DOI: 10.48011/asba.v2i1.1513

On Evolutionary and Swarm Computation for Solving the Dynamic Economic Dispatch Problem: a Rapid Review

Omar Andres Carmona Cortes * **Daniel Lima Gomes Junior** *
Osvaldo Ronald Saavedra **

* *Departamento de Computação, Instituto Federal de Educação, Ciência e Tecnologia do Maranhão, São Luís, MA (e-mail: {omar,daniellima}@ifma.edu.br).*
** *Instituto de Energia Elétrica, Universidade Federal do Maranhão, São Luís, MA (e-mail: o.saavedra@ieee.org)*

Abstract: The Dynamic Economic Dispatch problem is a fundamental and challenging optimization problem in the field of power dispatch. Several techniques have been investigated for its optimization, especially metaheuristics. This rapid review focuses on evolutionary and swarm computation, focusing on 22 research publications for solving the single objective DED problem. Through this review, we discuss the techniques that have been used to solve the problem and how they tackled the DED constraints. We analyze the problem's complexity, showing if the problem being solved considers the valve effect, transmission losses, ramp rates, prohibited zones, and reserve spinning requirements. Also, we investigate the number of units used in each case study. Therefore, we believe that this review is relevant for driving researchers interested in using evolutionary algorithms or swarm intelligence to tackle the DED problem in all its complexity and how the proposed algorithms deal with constraints.

Keywords: Metaheuristics; Dynamic; Economic Dispatch; Optimization.

1. INTRODUCTION

The primary purpose of the Economic Dispatch (ED) (Pereira-Neto et al., 2005) (Barros et al., 2013) is to minimize the total energy production costs while various generator constraints are satisfied. However, generator output curves have a high degree of non-linearity and discontinuities due to the effect of "valve points" (Ribeiro Jr. et al., 2020).

As the operating costs of different generating units differ significantly in ED, it is challenging to schedule the best mix of generation from several units to attend a particular load demand at minimum cost for an entire day. So, when a specific demand is established, the problem is known as static ED.

The main drawback of the static approach is to establish a unique demand for 24 hours. Usually, different hours of the day demand different generation requirements. To attend them, the ED problem can be extended to the Dynamic Economic Dispatch (DED), in which each hour of the day may require a different power consumption, turning the problem into a more challenging one due to its high number of variables. For instance, let us consider a test-bed problem devised by five generators. Thus, to compute the cost of operation, we need 120 variables, *i.e.*, five generators times 24 hours. As we can see, the number of variables proportionally increases as we increase the number of generators. Further, limitations such as ramp rates and transmission losses turn the problem into a much more complex one.

Therefore, classical optimization methods are inefficient due to the high number of variables and, either, because classical algorithms cannot deal with constraints. Consequently, evolutionary and swarm techniques, such as Genetic Algorithms (GA) (Holland, 1975), Evolutionary Programming (Yao et al., 1999), Differential Evolution (Storn and Price, 1997), Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995), and Artificial Bee Colony (Karaboga, 2005), among others, represent an attractive manner of tackling this kind of optimization problem.

Despite the capability of evolutionary and swarm computation, there are still challenges to deal with because DED, in its full complexity, demands the algorithm to obey several constraints, as previously mentioned. Zaman et al. (2016) claims that in most of these approaches, the equality constraints are usually handled using the penalty-function technique. However, there are too many equality constraints in DED problems that are mutually coupled, making it challenging to generate feasible solutions and maintain feasibility after crossover and mutation operations in a metaheuristic such as GA. Therefore, programmers who want to solve DED must tackle all those constraint requirements.

In this context, we would like to contribute with the area guiding researchers in this challenging field, answering the following research questions:

Q1 What are the most used evolutionary and swarm techniques for solving the DED and how they deal with constraints?
Q2 What the complexity of the DED, and what are the most common case studies being solved?

For this sake, this paper is divided as follows: Section 2 presents the methodology of selecting and excluding papers; Section 3 shows the DED problem on its full complexity; Section 4 presents the results that answer the questions raised in this paper; finally, Section 5 illustrates the final remarks of this work.

