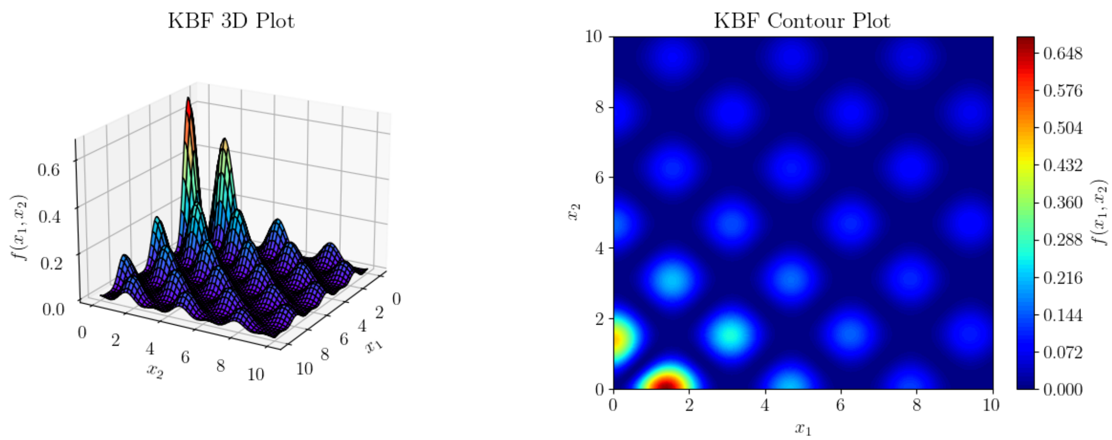
4M17 Coursework #2
Optimisation Algorithm Performance Comparison

Candidate No: 5730E

December 11, 2023

1 Abstract

This report conducts a comparative analysis of two optimisation algorithms applied to minimise Keane's Bump Function, (KBF). In particular, the study focuses on a Continuous Genetic Algorithm, (CGA), as well as an alternative algorithm not covered in the lectures: Q-learning, (QEG). Specifically, the Q-learning approach is adapted with the epsilon-greedy strategy to introduce a level of stochasticity into the optimisation process, thus aligning it with the requirements of this assignment.



(a) Surface plot.

(b) Contour plot.

Figure 1: The two-dimensional visualisation of the Keane's Bump Function, (KBF).

2 Introduction

2.1 Keane's Bump Function

To compare the performances of the two algorithms, the Keane's Bump Function, (KBF), is used as the objective function. In particular, the n-dimensional constrained optimisation problem is defined as the maximisation of:

$$f(\mathbf{x}) = \left| \frac{\sum_{i=1}^{n}(\cos(x_i))^4 - 2\prod_{i=1}^{n}(\cos(x_i))^2}{\sqrt{\sum_{i=1}^{n} i \cdot x_i^2}} \right| \tag{1}$$

$$\begin{aligned} \text{subject to} \quad & 0 \leq x_i \leq 10 \quad \forall i \in \{1, \ldots, n\} \\ & \prod_{i=1}^{n} x_i > 0.75 \\ & \sum_{i=1}^{n} x_i < \frac{15n}{2} \end{aligned} \tag{2}$$

The two-dimensional form of the function has been plotted in Figure 1. Some notable properties are as follows:

- The function is undefined at the origin, (0, 0). This is due to the division by zero in the denominator of Equation 1. Otherwise, the function is continuous and differentiable everywhere.

- The function is highly multi-modal. Its global maximum is located on the constaint boundary $x_n = 0$, where x_n denotes the final variable in the n-dimensional space. However, there are many local maxima located inside the feasible region, all of which have quite similar amplitudes.

- The function is nearly symmetric about the line $x_1 = x_2$. This stems from its construction in 1, using the sums of squared, symmetric terms, x_i^2, $(\cos(x_i))^2$, and $(\cos(x_i))^4$. This results in some invariance regarding the order of the input variables. Overall, the peaks consistently manifest in pairs, yet there is a notable pattern wherein one peak always surpasses its counterpart in magnitude.

Given the above properties, the KBF is a challenging function to optimise. The presence of multiple, similar-amplitude local maxima makes it difficult for an optimisation algorithm to converge to the global maximum. On the other hand, all control variables share the same nature, (continuous variables), and exhibit identical scales. Additionally, all constraints are of the inequality type, and the feasible space is non-disjoint.

