6-2011

Distributed Learning with Biogeography-Based Optimization

Carre Scheidegger
Cleveland State University

Arpit Shah
Cleveland State University

Daniel J. Simon
Cleveland State University, d.j.simon@csuohio.edu

Publisher's Statement

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-642-21827-9_21

Distributed Learning with Biogeography-Based Optimization

Carre Scheidegger, Arpit Shah, and Dan Simon

Cleveland State University, Electrical and Computer Engineering

Abstract. We present hardware testing of an evolutionary algorithm known as biogeography-based optimization (BBO) and extend it to distributed learning. BBO is an evolutionary algorithm based on the theory of biogeography, which describes how nature geographically distributes organisms. We introduce a new BBO algorithm that does not use a centralized computer, and which we call distributed BBO. BBO and distributed BBO have been developed by mimicking nature to obtain an algorithm that optimizes solutions for different situations and problems. We use fourteen common benchmark functions to obtain results from BBO and distributed BBO, and we also use both algorithms to optimize robot control algorithms. We present not only simulation results, but also experimental results using BBO to optimize the control algorithms of mobile robots. The results show that centralized BBO generally gives better optimization results and would generally be a better choice than any of the newly proposed forms of distributed BBO. However, distributed BBO allows the user to find a less optimal solution to a problem while avoiding the need for centralized, coordinated control.

1 Introduction

Biogeography is the study of the geographical distribution of plant and animal life. Alfred Wallace and Charles Darwin were some of the first to observe the patterns of biogeography and introduce the subject to the scientific world [1]. Biogeography did not evolve into the quantitative science that it is today until Robert MacArthur and Edward Wilson created models from their studies of island biogeography in the early 1960s [2]. Other scientists also contributed to the emergence of the theory of island biogeography, most notably Eugene Monroe in 1948 [12]. Biogeography has continued to develop after MacArthur and Wilson's research and it has recently been used as the motivating framework for the development of an evolutionary algorithm (EA) called biogeography-based optimization (BBO) [3]. In this paper, we apply BBO to experimental mobile robotics control optimization.

This paper gives an overview of the evolutionary algorithm called BBO, which is based on mathematical models of biogeography, and which has been developed to solve general optimization problems [3]. BBO has been applied to several real-world problems. In addition to experimental robot control tuning, as discussed in this paper and in [4], BBO has been applied to aircraft engine sensor selection [3], power system optimization [13, 14], groundwater detection [15], mechanical gear train design [16], satellite image classification [17], and neuro-fuzzy system training for biomedical

applications [8]. Recent research in the area of BBO has focused on putting it on a firm theoretical and mathematical foundation, including the derivation of Markov models [19, 20] and dynamic system models [21] that describe its behavior.

In this paper, we use BBO in computer generated simulations and on experimental mobile robots to study its performance. This paper also develops BBO's distributed counterpart, which is based on distributed learning and intelligence, and the simulation results gathered from the distributed algorithm. The distributed algorithm's communication and control is distributed through various BBO individuals rather than coordinated by a central computer. The development of distributed BBO (DBBO) has been motivated by the confluence of centralized BBO and concepts from distributed learning.

Distributed learning is a theory developed to explain how the human mind understands and learns [7] [9]. Human's mental capabilities are often assumed to be centralized inside the brain, but research has shown that outside social interactions greatly affect how the brain learns [7]. Distributed learning is an example of this type of environmental influence, and is often seen in humans working in teams to solve a common problem or complete a task [5] [6]. Distributed learning has been used in recent years to study how automated technology can be taught to perform human-like tasks using teams of robots that function together with nearly the same effectiveness as a team of humans [6] [7] [10] [11].

This distributed learning or intelligence in robotics is the ability of numerous entities to solve problems, perform tasks, learn, and understand by communicating with other entities in a group, rather than under the control of a centralized coordinator [6] [9] [10] [11]. Each organism is considered an individual with governing logic that allows it to perform a particular task, and the way in which these distributed learning groups carry out a task is dependent on the mode of communication by each entity [5] [6] [8] [10]. The study of distributed interaction is helpful in discovering and researching different ways to make artificial intelligences communicate to perform like a group of human beings, and has increasingly been applied to different types of systems. The advantages of distributed interaction in robotics open doors to new types of systems that do not need a central processor to control the team of robots, and allow flexibility for change and improvement. This flexibility in robotic applications of distributed intelligence is increasingly being studied because it has advantages that centralized systems do not. These advantages include autonomy, fault tolerance, and robustness.