2. METHODOLOGY

A rapid review is a technique for the synthesis of evidence for a comprehensive or systematic search of the literature; however, it requires a shorter time frame than traditional systematic approaches (Khangura et al., 2012). It addresses a research question or a set of research questions related to a single topic. In this particular case, the subject is using evolutionary or swarm algorithms for solving the DED problem, consequently answering the questions Q1 and Q2 presented previously in Section 1.

In Q1, we aim to identify what are the most used algorithms in solving the DED problem. Additionally, we want to distinguish how researchers have dealt with constraints, which is an essential part of adequately addressing the problem. In Q2, we intend to determine the systems and their complexity; thus, researchers can obtain directions to compare their algorithms. Therefore, considering Q1 and Q2, we devised the following search string: ***"Dynamic Economic Dispatch" AND (Swarm OR Evolutionary)***.

As suggested by Brereton et al. (2007), we used the search string in four electronic databases: IEEE Xplore, ACM Digital, Springer, and ScienceDirect. As time is a constraint in getting this job done, it is necessary to define criteria that narrow the search even more. In this context, we set the inclusion and exclusion criteria as follows.

2.1 Inclusion Criterion

The following inclusion criterion has carried out:

- Papers whose primary objective is to solve the DED problem using any evolutionary or swarm algorithms;
- Papers whose the DED problem belongs to the set of problems being solved using the referred algorithms;
- Papers published between 2016 and 2020;
- Papers solving the single fitness function model

2.2 Exclusion Criterion

As previously mentioned, we have to narrow the set of papers we want to analyze. Hence, the following exclusion criteria have been considered:

- Papers in languages other than English;
- Short papers (less than four pages);
- Papers computing only the emission cost;
- Papers whose techniques change the mathematical model, i.e., other than the model presented in Section 3;
- Bio-inspired algorithms that are not considered as evolutionary or swarm one.
- Papers handling only a part of the problem, such as papers handling only transmission losses;
- Papers combining models that change the mathematical model, such as those mixing the cost with other power generators, which ends up changing the mathematical model as well;
- Papers that presented surveys and reviews.

Next, we present the mathematical model of the DED problem in its full complexity, *i.e.*, including all possible constraints that make the problem harder to solve.

3. DYNAMIC ECONOMIC DISPATCH

The purpose of the DED problem optimization is to discover the best power dispatch in a power plant, *i.e.*, attending different power demands during the day, minimizing the cost of doing it. Thus, the objective is to compute the required power on each generator unit for 24 hours. Moreover, the generation of power must obey the constraints of the system. Therefore, the goal is to minimize the cost of production and, at the same time to satisfy all constraints: (i) balance limits; (ii) real power generation limits; (iii) unit ramp rate limits; (iv) prohibited zones; and, (v) spinning generators.

It is clear that when the demand changes, the power generation must change either. The problem is that changing the power generation has a cost associated with the process. This behavior turns the problem into one impossible to solve by a gradient-based method; therefore, the problem is entirely suitable to be tackled by evolutionary algorithms (Ribeiro Jr. et al., 2020). According to Kumar and Alwarsamy (2011), DED is a dynamic problem due to the dynamic nature of the power system and the considerable variation of load demands.

The cost of generating power is depicted by Equation 1, in which F is the power generation cost within the period, T is the number of intervals, N represents the number of generation units, and $F_{it}(P_{it})$ is the cost of the real power P_{it} in a time interval t. The function works on 24 intervals of 1 hour each.

$$\min F = \sum_{t=1}^{T} \sum_{i=1}^{N} F_{it}(P_{it}) \tag{1}$$

Thus, the cost of producing power is represented by Equation 2, in which i represents the generation unit, a_i, b_i, and c_i are cost coefficients, and P_i is the power output of this unit expressed in MW.

$$F_{it}(P_{it}) = a_i + b_i P_{it} + c_i P_{it}^2 \tag{2}$$

When we consider the valve effects, we have to add the expression $|e_i \sin f_i(P_{imin} - P_{it})|$ to the Equation 2. Then, we can rewrite the cost function as illustrated in

