Bai *et al. Journal of Cloud Computing* (2024) 13:59
https://doi.org/10.1186/s13677-024-00615-x

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Optimizing the resource allocation in cyber physical energy systems based on cloud storage and IoT infrastructure

Zhiqing Bai[1], Caizhong Li[2], Javad Pourzamani[3*], Xuan Yang[4] and Dejuan Li[4]

Abstract

Given the prohibited operating zones, losses, and valve point effects in power systems, energy optimization analysis in such systems includes numerous non-convex and non-smooth parameters, such as economic dispatch problems. In addition, in this paper, to include all possible scenarios in economic dispatch problems, multi-fuel generators, and transmission losses are considered. However, these features make economic dispatch problems more complex from a non-convexity standpoint. In order to solve economic dispatch problems as an important consideration in power systems, this paper presents a modified robust, and effective optimization algorithm. Here, some modifications are carried out to tackle such a sophisticated problem and find the best solution, considering multiple fuels, valve point effect, large-scale systems, prohibited operating zones, and transmission losses. Moreover, a few complicated power systems including 6, 13, and 40 generators which are fed by one type of fuel, 10 generators with multiple fuels, and two large-scale cases comprised of 80 and 120 generators are analyzed by the proposed optimization algorithm. The effectiveness of the proposed method, in terms of accuracy, robustness, and convergence speed is evaluated, as well. Furthermore, this paper explores the integration of cloud storage and internet of things (IoT) to augment the adaptability of monitoring capabilities of the proposed method in handling non-convex energy resource management and allocation problems across various generator quantities and constraints. The results show the capability of the proposed algorithm for solving non-convex energy resource management and allocation problems irrespective of the number of generators and constraints. Based on the obtained results, the proposed method provides good results for both small and large systems. The proposed method, for example, always yields the best results for the system of 6 power plants with and without losses, which are $15,276.894 and $15,443.7967. Moreover, the improvements made in the proposed method have allowed the economic dispatch problem regarding multi-fuel power plants to be solved not only with optimal results ($623.83) but also in less than 35 iterations. Lastly, the difference between the best-obtained results ($121,412) and the worst-obtained results ($121,316.1992) for the system of 40 power plants is only about $4 which is quite acceptable.

Keywords Energy optimization, Cloud storage, IoT, Prohibited operating zones, Multiple fuels and robust optimization method

*Correspondence:
Javad Pourzamani
javadpourzamani@mail.ir
Full list of author information is available at the end of the article

Introduction

Over the past decade, energy resource management and allocation have become key challenges for modern societies [1, 2]. The Economic Dispatch (ED) problem is one of the most crucial optimization issues in the operation and energy resource management and allocation of power systems. In this problem, considering the demanded load and transmission network losses, the output of each generator is determined to minimize fuel costs. Additionally, the output of each unit must satisfy power generation constraints [3]. This means that all information regarding the demanded load and each generator is gathered in one place and analyzed centrally. This aligns perfectly with the concept of cloud storage. Furthermore, to address the need for efficient information exchange, our approach involves a bidirectional flow of information. The cloud-based central system sends and receives all relevant data, while each unit receives and transmits information. This exchange is designed to be fast, secure, and intelligent, incorporating principles of IoT networks, which plays a significant role in our proposed methodology (see Fig. 1).

The mathematical methods, such as the lambda iteration method [5], gradient method [6], Newton method [7], linear programming [8], quadratic programming [9], and Lagrangian multiplier method [10], can be used to solve the economic dispatch problems with smooth and monotonic cost function [11]. In [12], dynamic economic dispatch problems have been solved by Lagrangian relaxation, while the Lagrangian multipliers have been updated through the quasi-newton process. The mathematical methods can find the best global solution in less number of iterations and hence converge faster [13]. However, in practical scenarios, the fuel cost function curves are non-convex and discontinue induced by valve point effect, losses, the impact of multiple fuels, and prohibited operating zones [14], which can hinder the applicability of these methods. To model the, a sinusoidal term is added to the traditional quadratic cost function of the economic dispatch problems [15], which brings about additional complexity. Moreover, as the number of decision variables increases, the mathematical complexity increases, and the problem becomes less tractable.

In order to address these issues, some meta-heuristics methods, such as fuzzy adaptive particle swarm optimization [16], hybrid genetic algorithm [17], and improved fast evolutionary programming [18] have been utilized. The non-smooth economic dispatch problems have been solved utilizing the fuzzy adaptive particle swarm optimization method in [16], while the Nelder-Mead process, like a local search algorithm, searches around the achieved solution in each iteration, which brings about more robustness [16]. In [17] the genetic algorithm, as the main optimization method, is used to tackle the economic dispatch problem with a non-smooth cost function. Sequential quadratic programming tunes the genetic algorithm in each trial run while the maximum entropy principle approximates the cost function to improve the process [17]. Authors in [18] have used improved fast evolutionary programming to solve the non-convex economic dispatch problem. Although these algorithms found the optimum solution, they need parameter tuning during the optimization process, which hinders their applicability of them. In addition, none of these algorithms considers the effect of multi-fuel generators and prohibited operating zones in solving the economic dispatch problem.

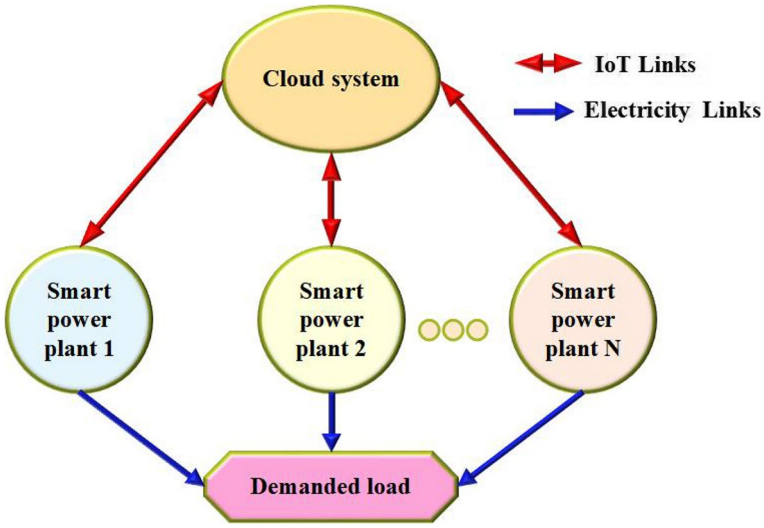

Fig. 1 Cloud-IoT infrastructure of energy resource allocation

Several works have either considered the prohibited operating zones or multi-fuel generators separately. The multi Tabu search algorithm equipped with salient mechanisms, genetic algorithm, and evolutionary strategy optimization algorithms proposed in [19–21] have solved the economic dispatch problem by considering only prohibited operating zones constraint. However, multiple fuel assumption, which makes the economic dispatch problems even more complex, is not considered in their framework. The optimal solution to the economic dispatch problem in the presence of valve point effects and multi-fuel generators is achieved by an improved genetic algorithm-multiplier updating method [22]. In fact, this method finds the optimal solution while it handles the equality and inequality constraint. However, the prohibited operating zones constraint, which is an important limiting factor in real-world scenarios, is not considered. In fact, generating power in these prohibited intervals may damage the generators. Therefore, a powerful and holistic method of solving non-convex and non-smooth economic dispatch problems with sophisticated constraints of having various combinations of generated power, considering the valve point effects, prohibited operating zones, and transmission loss is yet to be investigated.

