Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Biogeography-Based Optimization and the Solution of the Power Flow Problem

Rick Rarick, Dan Simon, F. Eugenio Villaseca, Bharat Vyakaranam
Department of Electrical and Computer Engineering
Cleveland State University
Cleveland, Ohio, USA
r.rarick@csuohio.edu

Abstract—**Biogeography-based optimization (BBO) is a novel evolutionary algorithm that is based on the mathematics of biogeography. Biogeography is the study of the geographical distribution of biological organisms. In the BBO model, problem solutions are represented as islands, and the sharing of features between solutions is represented as immigration and emigration between the islands. This paper presents an application of the BBO algorithm to the power flow problem for an IEEE 30-bus Test Case system. The BBO solution is compared with the solution of the same problem using a genetic algorithm (GA). The results of Monte Carlo simulations indicate that the BBO algorithm consistently performs better than the GA in determining an optimal solution to the power flow problem.**

Keywords—**Biogeography-Based Optimization, Power Flow Problem, Genetic Algorithms, Evolutionary Algorithms, Power Systems.**

I. INTRODUCTION

Classical algorithms for solving optimization problems rely strongly on initial values and convexity for their success in determining global optima. The algorithms frequently get trapped in local optima or diverge. Modern heuristic techniques such as genetic algorithms (GA) and, more generally, evolutionary algorithms, have proven to be effective in solving such problems [1]. These techniques are based on the science and mathematics of biological genetics which seeks to understand and model the way populations in nature such as insects, animals, and humans "solve" evolutionary problems.

The science of biogeography can be traced to the 19th century work of Alfred Wallace [2] and Charles Darwin [3]. Eugene Munroe was the first to introduce mathematical models of biogeography in 1948 [4], and Robert MacArthur and Edward Wilson were the first to extensively develop and publicize them in the 1960s [5]. Biogeography-based optimization [6] is founded on the observation that the migration of species among a group of neighboring islands, combined with mutation of the individual species, will tend over many generations to produce islands that attract and keep large numbers of species through immigration. Other islands will lose species through extinction or emigration and will sometimes become desolate. The BBO algorithm seeks to model this behavior in a way that causes an "optimal" island to emerge from the original population of islands.

An optimization problem that is of primary importance for power utilities is the optimal power flow (OPF) problem which was introduced in 1962 by the French engineer Jules Carpentier [7]. A power utility may own power generating plants fueled by coal, natural gas, and nuclear material, as well as hydroelectric plants. The cost of operation and the amount of emissions per kilowatt-hour of these plants will vary depending on the type of plant. Also, the demand for power will vary depending on the location of the plant, the time of day, and the season of the year. It is important to adjust the power output levels among the various generating plants in the system at any given time so as to minimize the operating costs and simultaneously meet three major constraints. First, the Law of Conservation of Energy requires that the power generated by the plants must equal the power absorbed by the loads plus the power losses that occur in the transmission system. Second, both the active and reactive power levels and the voltage levels at each bus in the system must be maintained within specified limits for efficiency, the protection of equipment, and the safety of the public and electrical workers. And third, the maximum amount of emissions that a plant is allowed to produce is regulated by law in most countries.

The mathematical model of a large power system network is nonlinear and nonconvex. The classical approach to solving such problems has been the gradient method using Lagrange multipliers. The solution of the optimization problem requires solving very large systems of nonlinear equations. It is not unusual to have 5,000 buses in a large power system with 10,000 unknown variables that must be constrained by as many equations. In addition there can be 10,000 inequality constraints on these variables. Inverting the large matrices and solving the large matrix equations is a very formidable task. The matrices are extremely sparse, and special numerical techniques must be employed to ensure accuracy and stability. The inequality constraints contribute an additional layer of difficulty to the problem [8]. Evolutionary algorithms avoid all of the above mathematical difficulties, and consequently they are being employed increasingly in the solution of large, intractable power system problems. The OPF research literature contains numerous adaptations of genetic algorithms [9], as well as more recent techniques such as evolutionary programming [10], particle swarm algorithms [11], and ant colony searches [12].