Equation 3, in which e_i and fi are constants of the valve effect of the generation unit i.

$$F_{it}(P_{it}) = a_i + b_i P_{it} + c_i P_{it}^2 + |e_i \sin f_i (P_{imin} - P_{it})| \quad (3)$$

As previously mentioned, the computation of cost is associated with constraints. Each one must be obeyed in order to produce a solution. Equation 4 aims to keep the operation limits of each generator, in which P_{imin} is the lower bound, P_{imax} is the upper bound of the generation unit i, $t = 1, 2...T$, and $i = 1, 2...N$.

$$P_{itmin} \leq P_{it} \leq P_{itmax} \quad (4)$$

Equation 5 expresses the first equality constraint that is the power balance constraint, in which P_{Dt} represents the total power demanded in a period t, P_{Lt} is the power loss during transmission in the same period, both are in MW.

$$\sum_{i=1}^{N} P_{it} - P_{Dt} - P_{Lt} = 0, \text{ in which } t = 1, 2...T \quad (5)$$

The transmission loss is computed by using Equation 6 in which B is a matrix of loss coefficients. Consequently, Equation 5 can be rewritten as $\sum_{i=1}^{N} P_{it} = P_{Dt} + P_{Lt}$

$$P_{Lt} = \sum_{i=1}^{n}\sum_{j=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} B_{0i} P_i + B_{00} \quad (6)$$

Equations 7 and 8 describe the ramp limit constraints of a unit i, in which UR_i is the increasing limit of a generation unit i, and DR_i is the decreasing limit.

$$P_{it} - P_{it-1} \leq UR_i, \text{ in which } i = 1, 2...N \quad (7)$$

$$P_{it-1} - P_{it} \leq DR_i, \text{ in which } i = 1, 2...N \quad (8)$$

Additionally, the DED problem can present prohibited zones, *i.e.*, the operation zone of generators can be discontinuous, which adds more difficulties to the algorithm that will solve the problem. The forbidden zones can be expressed as depicted in Equation 9, in which P_{itj} and P_{itz} are the limits of the prohibited zones.

$$P_{it} \in \begin{cases} P_{itmin} \leq P_{it} \leq P_{it,1} \\ P_{it(j-1)} \leq P_{it} \leq P_{itj} & j = 1, 2, \ldots, z \\ P_{itz} \leq P_{it} \leq P_{itmax} \end{cases} \quad (9)$$

Finally, the DED model can present reverse spinning requirements to protect the system against unexpected events such as load changes and failure in the operating units. Hence, to increase the system reliability, three new constraints can be included in the DED mathematical model as shown in Equations 10, 11, and 12.

$$\sum_{i=1}^{N} P_(it)^{max} - (P_{Dt} + P_{Loss} + SR_t) \geq 0 \quad (10)$$

$$\sum_{i=1}^{N} min(P_(it)^{max} - P_{it}, UR_i) - SR_t \geq 0 \quad (11)$$

$$\sum_{i=1}^{N} min(P_(it)^{max} - P_{it}, \frac{UR_i}{6}) - SR_t \geq 0 \quad (12)$$

Constraints 10 and 11 are frequently applied to satisfy the one-hour spinning reserve requirements (SR), and constraint 12 is used to fulfill the SR for the spinning generators in each time within 10 min is related to the ramp-up rate constraint of that unit (URi/6) (Zaman et al., 2016).

In the next section, we show and analyze the results of this review.

4. RESULTS AND DISCUSSION

The search has been conducted from April to June of 2020. The search string returned the following number of papers: IEEE Xplore - **63**, ScienceDirect - **86**, ACM Digital - **37**, and Springer **92**, totaling 278. After applying the inclusion and exclusion criteria, we selected 22 papers. Figure 1 shows how many papers we included per year in this study and how the number of papers is distributed over the years. As we can see, there are more papers in 2018 and 2019. The year 2020 presents a low quantity of paper, probably because we are still in the first semester.

