

Opposition-Based Elitist Real Genetic Algorithm for Optimal Power Flow

Saleh Almasabi
Electrical and Computer Engineering
Michigan State University
East Lansing, MI 48824
almasabi@msu.edu

Fares T. Alharbi
Electrical and Computer Engineering
Michigan State University
East Lansing, MI 48824
alharb15@msu.edu

Joydeep Mitra
Electrical and Computer Engineering
Michigan State University
East Lansing, MI 48824
mitraj@msu.edu

Abstract—Optimal power flow (OPF) algorithms are widely used in the operation of modern power systems, and numerous variations and enhancements have been developed over the last four decades. Yet, with decreasing time intervals and increasing complexities introduced by market policies and stochastic inputs, the need for further improvements in speed and performance of OPF algorithms persists. This paper proposes using elitist real genetic algorithm (ERGA) and opposition-based elitist real genetic algorithm (OB-ERGA) to solve the OPF problem. Also, inverse transformation and exponential transformation are implemented to investigate the convergence performance of the proposed methods. The combination of the OB-ERGA, ERGA and the fitness functions is tested on the IEEE 30-bus system to determine the effectiveness of the proposed approaches. The results are presented and compared with the existing evolutionary algorithms in the literature.

Index Terms—Evolutionary algorithms, Genetic algorithms (GAs), opposition-based genetic algorithms (OB-GAs), Optimal power flow (OPF).

I. INTRODUCTION

In power systems, optimal power flow (OPF) plays a crucial role in minimizing generation costs, maximizing revenues, and improving system efficiency and performance. Since the OPF problem was introduced in the sixties [1], researchers have developed several approaches to overcome the nonlinearity, non-convexities, and market complexities. One of the approaches is to linearize the problem and use the DC power flow model for the OPF. When the DC model is used, linear programming (LP) can be used to solve the OPF as in [2], [3]. However, the DC model ignores the reactive power in the system and assumes constant voltages throughout the network. Although the DC model results lack accuracy, it is widely used due to its simplicity and low computation cost.

On the other hand, the AC power flow model, many classical methods have implemented AC-based OPF by utilizing the quadratic programming approach [4], the sequential quadratic programming approach [5], and Newton's approach [6]. Although these methods have solved the OPF problem, they are dependent on the initial guess of the solution which may lead to a local optimal point in the search space and miss the global optimal point [7]. In addition, it is difficult to expand those methods to include multi-objective OPF such as environmental/economic dispatch.

Evolutionary algorithms however, can be expanded easily to include multi-objective optimization. Furthermore, those algorithms are more robust and can easily handle constraints and discontinuities. Unlike classical methods, evolutionary methods start with a population of points which evolve to better points in terms of their fitness values at each successive generation by mimicking the natural evolutionary process [8]. In [9], an improved genetic algorithm was used to solve the optimal power flow. The work presented in [10] has applied a refined GA, and [11] has applied an enhanced GA. The author in [12] used a fuzzy GA and a fuzzy PSO to solve the OPF. The above studies have used binary coded GAs in which the population points are converted from real space to binary space where the evolutionary process takes place.

Genetic algorithms (GA) usually imitate the evolution process by creating offsprings using the fittest parents and then repeat the process. Using the fittest among all the population instead of choosing the offsprings, is called the elitist GA. One of the popular algorithms that uses elitist selection is the non-dominated sorting genetic algorithm (NSGA II) [13]. The NSGA II is used for multi-objective optimization. Building on the work in [13], this paper proposes using elitist real genetic algorithm (ERGA) and opposition based elitist real genetic algorithm (OB-ERGA) to solve the OPF using two different fitness functions. Unlike NSGA II the proposed algorithms does not maintain the population diversity, instead chooses the fittest regardless of the diversity. The proposed methods provide fast convergence and very good results. The proposed methods are tested on the IEEE 30-bus system in Fig. 3 and compared with other OPF approaches in the literature that use evolutionary methods.

The rest of the paper is organized as follows. Section II, describes the problem statement, and discusses the fitness function. In section III, the optimization algorithms are presented. The case study and conclusion are presented in sections IV and V respectively.