Gap analysis

As it is obvious, providing a robust framework that can be utilized to solve complex economic dispatch problems in small- and large-scale systems is vital. Moreover, this framework should be easy implementing with an acceptable convergence speed. Hence, in this paper we seek to cover these challenges in solving economic dispatch problems.

One of the new methods of finding the global solution to such problems is to use the JAYA algorithm [23]. In this algorithm, the particle must move toward the best solution and avoid the worst one, in each iteration, to find the optimal solution. However, the JAYA algorithm is still not capable of providing the optimal solution in certain cases, which are shown in the results section of the presented paper, in detail. In fact, there are real-world scenarios where JAYA needs to be modified to provide the optimal solution.

In this paper, a newly modified optimization algorithm called modified JAYA is proposed to solve non-convex and non-smooth economic dispatch problems in small, medium, and large-scale systems where prohibited zones, losses, the impact of multiple fuels, and valve point effect are considered. Moreover, all possible moving states to get away/move towards the worst/best solution are considered. In addition, by adaptively selecting the population size through a mutation operator, the convergence rate and the effectiveness of the algorithm are improved. Furthermore, a new and proper process satisfies the exact demanded load, as the most important goal in energy resource management and allocation.

Research questions

- Is the proposed method effective in extracting the optimal solutions in small- and large-scale systems?
- Is the convergence speed suitable for finding optimal results in the proposed method?
- Is the proposed method effective in fulfilling the equal constraint, which is the provision of the demanded electrical power?
- How effective is the proposed method in the ED problem with multi-fuel power plants?

The rest of this paper is organized as follows. Section 2 illustrates the modeling of non-convex economic dispatch problems. Moreover, the role of multi-fuel generators and network losses is discussed. The proposed algorithm formulation is presented in Sect. 3. Section 4 presents the application of the modified JAYA algorithm in economic dispatch problems. Section 5 describes the test systems and simulation results. Finally, Sect. 6 draws conclusions.

Non-convex economic dispatch problem formulation

In order to better describe the advantages of the modified JAYA algorithm (MJAYA) algorithm, economic dispatch (ED) problems with convex and non-convex fitness functions are initially presented in subsections 2.1 and 2.2, respectively. In addition, the ED problem constraints, including power balance constraint, prohibited operating zones (POZs) Constraint, and output power constraint are introduced in subsection 2.3.

The ED problem with convex fitness function

The ED problems aim to minimize the total energy cost while all constraints are satisfied. Using a second-degree polynomial equation, the ED problem, aiming to minimize the cost of the generated power, is formulated by [24],

$$\begin{aligned} &Min\,H(X) = \sum_{i=1}^{Ng} F_i(p_{gi}) \\ &F_i(p_{gi}) = \alpha_i \times P_{gi}^2 + b_i \times P_{gi} + c_i \\ &X = [P_{g1}, P_{g2}, P_{g3}, ..., P_{gN_g}], \end{aligned} \tag{1}$$

where vector X is the decision variables. As stated before, ED problems with multi-fuel generators are also

investigated in this paper. As a result, such ED problems are represented by Eq. (2).

$$\text{Min } H(X) = \sum_{i=1}^{N_g} F_i(P_{gi})$$
$$F_i(P_{gi}) = \begin{cases} a_{i1} \times P_{gi}^2 + b_{i1} \times P_{gi} + c_{i1} & \textit{fuel type } 1 \\ a_{i2} \times P_{gi}^2 + b_{i2} \times P_{gi} + c_{i2} & \textit{fuel type } 2 \\ \vdots & \\ a_{iz} \times P_{gi}^2 + b_{iz} \times P_{gi} + c_{iz} & \textit{fuel type } z \end{cases} \quad (2)$$

where, $\{a_{i1}, ..., a_{iz}\}, \{b_{i1}, ..., b_{iz}\}, \{c_{i1}, ...c_{iz}\}$ being the cost coefficients of ith generator with a fuel type of $1, 2, ..., z$.

The ED problem with non-convex fitness function

When valves are opened to increase electrical power, it introduces mechanical losses which in turn brings about non-convexity and non-smoothness to the ED model. In order to take the network losses as well as POZs into account, we include them as constraints in the optimization problem. In addition, the valve point effect (VPE) effect is modeled by adding a sinusoidal term to the cost function [25]. The developed ED problem with the VPE effect is formulated in (3). Figure 2 represents the non-convex cost function with four VPEs.

$$\begin{aligned} &\text{Min } H(X) = \sum_{i=1}^{N_g} F_i(P_{gi}) \\ &F_i(p_{gi}) = a_i \times P_{gi}^2 + b_i \times P_{gi} + c_i + |e_i \times \sin(f_i \times (P_{gi\,\min} - P_{gi}))| \\ &X = [P_{g1}, P_{g2}, P_{g3}, \ldots, P_{gN_g}] \end{aligned} \quad (3)$$

Constraints

Power balance constraint

Supplying demanded load is one of the most important goals in energy resource management and allocation [26] and ED problems. In literature, this constraint is called the power balance constraint and is expressed as

$$\sum_{i=1}^{Ng} P_{gi} = P_D + P_L \quad (4)$$

where P_D denotes the total demanded load. Moreover, P_L is the total transmission network losses and can be approximated by

$$P_L = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} P_{gi} B_{ij} P_{gj} + \sum_{i=1}^{Ng} B_{0i} P_{gi} + B_{00} \quad (5)$$

Prohibited operating zones constraint

In practical situations, all units should not generate output power during POZ intervals. In fact, generating power in POZs may damage generators. The cost function with two POZs is shown in Fig. 3.

To apply this constraint, the output power between POZ intervals is fixed to the near boundary of the POZ

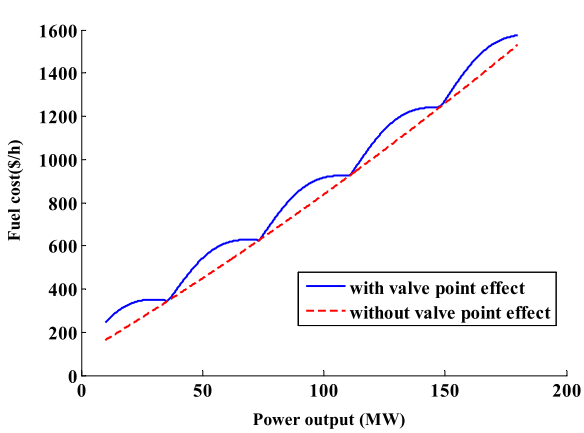

Fig. 2 The effect of VPEs on ED curve [4]

