International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:14, No:3, 2020

Upgraded Cuckoo Search Algorithm to Solve Optimisation Problems Using Gaussian Selection Operator and Neighbour Strategy Approach

Mukesh Kumar Shah, Tushar Gupta

Abstract**—**An Upgraded Cuckoo Search Algorithm is proposed here to solve optimization problems based on the improvements made in the earlier versions of Cuckoo Search Algorithm. Short comings of the earlier versions like slow convergence, trap in local optima improved in the proposed version by random initialization of solution by suggesting an Improved Lambda Iteration Relaxation method, Random Gaussian Distribution Walk to improve local search and further proposing Greedy Selection to accelerate to optimized solution quickly and by "Study Nearby Strategy" to improve global search performance by avoiding trapping to local optima. It is further proposed to generate better solution by Crossover Operation. The proposed strategy used in algorithm shows superiority in terms of high convergence speed over several classical algorithms. Three standard algorithms were tested on a 6-generator standard test system and the results are presented which clearly demonstrate its superiority over other established algorithms. The algorithm is also capable of handling higher unit systems.

Keywords**—**Economic dispatch, Gaussian selection operator, prohibited operating zones, ramp rate limits, upgraded cuckoo search

I. Introduction

ECONOMIC dispatch (ED) is one of the most fundamental optimization problems in electric power systems with the objective to minimize the total cost for power generation. It aims at economically allocating the load demand among the generators while satisfying several equality and inequality constraints in the systems.

In recent years, several metaheuristic nature-inspired algorithms for solving complex engineering optimization problems have been proposed [1], [2]. Metaheuristic algorithms are basically a part of stochastic optimization algorithms [3] that depend on randomization in generating possible solutions in search space. Therefore, these algorithms perform better by escaping local optima compared with deterministic algorithms [4]. In addition, metaheuristic algorithms are inspired by natural evolutionary behavior of some species. Due to the robustness of metaheuristic optimization algorithms, these are used in solving complex problems in different engineering systems to be optimized.

Compared with the existing studies, the following contributions have been made over standard Cuckoo Search (CS):

1) The Upgraded Cuckoo Search Algorithm (UCSA) utilizes the Random Gaussian Distribution Walk instead of the Lévy flights in exploitation of the search space to avoid the fixed step size used in the CS algorithm. With its Gaussian random diffusion behavior it has a promising performance in reaching the global optimum. The performance of exploitation of Upgraded Cuckoo Search (UCS) algorithm is enhanced further by utilizing greedy selection approach as used in Artificial Bee Colony (ABC) algorithm [5].
2) A "Study nearby Strategy" is adapted in the paper. When the best solution is no longer updated after a number of iterations, each solution can exchange the information with others randomly.
3) An improved Lambda Iteration Relaxation method is suggested in the paper to generate feasible solutions at the initial stage. In UCSA, all solutions must be feasible thus a relaxation method is designed to handle the equality constraint that may lead to infeasible solutions.

The rest of paper is organized as follows: Section II describes the ED problem. Section III implements UCSA to solve the ED problem. Section IV is dedicated to numerical simulations and results. Conclusions are given in Section V.

II. Problem Formulation

The objective of ED problem is to minimize the fuel cost of generators in electric power systems for a given load demand subject to various constraints.

A. Objective Function

The fuel function without valve-point loading of generators is given as:

$$min\ F_t = \sum_{i=1}^{D} F_i(P_i) = \sum_{i=1}^{D} a_i + b_i P_i + c_i P_i^2 \qquad (1)$$

where D is the total number of generators. Fi (Pi) is the fuel cost of the i^{th} generator with unit \$/h. Pi is the power in megawatt (MW) generated by the i^{th} generator, and a_i, b_i and c_i are respectively the cost coefficients of the i^{th} generator.