Section 2 in this paper gives an outline of BBO. It also proposes a distributed extension of BBO, which is one of the primary contributions of this paper. Section 3 presents fourteen benchmark function simulation results. Section 4 discusses the mobile robot system used for an experimental application of BBO and DBBO, including its design, hardware, task description, simulation results, and experimental results. Section 5 concludes with some discussion and suggestions for future work.

2 Biogeography-Based Optimization (BBO)

BBO is an evolutionary algorithm that is modeled on the theory of biogeography. These models of biogeography mathematically describe how species travel to and

from environments based on different factors of the environment [2]. These environmental factors can be represented quantitatively and are called suitability index variables (SIVs) and determine the suitability of the area for habitation. Examples of natural SIVs seen often in habitats are the amount of rainfall, the diversity of vegetation, and the temperature range. An area that is highly habitable is considered to have a high habitat suitability index (HSI) [3]. Habitats can be observed by scientists to develop mathematical models of migration, speciation, and extinction.

A high-HSI habitat is likely to have a large number of species. It has a high rate of emigration and a low rate of immigration due to its dense population. The opposite occurs in low-HSI habitats because of the habitat's sparse population. Emigration and immigration rates in a given habitat are proportional to the number of species that reside in that habitat. This habitat suitability concept that biogeography quantifies is what makes biogeography applicable to optimization problems in engineering and other fields of study. An individual in an evolutionary algorithm that has a high HSI (*performance,* or *fitness*) represents a good solution, and that individual will emigrate successful features to other individuals. Individuals that receive features from successful candidate solutions tend to increase their own fitness. Quantifiably applying the emigration and immigration of specific features from one individual to another depending on the individuals' HSI values generally creates better solutions. Biogeography-based optimization (BBO), which was introduced by Simon [3], is the implementation of this extension of biogeography to optimization.

2.1 Centralized BBO

Centralized BBO is the original BBO algorithm created to optimize solutions based on the theory of biogeography, and it uses the migration of traits to create better generations of candidate solutions to an optimization problem. As explained previously, a habitat's migration rates are dependent on the habitat's HSI. A habitat is analogous to a problem solution, and its HSI is analogous to the fitness of the solution. A solution's fitness determines its rates of immigration, λ, and emigration, μ, and is determined in a way that is similar to natural biogeography. BBO bases the migration rate of each candidate solution on the HSI of the solution, with high HSI giving a high emigration rate, and low HSI giving a high immigration rate.

BBO operates in a way that allows each generation of candidate solutions to improve from one generation to the next. Migration, mutation, and elitism are three characteristics of BBO that allow the population of candidate solutions to share information and keep the best solutions of each generation for the following generation. Migration is among the most influential and unique part of the BBO algorithm and it allows the individuals in the system to immigrate or emigrate data from other individuals in a single generation of the program. Mutation operates as in other evolutionary algorithms, and encourages diversity in the population and allows the replacement of a specific solution feature with another randomly generated solution feature. Mutation allows problem solutions to improve, but it also introduces the risk of degrading the solution. Elitism also operates as in other evolutionary algorithms, and it counteracts the risks of mutation. Elitism saves the best problem solutions at each generation, and replaces the worst solutions in the next generation with these elite solutions [3]. The listing of Algorithm 1 depicts a single generation of centralized BBO.