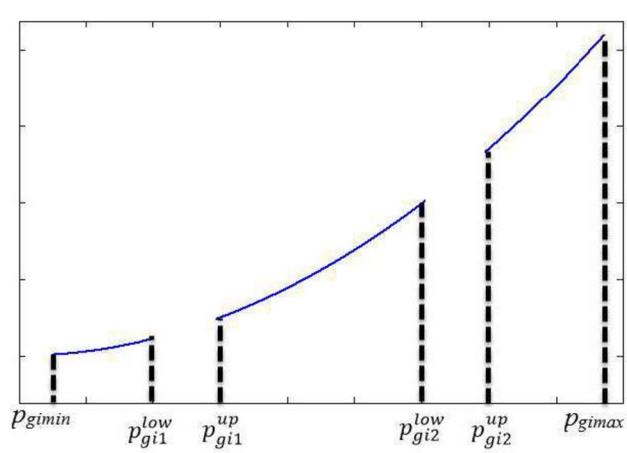

Fig. 3 The cost function with two POZs [4]

interval. The detailed expression of the POZ constraint is given by

$$P_{gi} = \left\{ \begin{array}{l} P_{gi\,\min} \le P_{gi} \le P_{gi,l-1}^{Low} \\ P_{gi,l-1}^{Up} \le P_{gi} \le P_{gi,l}^{Low} \\ . \\ . \\ . \\ P_{gi,L_i}^{Up} \le P_{gi} \le P_{gi\max} \end{array} \right\} l = 2,3,...,L_i \quad i = 1,2,...,N_g \tag{6}$$

$$i = 1,2,...,N_g$$

Output power constraint

The output power of each generator is bounded to certain values as in (7).

$$P_{gi\ \min} \le P_{gi} \le P_{gi\ \max} \tag{7}$$

Modified JAYA algorithm

In order to effectively introduce the MJAYA algorithm and showcase its advantages, we first provide a brief overview of the conventional JAYA algorithm.

Original JAYA Algorithm

JAYA is a Sanskrit word meaning victorious [23]. In JAYA, as a new and simple algorithm, the particles move towards the best solution and retreat from the worst one in each iteration, until the optimal solution is eventually achieved [22, 26]. The main advantages of this algorithm are simple implementation, low computational complexity, and the ability to work without control parameters [23]. This algorithm changes the position of candidate solutions by

$$\begin{aligned} X_{s,t,k}^{new} = X_{s,t,k} + r_{1,s,t}\,(X_{s,t,best} - |X_{s,t,k}|) \\ - r_{2,s,t}\,(X_{s,t,worst} - |X_{s,t,k}|) \end{aligned} \tag{8}$$

where $X_{s,t,best}$ and $X_{s,t,worst}$ are the best and worst solutions of $H(X)$ in the kth iteration, respectively.

Modified JAYA algorithm

As mentioned previously, power balance constraints, increasing the dimension of power network decision variables, and other critical complications, hinder the JAYA algorithm in performance and quality. Therefore, in order to circumvent the aforementioned drawbacks, three changes need to be applied to the conventional JAYA algorithm. The proposed modifications are presented in the following subsections.

Increment of the algorithm ability

Unlike the JAYA algorithm, the proposed algorithm checks all possible moving states for i^{th} solution $\left(X_{i,k}^{new}\right)$ according to (9),

$$X_{i,k}^{new} = \begin{pmatrix} X_{1,i,k}^{new} = X_{i,k} + r_1\,(X_{best} - |X_{i,k-1}|) - r_2\,(X_{worst} - |X_{i,k-1}|) \\ X_{2,i,k}^{new} = X_{i,k} + r_3\,(X_{best} - |X_{i,k-1}|) + r_4\,(X_{worst} - |X_{i,k-1}|) \\ X_{3,i,k}^{new} = X_{i,k} - r_5\,(X_{best} - |X_{i,k-1}|) - r_6\,(X_{worst} - |X_{i,k-1}|) \\ X_{4,i,k}^{new} = X_{i,k} - r_7\,(X_{best} - |X_{i,k-1}|) + r_8\,(X_{worst} - |X_{i,k-1}|) \end{pmatrix}. \tag{9}$$

In addition, an augmented matrix is produced as follows

$$\tilde{X}_{i,k}^{new} = \begin{pmatrix} X_{i,k}^{new} \\ X_{i,k-1} \end{pmatrix}. \tag{10}$$

Consequently, the cost vector of the achieved solution in the kth iteration is calculated.

$$\cos t_{i,k}^{*} = \begin{pmatrix} \cos t_{1,i,k}^{new} \\ \cos t_{2i,k}^{new} \\ \cos t_{3,i,k}^{new} \\ \cos t_{4,i,k}^{new} \\ \cos t_{5,i,k}^{new} \end{pmatrix} \tag{11}$$

Then, the corresponding $\tilde{X}_{i,k}^{new}$ to the least cost is chosen as the ith solution. The corresponding pseudo-code for MJAYA algorithm is provided in Algorithm 1, lines 3, 14 and 15.

Increment of convergence speed

One of the important factors determining the convergence speed of algorithms is the number of swarm populations. An algorithm may not find the optimum solution if a very small population size is chosen. In addition, increasing the size of the population prolongs the execution time. In fact, there is a tradeoff between the accuracy of the optimal solution and the speed of the convergence. Therefore, it is recommended to select the population size, adaptively. We can start with small swarm sizes, due to the inherent diversity of the population in the initial stage. After a few iterations, the resemblance among the swarms will increases and we can add new swarms at any iteration. This can guarantee that the algorithm never remains in local minima. To increase the effectiveness and reduce the execution time of the MJAYA algorithm, the population size can be determined using (12). The corresponding pseudo-code is provided in Algorithm 1 lines 17 and 18.

$$\begin{aligned} \Delta N &= N_{\max} - N_{\min} \\ N &= round\left(\frac{\Delta N \times iteration\ number}{Max\ iteration} + N_{\min}\right) \end{aligned} \tag{12}$$

In Eq. 12, the lowest and highest population sizes are $N_{\min}$ and $N_{\max}$, respectively. In addition, N represents the size of the population that is adopted in each iteration.

Increment of algorithm effectiveness

The mutation operator is another important strategy to skip local minima. The mutation operator changes the position of the candidate solutions and almost boosts the population search in the whole space of the problem. Like population size, it is better to apply less mutation in the beginning and more at the end of the optimization procedure to achieve better performance and higher convergence speed. Therefore, three mutated solutions for the ith solution are provided in (13).

$$\begin{aligned} X^1_{mutated} &= X_{b_1} + rand \times (X_{b_2} - X_{b_3}) \\ X^2_{mutated} &= X^1_{mutated} + rand \times (X_{best} - X_{worst}) \\ X^3_{mutated} &= X_{b4} + rand \times (X_{best} - X_{b_5}). \end{aligned} \tag{13}$$

where $b_1 \neq b_2 \neq b_3 \neq b_4 \neq b_5$ are random constants and unequal to b_i. Additionally, X represents the decision variable, here the generation output. *rand* is also denoting a random amount. The cost functions of the mutated and the ith solutions are evaluated and the one with the lowest value is substituted for the ith solution according to the pseudo-code in Algorithm 1, lines 4–15,

Proposed Modified JAYA

```
1) while iteration<maximum iteration
2)    for i=1:N
3)       select i^th solution and generate four solutions according to equation 9
4)       apply mutation operator for i^th selected solution as (11) and apply this as following
5)       if iteration< 1/3 × (maximum iteration)
6)          select one of mutated solution randomly
7)       else if 1/3 × (maximum iteration) <iteration< 2/3 × (maximum iteration)
8)          select two mutated solutions randomly
9)       else
10)         select all mutated solutions
11)      end
12)      end else if
13)   end else
14)   evaluate cost functions of i^th, mutated and the mentioned four solutions
15)   the least cost function is chosen as the i^th solution
16)   end for
17)   change the population size according to (10)
18)   generate random solutions for new population
19) end while
```

Algorithm 1. The pseudo-code of the MJAYA algorithm

Application of the MJAYA algorithm in solving economic dispatch problem

In this section, the application of the proposed method to solve the ED problem, an important problem in energy optimization, is described in detailed steps.