B. Equality Constraint

In order to balance the power, the total generated power should meet the power demand and transmission loss (TL).

$$\sum_{i=1}^{D} P_i = P_T + P_L \qquad (2)$$

Mukesh Kumar Shah is Assistant Professor with the Department of Electrical Engineering, MNIT, Jaipur, India (e-mail: author@ mkshah.ee@mnit.ac.in).
Tushar Gupta is Application Developer with the Oracle India Private Limited, CO 80309 USA, Hyderabad, India (e-mail: tusharyash1997@gmail.com).

where P_T (MW) is the total power demand and P_L (MW) is the TL which can be computed by using B-coefficients [6] and is given by

$$P_L = \sum_{i=1}^{D}\sum_{j=1}^{D} P_i B_{ij} P_j + \sum_{i=1}^{D} B_{0i} P_i + B_{00} \quad (3)$$

where B_{ij}, B_{0i} and B_{00} are the loss coefficients which are constant under normal operational conditions.

C. Inequality Constraint

The output power of each generator has a lower limit and an upper one

$$P_i^{min} \le P_i \le P_i^{max}, \;\; i = 1,2,3 \dots \dots D \quad (4)$$

where P_{imin} and P_{imax} are the minimum and maximum power, respectively, in MW generated by the i^{th} generator.

D. Prohibited Operating Zones

Each generator may have certain prohibited operating zones (POZ) caused by opening or closing its steam valve. The feasible operating zones of generator i can be described as:

$$P_i \in \begin{cases} P_i^{min} \le P_i \le P_{i,1}^{l} \\ P_{i,j-1}^{u} \le P_i \le P_{i,j}^{l} \\ P_{i,n_j}^{u} \le P_i \le P_i^{max} \end{cases} \quad , \; j = 2,3,\dots\dots,n_j \quad (5)$$

where n_j is the number of POZ of the i^{th} generator, and $P_{i,j}^{l}$ and $P_{i,j}^{u}$ are the lower and upper bounds of power in the j^{th} POZ by the i^{th} generator, respectively.

E. Ramp Rate Limits

Practically, all generators should satisfy the physical limitation of starting up and shutting down by using ramp rate limits (RRL). The increase and reduction of power generation in each generator are limited by

$$P_i - P_i^0 \le U_i \quad (6)$$

$$P_i^0 - P_i \le L_i \quad (7)$$

where P_i^0 is the previous output power. U_i and L_i are the up-ramp limit and the down-ramp limit of the i^{th} generator, respectively.

Combining (6) and (7) with (4) results in the change of the generation limits to

$$\underline{P_i} \le P_i \le \overline{P}_i, \quad \mathrm{i} = 1,2,\dots,\mathrm{D} \quad (8)$$

where

$$\underline{P_i} = \max\left(P_i^{min}, P_i^0 - L_i\right), \quad \overline{P}_i = \min(P_i^{max}, P_i^0 + U_i). \quad (9)$$

III. UCSA

The standard CS Algorithm [7] presents algorithm's three main steps as: (i) Improved Lambda Iteration Relaxation Method, (ii) Random Gaussian Distribution Walk, (iii) Better Selection by Greedy Approach (iv) Study Nearby Strategy

A. Improved Lambda Iteration Relaxation Method

To obtain feasible and easy to repairable solutions at the beginning, we can modify Classical Lambda Iterations as:

1. Calculate $\lambda_i = b_i + 2c_i p_i$ for every generator.
2. Calculate lambda at maximum λ_i^{max} and minimum λ_i^{min} generator limit for every generator unit.
3. Calculate mean lambda λ_{mean} and variance lambda λ_{var}
4. Calculate $\lambda = \lambda_{mean} + randn \times sqrt(\lambda_{var})$
5. Then iterate to every generator and if find $\lambda_i < \lambda_i^{min}$ then $temp = \lambda_{i,min} + 0.5 \times rand \times \left(\lambda_{i,max} - \lambda_{i,min}\right)$ else if $\lambda_i > \lambda_i^{max}$ then $temp = \lambda_{i,max} + 0.5 \times rand \times (\lambda_{i,max} - \lambda_{i,min})$
6. New solution would be as $X_k^i = (temp - b_i)/(2c_i)$

B. Random Gaussian Distribution Walk

The new solution is generated as follows:

$$X_i^{t+1} = Gaussian(Best^t, \sigma) + (r_1 * Best^t - r_2 * X_i^t) \quad (10)$$

where X_i^{t+1}: the new solution and X_i^t: i^{th} solution in generation t. $Best^t$ refers to the best solution in generation t and both r_1 and r_2 are random numbers between [0,1]. In Gaussian distribution, the mean is assigned to its best nest and the standard deviation σ is chosen as product of ratio of logarithm of t and t with difference of X_i^t and $Best^t$.