The OPF problem is usually solved in two stages. First, numerical techniques such as the Newton-Raphson method are used to solve the so-called power flow problem (not the same as the OPF problem). This part of the solution guarantees that the equality constraints arising from the conservation of energy principle mentioned above are satisfied. Then the solution to this power flow problem is used as the starting point (initial

This work was supported by NSF Grant 0826124 in the CMMI Division of the Engineering Directorate and by Department of Energy Contract DE-FC26-06NT42853.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

guess) for the various numerical techniques used to solve the OPF problem. In this preliminary study, the GA and the BBO algorithms were restricted to the power flow problem. In addition to demonstrating a new way of solving the power flow problem, we compare the performance of the GA to the new BBO algorithm in a practical power systems application and show the superior performance of the latter via Monte Carlo simulations.

The remainder of the paper is organized as follows. Section II describes the BBO algorithm and lists the pseudocode required for its implementation. Section III explains the formulation of the power flow problem and the application of both algorithms to the problem. In Section IV, the results of the MATLAB® simulations are given and the performance of the algorithms is compared. Some concluding remarks and topics for future investigations are given in Section V.

II. The Biogeography-Based Optimization Algorithm

In a group of neighboring islands, species of plants and animals will migrate over time between the islands by various means, being carried along by driftwood, fish, birds, and the wind. Over evolutionary periods of time, some islands may tend to accumulate more species than others because they posses certain environmental features that are more suitable to sustaining those species than islands with fewer species. This ability to sustain larger numbers of species can be associated with a fitness measure that we can quantify by assigning an island suitability index (*ISI*) to each island. The value of the *ISI* depends on many features of the island. If a value is assigned to each feature, then the *ISI* is a function of these values. Each of these values is represented by a suitability index variable (*SIV*). These mappings are summarized as follows:

$$\text{Island} \rightarrow \left(\text{feature}_1, \ldots, \text{feature}_n\right) \rightarrow \left(SIV_1, \ldots, SIV_n\right) \rightarrow ISI$$

An island with a large number of species (a large *ISI*) has an abundance of species which can emigrate to other islands, so its rate of emigration, denoted by μ, is correspondingly large. The island is also less likely to be able to sustain further immigration of species because of the growing demand on its finite environmental resources, so its immigration rate, denoted by λ, is small. For many applications it suffices to assume a linear relationship between an island's *ISI* and its immigration and emigration rates and that these rates are the same for all islands under consideration (the population). These relationships are depicted in Figure 1.

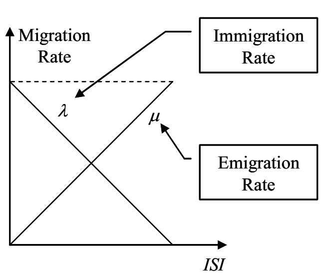

Figure 1: Island Migration Rates vs. *ISI*

In order to apply the BBO concept to an optimization problem, the *n*-tuple $\left(SIV_1, \ldots, SIV_n\right)$ associated with the features of an island is viewed as a possible solution to the optimization problem. In other words, the set of all such *n*-tuples is the search space from which an optimal solution will be determined. The value of the *ISI* for a particular island is viewed as the value of the objective function associated with that solution. The goal of the BBO algorithm, then, is to determine the solutions which maximize the *ISI* over the entire search space.

We can use the migration rates of each solution to probabilistically share features between islands. For each *SIV* (feature) in each island (solution), we probabilistically decide whether or not to immigrate. If immigration is selected for a given *SIV*, then the emigrating island is selected probabilistically using roulette wheel selection [9] normalized by μ. After the migration operation, a mutation operation is probabilistically applied to the island to increase diversity in the population. This gives the algorithm shown in Figure 2 as a conceptual description of one generation using this approach. We use the notation $y_k(s)$ to denote the *s*-th feature of the *k*-th island in a population *y* of islands. Migration and mutation of the entire population take place before any of the solutions are replaced in the population, which requires the use of the temporary population *z*.