Step 1: Define input parameters

The generator data, fuel cost coefficients, POZs intervals of each unit, the amount of demanded load, and loss coefficients are defined.

Step 2: Initialize the population

A random population is initialized using (14) and (15).

$$P_{rand} = \begin{bmatrix} p_{g1,1} \cdots p_{g1,Ng} \\ p_{g2,1} \cdots p_{g2,Ng} \\ \cdot \\ \cdot \\ \cdot \\ p_{gN_{swarm},1} \cdots p_{gN_{swarm},Ng} \end{bmatrix} = \begin{bmatrix} P_{g1} \\ P_{g2} \\ \cdot \\ \cdot \\ \cdot \\ P_{gNg} \end{bmatrix}. \tag{14}$$

$$P_{gi} = P_{gi\min} + rand(1,n) \times (P_{gi\max} - P_{gi\min}), \tag{15}$$

where P_{gi} denotes the ith candidate solution vector for the generated power. Additionally, for each generator, $P_{gi\min}$ and $P_{gi\max}$ represent the minimum and maximum boundaries of generation.

Step 3: Satisfy the power balance constraint

In order to satisfy the active power and load balance requirements, the power balance constraint must be met [27]. In Fig. 4, a flowchart is portrayed where this constraint is satisfied eventually.

Step 4: Evaluate the fitness function

We integrate the constraints into the cost function using the method of Lagrangian multipliers. Therefore, using Lagrangian duality, the dual problem of the original optimization problem (2) can be given by.

$$Min\, L(X,\lambda,u) = F(X) + \sum_{i=1}^{m} \lambda_i \times h_i(X) + \sum_{j=1}^{p} \mu_j \times g_j(X) \tag{16}$$

where, λ_i and μ_j are Lagrangian multipliers. In addition, $h_i(X)$ and $g_j(X)$ are equality and inequality constraints, respectively. Note that, in (16), there are m equality and p inequality constraints integrated into the cost function.

In ED problems, the generated power can be controlled manually through a variable. Hence, if the suggested solution by MJAYA is not between the minimum/maximum values, the output power is set to the minimum/maximum boundary. In addition, considering the fact that the equality constraint has already been satisfied through step 3, the cost function is obtained as

$$Min\ L(X,\lambda,u) = F(X). \tag{17}$$

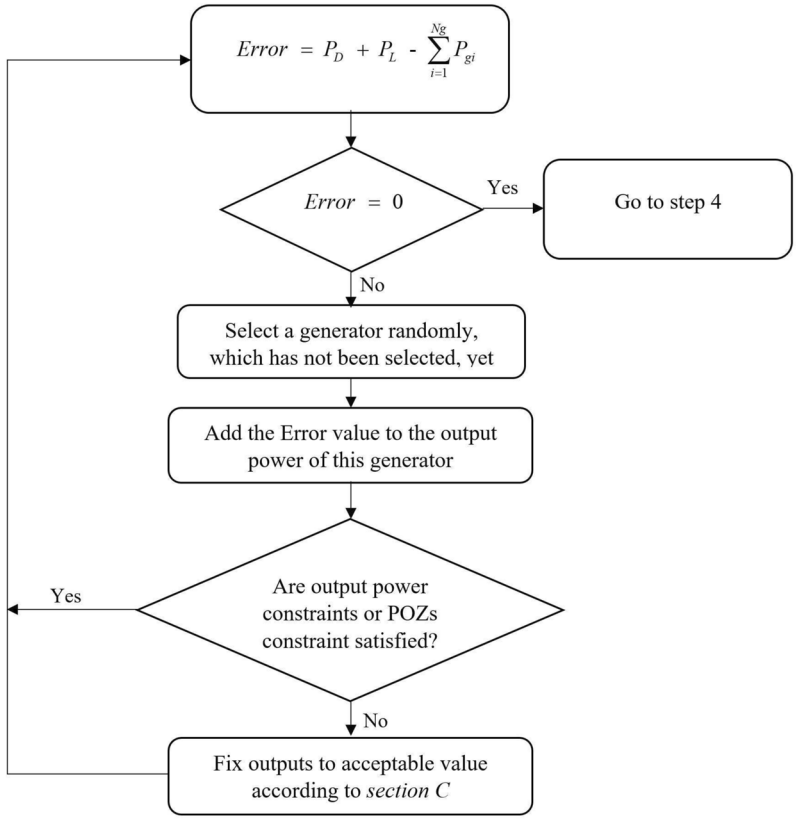

Fig. 4 The flowchart of equality constraint satisfaction

Step 5: Obtain the worst and best solutions

In this step, the solution with minimum/maximum cost is selected as the best/worst solution.

Step 6: Apply the MJAYA to the particles according to the pseudo-code represented in Algorithm 1
Step 7: Satisfy the Equality constraint as in Step 3
Step8: Evaluate the fitness function for new solutions
Step 9: Update the best and worst solutions

In each iteration, the best-obtained solution is considered the global optimal solution.

Step 10: iteration = iteration + 1.

If the convergence condition or maximum iteration is reached, then the process is terminated; otherwise, the algorithm returns to step 6.

Step 11: Print the best cost for the best power output for each generator

Simulation results

In this section, we study the effectiveness of the MJAYA algorithm by evaluating the energy optimization problem in the ED problem, for five different cases. For a quick reference, these cases are given in Table 1. Moreover, a PC with the following specification is utilized.

RAM: 8 GB, CPU: 2.6 GHz.

Case I: 6 generators with and without loss

This system contains six generators without a valve point effect. The demanded load is 1263 MW and the loss coefficients of this system are presented in (18). In addition, the POZs information is presented in Table 2. The obtained optimum solutions by MJAYA, JAYA, TLBO, PSO, GA, DE, and TS algorithms for the system with and without losses are presented and compared in Tables 3 and 4, respectively.

Table 1 Simulation case studies

Case No	Simulation/System setup
1	6 generators with and without loss
2	10 generators, which are fed by multiple types of fuels. VPE is included, as well
3	13 generators with 1800 MW and 2520 MW load demands, in two states
4	40 generators with 10500 MW load demand
5	large scale systems with 80 and 120 power-generating units in two states

Table 2 POZs for case I

Generators	Zone I	Zone II
1	[210–240]	[350–380]
2	[90–110]	[140–160]
3	[150–170]	[210–240]
4	[80–90]	[110–120]
5	[90–110]	[140–150]
6	[75–85]	[100–105]

Table 3 The obtained optimum results for 6 generators without loss

Method	Best ($)	Worst ($)	Mean ($)
MJAYA	**15,276.894**	**15,276.894**	**15,276.894**
JAYA	15,288.8565	15,288.8565	15,288.8565
TLBO	15,292.755	15,292.755	15,292.755
PSO	15,292.755	15,292.755	15,292.755