C. Greedy Selection

The performance of exploitation of UCS algorithm is enhanced further by utilizing greedy selection approach as used in ABC algorithm [5].

Each solution of every iteration is calculated as

$$v_{ij}^t = x_{ij}^t + r_3 * (x_{ij}^t - x_{kj}^t) \quad (11)$$

where V_i^t is the new solution, r_3 is random number between [0,1] and j is a random chosen parameter index.

D. Study nearby Strategy

Discarding some solutions is generally based on

$$K = \begin{cases} 1 & if \;\; r_5 > P_\alpha \\ 0 & otherwise \end{cases} \quad (12)$$

K is an updated coefficient based on the probability of discovering an alien egg in its nest.

New replacing solutions for every i^{th} generator can be then

$$X_{k'}^i = (1-k) \times X_k^i + K \times \gamma_{k'}^i \quad (13)$$

where exemplar $\Upsilon = [\gamma_1, \gamma_2, \dots\dots\dots, \gamma_N]^T$ is calculated by comparing any two random solution set and taking ahead the superior one in next iteration through exemplar set. If after few iterations, global solution does not reach and current solution is repeated again, another exemplar can be remodeled. It will also save time in reaching to the global solution.

The main steps of UCSA are illustrated in Fig. 1.

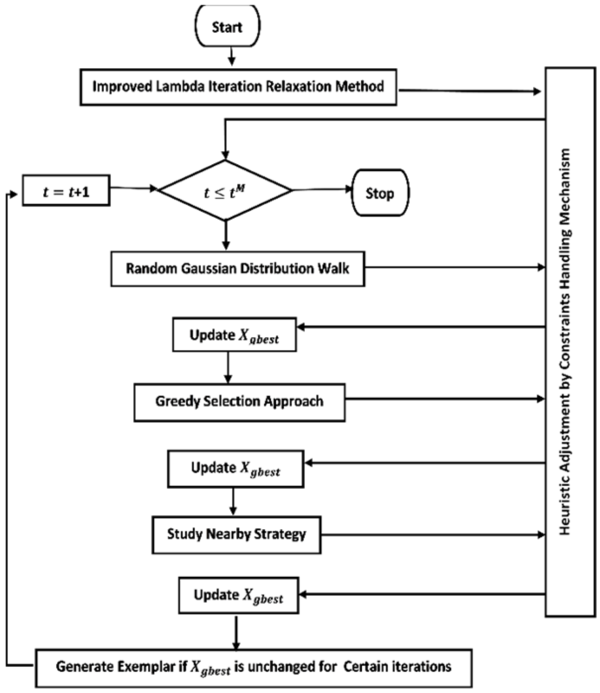

Fig. 1 Flow chart of UCSA

IV. Numerical Simulations and Results

In order to demonstrate the efficiency and robustness of UCSA for solving the ED problem, a 6-unit system is considered, and the results are compared with several state-of-the-art ED algorithms in the literature. All case studies are implemented in MATLAB R2015a, on a personal computer with Intel i5 2.5GHz processor, 8GB of RAM and Windows 10 Home. Due to the stochastic nature of an evolutionary algorithm in each case, 50 independent trials are conducted to calculate the best, mean, and worst fuel costs, and its standard deviation for each test system.

The tested system is a 6-unit system which has a demand of 1263 MW with POZ, TL and RRL. Number host nest are taken to be 25, probability of discarding cuckoo eggs is taken 0.25 and algorithm is run for 200 iterations. Its input data are taken from [8] and [9]. The best generation values, TL and optimal cost obtained by UCSA are presented in Table I. Note that all system constraints, such as POZ and RRL are satisfied. The total generation cost and the corresponding TL are 15449.8995$/h and 12.9582MW, respectively.