The problem becomes additionally complex with the inclusion of the contraints delineated in 2. The feasible region has been carved out and visualised in Figure 2. The constraints are of the inequality type, and the feasible space is non-disjoint.

These properties make the KBF a suitable candidate for the comparative analysis of the two optimisation algorithms, as discussed in the previous work [1].

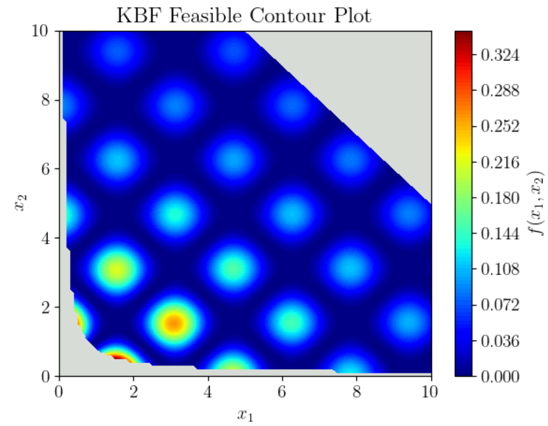


Figure 2: The three-dimensional visualisation of the Keane's Bump Function, (KBF).

2.2 Continuous Genetic Algorithm

The discrete nature of the GA presented in [4] makes it unsuitable for the optimisation of the KBF. An implementation of a Continuous Genetic Algorithm, (CGA), is used instead, which lends itself better to the problems presented in 1-2.

The CGA, a technique inspired by natural selection and genetics, presents itself as particularly well-suited to tackling challenges associated with multiple local optima. Furthermore, the algorithm lends itself well to parallelisation with low implementation effort, and offers ample opportunities for modifications and adaptations, supported by a rich body of literature on the subject.

The primary difference between the CGA and the GA in [4] is the representation of individuals, (solutions of the state space), within the population. Rather than representing an individual as a vector of binary values or bits, (0s and 1s), the CGA uses a real-valued vector of floating-point numbers to represent each individual, as discussed in [2]. This allows for a direct representation of the problem, and eliminates the need for a decoding function, which reduces overhead in function evaluations.

This adjustment marks a significant departure from conventional GAs, aligning the algorithm more closely with Evolution Strategies (ES), another member of the evolutionary algorithms family presented in [5]. However, the algorithm presented in 2.2.1 is still classified as a GA in accordance with the differences presented in [3], given that mutation does not serve as the primary search mechanism for exploring the state space. Instead, it functions as a nonadaptive, background operator.

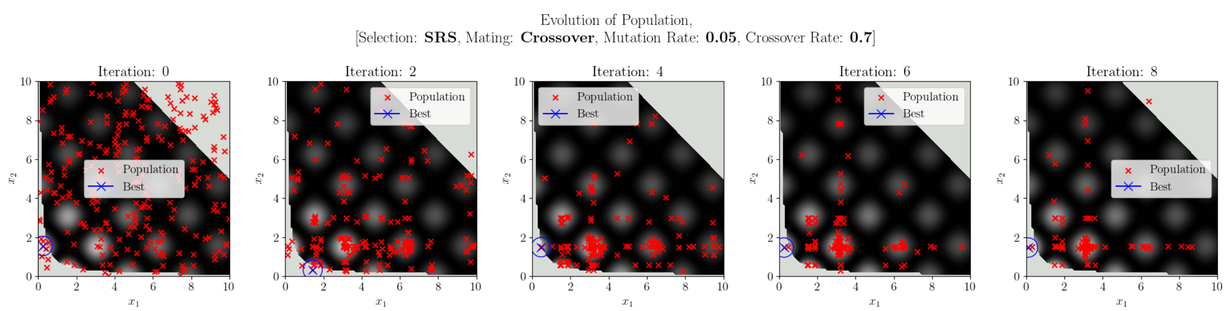
2.2.1 Implementation

In accordance with [4], a vector solution of the state space will be referred to as an *individual* or *chromosome*, and a matrix of individuals will be referred to as a *population*.

