



ARTICLE

# Harnessing Trend Theory to Enhance Distributed Proximal Point Algorithm Approaches for Multi-Area Economic Dispatch Optimization

Yaming Ren<sup>1,\*</sup> and Xing Deng<sup>2</sup>

<sup>1</sup>College of Mechanical and Control Engineering, Guilin University of Technology, Guilin, 541006, China

<sup>2</sup>School of Computer, Jiangsu University of Science and Technology, Zhenjiang, 212003, China

\* Corresponding Author: Yaming Ren. Email: renyaming1981@gmail.com

Received: 18 October 2024; Accepted: 12 December 2024; Published: 06 March 2025

**ABSTRACT:** The exponential growth in the scale of power systems has led to a significant increase in the complexity of dispatch problem resolution, particularly within multi-area interconnected power grids. This complexity necessitates the employment of distributed solution methodologies, which are not only essential but also highly desirable. In the realm of computational modelling, the multi-area economic dispatch problem (MAED) can be formulated as a linearly constrained separable convex optimization problem. The proximal point algorithm (PPA) is particularly adept at addressing such mathematical constructs effectively. This study introduces parallel (PPPA) and serial (SPPA) variants of the PPA as distributed algorithms, specifically designed for the computational modelling of the MAED. The PPA introduces a quadratic term into the objective function, which, while potentially complicating the iterative updates of the algorithm, serves to dampen oscillations near the optimal solution, thereby enhancing the convergence characteristics. Furthermore, the convergence efficiency of the PPA is significantly influenced by the parameter  $c$ . To address this parameter sensitivity, this research draws on trend theory from stock market analysis to propose trend theory-driven distributed PPPA and SPPA, thereby enhancing the robustness of the computational models. The computational models proposed in this study are anticipated to exhibit superior performance in terms of convergence behaviour, stability, and robustness with respect to parameter selection, potentially outperforming existing methods such as the alternating direction method of multipliers (ADMM) and Auxiliary Problem Principle (APP) in the computational simulation of power system dispatch problems. The simulation results demonstrate that the trend theory-based PPPA, SPPA, ADMM and APP exhibit significant robustness to the initial value of parameter  $c$ , and show superior convergence characteristics compared to the residual balancing ADMM.

**KEYWORDS:** Multi-area economic dispatch problem; proximal point algorithm; trend theory

## 1 Introduction

Economic dispatch (ED) is a classical problem in the field of power systems [1,2]. Additionally, satisfying the corresponding constraints is a necessary condition for a feasible ED solution [3]. However, in actual power systems, the local energy supply often cannot meet the local demand for power energy. Taking China as an example, the southeast coastal area is one of the most densely populated and economically developed regions. This region, which accounts for 5.4% of China's land area, generates more than one-third of China's economic aggregate and 40% of its fiscal revenue. While this huge capacity implies a huge demand for energy, the energy produced in this region constitutes only 0.4% of the country's total energy. Concurrently, hydropower is concentrated in the southwest, solar and wind power in the northwest, and coal resources in the north and northwest of China. Consequently, China's power grid employs a multi-area interconnected



power network. This network allows the abundant coal and water resources in the southwest and northwest to be easily converted into electricity and transported to the economically developed southeast coastal areas. Specifically, multi-area interconnected power networks connect different regional power networks through tie lines [4].

Regarding the multi-area economic dispatch problem (MAED), traditional mathematical programming techniques [5] are ostensibly amenable to its resolution. However, given the substantial number of variables and constraints typically associated with the multi-area economic dispatch problem, a distributed approach is recommended. Zhang et al. successfully implemented a parallel solution for multi-area interconnected power systems, leveraging the combination of a regularized term and the primal-dual interior-point method [6]. Lu et al. introduced a comprehensively decentralized optimal power flow (OPF) algorithm tailored for multi-area interconnected power systems. This innovative approach relies on the distributed interior point method, effectively transforming the resolution of the regional correction equation into a parametric quadratic programming problem [7]. Ye et al. proposed a novel distributed solution strategy for multi-area AC power flow, predicated on the quadratic Taylor expansion [8].

Traditional mathematical programming methods have indeed showcased their efficacy in obtaining distributed solutions for the multi-area economic dispatch problem (MAED). However, when integrating non-differentiable and discontinuous factors such as the valve-point effect (VPE) [9], prohibited operating zone (POZ) [10], multi-fuel operation (MFO) [11], and ramp rate (RR) [12] into MAED, the intrinsic limitations of these conventional mathematical planning methods become apparent [13]. In such contexts, meta-heuristic optimization methodologies emerge as a superior alternative to traditional mathematical planning approaches, providing a more robust and flexible solution to manage the complexities introduced by these factors. Examples of such methodologies include black widow optimization [14], grey wolf algorithm [15], squirrel search algorithms [16], hybrid imperialist competitive algorithms combined with particle swarm optimization methods [17], particle swarm optimization [18], bat algorithms [19] and honey bee mating optimization [20]. Nonetheless, meta-heuristic optimization methods often lack explicit convergence criteria and necessitate adept handling of constraint conditions by users. Moreover, when confronting high-dimensional optimization problems, these methods typically demand larger populations and a greater number of iterations.

In the context of MAED, the presence of tie lines facilitates the transmission of electric energy between disparate regions. From a mathematical perspective, tie-line constraints are tantamount to coupling constraints between areas. In actual power networks, each regional power grid is under the jurisdiction of a specific power company, and the internal data of these regions are not disclosed to the public. In China, the power network comprises the State Grid Corporation of China (SGCC) and the China Southern Power Grid (CSG), which operate independently of each other. Similarly, the European Network of Transmission System Operators for Electricity (ENTSO-E) is a consortium consisting of 39 transmission system operator (TSO) members from 35 countries. Given national security concerns and competition among companies, implementing a fully distributed algorithm that exchanges tie-line information between adjacent power networks, rather than aggregating global information, is more apropos.

Tie-lines primarily serve to interconnect disparate regions and equilibrate power flows within the network, with these interconnections and balancing dynamics often being describable through linear equalities. Decomposition of the MAED problem into a constellation of sub-optimization problems based on tie lines for distributed resolution offers the flexibility to select bespoke optimization methodologies for each sub-problem, distinct from traditional mathematical programming methods and meta-heuristic optimization approaches. However, a salient challenge emerges in the orchestration of these sub-optimization problems to ensure that the solution derived from the distributed optimization algorithm is congruent with or closely

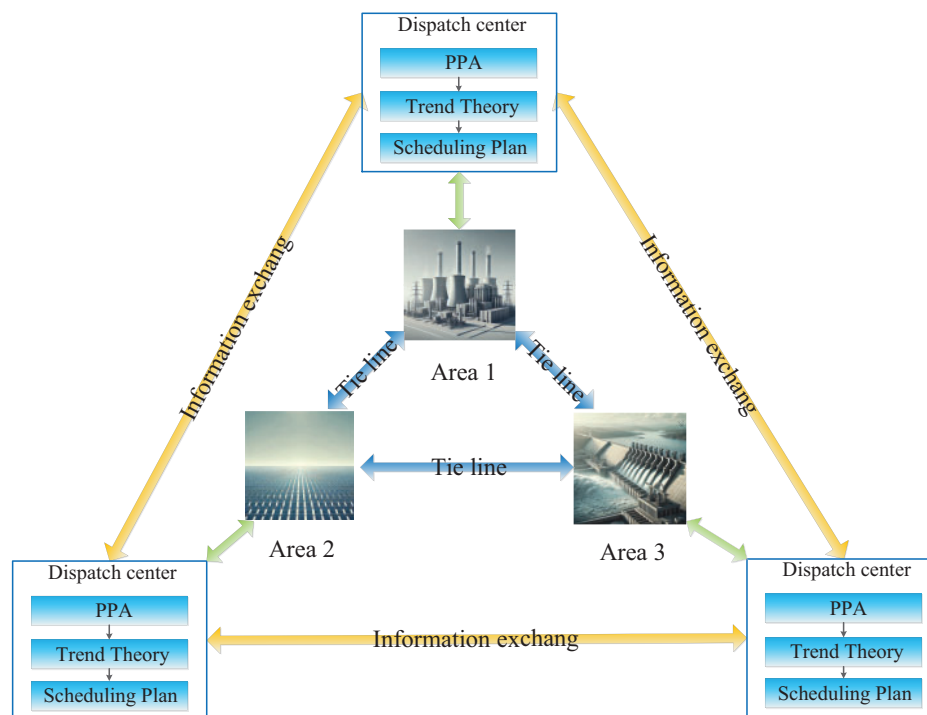
approximates the solution of the original optimization problem. The Alternating Direction Method of Multipliers (ADMM) has evinced an exceptional capacity to decompose large-scale optimization problems characterized by linear coupling constraints, leading to its broad application across various industries, including lower and upper bound limit analysis [21], robot trajectory optimization [22], matrix optimization [23], estimating the unknown source term in the time-fractional diffusion equation [24], graph matching [25], geometric inverse problems [26] and electromagnetic inverse scattering problems [27].