II. PROBLEM STATEMENT

This section presents the AC power flow model with constraints. It also discusses the implantation of the inverse transformation and the decaying exponential for the fitness

functions. Both fitness functions are used to determine which is more suitable in terms of accuracy and convergence rate.

A. Objective Function and Constraints

The OPF is a nonlinear and non-convex problem [11]. The goal is to find a solution (global minimum), such that no other point in the search space can compete with it in minimizing the cost function in (1), and satisfying the inequality and equality constraints in (2)-(7). Table I summarizes some of the characteristics of the IEEE 30-bus system.

$$\min \sum_{j=1}^{N_g} (a_j P_j^2 + b_j P_j + c_j) \quad (1)$$

subject to:

$$P_{g,i}^{min} \leq P_{g,i} \leq P_{g,i}^{max} \quad (2)$$

$$Q_{g,i}^{min} \leq Q_{g,i} \leq Q_{g,i}^{max} \quad (3)$$

$$V_i^{min} \leq V_i \leq V_i^{max} \quad (4)$$

$$-\pi \leq \delta_i \leq \pi \quad (5)$$

$$P_i(\delta, V) + P_{d,i} - P_{g,i} = 0 \quad (6)$$

$$Q_i(\delta, V) + P_{(L,i)} - P_{(g,i)} = 0 \quad (7)$$

where

$$P_i(\delta, V) = |V_i| \sum_{j=1}^N |Y_{ij}| |V_j| \cos(\delta_i - \delta_j - \theta_{ij}) \quad (8)$$

$$Q_i(\delta, V) = |V_i| \sum_{j=1}^N |Y_{ij}| |V_j| \sin(\delta_i - \delta_j - \theta_{ij}) \quad (9)$$

Where, P_i and Q_i are the nonlinear equation of the active and reactive power injected into bus i , N_g is the number of generators; N is the total number of buses in the system, $P_{g,i}$ is the real power generated at the generation unit i , and $Q_{g,i}$ is the reactive power generated at the generation unit i . V_i is the voltage magnitude at bus i and δ_i is the voltage angle at bus i ; V_i^{min} and V_i^{max} are the minimum and maximum voltage magnitudes allowed at bus i respectively. $P_{g,i}^{min}$ and $P_{g,i}^{max}$ are the minimum and maximum real power that could be generated at generation bus i respectively; $Q_{g,i}^{min}$ and $Q_{g,i}^{max}$ are the minimum and maximum reactive power that could be generated at generation bus i respectively.

B. Fitness Function

In order to use evolutionary algorithms, each candidate solution needs to be evaluated in terms of the objective function and constraint violations to determine the feasibility of the solution. If a candidate solution violates one constraint or more, penalty term(s) will be added to the cost associated with this candidate. The cost and the constraints violations are companied into one function which is called the fitness function. In this paper, the IEEE 30-Bus system cost coefficients, as shown in Table II, are used to determine the total cost in (1).

The evolutionary algorithms are based on maximizing the fitness of a function. However, the OPF objective is to minimize the dispatched power cost. Therefore, the objective function in (1) is used with the inverse transformation as presented in (10). The performance for the inverse transformation will be compared with the decaying exponential transformation in (11).

$$F_1(x) = \frac{k_1}{(1 + TotalCost + TotalPenalties)} \quad (10)$$

$$F_2(x) = e^{-(TotalCost + TotalPenalties)/K_2} \quad (11)$$

where, k_1 and k_2 are constants.

III. OPTIMIZATION ALGORITHMS

In this section, the ERGA and OB-ERGA are introduced. The main advantage for the opposition based algorithm is to speed up the convergence rate. However, a faster convergence rate does not guarantee better solutions. Both algorithms are used to check the accuracy and convergence rate.

A. Elites Real Genetic Algorithm (ERGA)

In the binary coded GA the selection, crossover and mutation operations are performed in the binary space. The accuracy of the results is highly dependent on the number of bits in the binary space (string) representing each point in the real space. The binary coded GA needs larger strings (larger number of bits) to achieve better accuracy. The accuracy of each variable in the string is shown in (12) [8].