$z \leftarrow y$
For each island z_k
  For each SIV s
    Use λ_k to decide whether to immigrate to $z_k(s)$
    If immigrating to $z_k(s)$ **then**
      Use μ to select the emigrating island y_j
      $z_k(s) \leftarrow y_j(s)$
    End if
    Probabilistically decide whether to mutate $z_k(s)$
  Next SIV $s+1$
Next island z_{k+1}
$y \leftarrow z$

Figure 2: One Generation of the BBO Algorithm

III. The Power Flow Problem

Power systems are comprised of generators, transformers, transmission lines, and electrical loads connected at circuit nodes called buses. A simplified three-bus system is depicted in Figure 3. The system is assumed to be a balanced three-phase system in sinusoidal steady-state operation. The *power flow problem* is defined to be the calculation of the voltage magnitude and phase angle at each bus in the system [13]. From these voltages and angles, combined with the known transmission line admittances, the active and reactive power flowing into the network from each bus can be calculated.

The description of the power flow problem requires the introduction of the following terminology and notation. Four electrical quantities are associated with each bus. At Bus *i*, V_i is the magnitude of the bus voltage, θ_i is the voltage phase angle, and P_i and Q_i are the net active and reactive powers entering the network from the bus, respectively. For the standard IEEE 30-bus system used in our simulations [16], the voltage magnitude and the phase angle vectors are denoted by

$$V = (V_1, V_2, \ldots, V_{30}), \qquad \theta = (\theta_1, \theta_2, \ldots, \theta_{30}) \tag{1}$$

The explicit dependence of the active and reactive powers on the voltages, angles, and admittances in the system is given by the *power flow equations*

$$\begin{aligned} P_i(V,\theta) &= V_i \sum_{k=1}^{N_b} V_j \left[G_{ik}\cos(\theta_{ik}) + B_{ik}\sin(\theta_{ik}) \right] \\ Q_i(V,\theta) &= V_i \sum_{k=1}^{N_b} V_j \left[G_{ik}\sin(\theta_{ik}) - B_{ik}\cos(\theta_{ik}) \right] \end{aligned} \tag{2}$$

where i) for $i \neq k$, $G_{ik} + jB_{ik}$ is the negative of the admittance connected between Bus *i* and Bus *j*; ii) for $i = k$, $G_{ii} + jB_{ii}$ is the sum of all admittances connected to Bus *i*; and iii) N_b is the number of buses in the system. At each bus there are four unknown quantities: V_i, θ_i, P_i, and Q_i. The system of nonlinear equations in (2) has $4N_b$ unknowns and only $2N_b$ equations, and therefore it is underdetermined.

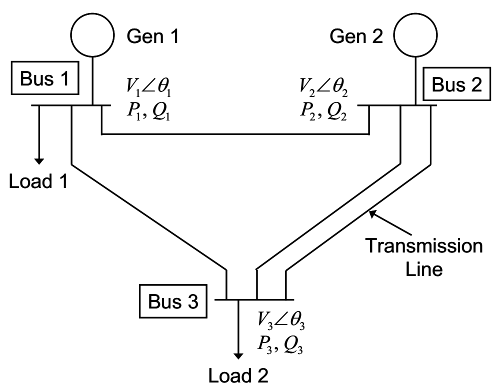

Figure 3: A 3-Bus Power System

The general practice in the analysis and design of power systems is to specify two of the four unknown quantities at each bus so that the system in (2) can be solved. Depending on which quantities are specified, the buses are divided into three categories: slack buses, *PQ* buses, and *PV* buses. These categories are described next.