For multi-area interconnected power networks, tie-line constraints are often expressible as linear equality constraints. The Augmented Lagrangian method can conveniently incorporate these linear constraints into the objective function, introducing an indivisible quadratic term in the process. The Alternating Direction Method of Multipliers (ADMM) leverages a serial algorithm to achieve distributed computation of the quadratic term, thereby realizing a distributed solution for multi-area interconnected power networks. Wang et al. implemented a partitioned black-start and restoration methodology for power systems predicated on the ADMM [28]. Khojasteh et al. employed ADMM to ascertain optimal energy trading strategies for community participants [29]. Mak et al. proposed an innovative decentralized machine learning approach, namely ML-ADMM, to expedite the convergence velocity of ADMM when addressing the AC-OPF problem [30]. Compared to the serial ADMM, the parallel Auxiliary Problem Principle (APP) [31] has also demonstrated high efficiency in decentralization and has been applied in the power systems field, such as transmission investment [32], optimal power flow [33], and voltage optimization [34]. In this paper, the proximal point algorithm (PPA) is deployed to construct a theoretical framework for resolving the MAED. Within the PPA framework, devising distributed iterative strategies that align with specific requirements becomes straightforward by exploiting the problem's structure. Similar to ADMM and APP, the distributed PPA iterative strategy conveniently dissects the MAED problem into a sequence of optimization subproblems and Lagrange multiplier updates. For these optimization subproblems, users are at liberty to employ efficient solution techniques [35] tailored to actual exigencies. A notable advantage is that the convergence proof process for the PPA is remarkably straightforward, enabling a clear and concise understanding of its theoretical underpinnings. In contrast, the convergence proofs for ADMM and APP tend to be more intricate.

For the PPA, the parameter  $c$  denotes the update step of the Lagrangian operator during the iteration process and concurrently serves as the penalty parameter for the quadratic term introduced by the Augmented Lagrangian function. Clearly, the value of parameter  $c$  will inevitably impact the convergence efficiency of PPA. Similar to the PPA algorithm, ADMM also confronts the selection of parameter  $c$ . Reference [36] employs the concept of primal-dual residual balancing to adjust the parameter  $c$  to ensure the stability of the algorithm's efficiency. For the MAED problem, linear coupling only considers the coupling relationship between adjacent two areas, which is a convex two-partitionable optimization problem. For this specific form, the dual residuals of the ADMM algorithm can be expressed as a formula that includes the penalty parameter  $c$ , thus achieving primal-dual residual balance. However, the serial and parallel PPA algorithms proposed in this paper and APP algorithms, due to the introduction of quadratic terms (the quadratic terms introduced by the PPA proposed in this paper and APP algorithms are distinct), cannot express the dual residuals as a single formula containing the penalty parameter  $c$ , and therefore cannot use primal-dual residual balance for parameter adjustment. Recognizing that the pivotal nexus between centralized and distributed optimization problems lies in the linearly coupled constraints among different regions, there is an inherent aspiration for these constraints to converge rapidly toward zero during the iterative process. Drawing inspiration from the concept of "trend theory" observed in stock markets, this study proposes a novel parameter adjustment methodology, termed the trend-based parameter adjustment approach. Specifically, if the linearly coupled constraints are swiftly approaching zero during the iterative

process, no adjustments to the parameters are necessary. However, if the trend of the linearly coupled constraints deviates from the anticipated pattern, prompt adjustments to the parameters are warranted. The graphical representation of the method proposed in this paper is depicted in Fig. 1. The principal contributions of this research are delineated as follows:

- (a) Within the PPA framework, addressing the MAED problem, this paper achieves both efficient and flexible parallel and serial distributed iterative strategies through the meticulous design of the positive definite matrix  $G$ , thereby significantly simplifying the complexity of convergence analysis. Simulation results further confirm that, compared to traditional serial ADMM methods and parallel APP methods, the serial and parallel algorithms proposed in this paper based on PPA exhibit a significant advantage in convergence speed, reaching the optimal solution more rapidly.
- (b) This study introduces a novel parameter adjustment strategy: the trend-based approach. In contrast to the primal-dual residual balancing method derived from the ADMM method, which necessitates the calculation of both primal and dual residuals, the trend-based approach streamlines the process by focusing exclusively on the primal residuals, thus diminishing computational demands. Additionally, the trend-based method demonstrates improved adaptability, facilitating seamless integration with both ADMM and APP methods, a feature not shared by the primal-dual residual balancing method. Moreover, simulation results confirm that the trend theory method boasts superior convergence characteristics.



**Figure 1:** Graphical representation of the proposed approach

The subsequent sections of this paper are meticulously structured as delineated below, Within Section 2, the convergence validation of the Proximal Point Algorithm (PPA) is initially proffered. This is followed by the elucidation of parallel and serial distributed iterative methodologies engineered to address the

complexities of MAED. Section 3 introduces innovative strategies for the dynamic adjustment of parameters based on iterative feedback, thereby mitigating the deleterious effects of suboptimal parameter selection on algorithmic convergence. Section 4 encompasses a suite of specific simulation scenarios aimed at substantiating the efficacy and veracity of the proposed parallel and serial distributed iterative strategies for MAED. Section 5 encapsulates the salient conclusions drawn from the study.

## 2 Proximal Point Algorithm for MAED

### 2.1 Basic Properties

For a convex problem such as:

$$\min \{f(x) | x \in X\} \tag{1}$$

Solving Eq. (1) is equivalent to finding  $x \in X$  which satisfies the following variable inequality:

$$(x' - x) \nabla f(x) \geq 0, \forall x' \in X \tag{2}$$

where  $\nabla f(\cdot)$  denotes the gradient of  $f(\cdot)$ .

For solving Eq. (2), the proximal point algorithm can be expressed as follows:

$$(x' - x^{k+1}) \{ \nabla f(x^{k+1}) + M(x^{k+1} - x^k) \} \geq 0, \forall x' \in X \tag{3}$$

where  $M$  is positive definite matrix,  $x^{k+1}$  is generated by given  $x^k$ .

Let  $x^*$  be the solution of Eq. (1), when the sequence  $\{x^k\}$  satisfies

$$\|x^k - x^*\|_M^2 - \|x^{k+1} - x^*\|_M^2 \geq \alpha \|x^k - x^{k+1}\|_M^2, \alpha > 0 \tag{4}$$

where  $M$  is positive definite matrix,  $\|x\|_M = \sqrt{x^T M x}$  denotes  $M$ -norm of vector  $x$ . Then, the sequence contractive to  $x^*$ .