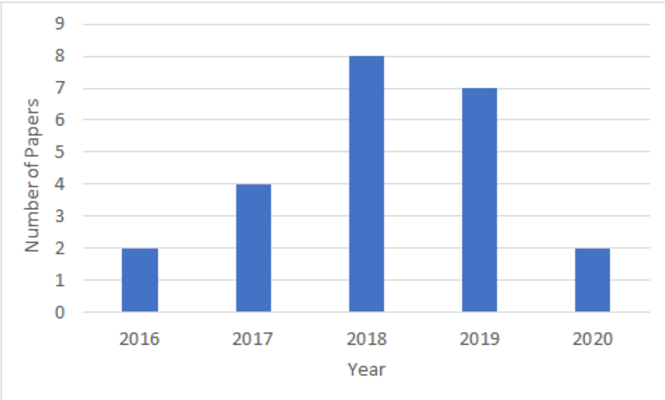

Fig. 1. Number of papers per year

4.1 Q1 - Algorithms Features

Table 1 summarizes the results for Q1, which is divided into two parts. The first one presents information on the algorithm, *i.e.*, the name, and if it is an enhanced or hybrid one. The second part shows how the algorithms deal with constraints, *i.e.*, if there is an initialization heuristic, if the algorithm uses a repair method in case of a constraint violation, and how the constraints are handled.

According to Table 1, the main base algorithms have been found solving the DED are: Genetic Algorithms, Evolutionary Programming, Differential Evolution, Invasive Weed Optimization (IWO) (Mehrabian and Lucas, 2006), Particle Swarm Optimization, Artificial Bee Colony, Firefly Algorithm (FA) (Yang, 2009), Ant Lion Optimizer (ALO) (Mirjalili, 2015), Moths Search Algorithm (MSA) (Wang,

Table 1. Evolutionary and Swarm Computation Techniques, and dealing with constraints (Q1)

#	Paper	Algorithms	Hybrid	Variation	Init Heuristic	Repair	Constraints
1	(Zaman et al., 2016)	GA, DE-Se-A	No	Yes	Yes	Yes	$\epsilon-$constraints
2	(Zaman et al., 2016)b	GA, DE-Se-A	No	No	No	Yes	Slack Generator
3	(Aydin et al., 2017)	ABC	No	Yes	No	Yes	No
4	(Gupta and Goyal, 2017)	PSO	No	No	No	No	No
5	(Xie et al., 2017)	PSO	No	No	No	No	No
6	(Sun and Wang, 2017)	PSO	No	Yes	No	No	Penalties
7	(Xiong and Shi, 2018)	BBOSB	Yes	Yes	No	No	Slack Generator
8	(Behera et al., 2018)	CFBPSO	No	Yes	No	No	No
9	(Marzbani and Samet, 2018)	IWO	No	No	No	No	No
10	(Wang, 2018)	MSA	No	No	No	No	Penalty
11	(Pattanaik et al., 2018)	GA	No	Yes	No	No	Slack Generator
12	(Zou et al., 2018)	DE	No	Yes	No	Yes	Penalty
13	(Fergougui et al., 2018)	GA, PSO	No	No	No	No	No
14	(Pürlü and Türkay, 2018)	GA, PSO	No	No	No	No	No
15	(Pattanaik et al., 2019)	DE, PSO, EP, GA	No	No	No	No	Slack Generator
16	(Basu, 2019)	EP, CFCEP	No	Yes	No	No	No
17	(Mostefa et al., 2019)	FA	No	No	No	No	$\epsilon-$constraints
17	(Dhifaoui et al., 2019)	ALO	No	No	No	No	Multiobjective
19	(Shen et al., 2019)	DE	No	Yes	Yes	No	Penalty
20	(Haripuddin et al., 2019)	ABC	No	No	No	No	No
21	(Gupta et al., 2020)	PSO, IPSO	No	Yes	No	No	No
22	(Stanovov et al., 2020)	DE	No	Yes	No	No	$\epsilon-$constraints