Table 4 Optimum results for 6 generators considering losses

Method	Best ($)	Worst ($)	Mean ($)
MJAYA	**15,443.7967**	**15,443.7967**	**15,443.7967**
JAYA	15,454.6673	15,454.6673	15,454.6673
GA [28]	15,469	15,524	15,469
TS [19]	15,454.89	15,498.05	15,472.56
DE [28]	15,449.766	15.449.874	15,449.77

$$B_{ij} = \begin{bmatrix} 0.0017 & 0.0012 & 0.0007 & -0.0001 & -0.0005 & -0.0002 \\ 0.0012 & 0.0014 & 0.0009 & 0.0001 & -0.0006 & -0.0001 \\ 0.0007 & 0.0009 & 0.0031 & 0.0000 & -0.0010 & -0.0006 \\ -0.0001 & 0.0001 & 0.0000 & 0.0024 & -0.0006 & -0.0008 \\ -0.0005 & -0.0006 & -0.001 & -0.0006 & 0.0129 & -0.0002 \\ -0.0002 & -0.0001 & -0.0006 & -0.0008 & -0.0002 & 0.015 \end{bmatrix}$$
$$B_{io} = 0.001 \times [-0.3908 - 0.12970.70470.05910.2161 - 0.6635]$$
$$B_{oo} = 0.056 \tag{18}$$

As mentioned previously, the most important goal in ED problems is supplying demanded load while the power network loss is considered. The exact value of losses must be calculated by solving power flow equations. However, in ED problems, this is estimated by (5). It is obvious that the generated power variation changes the net loss. Hence, it is important to find a proper output in the way that the power balance constraint is met while the cost function is minimized. As it can be seen in Table 3, the MJAYA algorithm outperforms the conventional JAYA and the other methods. The reason is mainly due to the modifications applied to the JAYA algorithm. In fact, the proposed MJAYA algorithm performs better, in not getting stuck in local minima, compared to JAYA.

Moreover, Table 4 represents a comparison between MJAYA, JAYA, GA, TS, and DE algorithms in 6 generators system considering losses. It has been shown that the performance of MJAYA is better than the other algorithms. It should be noted that computational time to extract results for JAYA algorithm is 1.35 s, while it is 0.96 s for the proposed modified JAYA algorithm.

Table 5 The comparison of MJAYA and other methods result in case II

Power generation (MW)	MJAYA	JAYA	CGA_MU [22]	IGA_MU [22]	Fuel type
P1	218.59	220.52	222.01	219.12	2
P2	211.66	209.18	211.63	211.16	1
P3	280.65	280.73	283.94	280.65	1
P4	239.50	239.48	237.80	238.47	3
P5	279.93	279.94	280.44	276.41	1
P6	239.63	239.23	236.03	240.46	3
P7	287.72	288.00	292.04	287.73	1
P8	239.63	240.84	241.97	240.76	3
P9	426.77	426.19	424.20	429.33	3
P10	275.86	275.84	269.90	275.85	1
Cost ($/h)	**623.83**	**624.07**	**624.71**	**654.57**	

Case II: 10 multi-fuel generators with 2700 MW load demand

The ED problem with multi-fuel generators is analyzed in the proposed method. In this case, the cost coefficients of generators are changed due to the utilization of different sorts of fuels such as coal, natural gas, and oil. It is necessary to select the appropriate fuel type to have an economically efficient cost.

The current system contains 10 generators and the demanded load is 2700 MW. The first generator is fed by two types of fuels while the others are fed by three types of fuels [22]. Table 5 provides the results of the MJAYA and JAYA algorithms in this case and the results are compared to the proposed methods in [22], as well.

It can be seen in Table 5 that JAYA and MJAYA's results are reasonable. However, MJAYA yields the optimal solution which depicts the effectiveness and advantages of MJAYA modifications. In addition, the convergence curves for both methods are shown in Fig. 5 to illustrate the enhancement of MJAYA compared to JAYA. As we can see, MJAYA converges faster and smoother compared to the conventional JAYA.

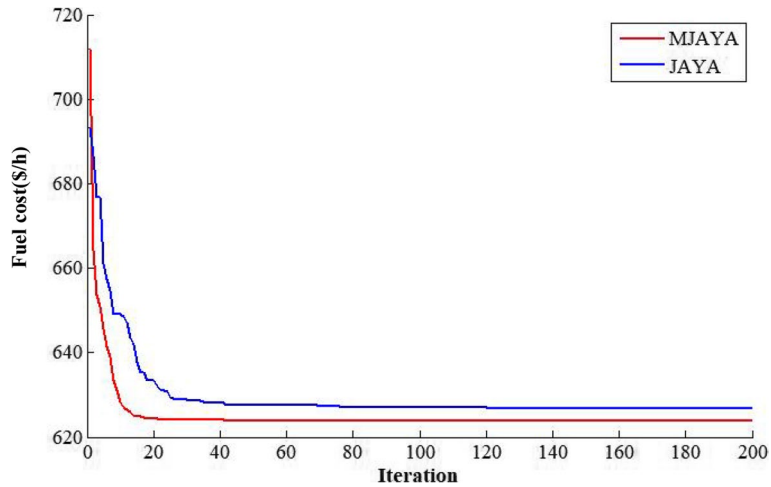

Fig. 5 The convergence curve of JAYA and MJAYA for case III

Furthermore, MJAYA obtains the optimum cost function by about 40 iterations; while JAYA achieves a sub-optimal solution in higher iteration (i.e., 120 iterations). Therefore, the proposed modifications in MJAYA are effective to increase the speed and accuracy of the algorithm. It should be noted that computational time to extract results for JAYA algorithm is 3.85 s, while it is 2.74 s for the proposed modified JAYA algorithm.

Case III: 13 generators with 1800 MW and 2520 MW load demands

There are thirteen generators in this case, while VPE is taken into account. The generators data for this system is given in [29]. This system has many local minima and some of the algorithms are stuck in one of these points [16 17, 18, 30]. To examine the performance of the proposed method, it is applied to this system with two different demanded load amounts, namely 1800 MW and 2520 MW. Moreover, Tables 6 and 7 present a comparison of MJAYA's results and some well-known methods, respectively.

It has been shown in Tables 6 and 7 that the worst cost obtained by the proposed algorithm is lower than the best solution obtained by the other algorithms. In addition, the obtained worst, mean and best costs by MJAYA are close to each other. In other words, the variance of the costs is low, meaning that the proposed algorithm is proper enough to be utilized in energy optimization. It should be noted that computational time to extract results for the MJAYA algorithm are 3.56 and 2.49 s, respectively.

Table 6 Results for case III with 1800 MW load demand

Method	Best ($)	Worst ($)	Mean ($)
MJAYA	**17,963.85**	**17,963.88**	**17,963.86**
PSO [16]	18,030.72	18,205.92	18,401.35
CEF [31]	18,048.21	18,190.23	18,404.04
EP-SQP [17]	17,991.03	18,106.93	-

Table 7 Results for case III with 2520 MW load demand

Method	Best ($)	Worst ($)	Mean ($)
MJAYA	**24,169.50**	**24,169.50**	**24,169.50**
PSO [22]	24,262.73	24,271.92	24,277.81
GA-SA [17]	24,275.71	-	-
UHGA [17]	24,172.25	-	-
ESO [21]	24,179.59	-	-

Case IV: 40 generators with 10500 MW load demand

This system contains 40 generators considering VPEs. The total demanded load is 10,500 MW and the data for this system is given in [18]. This system includes more elements of creating non-convexity in the problem, meaning that there are more local minima compared to the previous case studies. This case is of great importance and very close to practical energy resource management and allocation scenarios. MJAYA algorithm is applied to this case to compute the optimum solution. Results are shown and compared to the well-known algorithms suggested by [16, 17], in Table 8.