### 2.2 Proposed Method

In this paper, the mathematical model of the multi-area economic dispatch problem (MAED) can be found as follows:

$$\begin{aligned} &\min f(x_1) + g(x_2) \\ &s.t \ Ax_1 + Bx_2 = b \\ &\quad x_1 \in \Omega_1 \text{ and } x_2 \in \Omega_2 \end{aligned} \tag{5}$$

where  $f: R^{n_1} \rightarrow R$  and  $g: R^{n_2} \rightarrow R$  are closed differentiable convex functions.  $x_1 \in R^{n_1}$  and  $x_2 \in R^{n_2}$  are closed convex sets.  $A \in R^{l \times n_1}$  and  $B \in R^{l \times n_2}$  are given matrices, and  $b \in R^l$  is given vector. Eq. (5) is a classical two-region (region 1 and region 2) mathematical model of the MAED problem.  $x_1$  and  $x_2$  represent the active power output of the generator sets in region 1 and region 2.  $\Omega_1$  and  $\Omega_2$  represent feasible domains of variables  $x_1$  and  $x_2$  in region 1 and region 2, including upper and lower limit constraints of variables and active power balance constraints. In fact, region 1 and region 2 are connected by tie lines. Assuming that the active power flow traverses from region 1 to region 2, the equation constraint signifies that the power dispatched by region 1 is numerically equivalent to the power received by region 2.  $f(x_1)$  and  $g(x_2)$  represent the generation cost of regions 1 and 2, and the total generation cost is the sum of the generation cost of the two regions.

If Eq. (5) is strictly convex, the Lagrange function can indeed be employed to handle the equality constraints, thereby reducing Eq. (5) to the task of finding the subsequent saddle point. This transformation leverages the properties of strict convexity to ensure that the Lagrange function has a unique saddle point, which corresponds to the optimal solution of the original problem. The saddle point problem can then be solved using various optimization techniques that are designed to find such points, where the function value is both maximized with respect to the minimization variables and minimized with respect to the maximization variables.

$$(x_1^{k+1}, x_2^{k+1}) = \arg \min \{f(x_1) + g(x_2) - \lambda^{T,k} (Ax_1 + Bx_2 - b) \mid x_1 \in \Omega_1, x_2 \in \Omega_2\} \quad (6)$$

where  $c > 0$  is given fixed penalty factor for linear constraint.  $\lambda$  is the Lagrangian multiplier.

Then,  $\lambda^{k+1}$  is obtained by updating

$$\lambda^{k+1} = \lambda^k - c (Ax_1^{k+1} + Bx_2^{k+1} - b) \quad (7)$$

Eq. (6) is equivalent to solving the following variational inequality problem:

$$f(x_1) + g(x_2) - f(\tilde{x}_1^k) - g(\tilde{x}_2^k) + (u - \tilde{u}^k)^T \begin{pmatrix} -A^T \lambda^k \\ -B^T \lambda^k \end{pmatrix} \geq 0, \quad \forall x_1 \in \Omega_1, x_2 \in \Omega_2 \quad (8)$$

where

$$u = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad (9)$$

In general, the strict convexity of Eq. (5) cannot be ensured. Under such circumstances, equivalence between the solutions of the saddle point Eqs. (6) and (7) and the original Eq. (5) cannot be guaranteed. However, if Eq. (5) is assumed to be convex, the augmented Lagrange function can be employed to handle the equality constraints, thereby rendering Eq. (5) equivalent to resolving the subsequent saddle point issue.

$(x_1^{k+1}, x_2^{k+1})$  is obtained by solving

$$\min \left\{ \left\{ f(x_1) + g(x_2) - \lambda^{T,k} (Ax_1 + Bx_2 - b) + \frac{c}{2} \|Ax_1 + Bx_2 - b\|^2 \mid x_1 \in \Omega_1, x_2 \in \Omega_2 \right\} \right\} \quad (10)$$

where  $c > 0$  is given fixed penalty factor for linear constraint.  $\lambda$  is the Lagrangian multiplier. The Euclidean norm of vector  $x$  will be denoted by  $\|x\|$ , i.e.,  $\|x\| = \sqrt{x^T x}$ .

Then,  $\lambda^{k+1}$  is obtained by updating

$$\lambda^{k+1} = \lambda^k - c (Ax_1^{k+1} + Bx_2^{k+1} - b) \quad (11)$$

Similarly, Eq. (10) is equivalent to solving the following variational inequality problem [37].

$$f(x_1) + g(x_2) - f(x_1^{k+1}) - g(x_2^{k+1}) + (u - u^{k+1})^T \begin{pmatrix} -A^T \lambda^k + cA^T (Ax_1^{k+1} + Bx_2^{k+1} - b) \\ -B^T \lambda^k + cB^T (Ax_1^{k+1} + Bx_2^{k+1} - b) \end{pmatrix} \geq 0$$

$$\forall x_1 \in \Omega_1, x_2 \in \Omega_2 \quad (12)$$

where

$$u = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad (13)$$

For convenience, the following definition is given:

$$G = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix}, u = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, w = \begin{pmatrix} x_1 \\ x_2 \\ \lambda \end{pmatrix} \tag{14}$$

where  $G$  is a positive definite matrix.

Based on the description of Eqs. (12)–(14), for solving Eq. (10), the proximal point algorithm can be expressed as follows:

$$f(x_1) + g(x_2) - f(x_1^{k+1}) - g(x_2^{k+1}) + (u - u^{k+1})^T \times \left\{ \begin{pmatrix} -A^T \lambda^k + cA^T (Ax_1^{k+1} + Bx_2^{k+1} - b) \\ -B^T \lambda^k + cB^T (Ax_1^{k+1} + Bx_2^{k+1} - b) \end{pmatrix} + \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix} (u^{k+1} - u^k) \right\} \geq 0 \quad \forall x_1 \in \Omega_1, x_2 \in \Omega_2 \tag{15}$$

According to PPA, rather than solving problem (10) directly, an alternative method is employed by solving Eq. (15) to obtain the solution for  $u^{k+1}$ .

**Lemma 1.** Let sequence  $\{w^k\}$  be generated by the iterative scheme of PPA. Then, the following can be obtained:

$$\|w^k - w^*\|_M^2 - \|w^{k+1} - w^*\|_M^2 \geq \|w^k - w^{k+1}\|_M^2 \tag{16}$$

where  $w^*$  denotes the solution of Eq. (1), matrix  $M$  is denoted by

$$M = \begin{pmatrix} G & 0 \\ 0 & \frac{1}{c}I_l \end{pmatrix} \tag{17}$$

where  $I_l \in R^{l \times l}$  is identity matrix.

Proof: In fact, the proposed iterative scheme PPA can be explained as mixed variational inequality problem [38]:

Finding  $w^{k+1}$  which is generated by given  $w^k$  and satisfies

$$f(x_1) + g(x_2) - f(x_1^{k+1}) - g(x_2^{k+1}) + (w - w^{k+1})^T \{R(w^{k+1}) + H(w^{k+1} - w^k)\} \geq 0, \quad \forall w \in W \tag{18}$$

where

$$R(w^{k+1}) = \begin{pmatrix} -A^T \lambda^{k+1} + cA^T (Ax_1^{k+1} + Bx_2^{k+1} - b) \\ -B^T \lambda^{k+1} + cB^T (Ax_1^{k+1} + Bx_2^{k+1} - b) \\ (Ax_1^{k+1} + Bx_2^{k+1} - b) \end{pmatrix}, H = \begin{pmatrix} G_{11} & G_{12} & A^T \\ G_{21} & G_{22} & B^T \\ 0 & 0 & \frac{1}{c}I_l \end{pmatrix} \tag{19}$$

Setting  $w = w^*$  in Eq. (18), subsequently, one can derive the following:

$$(w^* - w^{k+1})^T M (w^{k+1} - w^k) \geq f(x_1^{k+1}) + g(x_2^{k+1}) - f(x_1^*) - g(x_2^*) + (w^{k+1} - w^*)^T F(w^{k+1}) \tag{20}$$

where  $F(w) = (-A^T \lambda, -B^T \lambda, Ax_1 + Bx_2)^T$ .