2018), and Bio-geography-Based Optimization (BBO) (Simon, 2008). Figure 2 presents the results in terms of numbers. In this context, we can see that PSO is the most used one, followed by GA, DE, and ABC. The other metaheuristics are newer than the winners, maybe that is why they are still not being widely used. Also, eleven works have implemented variations, *i.e.*, presented enhanced algorithms, such as Zaman et al. (2016) and (Zaman et al., 2016) that presented a self-adaptive DE, in which the DE parameters are encoded into the individuals' genes. Only one work (Xiong and Shi, 2018) showed a hybrid metaheuristic that mixes BBO with Brain Storm Optimization (BSO) (Shi, 2011).

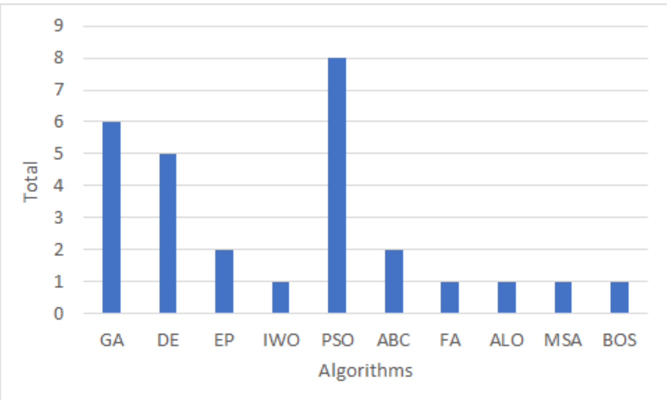

Fig. 2. Number of main algorithms

Concerning constraints, only Zaman et al. (2016)'s and Shen et al. (2019)'s works use an initialization heuristic. Usually, all generators are initialized using the constraint shown in Equation 11; however, this constraint does not guarantee that the other ones are also satisfied. Thus, an initialization heuristics intent to create solutions that satisfy all of them. Further, only four works apply a repair algorithm, trying to keep all the solutions satisfying the constraints after genetic or movement operators. On the one hand, repairing methods try keeping the feasibility of solutions. On the other hand, it can demand much more computation, and there is no guarantee that the repair can be done.

As we can see in the referred table, some works used the following method to handle constraints: $\epsilon-$constraints, Slack Generator, Multiobjective, and Penalties. In $\epsilon-$constraints, the violations in constraints are allowed but limited to an ϵ value, which usually decreases as the algorithm's iterations go by. In Slack Generator, a generator is chosen to compensate for the power loss. Typically, the last generator compensates for the losses in balance constraint; nonetheless, other approaches can be used such that one that is presented in Xiong and Shi (2018)'s work. Furthermore, the slack generator is associated with other handling constraints methods because it might violate the ramp limit or the prohibit zones.

The multiobjective approach, done in Dhifaoui et al. (2019)'s work, is an interesting way of handling constraints by transforming them and the cost function into a multiobjective problem. Finally, in penalties, a certain value is added to the cost function for each violated constraint. Either, many works in Table 1 do not inform how they manipulated constraints; however, we believe that they used penalty functions because it is the most common way of doing it, as mentioned by Zaman et al. (2016).

4.2 Q2 - DED Features

Table 2 refers to the complexity of the problem being solved on each work. In the simplest form, the DED only considers the valve effect that affects only the cost function. Ramp Limits, Losses, Prohibit Zones and Reverse Spinning requirements (SR) are constraints that increase the search space complexity. The table also presents the number of units used as a test problem.