For the sake of the mathematical notation involved, it is assumed that every bus in the system has both a generator and a load connected to it. Under this assumption, four additional quantities are defined for Bus *i*: the active and reactive powers supplied by the generator, P_{Gi} and Q_{Gi}, and the active and reactive powers absorbed by the load, P_{Di} and Q_{Di}. If a particular bus does not have a generator or a load, then we set the corresponding quantities to zero. Using this notation, the *power balance equations* are defined at Bus *i* as

$$\begin{aligned} P_i &= P_{Gi} - P_{Di} \\ Q_i &= Q_{Gi} - Q_{Di} \end{aligned} \tag{3}$$

That is, the power injected into the network at Bus *i* is equal to the power supplied by the generator minus the power absorbed by the load.

Every power system has a reference bus called a *slack bus*. The slack bus has a generator connected to it and is usually denoted as Bus 1. The voltage magnitude V_1 and angle θ_1 are specified with $\theta_1 = 0$ being specified as the reference angle for all voltages and currents in the system. The active and reactive powers of the generator, P_{G1} and Q_{G1}, are not specified, so P_1 and Q_1 are not specified but are left as "free" variables in order to guarantee that the system of equations in (2) has a solution.

At every bus in the system to which a load is connected, the powers, P_{Di} and Q_{Di}, absorbed by the load are specified in the design requirements. Some of the buses will have a generator connected to them, and the active power P_{Gi} supplied by the generator is specified except for the generator at the slack bus. And some of these generators may also have the reactive power Q_{Gi} specified by design because of the type of power plant or other conditions requiring both the active and reactive powers to be maintained at a certain level. So in these cases, both P_{Gi} and Q_{Gi} are specified. Buses for which all four powers, P_{Gi}, Q_{Gi}, P_{Di}, and Q_{Di}, are specified are called *PQ* buses.

At the remaining buses to which a generator is connected, only the active power of the generator P_{Gi} is specified; the reactive power Q_{Gi} is left unspecified. The voltage at these buses is controlled to specified values by adjusting the generator's output. These are called *PV* buses.

We can combine the generator and load power specifications into a single quantity called the *specified power* at Bus *i*. It is defined as the power generated at the bus minus the load connected to the bus:

$$\begin{aligned} P_i^{\text{sp}} &= P_{Gi} - P_{Di} \\ Q_i^{\text{sp}} &= Q_{Gi} - Q_{Di} \end{aligned} \tag{4}$$

The first equation in (4) is well defined for all buses except the slack bus, namely all *PQ* and *PV* buses. The second is well defined for all buses for which Q_{Gi} is specified, namely the *PQ* buses.

In order to solve the power flow equations in (2) iteratively, an initial value for the voltage-angle vector (V,θ) is generated to start the algorithm. The values of P_i and Q_i can then be calculated for all buses in the system using (2). These quantities represent the net *estimated* power entering the network from Bus *i* and are denoted by P_i^{est} and Q_i^{est}. The *power mismatch* is then defined as

$$\begin{aligned} \Delta P_i &= \left| P_i^{\text{est}} - P_i^{\text{sp}} \right| = \left| P_i - (P_{Gi} - P_{Di}) \right| \\ \Delta Q_i &= \left| Q_i^{\text{est}} - Q_i^{\text{sp}} \right| = \left| Q_i - (Q_{Gi} - Q_{Di}) \right| \end{aligned} \tag{5}$$

Mismatches are nonzero when the power estimated from a given voltage-angle vector (V,θ) differs from the specified power at the bus. The goal of the power flow problem is to find a vector (V,θ) such that the *power balance equation* at each bus is satisfied:

$$\begin{aligned} \Delta P_i(V,\theta) &= 0 \\ \Delta Q_i(V,\theta) &= 0 \end{aligned} \tag{6}$$

For *PV* buses, Q_{Gi} is not specified, and for slack buses, neither P_{Gi} nor Q_{Gi} are specified. In these cases, a mismatch for the corresponding quantity is not defined, and there is no requirement that the power balance equations associated with these buses be satisfied when solving the power flow problem [15].