According to [38], one can derive the following:

$$(w^{k+1} - w^*)^T F(w^{k+1}) \geq (w^{k+1} - w^*)^T F(w^*) \quad (21)$$

$$f(x_1^{k+1}) + g(x_2^{k+1}) - f(x_1^*) - g(x_2^*) + (w^{k+1} - w^*)^T F(w^*) \geq 0 \quad (22)$$

Combing Eqs. (20)–(22), one can derive the following:

$$\begin{aligned} & (w^* - w^{k+1})^T M(w^{k+1} - w^k) \geq 0 \\ \Rightarrow & (w^k - w^* + w^{k+1} - w^k)^T M(w^k - w^{k+1}) \geq 0 \\ \Rightarrow & (w^k - w^*)^T M(w^k - w^{k+1}) \geq (w^k - w^{k+1})^T M(w^k - w^{k+1}) \end{aligned} \quad (23)$$

According to Eq. (23), one can derive the following:

$$\begin{aligned} & \|w^k - w^*\|_M^2 - \|w^{k+1} - w^*\|_M^2 \\ = & \|w^k - w^*\|_M^2 - \|(w^k - w^*) - (w^k - w^{k+1})\|_M^2 \\ = & \|w^k - w^*\|_M^2 - \|w^k - w^*\|_M^2 + 2(w^k - w^*)^T M(w^k - w^{k+1}) - \|w^k - w^{k+1}\|_M^2 \\ = & 2(w^k - w^*)^T M(w^k - w^{k+1}) - \|w^k - w^{k+1}\|_M^2 \\ \geq & \|w^k - w^{k+1}\|_M^2 \end{aligned} \quad (24)$$

So far, the proof of Lemma 1 has been completed.

So one can derive the following:

$$\lim_{k \rightarrow \infty} w^k = w^* \quad (25)$$

## 2.3 The Customized PPA

In fact, the complex high-dimensional Eq. (15) can be transferred into low-dimensional sub-problems provided that the proximal matrix  $G$  is chosen judiciously.

### 2.3.1 Parallel PPA

If matrix  $G$  is chosen as follows:

$$G = \begin{pmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{pmatrix} = \begin{pmatrix} \beta I_{n_1} & -cA^T B \\ -cB^T A & \beta I_{n_2} \end{pmatrix} \quad (26)$$

where  $G_{11} = \beta I_{n_1}$ ,  $G_{22} = \beta I_{n_2}$  is chosen to ensure that  $G$  is a positive definite matrix.  $G_{12} = -cA^T B$  and  $G_{21} = -cB^T A$  is chosen to ensure that the solution of problem (15) with the particular choice Eq. (26) is equivalent to solving the following sub-problems:

$$f(x_1) - f(x_1^{k+1}) + (x_1 - x_1^{k+1})^T \{(-A^T \lambda^k + cA^T (Ax_1^{k+1} + Bx_2^k - b)) + \beta(x_1^{k+1} - x_1^k)\} \geq 0 \quad (27)$$

$$g(x_2) - g(x_2^{k+1}) + (x_2 - x_2^{k+1})^T \{(-B^T \lambda^k + cB^T (Ax_1^k + Bx_2^{k+1} - b)) + \beta(x_2^{k+1} - x_2^k)\} \geq 0 \quad (28)$$



Then, with the concept of variational inequality, one conclusion can be reached that the solution of Eqs. (27) and (28) is equivalent to that of the following independent convex programming sub-problems.

$$x_1^{k+1} = \arg \min \left\{ f(x_1) - \lambda^{T,k} Ax_1 + \frac{c}{2} \|Ax_1 + Bx_2^k - b\|^2 + \frac{\beta}{2} \|x_1 - x_1^k\|^2 \mid x_1 \in \Omega_1 \right\} \quad (29)$$

$$x_2^{k+1} = \arg \min \left\{ g(x_2) - \lambda^{T,k} Bx_2 + \frac{c}{2} \|Ax_1^k + Bx_2 - b\|^2 + \frac{\beta}{2} \|x_2 - x_2^k\|^2 \mid x_2 \in \Omega_2 \right\} \quad (30)$$

---

**Strategy 1:** Parallel proximal point algorithm (Parallel PPA)

---

```

1   Input: Initial points  $x_1^0$   $x_2^0$   $\lambda^0$ 
2       Maximum number of iterations  $k_{\max}$ , Tolerance  $\eta$ 
3   Output: Solution  $w^{k+1}$  of the problem
4   Initialize  $k = 0$ 
5   Repeat
6       Solving Eq. (29) yields  $x_1^{k+1}$ 
7       Solving Eq. (30) yields  $x_2^{k+1}$ 
8       Update  $\lambda^{k+1} = \lambda^k - c(Ax_1^{k+1} + Bx_2^{k+1} - b)$ 
9       If  $\max \left\{ \|x_1^{k+1} - x_1^k\|_2^2, \|x_2^{k+1} - x_2^k\|_2^2, \|\lambda^{k+1} - \lambda^k\|_2^2 \right\} < \eta$ 
10          break
11      end
12       $k = k + 1$ 
13      Until  $k > k_{\max}$ 
14      Return  $w^{k+1}$  as the solution

```

---

2.3.2 Serial PPA

Similarly, if matrix  $G$  is chosen as follows:

$$G = \begin{pmatrix} \beta I_{n_1} & -cA^T B \\ 0 & \beta I_{n_2} \end{pmatrix} \quad (31)$$

where  $\beta$  is chosen to ensure that  $G$  is positive definite matrix.  $G_{12} = -cA^T B$  and  $G_{21} = 0$  is chosen to ensure that the solution of Eq. (15) with the particular choice Eq. (31) is equivalent to solving the following sub-problems:

$$x_1^{k+1} = \arg \min \left\{ f(x_1) - \lambda^{T,k} Ax_1 + \frac{c}{2} \|Ax_1 + Bx_2^k - b\|^2 + \frac{\beta}{2} \|x_1 - x_1^k\|^2 \mid x_1 \in \Omega_1 \right\} \quad (32)$$

$$x_2^{k+1} = \arg \min \left\{ g(x_2) - \lambda^{T,k} Bx_2 + \frac{c}{2} \|Ax_1^{k+1} + Bx_2 - b\|^2 + \frac{\beta}{2} \|x_2 - x_2^k\|^2 \mid x_2 \in \Omega_2 \right\} \quad (33)$$

---

**Strategy 2:** Serial proximal point algorithm (Serial PPA)

---

```

1   Input: Initial points  $x_1^0$   $x_2^0$   $\lambda^0$ 
2       Maximum number of iterations  $k_{\max}$ , Tolerance  $\eta$ 
3   Output: Solution  $w^{k+1}$  of the problem
4   Initialize  $k = 0$ 

```

---

(Continued)

**Strategy 2 (continued)**


---

```

5      Repeat
6      Solving Eq. (32) yields  $x_1^{k+1}$ 
7      Solving Eq. (33) yields  $x_2^{k+1}$ 
8      Update  $\lambda^{k+1} = \lambda^k - c(Ax_1^{k+1} + Bx_2^{k+1} - b)$ 
9      If  $\max\{\|x_1^{k+1} - x_1^k\|_2^2, \|x_2^{k+1} - x_2^k\|_2^2, \|\lambda^{k+1} - \lambda^k\|_2^2\} < \eta$ 
10     break
11     end
12      $k = k + 1$ 
13     Until  $k > k_{\max}$ 
14     Return  $w^{k+1}$  as the solution