Table 2. Complexity of the DED problem (Q2)

#	Paper	Valve Effects	Ramp Limits	Losses	Prohibit Zones	SR	Units
1	(Zaman et al., 2016)	Yes	Yes	Yes	No	No	5, 10, 30, 100, 150
2	(Zaman et al., 2016)b	Yes	Yes	Yes	Yes	Yes	5,6,7,10,19
3	(Aydin et al., 2017)	Yes	No	Yes	No	No	3, 5, 6, 13, 40
4	(Gupta and Goyal, 2017)	Yes	Yes	Yes	No	No	5
5	(Xie et al., 2017)	No	No	No	No	No	10
6	(Sun and Wang, 2017)	No	Yes	Yes	No	No	5
7	(Xiong and Shi, 2018)	Yes	Yes	Yes	No	No	5, 10
8	(Behera et al., 2018)	No	No	No	No	No	10
9	(Marzbani and Samet, 2018)	No	Yes	Yes	No	No	6
10	(Wang, 2018)	Yes	Yes	Yes	No	No	5, 9
11	(Pattanaik et al., 2018)	Yes	Yes	Yes	No	No	5
12	(Zou et al., 2018)	Yes	Yes	Yes	Yes	No	5, 10, 30
13	(Fergougui et al., 2018)	Yes	No	Yes	No	No	10
14	(Pürlü and Türkay, 2018)	Yes	No	Yes	No	No	3, 10
15	(Pattanaik et al., 2019)	Yes	Yes	Yes	No	No	10
16	(Basu, 2019)	Yes	Yes	Yes	No	No	10
17	(Mostefa et al., 2019)	Yes	Yes	Yes	No	No	10
18	(Dhifaoui et al., 2019)	Yes	Yes	Yes	No	No	5,10
19	(Shen et al., 2019)	Yes	Yes	Yes	Yes	No	5, 10
20	(Haripuddin et al., 2019)	No	Yes	Yes	No	No	7
21	(Gupta et al., 2020)	Yes	Yes	Yes	No	No	5
22	(Stanovov et al., 2020)	Yes	Yes	Yes	No	No	5, 9

As we can see, only Zaman et al. (2016) deals with DED in its full complexity. Moreover, the problem's most common complexity is considering the valve effects, ramp limits, and losses. Only three works handle the prohibited zones, and only one research examines the reserve spinning requirements. Regarding the number of units or generators, Figure 3 shows which the number of units is most common in the studies. The most recurrent units are 10 and 5 generators, followed by 6, 7, 9, and 30. More than 30 are uncommon problems and quite hard to deal with regarding computational time and showing results.

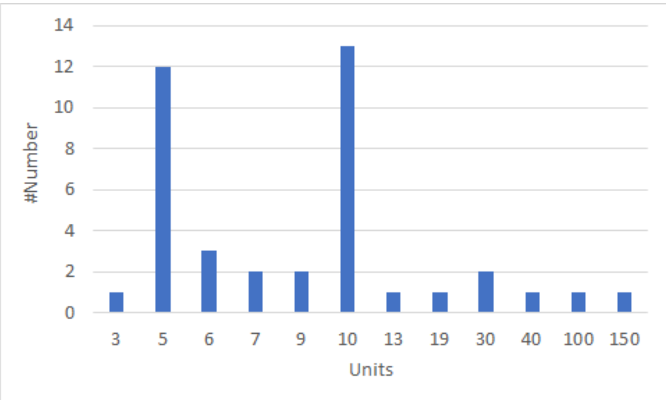

Fig. 3. Number of units used in the DED

5. CONCLUSIONS

This paper presented a rapid review of evolutionary and swarm computation for solving de Dynamic Economic-Dispatch Problem. The most common metaheuristics are PSO, GA, and DE. However, new metaheuristics, such as ALO and MSA, have been studied as well. We believe that attributing a penalty in the cost function for violating the constraints is the most common form of handling constraints, followed by Slack Generators. On the other hand, repairing solutions is not common, probably because it can increase the required computing time excessively, and there is no guarantee that the solution might be repaired. Nonetheless, interesting solutions have been proposed, such as $\epsilon-$constraints and the multiobjective approach. The most common problem solved is the DED with valve effect and ramp limits considering transmission losses and using 5 and 10 units (generators).

This research's main difficulty is that some papers do not provide enough information to identify all questions done in this work. Sometimes we have to go after one or more related works to correctly identify the base algorithm, the number of generators, or how they dealt with constraints. Especially those conference papers that are abided by restricted space rules. Some lack of information also impacts the experiment's reproducibility, especially in those works that do not provide information about how they treated constraints.