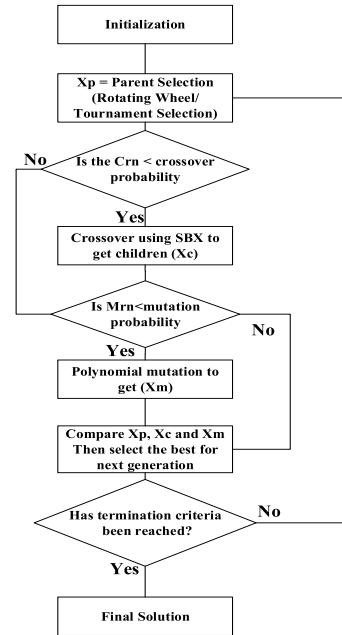


Fig. 1. Flow chart algorithm for the purposed ERGA

$$\epsilon = (x_i^U - x_i^L)/(2^{l_i} - 1) \quad (12)$$

Where x_i^U and x_i^L are the upper and lower bounds of the variable x_i . l is the number of bits occupied by x_i , and ϵ is the accuracy.

Applying genetic algorithm directly on the real space would, obviously solve the accuracy problem. In the real coded GA, the first operation (selection) could be done directly on the real space without the need of converting to binary space; however, the crossover and the mutation operations could not be applied directly in the real space. To overcome this problem, a simulated binary crossover operator introduced in [14] and [8] is used with crossover probability $P_c = 0.7$. The parents are crossed variable wise as in [14] and [8]. For the mutation in the real space, a polynomial mutation operator proposed in [15] is used with a mutation probability of $P_m = 0.2$ and $\eta = 20$. The mutation is performed variable wise as in [15] and [8]. Compared to the traditional real genetic algorithm, *ERGA* compares the current population with the offspring that are created by the crossover operation as well as the offspring created by the mutation operation. The *ERGA* is as follows: After initializing the first random population, the selection operator uses roulette wheel selection to randomly select the parents for the mating pool. Then, with a crossover probability P_c , the genes of two parents are exchanged, using SBX crossover as in (13). The mating process generates the offsprings population X_c . These offsprings will be compared with the whole population later on. After that, a small mutated population will be generated from the parents based on the mutation probability P_m . The mutated population X_m is generated using a polynomial mutation process as in (14) and (15); more details the could be found in [8]. After obtaining the mutated population X_m , the whole population is compared to select the elites. The output of each iteration are the best among the parents (X_p) and all the offspring that are generated from the current parents (X_c and X_m). The flow chart in Fig. 1 summarizes the *ERGA*. Note that Crn and Mrn are random numbers.

$$X_c = \begin{cases} \frac{1}{2} \times [((1 + \beta) \times P_i) + ((1 - \beta) \times P_j)] \\ \frac{1}{2} \times [((1 - \beta) \times P_i) + ((1 + \beta) \times P_j)] \end{cases} \quad (13)$$

where

β = distribution index
 P_i, P_j = are the chosen pair of parents
 X_c = the offsprings from the parents

$$dr = \begin{cases} [((2 \times rand)^{\frac{1}{1+\eta}})^{-1}], \text{ if } rand \leq 0.5 \\ [1 - ((2 \times rand)^{\frac{1}{1+\eta}})], \text{ if } rand > 0.5 \end{cases} \quad (14)$$