```

---

**3 Trend Theory-Driven PPA**

Empirical applications have demonstrated that the iterative strategy of the Proximal Point Algorithm (PPA) requires a substantial number of iterations to reach the optimal solution when the penalty parameter  $c$  is either too small or too large. However, the optimal values of this parameter cannot be predetermined. Consequently, there is a compelling necessity to propose a novel method for the dynamic adjustment of the penalty parameter during the iteration process. Fortunately, the work of Boyd et al. [39] provides a precedent by employing the balance of residuals between the primal and dual problems to adjust the parameter  $c$ . According to the aforementioned exposition, adjusting parameters based on the concept of primal-dual residual balancing necessitates the concurrent computation of residuals for both the primal and dual problems [36]. Furthermore, when constructing various distributed iterative strategies, whether parallel or serial, the customization of positive definite matrices  $G$  results in divergent formulas for computing dual residuals. This undoubtedly increases the complexity associated with the implementation of the algorithm. Reverting to the mathematical model of the multi-area economic dispatch (MAED) problem, if the linear coupling constraint is disregarded, the original optimization problem can be decomposed directly into two independent sub-optimization problems. Ensuring the validity of the linear coupling constraint is pivotal for achieving distributed optimization. In Section 2, the augmented Lagrangian function method is employed to incorporate the linear coupling constraint into the objective function. At this juncture, the parameter  $c$ , introduced by the augmented Lagrangian function, assumes the role of a penalty parameter. An ideal state would be for the numerical values obtained from calculating the linear coupling conditions during the iteration process to continuously approach the ideal value of zero. However, during the actual iteration process, the numerical trend of the linear coupling constraint often exhibits unexpected patterns. Specifically, three distinct scenarios may arise. Firstly, the trend may approach zero, which is the desired outcome. Secondly, the trend may unexpectedly diverge from zero, moving further away. Lastly, the trend may oscillate within a certain range, neither converging towards nor diverging from zero.

In response to the three scenarios mentioned above, inspired by trend theory in the stock market, this study proposes a parameter adjustment strategy based on trend theory. The core tenet of trend theory is to timely cut losses when faced with adverse market movements, while holding onto profitable investments for an extended period in anticipation of greater gains. Trend theory can be succinctly encapsulated as the notion that stock prices tend to persist in an upward or downward trajectory once established by the market, until a clear reversal signal emerges. In upward trends, the widely adopted strategy among stock traders is to sell promptly once the stock price breaks downward through the upward trend line, an approach firmly rooted in the fundamental principles of trend theory. Generally, a breach of the upward trend line by stock prices is viewed as a sell signal, suggesting a possible end to the upward momentum. However, one cannot

definitively state that the stock has entered a downward trend. Inspired by trend theory, the changing trend of equation coupling constraints is likened to the upward momentum of a stock. When this trend aligns with expectations (the numerical results of the linear coupling conditions quickly approach zero), the parameter is maintained to allow profits to accumulate. Conversely, if the trend diverges, adjustments to the parameter are promptly made. The parameter  $c$  serves as a penalty term in the objective function. If the equality coupling constraints don't approach zero as swiftly as expected, the penalty term may not be sufficiently effective. A straightforward corrective action would be to elevate the value of the penalty parameter  $c$ .

Considering that parameter  $c$  also serves as the update step size for the Lagrange multiplier, if the step size is too large, the resulting iterative points may oscillate with significant errors near the optimal point, which can make convergence to a small neighborhood around the optimal solution difficult. Conversely, if the step size is too small, more iterations may be required to converge to such a small neighborhood. In light of the trend-based parameter adjustment method proposed in this study, a smaller initial value for the parameter is recommended with the concurrent establishment of an upper limit. Based on the aforementioned analysis, the trend-based parameter adjustment strategy proposed in this study is as follows:

$$c = \begin{cases} 2c & \text{if } \|Ax_1^{k+1} + Bx_2^{k+1} - b\|_2 \geq 0.5 \|Ax_1^k + Bx_2^k - b\|_2 \text{ and } k \leq k_{stop} \text{ and } c < 100 \\ c & \text{other} \end{cases} \tag{34}$$

where the meanings of the symbols in Eq. (34) have already been explained in Section 2 and will not be repeated here.

Simultaneously, the consideration of linear coupling constraints as residuals of the original optimization problem effectively illustrates the degree of approximation between solutions derived from distributed optimization algorithms and those obtained through centralized methods. Consequently, in this study, the residuals of the original optimization problem have been incorporated into the existing convergence criteria for enhanced accuracy. The refined formula for convergence determination is presented as follows:

$$\max \left\{ \|x_1^{k+1} - x_1^k\|_2^2, \|x_2^{k+1} - x_2^k\|_2^2, \|\lambda^{k+1} - \lambda^k\|_2^2, \|Ax_1^{k+1} + Bx_2^{k+1} - b\|_2^2 \right\} < \eta \tag{35}$$

At the same time, to facilitate comparative analysis, the convergence criteria in both Strategy 1 and Strategy 2 have been modified to Eq. (35).

---

**Strategy 3:** Parallel PPA-based on trend theory

---

- 1     **Input:** Initial points  $x_1^0$   $x_2^0$   $\lambda^0$
  - 2     Maximum number of iterations  $k_{max}$ , Tolerance  $\eta$
  - 3     **Output:** Solution  $w^{k+1}$  of the problem
  - 4     Initialize  $k = 0$
  - 5     **Repeat**
  - 6     Solving Eq. (29) yields  $x_1^{k+1}$
  - 7     Solving Eq. (30) yields  $x_2^{k+1}$
  - 8     Update  $\lambda^{k+1} = \lambda^k - c (Ax_1^{k+1} + Bx_2^{k+1} - b)$
  - 9     Update  $c$  using Eq. (34)
  - 10    **If**  $\max \left\{ \|x_1^{k+1} - x_1^k\|_2^2, \|x_2^{k+1} - x_2^k\|_2^2, \|\lambda^{k+1} - \lambda^k\|_2^2, \|Ax_1^{k+1} + Bx_2^{k+1} - b\|_2^2 \right\} < \eta$
  - 11    break
  - 12    **end**
- 

(Continued)

**Strategy 3 (continued)**


---

```

13       $k = k + 1$ 
14      Until  $k > k_{max}$ 
15      Return  $w^{k+1}$  as the solution

```

---

**Strategy 4: Serial PPA-based on trend theory**


---

```

1      Input: Initial points  $x_1^0$   $x_2^0$   $\lambda^0$ 
2          Maximum number of iterations  $k_{max}$ , Tolerance  $\eta$ 
3      Output: Solution  $w^{k+1}$  of the problem
4      Initialize  $k = 0$ 
5      Repeat
6      Solving Eq. (32) yields  $x_1^{k+1}$ 
7      Solving Eq. (33) yields  $x_2^{k+1}$ 
8      Update  $\lambda^{k+1} = \lambda^k - c (Ax_1^{k+1} + Bx_2^{k+1} - b)$ 
9      Update  $c$  using Eq. (34)
10     If  $\max \left\{ \|x_1^{k+1} - x_1^k\|_2^2, \|x_2^{k+1} - x_2^k\|_2^2, \|\lambda^{k+1} - \lambda^k\|_2^2, \|Ax_1^{k+1} + Bx_2^{k+1} - b\|_2^2 \right\} < \eta$ 
11         break
12     end
13      $k = k + 1$ 
14     Until  $k > k_{max}$ 
15     Return  $w^{k+1}$  as the solution