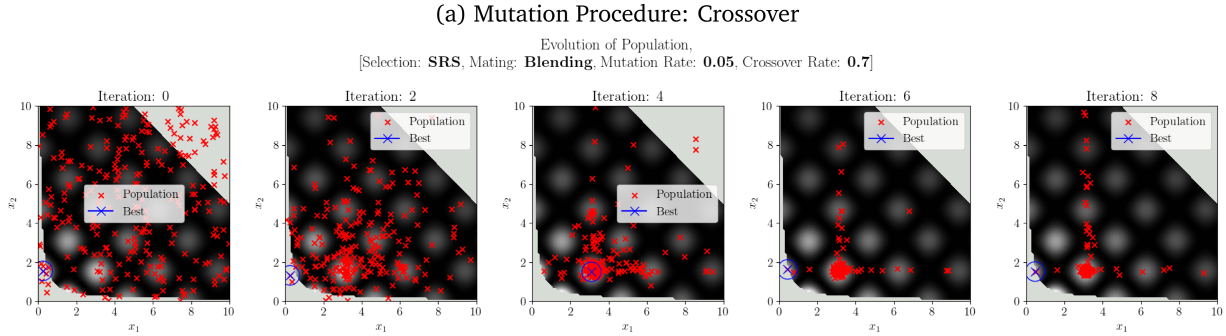
2.2.2 Choosing the Selection Method and Mutation Procedure

Selection Method	Mating Procedure	Iterations	Final Avg. Fitness	Final Min. Fitness
Proportional	Crossover	10	-0.5638	-0.6609
		100	-0.5956	-0.6609
	Blending	10	-0.5458	-0.6559
		100	-0.5356	-0.6243
Tournament	Crossover	10	-0.0203	-0.2896
		100	-0.0079	-0.0867
	Blending	10	-0.0326	-0.1305
		100	-0.0720	-0.2075
SRS	Crossover	10	-0.5671	-0.6609
		100	-0.5865	-0.6609
	Blending	10	-0.5557	-0.6536
		100	-0.5355	-0.6293

Table 1: Your Table Caption Here



(a) Mutation Procedure: Crossover



(b) Mutation Procedure: Blending

Figure 3: Evolution of the population over 10 iterations using Stochastic Remainder Selection without Replacement (SRS). SRS was found to be the most effective selection method, when compared to Proportional Selection and Tournament Selection, as seen in Figures 4 and 5, respectively.

3 Methodology

4 Results

5 Discussion

6 Conclusion

References

[1] M.A. El-Beltagy and A.J. Keane. A comparison of various optimization algorithms on a multilevel problem. *Engineering Applications of Artificial Intelligence*, 12(5):639–654, 1999.

[2] Randy L. Haupt, S. E. Haupt, and Randy L. Haupt. *Practical Genetic Algorithms*, chapter 3: The Continuous Genetic Algorithm, pages 51–66. John Wiley & Sons, Ltd, 2003.

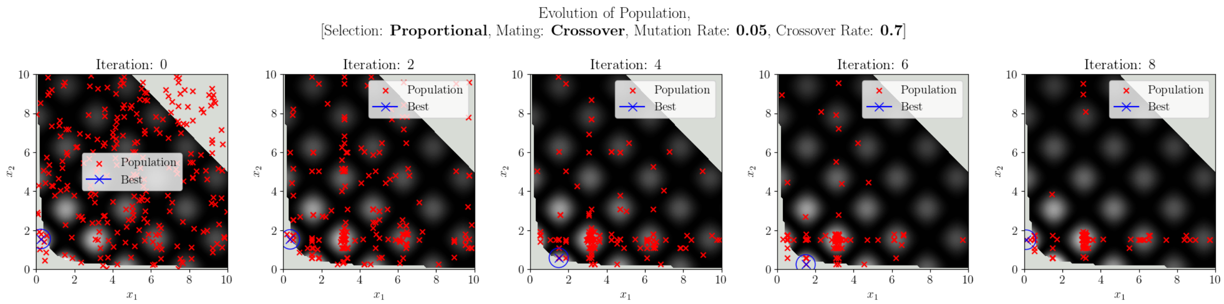
[3] Frank Hoffmeister and Thomas Bäck. Genetic algorithms and evolution strategies: Similarities and differences. In Hans-Paul Schwefel and Reinhard Männer, editors, *Parallel Problem Solving from Nature*, pages 455–469, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg.

[4] Geoff Parks. Genetic algorithms: Lecture notes. Cambridge University, 4M17 Practical Optimisation Module, 2023. Unpublished.