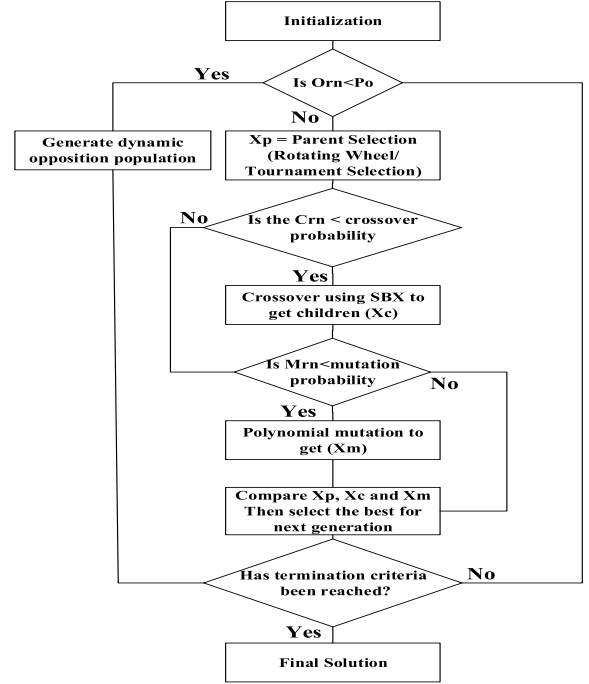


Fig. 2. Flow chart algorithm for the purposed OB-ERGA

$$X_{m_{i,j}} = \begin{cases} [(P_{i,j}) + dr \times ((P_{i,j}) - Y_{min,j})], \text{ if } rand \leq 0.5 \\ [(P_{i,j}) + dr \times (Y_{max,j} - (P_{i,j}))], \text{ if } rand > 0.5 \end{cases} \quad (15)$$

where

$X_{m_{i,j}}$ = mutated candidate i for variable j
 $rand$ = random number
 dr = mutation variable constant
 P_i = is the random chosen parent based on the mutation probability
 $Y_{min,j}$ = is the global minimum of variable j
 $Y_{max,j}$ = is the global maximum of variable j

B. Opposition Based Elites Real Genetic Algorithm (OB-ERGA)

The opposition search technique has helped to speed up convergence to the optimal point as proposed in [16]. It is more likely that the opposite of the initial population (since it is random) will contain some points which are better than some of the points in the initial random guess. The opposite of a point in a search space is given in (16) [16]. The updated initial population will be the best points among initial guess and its opposite.

$$X_i^{Opp} = X_i^{Min} + X_i^{Max} - X_i \quad (16)$$

In the opposition based genetic algorithm, the opposition will be used along with each iteration with a probability of

P_o . At each iteration, a random number $j \in [0, 1]$ is generated. If $j > P_o$, no opposition takes place; the population will continue evolving as in *ERGA* algorithm. However, if $j \leq P_o$, a dynamic opposite population (17) would be generated [16]. Then, the best among the current population and their opposite will be selected as the output of the current iteration, and no evolving operation would take place. The opposition probability in this case is $P_o = 0.4$.

$$X_i^{Opp} = a_{i,n}^{Min} + a_{i,n}^{Max} - X_i \quad (17)$$

As before, $a_{i,n}^{Min}$ and $a_{i,n}^{Max}$ are the minimum and maximum of X_i at iteration n . This dynamic opposition prevents the algorithm from jumping out of a good region that was found from the previous iterations when the opposition takes place. The flow chart in Fig. 2 summarizes the OB-ERGA algorithm; note that Orn , Crn , and Mrn are random numbers.

IV. CASE STUDY

The two proposed approaches ERGA and OB-ERGA are tested to solve the OPF for the IEEE 30-bus system. The testing involves a comparison of alternating the fitness functions (10) and (11) in the two proposed algorithms to test their effect on the convergence performance. The same initial random guess has been used in the all the cases, but note that the results show progress from iteration 1. The population size was chosen to be 10 with a maximum of 200 generations.

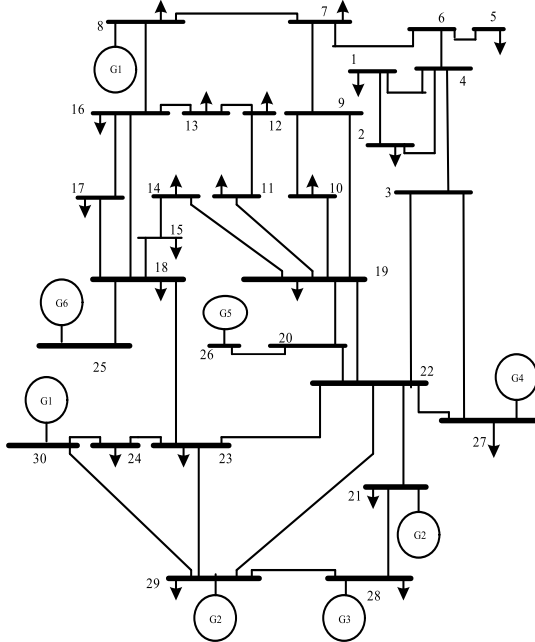


Fig. 3. IEEE 30-bus system

The results in Fig. 4 and Fig. 5 shows that using the decaying exponential function (10), gives a better convergence rate than the inverse transformation (11) for both ERGA and