```

---

**4 Application**

Section 2 gives the simplified mathematical model of MAED, while this section provides the MAED model with physical meaning that corresponds to the simplified model as follows:

$$\min \{f(x_1) + g(x_2) | Ax_1 + Bx_2 = b, x_1 \in \Omega_1, x_2 \in \Omega_2\} \quad (36)$$

where

$$A = \left( \underbrace{0, \dots, 0, 1}_{N_1} \right)^T, B = \left( \underbrace{0, \dots, 0, -1}_{N_2} \right)^T, b = 0 \quad (37)$$

$$x_1 = (P_1, P_2, \dots, P_{N_1}, P_{b1}), x_2 = (P_{N_1+1}, P_{N_1+2}, \dots, P_{N_1+N_2}, P_{b2}) \quad (38)$$

$$f(x_1) = \sum_{i=1}^{N_1} (a_i^2 P_i^2 + b_i P_i + c_i), g(x_2) = \sum_{i=N_1+1}^{N_1+N_2} (a_i^2 P_i^2 + b_i P_i + c_i) \quad (39)$$

$$\Omega_1 = \left\{ x_1 \mid \sum_{i=1}^{N_1} P_i + P_{b1} = load_1, P_{i, \min} \leq P_i \leq P_{i, \max}, 1 \leq i \leq N_1; |P_{b1}| \leq T_{12} \right\} \quad (40)$$

$$\Omega_2 = \left\{ x_2 \mid \sum_{i=N_1+1}^{N_1+N_2} P_i - P_{b2} = load_2, P_{i, \min} \leq P_i \leq P_{i, \max}, N_1 + 1 \leq i \leq N_1 + N_2; |P_{b2}| \leq T_{12} \right\} \quad (41)$$

where  $P_i$  is the active output of  $i$ -th unit.  $P_{b1}$  and  $P_{b2}$  denote transfer power flow between area 1 and area 2.  $T_{12}$  denotes the maximum transfer power flow between area 1 and area 2.  $P_{i, \max}$ ,  $P_{i, \min}$  are the upper and lower limits of  $i$ -th unit.  $load_i$  denotes the system load for area  $i$ .  $a_i$ ,  $b_i$  and  $c_i$  are given parameters.  $N_i$  denotes the number of units in area  $i$ .  $f(x_1)$  and  $g(x_2)$  represent the generation cost of areas 1 and 2, and the total generation cost is the sum of the generation cost of the two regions.

According to Lemma 1, a sufficient condition for ensuring the convergence of the PPA algorithm is that matrix  $G$  is positive definite. In fact, for PPA, when matrix  $G$  is diagonally dominant, matrix  $G$  necessarily becomes positive definite. For parallel PPA, the customized matrix  $G$  is given by (26). When Eq. (37) is concurrently considered and the condition  $\beta > c$  is satisfied, the positive definiteness of the matrix represented by matrix (26) is ensured. At the same time, according to the description of parallel PPA, the parameter  $\beta$  in the matrix (26) appears as quadratic terms  $\frac{\beta}{2} \|x_1 - x_1^k\|^2$  and  $\frac{\beta}{2} \|x_2 - x_2^k\|^2$  in the objective function. The existence of  $\frac{\beta}{2} \|x_1 - x_1^k\|^2$  and  $\frac{\beta}{2} \|x_2 - x_2^k\|^2$  act as penalty functions when  $(x_1, x_2)$  is far away from  $(x_1^k, x_2^k)$ . In order to reduce the penalty effect of  $\frac{\beta}{2} \|x_1 - x_1^k\|^2$  and  $\frac{\beta}{2} \|x_2 - x_2^k\|^2$ , the parameter  $\beta$  should be as small as possible within a reasonable range. Therefore, a suggestion is made to set  $\beta/c = 1.1$  for parallel PPA. Similar to parallel PPA, the same suggestion is made to set  $\beta/c = 1.1$  for serial PPA.

In this paper, two test systems are delineated for the purpose of algorithmic evaluation. Test System 1: This system encompasses ten generators dispersed across three distinct regions. The collective load demand for the entire system is 2700 MW, with a tie-line power transfer capacity of 100 MW between any two areas. Area 1, which houses four generators, accounts for 50% of the total load demand. In contrast, areas 2 and 3, each equipped with three generators, contribute approximately 25% to the overall demand. Detailed unit data can be referenced from [40]. Test System 2: Drawing data from [41], this system comprises forty generators distributed among four areas, meeting a total load demand of 10,500 MW. Area 1, which includes the first ten generators, supports 15% of the total load. Area 2, with the subsequent ten generators, bears 40% of the demand. Area 3, consisting of the third set of ten generators, is responsible for 30% of the load. Lastly, area 4, with the remaining ten generators, manages an additional 15% of the total demand. Regarding power flow constraints, the limit between areas 1 and 2 is established at 200 MW. Similarly, the limits between areas 1 and 3, and areas 2 and 3, are also set at 200 MW. In the case of power flow between area 4 and each of the other three areas (areas 1, 2, and 3), the restriction is set to 100 MW.

To verify the effectiveness of the proposed algorithm in this study, the following eight strategies were adopted to solve Test System 1 and Test System 2 (Table 1):

Strategy 1: Parallel distributed algorithm based on the PPA principle utilizes a fixed value for parameter  $c$  throughout the iteration process (PPPA).

Strategy 2: Serial distributed algorithm based on the PPA principle utilizes a fixed value for parameter  $c$  throughout the iteration process (SPPA).

Strategy 3: Parallel distributed algorithm based on PPA principle and trend theory (PPPA-TT).

Strategy 4: Serial distributed algorithm based on PPA principle and trend theory (SPPA-TT).

Strategy 5: Serial distributed algorithm based on ADMM and residual balancing theory (ADMM-RB).

Strategy 6: Serial distributed algorithm based on ADMM and trend theory (ADMM-TT).

Strategy 7: Parallel distributed algorithm based on APP (APP).

Strategy 8: Parallel distributed algorithm based on APP and trend theory (APP-TT).

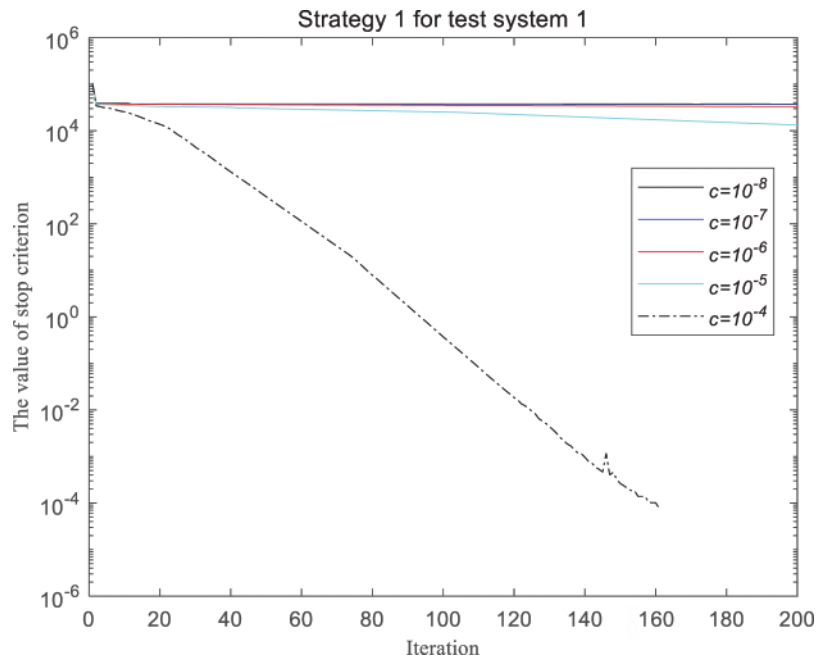
To facilitate the comparison of results across different strategies, the simulation uniformly sets  $K_{max}$  to 200,  $k_{stop}$  to 200, and  $\eta$  to  $10^{-4}$ . The strategy based on trend theory suggests a smaller initial value for parameter  $c$ . Likewise, to facilitate the comparison of different strategies, the initial value of parameter  $c$  is set to  $10^{-8}$ ,  $10^{-7}$ ,  $10^{-6}$ ,  $10^{-5}$ , and  $10^{-4}$ , respectively. The stop criterion is uniformly adopted as Eq. (35). For all strategies, each sub-optimization problem is solved using the *fmincon* function based on MATLAB software.