Since the voltage magnitude is specified at *PV* buses, the given voltage magnitude vector *V* can be used to define the *voltage mismatch* [14] as

$$\Delta V_i = \left| V_i - V_i^{\mathrm{sp}} \right| \tag{7}$$

where V_i is the *i*-th component of the vector *V* in (1). As in (6), a *voltage balance equation* can be then defined:

$$\Delta V_i = 0 \tag{8}$$

Note that V_i is not estimated from the power flow equations (2) as were P_i^{est} and Q_i^{est}, but is obtained from the estimate for *V* in the algorithm when solving the power flow problem.

Typically numerical methods such as the Newton-Raphson method are used to solve the nonlinear system in (2) [8][13][15]. As mentioned above, an initial guess for the voltage-angle vector (V,θ) is required to start the algorithm. This initial solution vector is used in (5) and (7) to generate a mismatch or estimation error. In numerical approaches, a gradient method is then applied to the mismatch to generate a new estimate for (V,θ), and the process is iterated.

In GA or BBO algorithms, an initial population of individuals (chromosomes for the GA and islands for the BBO) is generated by assigning a random vector (V,θ) to every individual in the population. The components of the vector are real-valued voltages and phase angles constrained to be within some prescribed limits based on considerations from the power system specifications. The components of these vectors comprise the genes in the GA and the features (*SIV*s) in the BBO algorithm. The mismatches can then be calculated for each individual and a cost assigned from a weighted sum of the squares of the mismatches:

$$c(V,\theta) = \sum_{i \in PV \cup PQ} \Delta P_i^2 + \lambda_Q \sum_{i \in PQ} \Delta Q_i^2 + \lambda_V \sum_{i \in PV} \Delta V_i^2 \tag{9}$$

Here *PV* and *PQ* are index sets for the *PV* and *PQ* buses, respectively. This cost is used as the objective function in both the GA and the BBO algorithms. The goal of each algorithm is to find a global minimum for the cost $c(V,\theta)$ in as few generations as possible. For the power flow problem, the global minimum is known to be zero from (6) and (8).

The GA and BBO algorithms require a fitness function for roulette wheel calculations. Since the theoretical minimum of $c(V,\theta)$ is zero, the fitness function was chosen to be

$$f(V,\theta) = \frac{100}{c(V,\theta) + 0.005} \tag{10}$$

to avoid extremely large fitness values or divide-by-zero errors.

IV. Simulation Results

The GA and BBO algorithms were applied to the power flow problem using the MATLAB® programming environment. The standard IEEE 30-bus system, based on the data from a real power system in Virginia in the 1960s, was used as a test case for the study. A diagram and the electrical specifications of the system are available at the Power Systems Test Case Archives website maintained by the University of Washington [16]. The data file containing the electrical specifications is formatted in the IEEE Common Format for the exchange of solved load flow data [17].

The GA population size was chosen as 100. Each chromosome in the population has 60 genes made up of the 30 voltage magnitudes and 30 voltage phase angles in the vector

$$X = (V_1, V_2, \ldots, V_{30}, \theta_1, \theta_2, \ldots, \theta_{30}) \tag{11}$$

The slack bus voltage V_1 was fixed at 1 p.u. (*per unit*, a normalized unit of voltage measurement) and its angle θ_1 was fixed at zero degrees. The weights assigned to the cost function were $\lambda_Q = 1$ and $\lambda_V = 100$.

For the reproduction operation, a one-point crossover scheme was used. Each pair of parent chromosomes can potentially generate two offspring depending on the crossover probability. Fifty parents were randomly chosen by roulette wheel selection based on the fitness expression in (10). Crossover was set to occur with a probability of 0.70 at a randomly determined crossover point in the chromosome.