TABLE I
IEEE 30-BUS SYSTEM SUMMARY

Number of buses	30
Number of branches	41
Total number of buses (Number of variables)	06
Inequality voltage constraints	30
Inequality angle constraints	30
Real power generation inequality constraints	06
Reactive generation inequality constraints	06
Real power balance equality constraints	30
Reactive power balance equality constraints	30
Equality constraint of total generated power balance	01
Total number of equality constraints	66
Total number of inequality constraints	61
Total number of all constraints	127

TABLE II
COST COEFFICIENTS OF THE GENERATORS

Bus	a_i	b_i	c_i
1	0.00375	2	0
2	0.0175	1.75	0
5	0.0625	1	0
8	0.0083	3.25	0
11	0.025	3	0
13	0.025	3	0

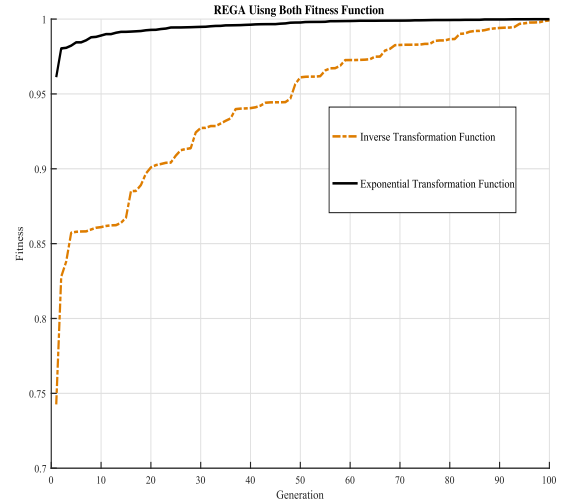


Fig. 4. Comparisons between the inverse and the exponential transformation functions using REGA

TABLE III
IEEE 30-BUS OPF USING OB-ERGA AND ERGA

	MATPOWER [17]	ERGA with F_1	ERGA with F_2	OB-ERGA with F_1	OB-ERGA with F_2
P_1	176.21	179.07	176.64	176.76	177.03
P_2	48.86	48.09	49.40	48.36	48.93
P_5	21.54	21.49	21.56	21.47	21.39
P_8	22.32	18.99	21.85	21.55	22.32
P_{11}	12.29	12.24	11.73	11.69	12.2
P_{13}	12.00	12.73	12.29	12.21	12.01
Total power	293.22	292.61	292.42	292.47	292.48
Total cost	803.62	800.79	800.70	800.69	800.68

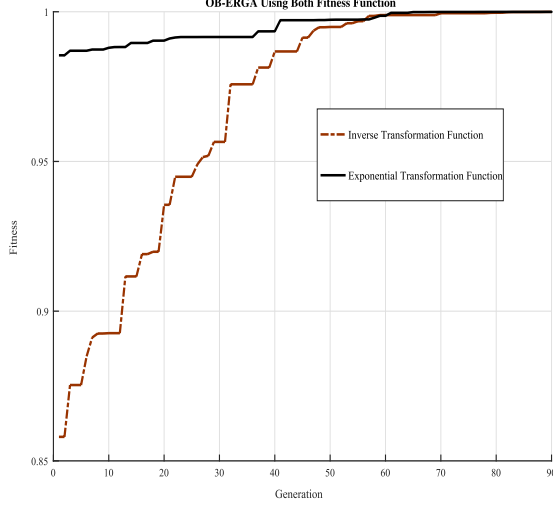


Fig. 5. Comparison between the inverse and the exponential transformation functions using OB-ERGA

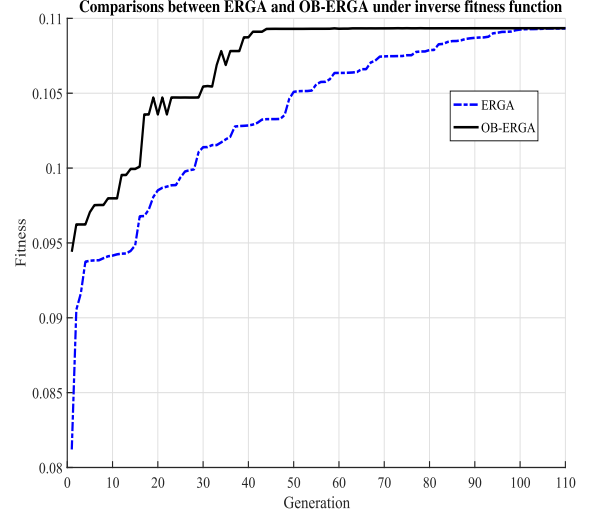


Fig. 6. Comparisons between ERGA and OB-ERGA under inverse fitness function (F_1)