Algorithm 1. Basic description of the BBO algorithm for one generation

For each candidate problem solution P_i
    Calculate immigration probability λ_i and emigration probability μ_i :
        $\mu_i \in [0, 1]$ is proportional to the fitness of P_i, and $\lambda_i = 1 - \mu_i$
Next candidate solution: $i \leftarrow i+1$
For each candidate problem solution P_i
    For each solution variable v in P_i
        Use immigration probability λ_i to decide whether to immigrate to P_i
        If immigrating to P_i
            Select P_k for emigration according to the probability μ_k
            P_k emigrates data to P_i : $P_i(v) \leftarrow P_k(v)$
        End immigration
    Next solution variable
    Mutate P_i probabilistically based on mutation probability
Next candidate solution: $i \leftarrow i+1$

2.2 Distributed BBO

One of the main contributions of this paper is the development of distributed biogeography-based optimization (DBBO). Distributed BBO is based on the centralized BBO algorithm, but has been motivated by the theory of distributed systems. DBBO has goals similar to centralized BBO, which is to optimize a task or problem solution. However, it does not use a centralized computer for control, but rather each individual in DBBO is capable of performing the evolutionary algorithm on its own. Although distributed BBO is very closely related to the original centralized implementation,

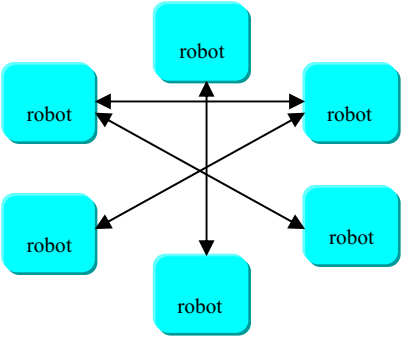

Fig. 1. Peers randomly choose one another for communication and feature-sharing in distributed BBO

it allows evolution in cases in which the use of a central computer is not ideal or possible. Figure 1 shows an example of a team of robots communicating peer-to-peer and not through a centralized coordinator.

The DBBO algorithm runs on each individual separately. We demonstrate this concept using mobile robots in this paper. The individuals were programmed to share information from peer to peer, rather than through a central computer. A peer contains a single problem solution (for example, a single robot with its control algorithm), and a few peers are randomly chosen to communicate and share information with each other. DBBO has the same basic characteristics as BBO and it also uses mutation to diversify the solution sets. Centralized BBO uses elitism; however, elitism is not used in distributed BBO because only a few individuals (as few as two) communicate with each other at each generation. Algorithm 2 shows a simple description of a single generation of the DBBO algorithm.

Algorithm 2. Basic description of the DBBO algorithm for one generation

Select m peers $\{P_i\}$ for communication with each other
Revise each peer's best and worst cost estimates. For each i,
    $\text{MinEst}_i = \min_{k \in I} \{\text{MinEst}_k\}$ and $\text{MaxEst}_i = \max_{k \in I} \{\text{MaxEst}_k\}$, where
    I is the set of all peers of robot i
Calculate each peer's likelihood to immigrate, λ, and emigrate, μ :
    $\mu_i \in [0, 1]$ is proportional to the fitness of P_i relative to its peers, and $\lambda_i = 1 - \mu_i$
For each peer P_i
    For each solution variable v
        Use immigration probability λ_i to decide whether to immigrate to P_i
        If immigrating to P_i
            Select P_k for emigration according to the probability μ_k
            P_k emigrates data to P_i : $P_i(v) \leftarrow P_k(v)$
        End immigration
    Next solution variable
    Mutate P_i according to mutation probability
Next peer

3 Benchmark Simulation Results

As in previous research with BBO [3], we used 14 common benchmark functions to simulate the BBO and DBBO algorithms. Distributed BBO was used with three different numbers of peers (2, 4, and 6). We were able to compare the results of BBO against DBBO/2, DBBO/4, and DBBO/6 for the benchmark functions from 100 Monte Carlo simulations. The results from the Monte Carlo simulations were analyzed based on the best cost functions returned by BBO and DBBO. The same parameters were used in all four runs of the BBO and DBBO algorithms. After analysis

of the best combination of variables, we chose a population size of 50, we used 20 independent variables (i.e., each benchmark has a dimension of 20), and we used a 1% probability of mutation for each independent variable.

Figures 2 and 3 are normalized plots of the cost function values calculated from the simulation runs of the benchmark functions. The value used for normalization was the minimum value achieved for each benchmark function by BBO and the three DBBO versions. However, if a minimum value was 0, we instead normalized to the second smallest value for that benchmark. We analyzed the minimum cost and average cost over the 100 Monte Carlo simulations. The minimum cost figure shows the minimum cost achieved by each algorithm for each benchmark after 100 simulations. BBO minimized the cost functions of each benchmark function most often and generally out-performed DBBO. The average cost in Figure 3 shows which BBO version optimized best on average.