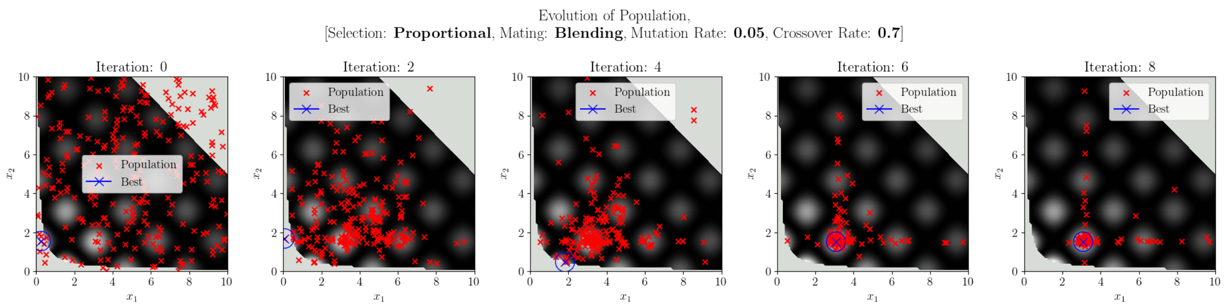
[5] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning, 2017.

7 Appendix

7.1 Supplementary Figures

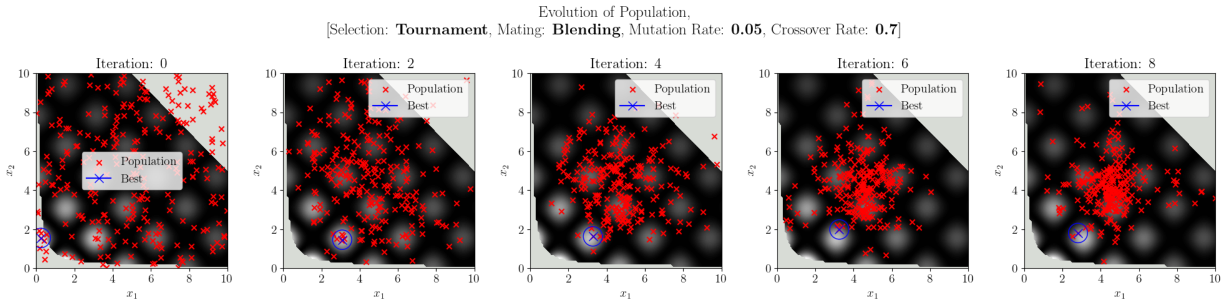


(a) Mutation Procedure: Crossover

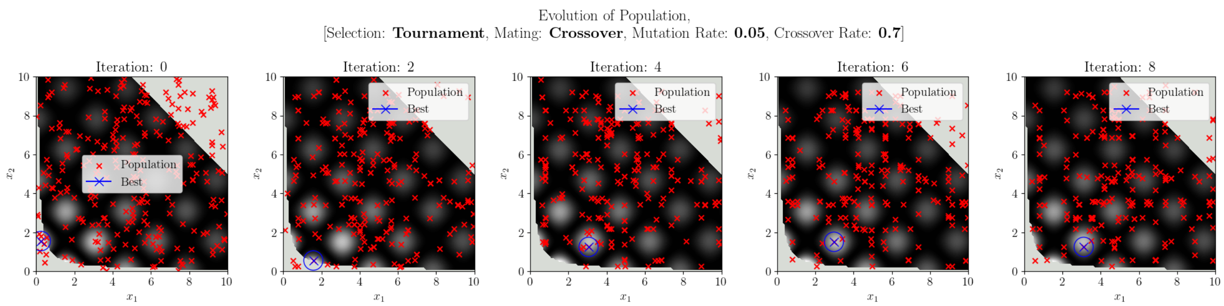


(b) Mutation Procedure: Blending

Figure 4: Evolution of the population over 10 iterations using proportional selection. Proportional selection proved to be an effective selection method. However, it produced a larger final fitness value than SRS, and was therefore not chosen as the primary selection method.



(a) Mutation Procedure: Blending



(b) Mutation Procedure: Crossover

Figure 5: Evolution of the population over 10 iterations using Tournament Selection. Using Tournament Selection, the algorithm was unable to converge to a solution. Even 100 iterations produced a suboptimal solution, showcasing convergences to local optima, rather than the global optimum. As such, it was disregarded as a viable selection method.