OB-ERGA. As for comparing ERGA and OB-ERGA, Fig. 6 and Fig. 7 show that the OB-ERGA is faster than the ERGA regardless of the fitness function. This is due to the fact that the opposition based GA continues to widely explore the search space with an opposition probability $P_o = 0.4$, which would make a faster convergence. The optimal results for the OB-ERGA and the ERGA are shown in Table III. The exponential decaying transformation has helped for faster convergence in both algorithms. The convergence rate of OB-ERGA is much better than the ERGA under both fitness functions. In Table IV, the Optimal results for the OB-ERGA, are compared with existing evolutionary algorithm in the literature. Compared to the literature not only OB-ERGA, but also ERGA have shown better convergence rates. To illustrate [12] uses a population size of 30 and 100 for the GA and GA-Fuzzy with a 300 maximum iterations which converges around the 100th generation. The purposed OB-ERGA and ERGA converge around the 65th and the 100th generation respectively with only a third of the population size in [12].

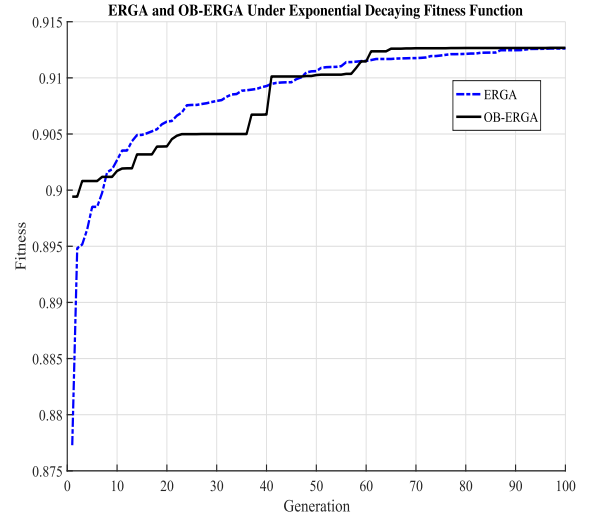


Fig. 7. Comparisons between ERGA and OB-ERGA under exponential decaying fitness function (F_2)

TABLE IV
IEEE 30-BUS OPF WITH OTHER EVOLUTIONARY ALGORITHMS IN THE LITERATURE

	Tab search [18]	GA [11]	GA [19]	GA-Fuzzy [19]	GA [12]	GA-Fuzzy [12]	PSO [12]	PSO-Fuzzy [12]	OB-ERGA (F_2)
P_1	176.04	176.20	175.64	174.97	175.16	178.17	178.60	172.96	177.03
P_2	48.76	48.75	48.94	50.35	49.03	45.16	45.77	49.34	49.93
P_5	21.56	21.44	21.18	21.45	19.52	20.65	20.36	20.26	21.39
P_8	22.05	21.95	22.65	21.18	19.68	21.29	21.32	23.77	20.90
P_{11}	12.44	12.42	12.43	12.67	17.10	15.16	13.59	12.63	12.21
P_{13}	12.00	12.02	12.00	12.11	12.00	12.00	12.84	13.19	12.01
Total power	292.76	292.84	292.84	292.72	292.49	292.43	292.48	292.15	292.48
Total cost	802.29	802.06	802.38	802.00	801.96	801.21	800.96	800.72	800.68

V. CONCLUSION

This paper presented two evolutionary algorithms, OB-ERGA and ERGA to solve the OPF problem while satisfying all the system constraints. The proposed algorithms can handle discrete constraints and stochastic processes. Moreover, by using the real space in the proposed algorithms, the accuracy and deteriorated solution issues associated with the binary space are avoided. Both the OB-ERGA and ERGA were tested on the modified IEEE 30-bus system and both proved at least as effective in performing the optimization as those reported in the literature. To improve and test the convergence speed, the inverse transformation and the decaying exponential transformation were coupled with both algorithms. It was shown that using OB-ERGA with the decaying exponential transformation results in the fastest convergence. In future work, the proposed methods will be expanded to include the multi-objective OPF and stochastic inputs.