TABLE I
Optimal Generations and Costs Obtained by UCSA

Unit	$\underline{P}_i(MW)$	$\overline{P}^i(MW)$	*POZ*	*Generation (MW)*
1	320	500	[210,240]; [350,380]	447.5038
2	80	200	[90,110]; [140,160]	173.3182
3	100	265	[150,170]; [210,240]	263.4628
4	60	150	[80,90]; [110,120]	139.0653
5	100	200	[90,110]; [140,150]	165.4734
6	50	120	[75,85]; [100,105]	87.1347
Cost ($/h): 15449.8995, TL (MW): 12.9582				

Table II shows the obtained best cost, mean cost, worst cost, standard deviation and time for this test system after 50 trial runs. These results are compared with other algorithms. TOL is the difference of power balance calculated by

$$TOL = \sum_{i=1}^{D} P_i - P_T - P_L \quad (14)$$

Through comparison, it can be found that the best, mean, worst costs and standard deviation obtained by UCSA are the least. Although the best, mean and worst costs are same for CS, MCSA and UCSA, the standard deviation, time and obtained by UCSA are much better than them. Moreover, it shows that the UCSA is more consistent and stable than the other algorithms.

Fig. 2 shows the generation cost convergence of the best solution with iterations for CS, MCSA and UCSA for a typical run. It can be seen that all CS, MCSA and UCSA enjoy smooth convergence, but UCSA is faster than MCSA and CSA. This algorithm is also tested on 13 and 20 unit system whose results data are shown in Tables III and IV respectively. The generator data of 13-unit system are taken from [10], and the loss coefficient is from [11] and the data of 20-unit test system can be found in [12]. The results again show the proposed algorithm superiority over other available algorithms.

TABLE II
Comparison between UCSA and Other Published Algorithms on 6 Units System

No.	*Algorithm*	*Best($/h)*	Mean ($/h)	Worst ($/h)	Std. dev.	Time (s)	*TOL* (MW
1	**UCSA**	**15449.8995**	**15449.8995**	**15449.8995**	**1.6003E−11**	**0.2044**	**−3.2634E−12**
2	MCSA [13]	15449.8995	15449.8995	15449.8995	1.6404E−11	0.2599	−3.6380E−12
3	MABC [14]	15449.8995	15449.8995	15449.8995	6.04E−8	0.62	7.208E−11
4	CS [15]	15449.8995	15449.8995	15449.8995	8.8315E−7	0.2514	−1.3642E−12

TABLE III
Comparison between UCSA and Other Published Algorithms on 13 Units System

No.	*Algorithm*	*Best($/h)*	Mean ($/h)	Worst ($/h)	Std. dev.	Time (s)	*TOL* (MW
1.	**UCSA**	**24512.5392**	**24512.5392**	**24512.5392**	**2.9382E-7**	**2.0098**	**4.1928E-13**
2	MCSA[13]	24514.8756	24514.8756	24514.8756	3.1191E−7	2.5592	4.5470E−13
3	MABC [14]	24514.8756	24514.8756	24514.8756	3.5E−7	117.6	2.91E−11
4	CS [15]	24514.9857	24516.9312	24538.1519	3.9436	2.7166	0

TABLE IV
COMPARISON BETWEEN UCSA AND OTHER PUBLISHED ALGORITHMS ON 20 UNITS SYSTEM

No.	***Algorithm***	***Best($/h)***	Mean ($/h)	Worst ($/h)	Std. dev.	Time (s)	*TOL* (MW
1	**UCSA**	**62456.6331**	**62456.6331**	**62456.6331**	**1.1008E-11**	**2.1358**	**-4.3467E-12**
2	MCSA[13]	62456.6331	62456.6331	62456.6331	1.2077E-11	2.4668	-5.0022E-12
3	CBA[16]	62456.6328	62456.6348	62501.6714	0.3809	1.16	-
4	CS [15]	62456.6331	62456.6331	62456.6331	2.8818E-8	2.1913	-2.2732E-12