Table 8 The results for 40 generators with 10,500 MW load demand

Method	Best ($)	Mean ($)	Worst ($)
MJAYA	**121,412.535**	**121,414.66**	**121,417.1992**
JAYA	121,841.481	121,928.839	122,054.950
PSO [17]	123,930.45	124,154.49	-
MPSO [17]	122,252.27	-	-
UHGA [17]	121,424.48	121,602.81	-
DE [16]	121,416.29	121,422.72	121,431.47
HDE [16]	121,698.51	122,304.3	-

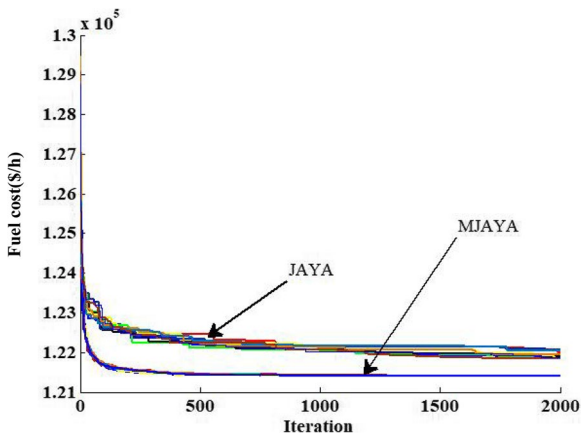

Fig. 6 The convergence characteristics of MJAYA and JAYA algorithms for 10 independent runs in case IV

As shown in Table 8, the best value obtained by MJAYA is better than the obtained values by other methods, meaning that MJAYA outperforms the other algorithms in such cases. Furthermore, Fig. 6 shows the convergence curve of JAYA and MJAYA algorithms for 10 independent runs. It should be noted that computational time to extract results for JAYA algorithm is 52 s, while it is 46.9 s for the proposed modified JAYA algorithm.

Robustness is another important feature that an algorithm must possess. In fact, an algorithm is robust if the best-obtained solutions in each trial run are close to each other. As shown in Fig. 6, the best costs achieved by MJAYA in trial runs are almost the same, meaning that the MJAYA algorithm is robust enough to handle complex energy optimization problems.

In order to check the effectiveness of the MJAYA algorithm in considering the output power constraint in the ED problem, Case IV is selected. Figure 7 shows the output power, minimum, and maximum generation limit of each unit in this case.

Case V: 80 and 120 generation units

Most of the existing algorithms for solving ED problems in large dimension problems are either unable to achieve the optimum solution or do not converge to the optimal solution with a reasonable number of iterations. To this end, the performance of the MJAYA algorithm in larger systems containing 80 and 120 generators is tested. The data of the generators and demanded load are double and triple the data of the 40 generators system (*Case IV*), respectively. The obtained results by MJAYA methods are shown in Table 9. It is seen that

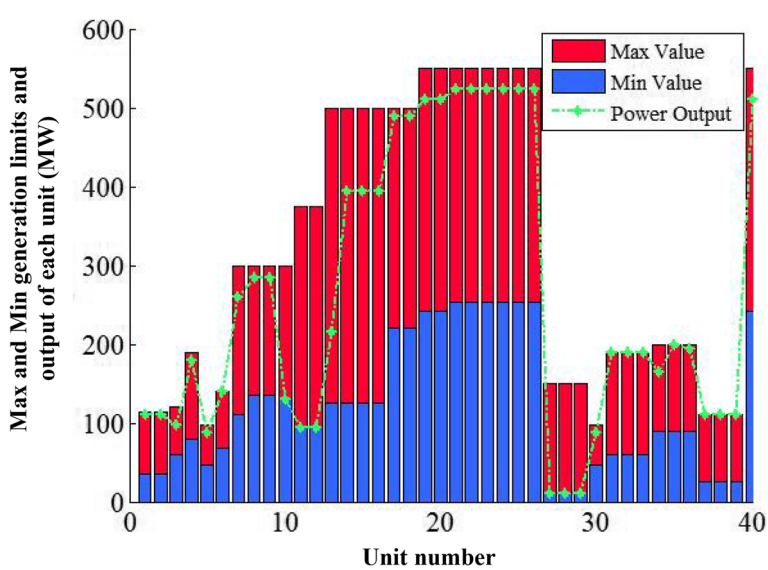

Fig. 7 The output power of each unit obtained from the MJAYA algorithm in case III

Table 9 The optimal solution for 80 & 120 generation units

No. of units	Best ($)	Mean ($)	Worst ($)
40 units	121,412	121,414	121,417
80 units	242,805.5709	242,807.65	242,811.2388
120 units	364,207.2382	364,215.28	364,223.3237

the proposed MJAYA algorithm does not suffer from enlarging the dimension.

The MJAYA algorithm benefits from three modifications which are mentioned in section III. In order to demonstrate the effect of each modification, Table 10 provides a comparison between them in the case when the number of generators is 80. In the first state, JAYA is implemented without any modification (JAYA1). The second stage is the first modification which is applied to JAYA (JAYA2). The mutation operator is implemented in the third state (JAYA3) while all modifications are applied to JAYA (MJAYA) in the last state.

As presented in Table 10, each of the obtained solutions in each state is reasonable. However, MJAYA's result is better than other states. The convergence characteristics of each state and also MJAYA for *case V* are shown in Fig. 8.

It is evident from Fig. 8 that by utilizing all modifications the optimal solution can be achieved in fewer iterations.

Table 10 The comparison of MJAYA's result and three modifications considered in the MJAYA algorithm

Method	JAYA 1	JAYA 2	JAYA 3	MJAYA
80 unit	249,578.9053 ($)	247,172.7983 ($)	244,878.953 ($)	242,805.5709 ($)

Discussion

According to the introduction section, solving complex optimization problems is a fundamental challenge for the JAYA algorithm. As a result, various papers seek to improve the performance and efficiency of the JAYA algorithm. The majority of papers recommend combining this algorithm with other optimization methods, or processes [32–35] that increase local search capability. In general, combining two methods causes complexity and often requires parameter setting. Despite obtaining the optimal results, this solution typically slows convergence, increases complexity, and increases calculation time. Thus, in this paper, we aim to improve the algorithm capability, convergence speed, and efficiency of the JAYA algorithm through some simple but straightforward modification process (see Sect. 3). In the proposed modification, no control parameters are required for the improvement, and the solution is faster than the combined algorithms [6–20]. It is also important to note that these modifications can be used for any other optimization algorithm and increase its efficiency. This is why the proposed method is new, and the results also prove it.

Future trend

The modified optimization algorithm proposed in this paper can be applied in a wide range of applications requiring high-performance computational intelligence techniques. The applications cover a wide range of real-world uses, including healthcare systems, industrial platforms, and smart cities. With this optimizer, it is possible to implement applications through IoT, edge/fog, and cloud architectures [36–45]. The proposed algorithm can be also used in other areas of computational sciences such as remote sensing systems and control systems, where there is an advanced problem of weight optimization [46–49].