In general, the average of each benchmark function and each algorithm was fairly good. Centralized BBO, however, usually obtained the best costs on average. However, on occasion BBO had a higher minimum cost than DBBO. This means that DBBO is sometimes more likely than BBO to get closer to the minimum. We intuitively expect BBO to outperform DBBO. But just because BBO has more candidate solutions to choose from when performing immigration does not guarantee that it will outperform DBBO. The fact that BBO has more candidate solutions to choose from can just as easily result in a detrimental immigration as a beneficial immigration. Although we usually see BBO outperform DBBO, there is no guarantee of this advantage, and further research is needed to determine the conditions under which BBO or DBBO will give better optimization results.

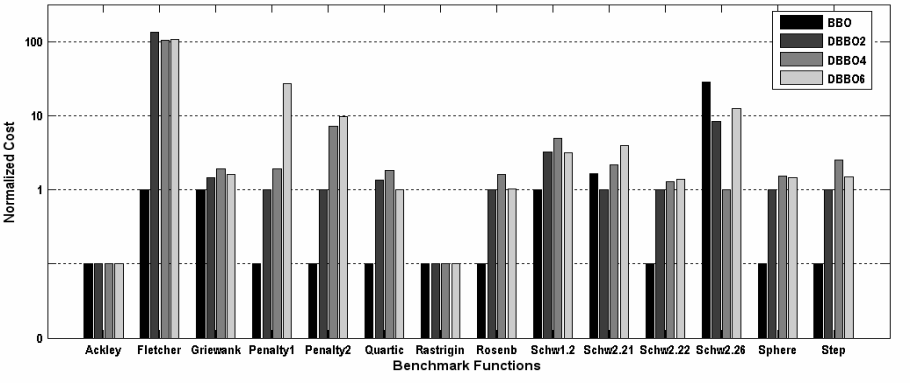

Fig. 2. Plot of the minimum cost function values (best performance) for BBO, DBBO/2, DBBO/4, and DBBO/6 over 100 Monte Carlo simulations. BBO usually performs better than DBBO, but not always.

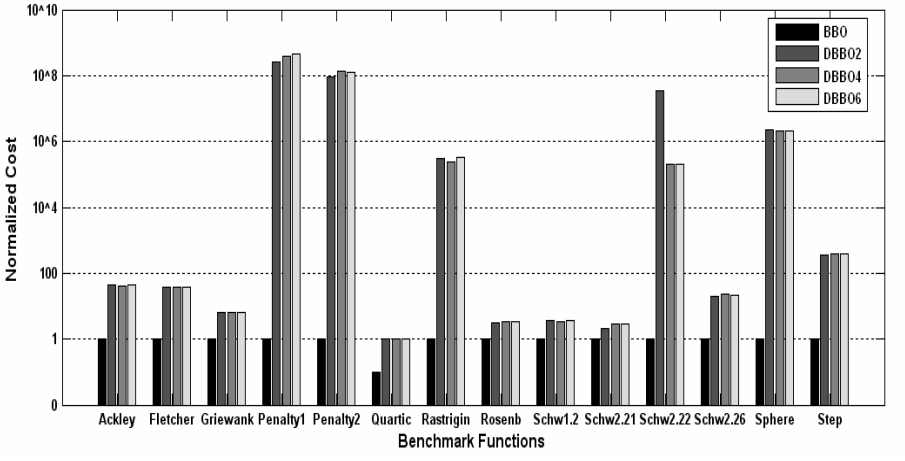

Fig. 3. Plot of the average cost function values for BBO, DBBO/2, DBBO/4, and DBBO/6 over 100 Monte Carlo simulations. BBO performs better than DBBO for every benchmark.

4 Robot Optimization Using BBO and DBBO

This section discusses the use of BBO and DBBO for robot controller optimization. Section 4.1 discusses the robot hardware that we used. Section 4.2 discusses the robot control task. Section 4.3 presents simulation results, and Section 4.4 presents experimental hardware results.