After the crossover step, a non-uniform mutation operation is applied to each chromosome in the population. The probability of selecting a given chromosome for mutation is 0.05. If a chromosome is chosen for mutation, then one of its genes, say $g \in X$, in (11) is changed to a new value g' according to the formula

$$\begin{aligned} & g' = a' + r(b' - a') \\ \text{where} \quad & a' = g - f(G)(g - a) \\ & b' = g + f(G)(b - g) \\ \text{and} \quad & f(G) = (1 - G / G\max)^2 \end{aligned} \tag{12}$$

and *r* is a uniformly distributed random number in [0, 1], *G* is the current generation number, $G_{\max}$ is the maximum number of generations, and *a* and *b* are the lower and upper limits associated with the gene g_i. Duplicate chromosomes were removed at each generation step and replaced with random mutations. Lastly, an elitism operation was applied in order to preserve the six fittest individuals from each generation.

The BBO algorithm is similar to the GA except that in BBO, the reproduction scheme of the GA is replaced by the migration scheme described in Figure 2. The same mutation, duplication removal, and elitism schemes are used in both algorithms. In BBO nomenclature, chromosomes are referred to as *islands* and genes as *features* or *SIV*s.

Many of the decisions and numerical values within both algorithms depend on stochastically generated quantities, so 700 Monte Carlo simulations were run to improve the reliability of the conclusions about performance. Only 100 generations were simulated in order to reduce simulation time, which was 462 minutes on a 2.3 GHz processor. This number of generations is extremely small for this type of problem. For example, a GA very similar to the one in this study was applied to the IEEE 14-bus standard system in [1] and required 10,000 generations to achieve a minimum cost very close to zero. But since the goal of this study is a comparison of the relative performance of the GA and the BBO algorithms, such time-consuming simulations were not required.

The overall performance of the two algorithms is shown in Figure 4. The plots were generated as follows. A cost is calculated for each individual in a population. Each population is associated with a generation, and each generation is associated with a Monte Carlo trial. Thus, a cost is generated for each individual in each generation in each trial; or, put another way, the cost is a function of the individual index, the generation index, and the trial index:

$$cost = cost(ind,\ gen,\ trial) \tag{13}$$

If we take the average of this cost, first over all individuals in a particular generation, and then over all Monte Carlo trials containing that generation, we obtain the average cost as a function of the number of generations:

$$AveCost(gen) = \operatorname*{ave}_{trials}\left(\operatorname*{ave}_{ind}\left[cost(ind,\ gen,\ trial)\right]\right) \tag{14}$$

This function is plotted in Figure 4 for the GA and BBO.

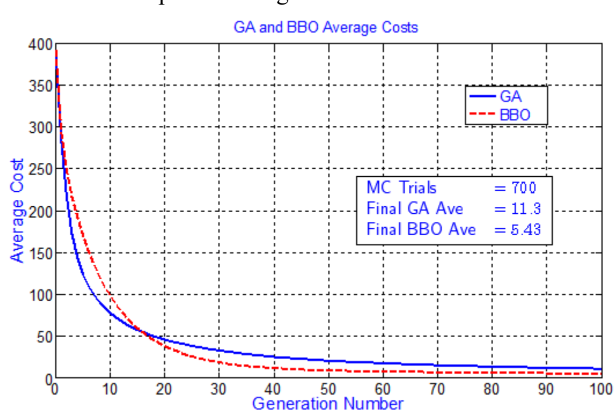

Figure 4: Average Cost of 700 Monte Carlo Trials

The plot shows that for about the first 16 generations, the GA performs better than the BBO, but thereafter the BBO performs better. This improved performance is consistently observed in simulations with 1000, 5000, and 10000 generations, although these simulations were not subjected to Monte Carlo trials because of prohibitive computation times. Nevertheless, in numerous non-Monte Carlo simulations with a large number of generations, the BBO always found a smaller minimum cost and had a smaller average cost, even when the crossover or mutation rates were changed.

Some further Monte Carlo results are shown in Figure 5 and Figure 6. Figure 5 shows the minimum cost achieved over all individuals, over all generations, and over all Monte Carlo simulations. Figure 6 shows the minimum average cost over all generations and over all trials. Both figures indicate that the BBO algorithm is able to find a better solution than the GA with respect to these statistical measures.