Future work to improve this review is: (i) add those works dealing only with emission cost; (ii) investigate other types of metaheuristics; and, (iii) expand the mathematical model to include also the multiobjective approaches; (iii) use this work to implement a variant of an algorithm for solving the DED; and (iv) implement an algorithm in GPU for solving big instance problems whit more than 30 units.

REFERENCES

Aydin, D., Yavuz, G., Özyön, S., Yaşar, C., and Stützle, T. (2017). Artificial bee colony framework to non-convex economic dispatch problem with valve point effects: A case study. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '17, 1311–1318. Association for Computing Machinery, New York, NY, USA.

Barros, R.S., Cortes, O.A.C., Lopes, R.F., and Silva, J.C.d. (2013). A hybrid algorithm for solving the economic dispatch problem. In *2013 BRICS Congress on*

Computational Intelligence and 11th Brazilian Congress on Computational Intelligence, 617–621.

Basu, M. (2019). Dynamic economic dispatch with demand-side management incorporating renewable energy sources and pumped hydroelectric energy storage. *Journal of Electrical Engineering*, 101(1), 1–17. doi:10.1007/s00202-019-00793-x.

Behera, S., Behera, S., and Barisal, A.K. (2018). Solution of dispatch problem with dynamic and static load using improved cfbpso. In *2018 International Conference on Applied Electromagnetics, Signal Processing and Communication (AESPC)*, volume 1, 1–4.

Brereton, P., Kitchenham, B.A., Budgen, D., Turner, M., and Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Systems and Software*, 80, 571–583.

Dhifaoui, C., Kahouli, O., and Abdallah, H.H. (2019). Multi-objective ant lion optimizer to solve the dynamic economic dispatch problem with valve point effect. In *19th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, 564–571.

Fergougui, A.E., Ladjici, A.A., Benseddik, A., and Amrane, Y. (2018). Dynamic economic dispatch using genetic and particle swarm optimization algorithm. In *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, 1001–1005.

Gupta, G.K. and Goyal, S. (2017). Particle swarm intelligence based dynamic economic dispatch with daily load patterns including valve point effect. In *2017 3rd International Conference on Condition Assessment Techniques in Electrical Systems (CATCON)*, 83–87.

Gupta, G.K., Goyal, S., and Yadav, I. (2020). A comparative study of dynamic economic load dispatch (deld) problem using pso and ipso for 24 hours load patterns. In *2020 International Conference on Power Electronics IoT Applications in Renewable Energy and its Control (PARC)*, 186–191.

Haripuddin, Suyuti, A., Said, S.M., and Syam Akil, Y. (2019). Dynamic economic dispatch for 150 kv sulselbar power generation systems using artificial bee colony algorithm. In *2019 International Conference on Information and Communications Technology (ICOIACT)*, 817–822.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI. Second edition, 1992.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical report, Erciyes University, Turkey.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *IEEE International Conference on Neural Networks*, volume 4, 1942–1948. IEEE.

Khangura, S., Konnyu, K., Cushman, R., Grimshaw, J., and Moher, D. (2012). Evidence summaries: the evolution of a rapid review approach. *Systematic Reviews*, 1(10), 1–10.

Kumar, C. and Alwarsamy, T. (2011). Dynamic economic dispatch – a review of solution methodologies. *European Journal of Scientific Research*, (4).

Marzbani, F. and Samet, H. (2018). Dynamic economic dispatch for a wind and run-of-the-river hydropower integrated system: An iwo implementation. In *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, 1–6.

Mehrabian, A. and Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics*, 1(4), 355–366.

Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 89, 228–249.

Mostefa, H., Mahdad, B., Srairi, K., and Mancer, N. (2019). Dynamic economic dispatch solution with firefly algorithm considering ramp rate limit's and line transmission losses. In M. Hatti (ed.), *Renewable Energy for Smart and Sustainable Cities*, 497–505. Springer International Publishing, Cham.