4.1 Robot Hardware

The physical application of BBO in this research is performed on mobile robots that have been used in previous research in which BBO has optimized the control parameters [4]. The robots are equipped with two DC motors and eight AA batteries. The batteries power the two motors and the circuit board. The main control base of the robots is the microcontroller, which is a Microchip PIC18F4520. The microcontroller is equipped to control the motors and communicate with a PC. In BBO the microcontroller communicates with a central PC using a wireless radio, the MaxStream 9Xtend radio. In BBO the radio sends all the necessary commands, parameters, and data to each individual robot. In DBBO the radio signals are sent between individual robots. Two voltage regulators are used on the robots to distribute a constant 5 volts to the microcontroller and to the motors. The voltage regulators ensure that the microcontroller receives enough current to function correctly. In order to power the motors the signal from the microcontroller is used to switch the motor power supply, so an H-bridge SN754410NE is also used. The final major hardware is the infrared sensors.

The infrared sensors are the main component that measure the distance of the robot from a wall during its tracking task using a light-emitting sensor and a light-detecting sensor. Figure 4 shows a photograph of the robots.

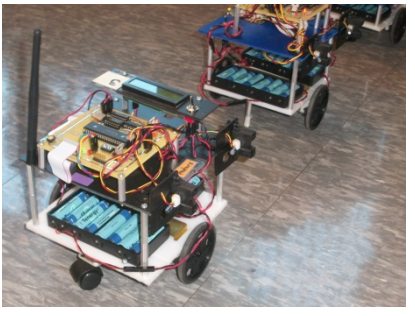

Fig. 4. Photograph of the mobile robots used for BBO and DBBO testing

4.2 PID Control

The robot controller is a proportional-integral-derivative (PID) controller, a very commonly used algorithm for control [22]. PID uses different factors to determine what a control system needs to do. The proportional gain K_p is generally a large number for performance purposes, and is the primary determinant of the amount of control signal. The higher the value of K_p the faster the controller responds to tracking errors. The integral gain K_i speeds up the output movement to the desired value and helps to reduce the steady-state error resulting from the proportional gain. The final parameter of PID control is K_d and is multiplied by the rate of change of the output of the system. It helps keep the controller stable and decrease the range of overshoot created by K_i and K_p. The robots used in our BBO and DBBO research use K_p and K_d terms.

The mobile robots used for testing BBO and DBBO are applied to a particular tracking task. The robots are programmed to follow a wall, using the PID controller to maintain a specified distance from the wall. The robot uses infrared sensors to measure its distance and angle from the. The robot controller uses a calculated error from the measured values to correct the motor output. The controller output controls the voltage input to the two motors (one for the left wheel and one for the right wheel). Using this control output, the robot is able to adjust its wheel speed to compensate for its error. See [4] for more details about the robots and their control algorithm.

4.3 Simulation

Simulation results were generated using a robot function in place of the benchmark functions used in Section 3. The robot function simulated a robot's program for the desired task of following a wall at a certain reference distance. To make the simulations close to a real robot application, the BBO and DBBO parameters were set differently than the benchmark simulation parameters. We ran BBO and DBBO with a

maximum of 50 function evaluations, a population size of 5, and a mutation rate of 15% per independent variable. The mutation rate was chosen to be relatively high because of the small population size. In the benchmark simulations the population size was much larger, but to have a realistic physical implementation a small population size of only 5 was chosen for the robot simulations. To create a fair chance of mutation of each problem solution, the mutation rate needed to be increased to a relatively high rate. The robot simulation function also had the following parameters that needed to be set: K_p and K_d minimums and maximums. The domain of K_p was set to [0, 2], and the domain of K_d was set to [0, 10]. Testing was done with 100 Monte Carlo simulations.