REFERENCES

- [1] J. Carpentier, "Contribution to the economic dispatch problem," *Bulletin de la Societe Francoise des Electriciens*, vol. 3, no. 8, pp. 431–447, 1962.
- [2] C. Coffrin and P. Van Hentenryck, "A linear-programming approximation of ac power flows," *INFORMS J. Comput.*, vol. 26, no. 4, pp. 718–734, 2014.
- [3] B. Stott and J. L. Marinho, "Linear programming for power-system network security applications," *IEEE Trans. PowerApp.Syst.*, vol. PAS-98, pp. 837–848, May 1979.
- [4] J. Nanda, D. P. Kothari, and S. C. Srivastava, "New optimal power-dispatch algorithm using fletcher's quadratic programming method," *Proc. Inst. Elect. Eng. C*, vol. 136, pp. 153–161, May 1989.
- [5] I. M. Nejdawi, K. A. Clements, and P. W. Davis, "An efficient interior point method for sequential quadratic programming based optimal power flow," *IEEE Trans. Power Syst.*, vol. 15, pp. 1179–1183, Nov 2000.
- [6] D. I. Sun, B. Ashley, B. Brewer, A. Hughes, and W. F. Tinney, "Optimal power flow by newton approach," *IEEE Trans. Power App. Syst.*, vol. PAS-103, pp. 2864–2880, Oct 1984.
- [7] J. A. Momoh, M. E. El-Hawary, and R. Adapa, "A review of selected optimal power flow literature to 1993. ii. newton, linear programming and interior point methods," *IEEE Trans. Power Syst.*, vol. 14, pp. 105–111, Feb 1999.
- [8] K. Deb, *Optimization for engineering design: algorithms and examples*. Prentice-Hall of India, 1998.
- [9] L. Lai, J. Ma, R. Yokoyama, and M. Zhao, "Improved genetic algorithms for optimal power flow under both normal and contingent operation states," *Int. J. Elect. Power Energy Syst.*, vol. 19, pp. 287–292, 6 1997.
- [10] S. Paranjothi and K. Anburaja, "Optimal power flow using refined genetic algorithm," *Elect. Power Compon. Syst.*, vol. 30, no. 10, pp. 1055–1063, 2002.
- [11] A. G. Bakirtzis, P. N. Biskas, C. E. Zoumas, and V. Petridis, "Optimal power flow by enhanced genetic algorithm," *Power Syst. IEEE Trans.*, vol. 17, no. 2, pp. 229–236, 2002.
- [12] S. Kumar and D. Chaturvedi, "Optimal power flow solution using fuzzy evolutionary and swarm optimization," *Int. J. Elect. Power Energy Syst.*, vol. 47, pp. 416–423, 2013.
- [13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.
- [14] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Compl. Syst.*, vol. 9, no. 2, pp. 115–148, 1995.
- [15] K. Deb and D. Deb, "Analysing mutation schemes for real-parameter genetic algorithms," *Int. J. Artif. Intell. Soft. Comput.*, vol. 4, no. 1, pp. 1–28, 2014.
- [16] S. Rahnamayan, H. R. Tizhoosh, and M. Salama, "Opposition-based differential evolution algorithms," in *CEC 2006. IEEE Congress on*, pp. 2010–2017, IEEE, 2006.
- [17] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "Mat-power's extensible optimal power flow architecture," *IEEE Trans. Power Syst.*, vol. 26, pp. 12–19, Feb 2011.
- [18] M. Abido, "Optimal power flow using tabu search algorithm," *Elect.Power Compon. Syst.*, vol. 30, no. 5, pp. 469–483, 2002.
- [19] A. Saini, D. K. Chaturvedi, and A. Saxena, "Optimal power flow solution: a ga-fuzzy system approach," *Int. J. Emerg. Elect. Power Syst.*, vol. 5, no. 2, 2006.