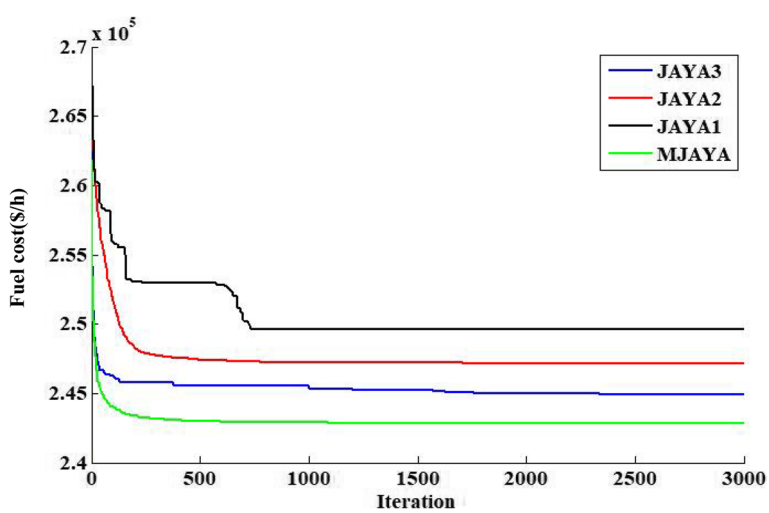

Fig. 8 The Convergence Characteristics for Different Modifications for Case V

Conclusion

The ED problem is one of the important problems in the energy optimization area which tries to supply the net demanded load with minimum fuel cost. In this paper, we considered some complicated and practical constraints such as VPEs, POZs, and multi-fuel generators. These constraints make ED problems more complex from a non-convexity and non-smoothness point of view. Hence, a new modified JAYA algorithm was introduced in this paper to handle such a complicated problem. Roughly speaking, the effectiveness and ability of MJAYA were increased by considering all moving possible states and adding a mutation operator. In addition, the convergence speed of the proposed algorithm was enhanced by adaptively utilizing population size. The MJAYA's results were compared with some well-known meta-heuristic algorithms in ED problems. The simulation results showed that the proposed MJAYA algorithm has a better performance in terms of accuracy and robustness of convergence as well as the capability of finding the optimal solution. The performance and effectiveness of this method were also demonstrated in different case studies of ED problems. It was shown that the proposed algorithm can be used to solve any constrained and unconstrained energy optimization problems.

Numerical results showed that the proposed algorithm has advantages such as simplicity, robustness, ability to work without control parameters, and capability in dealing with constraints irrespective of their difficulties.

Nomenclature

Indices

E Swarm index
i,j Generating unit indices
k Iteration index
l Prohibited operating zone index
s Decision variables index
t Candidate solution index

Constants

a_i, b_i, c_i, e_i, f_i Cost coefficients of ith generator
B_{ij} Loss coefficient associated with the production of ith and j^{th} generators
B_{0j} Loss coefficient associated with the production of ith generator
B_{00} Loss coefficient parameter (MW)
L_i Number of prohibited operating zones for i^{th} generator
N_g Number of power plants
N *Swarm size*
$P_{gi\ min}$ The lowest output power of ith generator (MW)
$P_{gi\ max}$ The highest output power of ith generator (MW)
$P_{gi,l}^{Low}$, $P_{gi,l}^{Up}$ Lower and upper bounds of the lth POZ for i^{th} generator
Rand(1,n) Uniformly distributed random vector of size 1by n (1×n)
$r_{1,s,t}$, $r_{2,s,t}$ Uniformly distributed random numbers

Variables

F(pg) Generating unit cost function
H(X) Objective function
P_{gi} The output of ith unit(MW)
$P_{ge,i}$ The output of ith unit for e^{th} swarm (MW)
$X_{s,t,k}$ Value of thevariable for the candidate during theiteration

Acronyms

ED Economic dispatch
VPE Valve point effect
POZ Prohibited operating zones

Acknowledgements
Acknowledgement: Key Research Project of Natural Science in Colleges and Universities of Anhui Provincial Department of Education (KJ2021A1163); Anhui Xinhua University—Intelligent Automation Research Institute (yjs202104); Major Teaching and Research Project of Quality Engineering Project of Education Department of Anhui Province, No. 2022jyxm649; Anhui Provincial Department of Education Quality Engineering Project School-Enterprise Cooperation Practice Education Base (2022xqhz028)

Authors' contributions
Z. B. wrote and revised the main manuscript; C. L. prepared Figs. 1, 2, 3, 4, 5, 6, 7 and 8; J. P. guided and reviewed the manuscript; X. Y. & D. L. prepared Tables 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10.

Funding
Not applicable.

Availability of data and materials
No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare no competing interests.

Author details
[1]Anhui Xinhua University, Hefei City 230088, China. [2]Weifang University of Science and Technology, Weifang City 262700, China. [3]Faculty of Administrative and Economics, Department of management, University of Isfahan, Isfahan, Iran. [4]Graduate School, Angeles University Foundation, 2009 Angeles City, Philippines.

Received: 7 January 2024 Accepted: 14 February 2024
Published online: 15 March 2024