The robot simulation results using BBO and DBBO/2, DBBO/4, and DBBO/6 are shown in Table 1. We analyzed the minimum cost, maximum cost, average cost, and standard deviation of the four algorithms for 100 optimization trials. The cost function analyzed is the sum of the rise time r of the controller and integral of the absolute tracking error [4]:

$$\text{Cost} = k_1 \int \left| e(t) \right| dt + k_2 r \tag{1}$$

Table 1. Cost values from 100 Monte Carlo simulations of the BBO and DBBO algorithms

	BBO	DBBO/2	DBBO/4	DBBO/6
Minimum Cost	7.48	7.23	7.30	7.16
Maximum Cost	7.99	8.12	8.07	8.10
Average Cost	7.68	7.78	7.77	7.76
Standard Deviation	0.119	0.169	0.147	0.193

As predicted from the benchmark results of the previous section, and as seen in Table 1, BBO performed better, on average, than the distributed versions of the algorithm. The centralized algorithm had the lowest worst-case cost value, performed best on average, and had the smallest standard deviation. However, it is interesting to note that the DBBO versions had better best-case performance than centralized BBO. Among the DBBO algorithms, they all performed very similarly, neither one significantly outperforming another. However, if one had to be chosen, 6-peer communications might be the best choice because of its low minimum cost value, and relatively low average cost. On average, BBO returned better results than any distributed algorithm. However, DBBO still is capable of performing well enough to be used in situations where a centralized processor is not available.

4.4 Experimental Results

We used four robots in our experiments. The four experimental robots' initial K_p and K_d values varied between each robot and were set randomly. The initial K_p values for

the robots were 0.93, 0.07, 0.18, and 0.12. The initial K_d values for the robots were 4.26, 6.36, 2.45, and 2.21. The K_p and K_d values change from one generation to the next as different robots communicate using the DBBO algorithm. The final values of K_p after 8 generations were 0.82, 0.07, 0.67, and 0.02, and the final values of K_d were 9.03, 3.41, 4.32, and 2.03. From their initial to their final values, the robots' control parameters changed as follows.

Robot 1: $K_p = 0.93 \rightarrow 0.82$
$K_d = 4.26 \rightarrow 9.03$
Robot 2: $K_p = 0.07 \rightarrow 0.07$
$K_d = 6.36 \rightarrow 3.41$
Robot 3: $K_p = 0.18 \rightarrow 0.67$
$K_d = 2.45 \rightarrow 4.32$
Robot 4: $K_p = 0.12 \rightarrow 0.02$
$K_d = 2.21 \rightarrow 2.03$

Figure 5 shows the experimental results from the robots using the distributed BBO algorithm DBBO/2 on four mobile robots. Two robots were communicating per generation in this particular experiment. The DBBO program shows successful optimization over 8 generations. Both the minimum cost of the four robots, and the average cost of the four robots are decreasing as the generation count increases. The decreasing cost values show that the robots are learning to have smaller fluctuations in their path as they track a certain distance from a wall. The best K_p and K_d values were 0.07 and 3.41 (robot #2). These values returned the smallest minimum cost. Figure 6 shows a simplified flowchart of the distributed BBO algorithm DBBO/2 as applied to robot controller optimization.

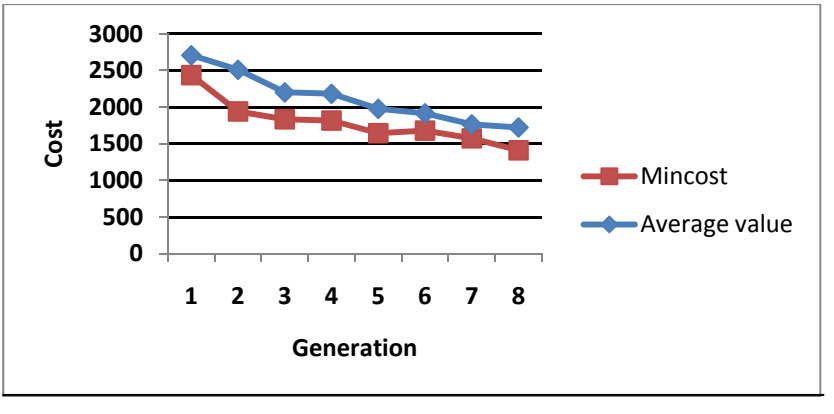

Fig. 5. Minimum and average cost value of the distributed BBO algorithm DBBO/2 on four mobile robots