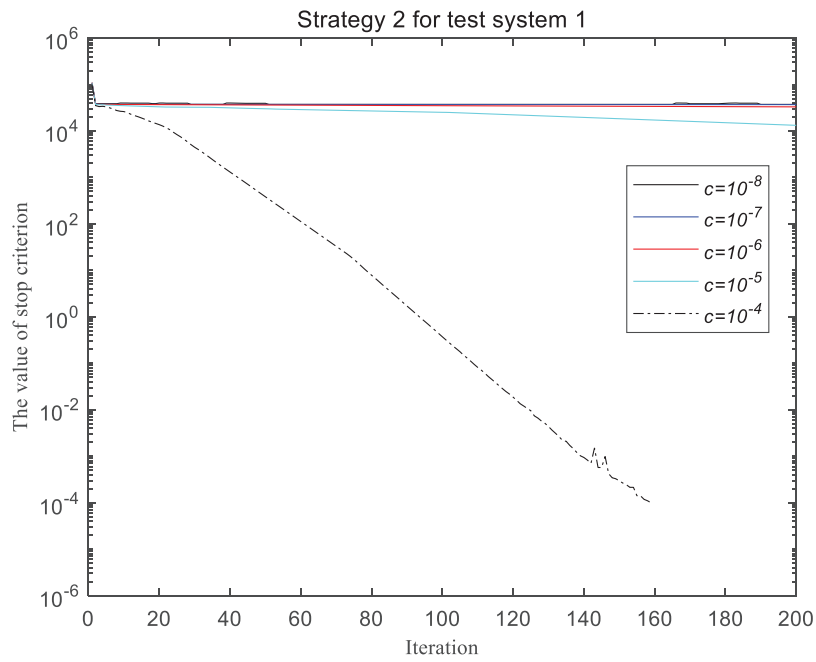
**Table 1:** Comparison of strategies 1–8

Strategy	Type	Algorithm	Para. $c$ Adj	Additions
1	Parallel	PPA	Fixed	–
2	Serial	PPA	Fixed	–
3	Parallel	PPA	Dynamic (Trend theory)	Trend theory
4	Serial	PPA	Dynamic (Trend theory)	Trend theory
5	Serial	ADMM	Dynamic (Residual balancing theory)	Residual balancing theory
6	Serial	ADMM	Dynamic (Trend theory)	Trend theory
7	Parallel	APP	Fixed	–
8	Parallel	APP	Dynamic (Trend theory)	Trend theory

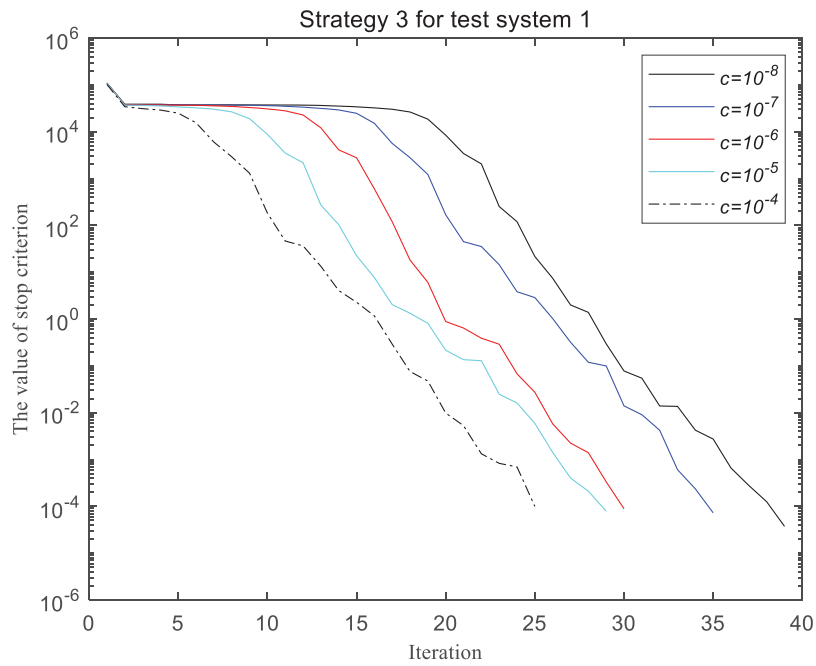
#### 4.1 Simulation Results for Test System 1

Figs. 2–9 depict the divergent trajectories of the stop criterion resulting from the application of strategies 1–8 to Test System 1. Within these figures, the horizontal axis corresponds to the iteration count, whereas the vertical axis signifies the numerical values of the stop criterion. Each figure encompasses five curves, illustrating the fluctuating trends of the stop criterion values across various initial configurations of parameter  $c$ .