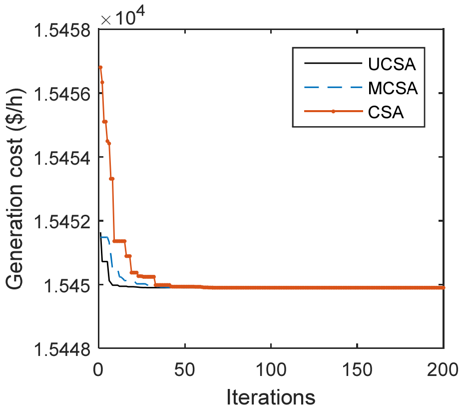

Fig. 2 Convergence characteristic of UCSA

V. CONCLUSION

In this paper, an UCSA is proposed. It has been designed in such a way that it has superior exploitation and exploration capability for producing better results. It can handle both convex and non-convex ED problems. The proposed algorithm has been run over 6-unit test system considering ramp limits, POZ and TL. Reported results in literature are verified by statistical results obtained. It is found that UCSA outperforms other algorithms in terms of reaching optimum solution, speed of convergence, efficiency and robustness. Thus, it has superior capability in solving optimisation problems.

REFERENCES

[1] C. Blum, J. Puchinger, G.R. Raidl and A. Roli, "Hybrid metaheuristics in combinatorial optimization: a survey," Appl. Soft Comput., vol. 11, pp. 4135–4151, 2011.
[2] I. Boussaïd, J. Lepagnot and P. Siarry, "A survey on optimization metaheuristics," Information Sciences, vol. 237, pp. 82–117, 2013.
[3] J. C. Spall, Introduction to stochastic search and optimization: estimation, simulation, and control, vol. 65, Hoboken:Wiley & Sons, 2005.
[4] A. R. Simpson, G. C. Dandy and L. J. Murphy, "Genetic algorithms compared to other techniques for pipe optimization," Journal Water Resource Planning and Management, vol. 120, pp. 423–43, 1994.
[5] A. Draa and A. Bouaziz, "An artificial bee colony algorithm for image contrast enhancement," Swarm and Evolutionary Computation, vol. 16, pp. 69-84, 2014.
[6] A. J. Wood and B. F. Wollenberg, Power Generation, Operation, and Control. Beijing: Tsinghua University Press, 2003, pp. 195.
[7] X. S. Yang and S. Deb, "Cuckoo search via levy flights," in´ Proc. World Congress on Nature and Biologically Inspired Computing, Kitakyushu, Japan, 2009.
[8] Z. L. Gaing, "Particle swarm optimization to solving the economic dispatch considering the generator constraints," *IEEE Trans. Power Syst.*, vol. 18, no. 3, pp. 1187−1195, Aug. 2003.
[9] Z. L. Gaing, "Closure to "discussion of 'particle swarm optimization to solving the economic dispatch considering the generator constraints'"," *IEEE Trans. Power Syst.*, vol. 19, no. 4, pp. 2122−2123, Nov. 2004
[10] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, "Evolutionary programming techniques for economic load dispatch," IEEE Trans. Evol. Comput., vol. 7, no. 1, pp. 83−94, Feb. 2003.
[11] A. Srinivasa Reddy and K. Vaisakh, "Shuffled differential evolution for large scale economic dispatch," Electr. Power Syst. Res., vol. 96, pp. 237−245, Mar. 2013.
[12] C. T. Su and C. T. Lin, "New approach with a Hopfield modeling framework to economic dispatch," IEEE Trans. Power Syst., vol. 15, no. 2, pp. 541−545, May 2000.
[13] Jian, Zhao, Shixin, Liu, Mengchu, Zhou, Xiwang, Guo, Liang, Qi, modified cuckoo search algorithm to solve economic power dispatch optimization problems, IEEE/CAA Journal of Automatica Sinica 2018, 5, 794 – 806, https://doi.org/10.1109/JAS.2018.7511138.
[14] D. C. Secui, "A new modified artificial bee colony algorithm for the economic dispatch problem," *Energy Convers. Manage.*, vol. 89, pp. 43 −62, Jan. 2015.
[15] X. S. Yang, *Nature-Inspired Optimization Algorithms.* Amsterdam, Holland: Elsevier Science Publishers B. V., 2014
[16] B. R. Adarsh, T. Raghunathan, T. Jayabarathi, and X. S. Yang, "Economic dispatch using chaotic bat algorithm," Energy, vol. 96, pp. 666− 675, Feb. 2016.