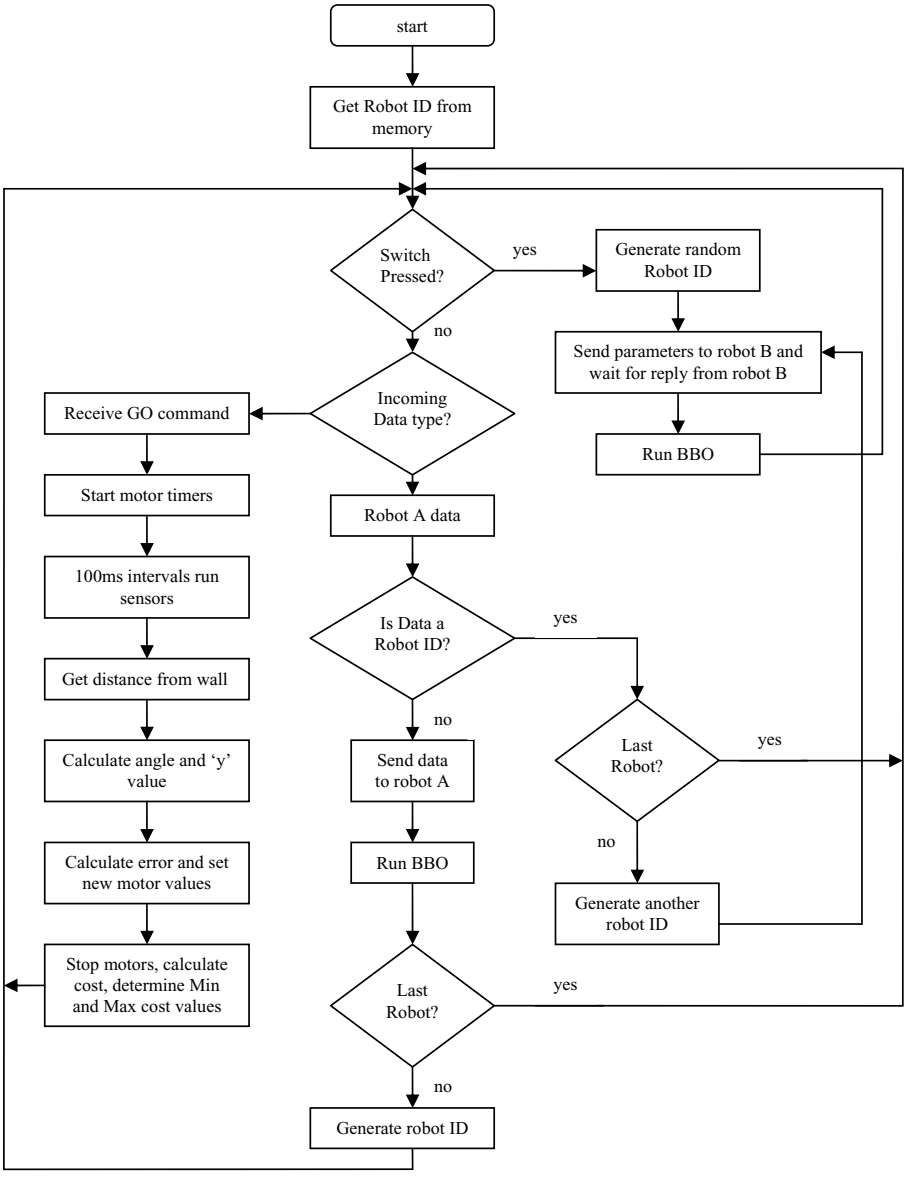

Fig. 6. Basic flow chart of the distributed BBO algorithm for robot control optimization. Pressing a switch on a random robot, which is called "Robot A," begins the DBBO process. Robot A randomly selects another robot, which is called "Robot B," with which to perform migration of solution features.

5 Conclusion

This work successfully extended BBO to its distributed counterpart, and presented simulations and experiments to see which would be better in different situations. The results from this paper show that BBO and DBBO are able to optimize benchmark functions and the real-world problem of robot controller tuning. Although distributed BBO offers more flexibility, centralized BBO returns the best results, on average. This topic can be researched further by using simulations and experiments to explore the effect of different BBO and DBBO parameter settings, including population sizes, mutation rates, and number of communicating peers in DBBO. Other future work includes using theoretical Markov modeling [19, 20] and dynamic system modeling [21], which has been performed for BBO, and extending it to DBBO.