References

1. Mokarram M, Mokarram MJ, Khosravi MR, Saber A, Rahideh A (2020) Determination of the optimal location for constructing solar photovoltaic farms based on multi-criteria decision system and Dempster-Shafer theory. Sci Rep 10(1):1–17
2. Mokarram M, Mokarram MJ, Gitizadeh M, Niknam T, Aghaei J (2020) A novel optimal placing of solar farms utilizing multi-criteria decision-making (MCDA) and feature selection. J Clean Prod 261(1):121098
3. Hu C, Wen G, Wang S, Fu J, Yu W (2023) Distributed multiagent reinforcement learning with action networks for dynamic economic dispatch. IEEE Trans Neural Netw Learn Syst 1:1–12
4. Mokarram MJ, Niknam T, Aghaei J, Shafie-Khah M, Catalão JPS (2019) Hybrid optimization algorithm to solve the nonconvex multiarea economic dispatch problem. IEEE Syst J 13(3):3400–3409
5. Chen CL, Wang SC (1993) Branch-and-bound scheduling for thermal generating units. IEEE Trans Energy Conversions 8(2):184–189
6. Dodu JC, Martin P, Merlin A, Pouget J (1972) An optimal formulation and solution of short-range operating problems for a power system with flow constraints. Proc IEEE 60(1):54–63
7. Wood AJ, Wollenberg BF, Sheblé GB, Power generation, operation, and control. Available: https://www.wiley.com/. Accessed 07 Aug. 2018
8. Jabr RA, Coonick AH, Cory BJ (2000) A homogeneous linear programming algorithm for the security constrained economic dispatch problem. IEEE Tranactions Power Syst 15(3):930–936
9. dos Santos Coelho L, Mariani VC (2006) Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. IEEE Trans Power Syst 21(2):989–996
10. El-Keib AA, Ma H, Hart JL (1994) Environmentally constrained economic dispatch using the LaGrangian relaxation method. IEEE Trans Power Syst 9(4):1723–1729
11. Mokarram MJ, Gitizadeh M, Niknam T, Niknam S (2019) Robust and effective parallel process to coordinate multi-area economic dispatch (MAED) problems in the presence of uncertainty. IET Gen Transmision Distribution 13(18):4197–4205
12. Li Z, Wu W, Zhang B, Sun H, Guo Q (2013) Dynamic economic dispatch using lagrangian relaxation with multiplier updates based on a quasi-newton method. IEEE Trans Power Syst 28(4):4516–4527
13. M. J. Mokarram, M. Nayeripour, T. Niknam and E. Waffenschmidt, "Multi-area economic dispatch considering generation uncertainty," in *2018 7th International Energy and Sustainability Conference (IESC)*, Cologne, Germany, pp. 1–6, 2018. https://doi.org/10.1109/IESC.2018.8439950.
14. Jayabarathi T, Raghunathan T, Adarsh BR, Suganthan PN (2016) Economic dispatch using hybrid grey wolf optimizer. Energy 111(1):630–641
15. Wang Y, Li B, Weise T (2010) Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems. Inf Sci 180(12):2405–2420
16. Niknam T (2010) A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem. Appl Energy 87(1):327–339
17. He D, Wang F, Mao Z (2008) Hybrid genetic algorithm for economic dispatch with valve-point effect. Electric Power Syst Res 78(4):626–633
18. Sinha N, Chakrabarti R, Chattopadhyay PK (2003) Evolutionary programming techniques for economic load dispatch. IEEE Trans Evol Comput 7(1):83–94
19. Pothiya S, Ngamroo I, Kongprawechnon W (2008) Application of multiple tabu search algorithm to solve dynamic economic dispatch considering generator constraints. Energy Convers Manage 49(4):506–516
20. Walters DC, Sheble GB (1993) Genetic algorithm solution of economic dispatch with valve point loading. IEEE Trans Power Syst 8(3):1325–1332
21. Pereira-Neto A, Unsihuay C, Saavedra OR (2005) Efficient evolutionary strategy optimisation procedure to solve the nonconvex economic dispatch problem with generator constraints. IEE Proc Gen, Transmission Distribution 152(5):653
22. Chiang C-L (2005) Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels. IEEE Trans Power Syst 20(4):1690–1699
23. VenkataRao R (2016) Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int J Industr Engine Comput 7(1):19–34
24. Niknam T, Firouzi BB, Mojarrad HD (2011) A new evolutionary algorithm for non-linear economic dispatch. Expert Syst Appl 38(10):13301–13309
25. Niknam T, Azizipanah-Abarghooee R, Aghaei J (2013) A new modified teaching-learning algorithm for reserve constrained dynamic economic dispatch. IEEE Trans Power Syst 28(2):749–763
26. Nikoobakht A, Aghaei J, Mokarram MJ, Shafie-khah M, Catalão JPS (2021) Adaptive robust co-optimization of wind energy generation, electric vehicle batteries and flexible AC transmission system devices. Energy 230(1):120781
27. Mokarram MJ, Gitizadeh M, Niknam T, Okedu KE (2021) A decentralized granular-based method to analyze multi-area energy resource management and allocation systems including DGs, batteries and electric vehicle parking lots. J Energy Storage 42(1):103128
28. Noman N, Iba H (2008) Differential evolution for economic load dispatch problems. Electric Power Syst Res 78(8):1322–1331
29. Aragón VS, Esquivel SC, CoelloCoello CA (2015) An immune algorithm with power redistribution for solving economic dispatch problems. Inform Sci 295(1):609–632
30. Niknam T, Mojarrad HD, Nayeripour M (2010) A new fuzzy adaptive particle swarm optimization for non-smooth economic dispatch. Energy 35(4):1764–1778
31. Wong KP (1994) Genetic and genetic/simulated-annealing approaches to economic dispatch. IEE Proc Gen, Trans Distribution 141(5):507
32. Alshammari BM, Anouar F, Khalid A, Tawfik G, Mihai O et al (2021) Robust design of dual-input power system stabilizer using chaotic jaya algorithm. Energies 14(17):5294
33. Ding Z, Li J, Hao H (2019) Structural damage identification using improved Jaya algorithm based on sparse regularization and Bayesian inference. Mech Syst Signal Process 132(1):211–231
34. Tu S, Muhammad W, Sadaqat UR, Talha M, Ghulam A et al (2021) Reinforcement learning assisted impersonation attack detection in device-to-device communications. IEEE Trans Veh Technol 70(2):1474–1479
35. Halim Z, Baig AR, Abbas G (2015) Computational intelligence-based entertaining level generation for platform games. Int J Comput Intel Syst 8(6):1128–1143
36. Zhou X, Liang W, Yan K, Li W, Wang K et al (2023) Edge-enabled two-stage scheduling based on deep reinforcement learning for internet of everything. IEEE Internet Things J 10(4):3295–3304
37. Zhou X, Hu Y, Wu J, Liang W, Ma J et al (2023) Distribution bias aware collaborative generative adversarial network for imbalanced deep learning in industrial IoT. IEEE Trans Industr Inf 19(1):570–580
38. Zhou X, Liang W, Li W, Yan K, Shimizu S et al (2022) Hierarchical adversarial attacks against graph-neural-network-based iot network intrusion detection system. IEEE Internet Things J 9(12):9310–9319
39. Zhou X, Xu X, Liang W, Zeng Z, Yan Z (2021) Deep-learning-enhanced multitarget detection for end-edge-cloud surveillance in smart IoT. IEEE Internet Things J 8(16):12588–12596
40. Zhou X, Liang W, Wang KIK, Yang LT (2021) Deep correlation mining based on hierarchical hybrid networks for heterogeneous big data recommendations. IEEE Trans Comput Soc Syst 8(1):171–178
41. Qi L, Yang Y, Zhou X, Rafique W, Ma J (2022) Fast anomaly identification based on multiaspect data streams for intelligent intrusion detection toward secure industry 4.0. IEEE Trans Industr Inf 18(9):6503–6511
42. Wu S, Shen S, Xu X, Chen Y, Zhou X et al (2022) Popularity-aware and diverse web apis' recommendation based on correlation graph. IEEE Trans on Comput Soc Syst 10(2):771–782
43. Dai H, Yu J, Li M, Wang W, Liu A et al (2022) Bloom filter with noisy coding framework for multi-set membership testing. IEEE Trans Knowl Data Eng 1(1):1–14
44. Kong L, Li G, Rafique W, Shen S, He Q et al (2022) Time-aware missing healthcare data prediction based on arima model. IEEE/ACM Trans Comput Biol Bioinf 1(1):1–10
45. Wang F, Wang L, Li G, Wang Y, Lv C et al (2022) Edge-cloud-enabled matrix factorization for diversified APIs recommendation in mashup creation. World Wide Web 25(5):1809–1829
46. Khosravi MR, Samadi S (2020) Reliable data aggregation in internet of ViSAR vehicles using chained dual-phase adaptive interpolation and data embedding. IEEE Internet Things J 7(4):2603–2610

47. Farhangkhah N, Samadi S, Khosravi MR, Mohseni R (2021) Overcomplete pre-learned dictionary for incomplete data SAR imaging towards pervasive aerial and satellite vision. Wireless Netw 1(1):1–13
48. Samuel M, Yahya K, Attar H, Amer A, Mohamed M et al (2023) Evaluating the performance of Fuzzy-PID control for lane recognition and lane-keeping in vehicle simulations. Electronics 12(3):724
49. Khosravi MR, Samadi S (2019) Data compression in ViSAR sensor networks using non-linear adaptive weighting. EURASIP J Wirel Commun Netw 2019(1):1–8

Publisher's Note