OPTIMAL POPULATION SIZE FOR GENETIC ALGORITHMS : AN INVESTIGATION

Michael O. Odetayo

Introduction. The performance of Genetic Algorithms (GAs) is affected by the parameters that are employed. In particular, the population size affects the performance and efficiency of GA-based systems [2]. An earlier researcher [2] claimed that a population size between 60 - 110 is optimal for the convergence of GA-based systems to optimal solution. This paper presents studies that do not support this claim. **GAPOLE,** a GA-based program, is used to build self-learning, self-adaptive, self-optimising controllers for a dynamic, multi-output, unstable system using different population sizes. It is argued that population size may need to be tuned from one application to the other.

The task. The task for GAPOLE is to induce control rules for a multi-output, unstable, dynamic system - a simulated pole-cart system [3, 4]. Formally the task is defined as follows [1, 3] :

A wheeled cart has a rigid pole hinged to its top. The cart is free to move right or left along a straight bounded track and the pole is free to move within the vertical plane parallel to the track. The cart is to be kept within the predefined limits of the track and the pole should be prevented from falling beyond a predefined vertical angle by applying a force of fixed magnitude to the left or right of the base of the cart (see fig I. below).

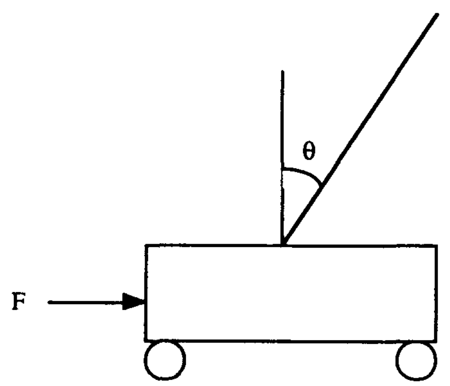

Fig I. A pole-cart system

The system is out of control if the cart has gone beyond ± 2.4 meters or the pole cart has fallen beyond ± 12° from the vertical [3,5].

Department of Computing Science, De Montfort University, The Gateway, Leicester, U.K.

<u>Experimental results and discussion</u>. GAPOLE maintained a population of learning controllers (chromosomes) which were made to evolve. It was regarded to have succeeded in balancing the pole-cart system as soon as it was able to hold the system continuously without a failure signal for 10,000 discrete time steps. **A run terminates** if one member (controller) held the pole-cart system for 10,000 time steps after all new members of the population have attempted to keep the system under control.

A summary of the performance of GAPOLE for population sizes 100, 150 , 300 and 400 when a force of 10 Newtons was applied to the base of the cart is shown in table I below.

Pop.	Generations			Failures			Time Taken (HR:MIN:SEC)		
Size	Minimum	Maximum	Mean	Minimum	Maximum	Mean	Minimum	Maximum	Mean
100	6	544	28	472	28853	1643	2:46	6:37:21	23:24
150	2	42	14	259	3571	1301	0:41	1:56:39	16:13
300	3	21	9	729	3808	1842	1:48	2:14:31	17:04
400	4	21	9	1228	5297	2394	2:58	2:55:25	23:06

Table I Perfromance Summary of GAPOLE

When Pushing Left and Right with a Force of 10 Newtons

From table I, we see that on average, GAPOLE sampled the least number of points (had the least number of failures) with a population size of 150 while the number of points sampled when using a size of 300 was almost the same as that sampled with a size of 100; it sampled the worst number of points with a size of 400. The program, on average, took the least number of generations to complete its task when a population size of 300 was used; it had worst number of generations with a size of 100 (about 3 times that of a size of 300 or 400, and twice that of a size of 150). The average completion time was highest when using a size of 100. The average completion times for sizes 150 and 300 were about the same.

It is clear from the table above that there is no need to continue experimenting with a size of 400.

Robustness of GAPOLE. In order to test the robustness of GAPOLE, uneven forces were applied to the base of the cart. A force of 5 Newtons was applied when going right while a force 10 Newtons was applied when going left. The summary of the program's performance for population sizes 100, 150, 300 are presented in table II below.

Pop.	Generations			Failures			Time Taken (HR:MIN:SEC)		
Size	Minimum	Maximum	Mean	Minimum	Maximum	Mean	Minimum	Maximum	Mean
100	4	321	43	290	17260	2382	0:48	13:50:21	41:12
150	6	97	30	702	7612	2477	3:16	2:09:10	23:50
300	4	51	16	876	8324	2876	1:41	1:53:02	19:29

Table II Perfromance Summary of GAPOLE

When using 5 Newtons to Push Right and 10 Newtons to Push Left.

From table II, we see that on average, GAPOLE sampled the least number of points with a population size of 100. However, on average, it took the least number of generations and the least time when using a size of 300. A size of 100 recorded the worst average computational time and had the highest number of generations. In addition, a population size of 100 had the worst deviations from the means. A population size of 300 is thus optimal along most dimensions when the forces are uneven. GAPOLE is therefore, most adaptive with a population size of 300.

From all the facts presented above, it is clear that a size of 300 is optimal for our task.

Conclusion. GAPOLE, a Genetic Algorithm-based program was used to automatically build self-learning, self-adaptive, self-optimising controllers for a multi-output, unstable, dynamic system - a simulated pole-cart system. The performance of GAPOLE using different population sizes under different conditions was investigated. A population size of 300 was found to be optimal for the convergence of the Genetic Algorithm-based program to the optimal solution. This result does not support an earlier claim that a population size between 60 - 110 is optimal across domains for the convergence of Genetic Algorithm-based systems to the optimal solutions. Therefore, one must conclude that this parameter may need to be tailored to a particular application.

References

[1] C. W. Anderson, Strategy Learning with Multilayer Connectionist Representations, in **Proceedings of the Fourth International Workshop on Machine Learning**, ed. Pat Langely, 1987, pp. 103 - 114.

[2] John J. Grefenstette, Optimization of Control parameters for Genetic Algorithms, in **IEEE Transactions on Systems, Man, And Cybernetics**, vol. Smc-16, No. 1, January/February 1986.

[3] Michael O. Odetayo, On Genetic Algorithms in Machine Learning and Optimisation, **Ph.D. Thesis**, University of Strathclyde, Glasgow, 1990.

[4] M. O. Odetayo and D. R. McGregor, Genetic Algorithm for Inducing Control Rules for a Dynamic System, in **Proceedings of the Third International Conference on Genetic Algorithms**, 1989, pp. 177 - 182.

[5] Claude Sammut, Experimental results From an Evaluation of Algorithms that Learn to Control dynamic systems, in **Proceedings of The Fifth International Conference on Machine Learning**, ed. John Laird, 1988, pp. 437 - 443.

Printed and published by the IEE, Savoy Place, London WC2R 0BL, UK.