References

1. Quammen, D.: The Song of the Dodo: Island Biogeography in an Age of Extinction. Simon & Schuster, New York (1997)
2. Mac Arthur, R.H., Wilson, E.O.: The Theory of Island Biogeography. Princeton University Press, Princeton (1967)
3. Simon, D.: Biogeography-Based Optimization. IEEE Transactions on Evolutionary Computation 12(6), 702–713 (2008)
4. Lozovyy, P., Thomas, G., Simon, D.: Biogeography-Based Optimization for Robot Controller Tuning. In: Igelnik, B. (ed.) Computational Modeling and Simulation of Intellect: Current State and Future Perspectives. IGI Global (in print, 2011)
5. Parker, L.E., Touzet, C.: Multi-Robot Learning in a Cooperative Observation Task, pp. 391–401 (2000)
6. Parker, L.E.: Distributed Intelligence: Overview of the Field and its Application in Multi-robot Systems. Journal of Physical Agents 2, 5–14 (2008)
7. Fischer, G.: Distributed Intelligence: Extending the Power of the Unaided, Individual Human Mind, pp. 7–14 (2006), `http://l3d.cs.colorado.edu/`
8. Van Dam, K.H., Verwater-Lukszo, Z., Ottjes, J.A., Lodewijks, G.: Distributed intelligence in autonomous multi-vehicle systems. International Journal of Critical Infrastructures 2, 261–272 (2006)
9. Valavanis, K.P., Saridis, G.N.: Intelligent Robotic Systems: Theory, Design and Application. Kluwer Acadamic, Boston (1992)
10. Weiss, G.: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. The MIT Press, Cambridge (1999)
11. O'Hare, G.M.P., Jennings, N.: Foundations of Distributed Artificial Intelligence. John Wiley and Sons, New York (1996)
12. Lomolino, M.V., Brown, J.H.: The Reticulating Phylogeny of Island Biogeography Theory. Q Rev. Biol., 357 – 390 (2009)
13. Rarick, R., Simon, D., Villaseca, F., Vyakaranam, B.: Biogeography-based optimization and the solution of the power flow problem. In: IEEE Conference on Systems, Man, and Cy-bernetics, pp. 1029–1034 (2009)
14. Roy, P., Ghoshal, S., Thakur, S.: Biogeography-based optimization for economic load dispatch problems. Electric Power Components and Systems (38), 166–181 (2010)

15. Kundra, H., Kaur, A., Panchal, V.: An integrated approach to biogeography based optimization with case based reasoning for retrieving groundwater possibility. In: 8th Annual Asian Conference and Exhibition on Geospatial Information, Technology and Applications (2009)
16. Savsani, V., Rao, R., Vakharia, D.: Discrete optimisation of a gear train using biogeography based optimisation technique. International Journal of Design Engineering (2), 205–223 (2009)
17. Panchal, V., Singh, P., Kaur, N., Kundra, H.: Biogeography based satellite image classification. International Journal of Computer Science and Information Security (6), 269–274 (2009)
18. Ovreiu, M., Simon, D.: Biogeography-based optimization of neuro-fuzzy system parameters for diagnosis of cardiac disease. In: Genetic and Evolutionary Computation Conference, pp. 1235–1242 (2010)
19. Simon, D., Ergezer, M., Du, D., Rarick, R.: Markov models for biogeography-based optimization. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics (41), 299–306 (2011)
20. Simon, D., Ergezer, M., Du, D.: Population distributions in biogeography-based optimization algorithms with elitism. In: IEEE Conference on Systems, Man, and Cybernetics, pp. 1017–1022 (2009)
21. Simon, D.: A Dynamic System Model of Biogeography-Based Optimization (2010) (submitted for publication)
22. Astrom, K., Hagglund, T.: PID Controllers: Theory, Design, and Tuning. International Society for Measurement and Control, Research Triangle Park, North Carolina (1995)