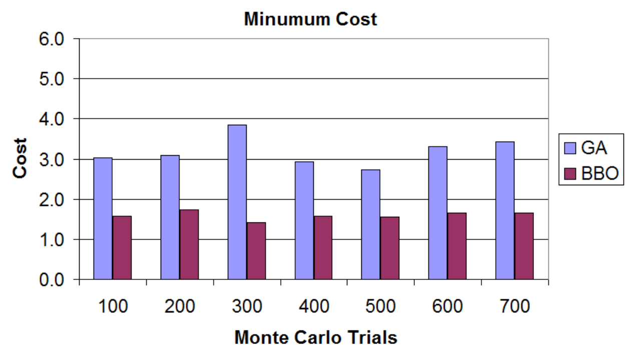

Figure 5: Minimum Cost from Monte Carlo Trials

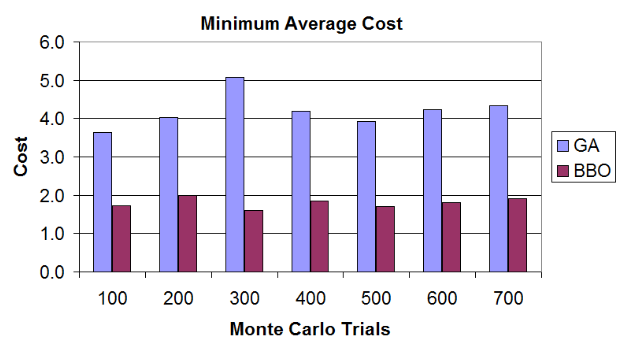

Figure 6: Minimum Average Cost from Monte Carlo Trials

V. Conclusion

The objective of this study was to compare the performance of a novel biogeography-based optimization algorithm to that of a traditional genetic algorithm in a practical application. Both algorithms were applied to a standard power flow problem in power systems analysis. Initial Monte Carlo testing indicates that the BBO consistently performs better than the GA with respect to certain basic statistical measures. The mathematical basis for this improved performance is discussed in [18].

For this study, the algorithms were applied to the relatively simple IEEE 30-bus test case so that the code could be tested for accuracy more efficiently and initial results generated in a reasonable amount time. Although traditional numerical methods for solving such problems have been successful, the difficulty in solving such problems increases rapidly as the number of buses increases or the system is stressed under heavy loads approaching the maximum operating point of the system. In the former situation, the Newton-Raphson method often converges to a local minimum which is very dependent on the starting solution. In the latter situation, the method is more likely not to converge at all because the Jacobian tends to be singular [14]. It is in these situations that population-based search algorithms have shown the most potential, since they are designed to search for global optima, and no derivatives or matrix inversions are involved. The contribution of this work, then, is that we have demonstrated the improved performance of the BBO algorithm over the genetic algorithm in a special case. We are now in the position to apply the algorithms to the much more difficult and important optimal power flow problem for a large number of buses where traditional approaches are less likely to succeed. This will also allow us to determine whether the BBO algorithm continues to perform better than the GA in these more difficult problems.

Since evolutionary algorithms are probabilistic, the first undertaking in future work would be to optimize the code for speed so that more Monte Carlo simulations could be applied to both of the algorithms that we tested. This would allow the student-*t* test [19] to be applied to the results from the two algorithms to determine the statistical significance of the differences between the results.

One of the strengths of population-based search algorithms is their ability to determine global optima with probabilities that can be determined in many situations. It would be beneficial to design experiments to measure the relative efficacy of the GA and BBO algorithms in finding a global minimum for the optimal power flow problem.

Many variations and enhancements are possible with both the GA and the BBO algorithms, and some of these should be implemented so that comparisons can be made to determine whether the BBO algorithm maintains its superior performance with these adaptations. One such variation fuses opposition-based learning [20] with biogeography-based optimization (OBBO) [21] and has shown promising initial results for the power flow problem and for many benchmark problems that are used to test evolutionary algorithms.