Pattanaik, J.K., Basu, M., and Dash, D.P. (2018). Improved real coded genetic algorithm for dynamic economic dispatch. *Journal of Electrical Systems and Information Technology*, 5(3), 349 – 362.

Pattanaik, J.K., Basu, M., and Dash, D.P. (2019). Dynamic economic dispatch: a comparative study for differential evolution, particle swarm optimization, evolutionary programming, genetic algorithm and simulated annealing. *Journal of Electrical Systems and Information Technology*, 6(1), 1–18. doi:10.1186/s43067-019-0001-4.

Pereira-Neto, A., Unsihuay, C., and Saavedra, O.R. (2005). Efficient evolutionary strategy optimisation procedure to solve the nonconvex economic dispatch problem with generator constraints. *IEE Proceedings - Generation, Transmission and Distribution*, 152(5), 653–660.

Pürlü, M. and Türkay, B.E. (2018). Dynamic economic dispatch with valve point effect by using ga and pso algorithm. In *2018 6th International Conference on Control Engineering Information Technology (CEIT)*, 1–6.

Ribeiro Jr., E.C., Cortes, O.A.C., and Saavedra, O.R. (2020). A parallel mix self-adaptive genetic algorithm for solving the dynamic economic dispatch problem. In *Simpósio Brasileiro de Sistemas de Energia*, 1–6. SBA.

Shen, X., Zou, D., Duan, N., and Zhang, Q. (2019). An efficient fitness-based differential evolution algorithm and a constraint handling technique for dynamic economic emission dispatch. *Energy*, 186, 115801.

Shi, Y. (2011). Brain storm optimization algorithm. In Y. Tan, Y. Shi, Y. Chai, and G. Wang (eds.), *Advances in Swarm Intelligence*, 303–309. Springer Berlin Heidelberg, Berlin, Heidelberg.

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, 12(6), 702–713.

Stanovov, V., Akhmedova, S., and Semenkin, E. (2020). Combined fitness–violation epsilon constraint handling for differential evolution. *Soft Computin*, 24(10), 7063–7079.

Storn, R. and Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4), 341—-359.

Sun, Y. and Wang, Z. (2017). Improved particle swarm optimization based dynamic economic dispatch of power system. *Procedia Manufacturing*, 7, 297–302. International Conference on Sustainable Materials Processing and Manufacturing, SMPM 2017, 23-25 January 2017, Kruger.

Wang, G. (2018). Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Computing*, 10(1), 151–164. doi: https://doi.org/10.1007/s12293-016-0212-3.

Xie, M., Cheng, P., and Ke, S. (2017). Power system dynamic economic dispatch with wind-solar integration based on particle swarm intelligent searching algorithm. In *2017 China International Electrical and Energy Conference (CIEEC)*, 252–257.

Xiong, G. and Shi, D. (2018). Hybrid biogeography-based optimization with brain storm optimization for non-convex dynamic economic dispatch with valve-point effects. *Energy*, 157, 424–435.

Yang, X.S. (2009). Firefly algorithms for multimodal optimization. In O. Watanabe and T. Zeugmann (eds.), *Stochastic Algorithms: Foundations and Applications*, 169–178. Springer Berlin Heidelberg, Berlin, Heidelberg.

Yao, X., Liu, Y., and Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3(2), 82–102.

Zaman, F., Elsayed, S.M., Ray, T., and Sarker, R.A. (2016). Configuring two-algorithm-based evolutionary approach for solving dynamic economic dispatch problems. *Engineering Applications of Artificial Intelligence*, 53, 105–125.

Zaman, M.F., Elsayed, S.M., Ray, T., and Sarker, R.A. (2016). Evolutionary algorithms for dynamic economic dispatch problems. *IEEE Transactions on Power Systems*, 31(2), 1486–1495.

Zou, D., Li, S., Kong, X., Ouyang, H., and Li, Z. (2018). Solving the dynamic economic dispatch by a memory-based global differential evolution and a repair technique of constraint handling. *Energy*, 147, 59 – 80.