Finally, different approaches to the migration operation could be tested. Instead of the linear approximations that we used for the migration rate curves λ and μ, more complicated curves can be constructed. Also, the migration strategy that we used can be described as an immigration-based strategy, but others are possible, such as the emigration-based, single immigration, and single emigration strategies discussed in [22].

References

[1] L. Lai and N. Sinha, "Genetic algorithms for solving optimal power flow problems," Chapter 17, Modern Heuristic Optimization Techniques, pp. 471 – 500, IEEE Press, 2008.

[2] A. Wallace, The Geographical Distribution of Animals, first published in 1876, Adamant Media Corporation, 2006.

[3] C. Darwin, The Origin of Species, first published in 1859, Gramercy, 1995.

[4] E. Munroe, "The geographical distribution of butterflies in the West Indies," PhD Dissertation, Cornell University, Ithaca, New York, 1948.

[5] R. MacArthur and E. Wilson, The Theory of Biogeography, Princeton University Press, 1967.

[6] D. Simon, "Biogeography-based optimization," IEEE Transactions on Evolutionary Computation Vol. 12, No. 6, pp. 702-713, December 2008.

[7] J. Carpentier, "Contribution to the study of economic dispatch," Bulletin of the French Society of Electricians, Vol. 3, No. 8, pp. 431-437, 1962.

[8] A. Wood and B. Wollenberg, Power Generation Operation and Control, Wiley-Interscience, NJ, 1996.

[9] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, MA, 1989.

[10] J. Yuryevich and K. Wong, "Evolutionary programming based optimal power flow algorithm," IEEE Transactions on Power Systems, Vol. 14, No. 4, pp. 1245-1250, November 1999.

[11] S. Agrawal, B. Panigrahi, and M. Tiwari, "Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch," IEEE Transactions on Evolutionary Computation, Vol. 12, No. 5, pp. 529-541, October 2008.

[12] K. Lenin and M. Mohan, "Ant colony search algorithm for optimal reactive power optimization," Serbian Journal of Electrical Engineering Vol. 3, No. 1, pp. 77 – 88, June 2006.

[13] J. Glover and S. Sarma, Power System Analysis and Design, Fourth Edition, CL-Engineering, 2007.

[14] K. Wong, A. Li, and M. Law, "Development of constrained genetic algorithm load flow method," IEE Proceedings – Generation, Transmission, and Distribution, Vol. 144, No. 2, pp. 91 – 99, Mar 1997.

[15] J. Grainger and W. Stevenson, Power System Analysis, McGraw-Hill Science, NY, 1994.

[16] R. Christie, Power Systems Test Case Archive, College of Engineering, University of Washington, WA (Dec. 2008). [Online]. Available: www.ee.washington.edu/research/pstca

[17] H. Pierce, Jr., "Common format for exchange of solved load flow data," IEEE Transactions on Power Apparatus and Systems, Vol. PAS-92, No. 6, pp. 1916 – 1925, Nov/Dec 1973.

[18] D. Simon, M. Ergezer, and D.Du, "Markov analysis of biogeography-based optimization algorithms," unpublished (Mar. 2009). [Online]. Available: http://academic.csuohio.edu/simond/bbo/markov

[19] J. Kennedy and R. Eberhart, Swarm Intelligence, Academic Press, CA, 2001.

[20] S. Rahnamayan, H. Tizhoosh, M. Salama, "Opposition-based differential evolution," IEEE Transactions on Evolutionary Computation Vol. 12, No. 1, pp. 64-79, Feb. 2008.

[21] M. Ergezer, D. Simon, and D. Du, "Opposition biogeography-based optimization, unpublished.

[22] D. Simon, "A probabilistic analysis of a simplified biogeography-based optimization algorithm," unpublished (Mar. 2009). [Online]. Available: http://academic.csuohio.edu/simond/bbo/simplified