**Figure 2:** The curve of stop criterion for test system 1 based on strategy 1

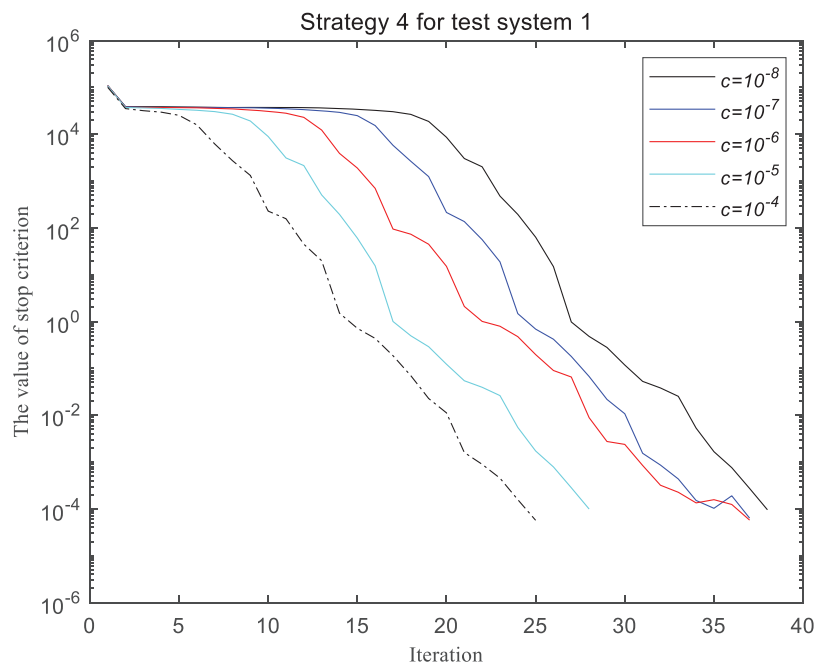


**Figure 3:** The curve of stop criterion for test system 1 based on strategy 2

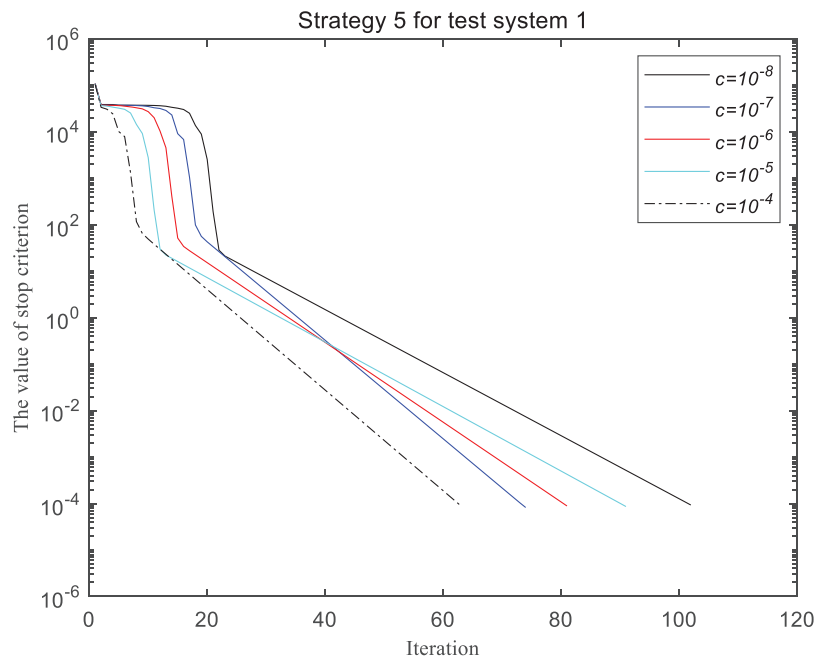


**Figure 4:** The curve of stop criterion for test system 1 based on strategy 3





**Figure 5:** The curve of stop criterion for test system 1 based on strategy 4



**Figure 6:** The curve of stop criterion for test system 1 based on strategy 5

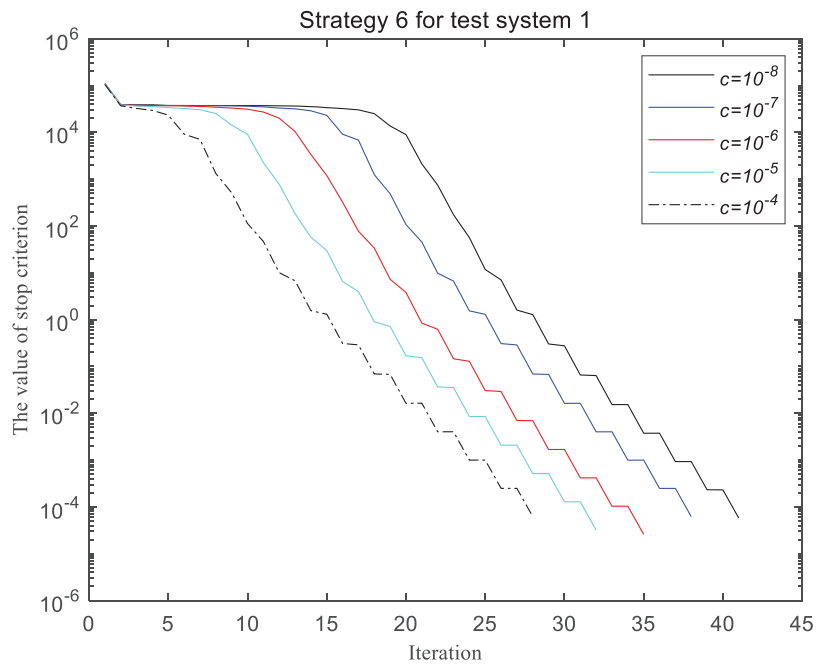


Figure 7: The curve of stop criterion for test system 1 based on strategy 6

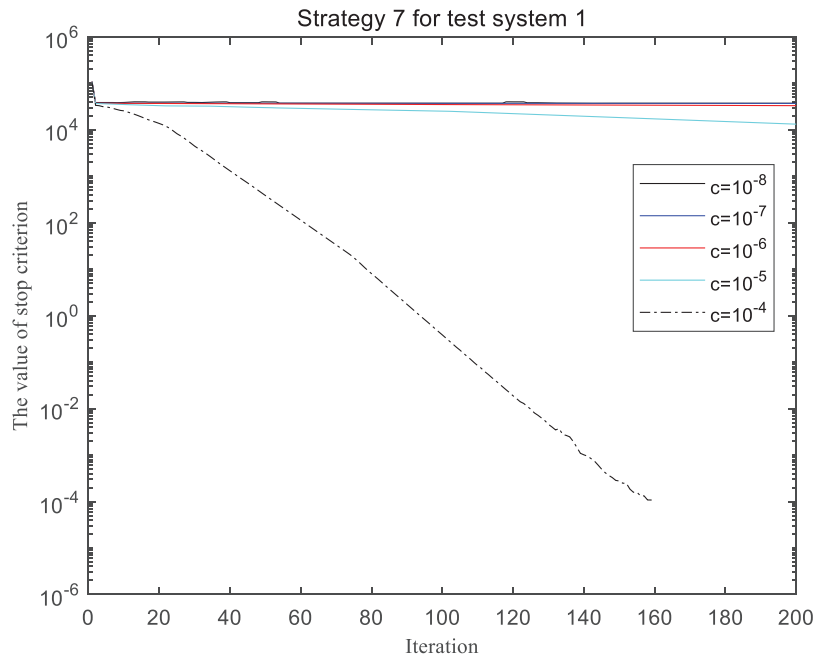
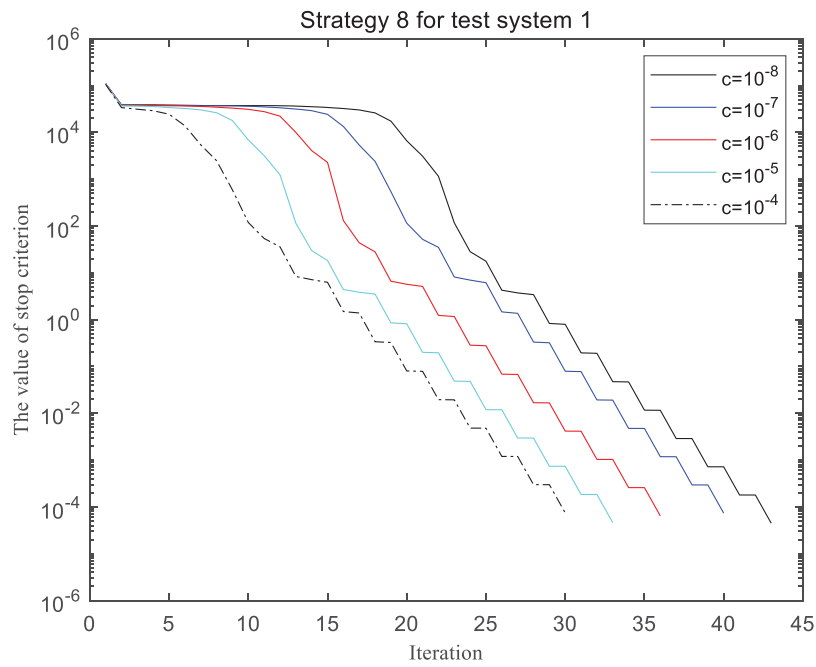


Figure 8: The curve of stop criterion for test system 1 based on strategy 7



**Figure 9:** The curve of stop criterion for test system 1 based on strategy 8

Tables 2–9 offer ancillary data supplementary to Figs. 2–9. Taking Table 2 as an exemplar, the inaugural column enumerates diverse parameter settings, facilitating direct comparisons. Columns 2, 3, 4, and 5 delineate the iteration count, stop criterion, program execution time and the optimal value of the objective function, respectively, for the corresponding algorithm upon completion with the specified parameter settings in the inaugural column. The maximum iteration count in this study is established at 200; should the iteration count surpass this threshold, the process is compelled to terminate, denoted as ‘Fail’. Concurrently, the ancillary data for Test System 1 based on strategies 1–8 are presented in Tables 2–9, respectively, with each column in Tables 3–9 maintaining the same significance as in Table 2.

**Table 2:** The results for test system 1 based on strategy 1

Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	Fail	Fail	Fail	Fail
$c = 10^{-7}$	Fail	Fail	Fail	Fail
$c = 10^{-6}$	Fail	Fail	Fail	Fail
$c = 10^{-5}$	Fail	Fail	Fail	Fail
$c = 10^{-4}$	161	$7.62 \times 10^{-5}$	10.51	$0.718 \times 10^3$

**Table 3:** The results for test system 1 based on strategy 2

Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	Fail	Fail	Fail	Fail
$c = 10^{-7}$	Fail	Fail	Fail	Fail
$c = 10^{-6}$	Fail	Fail	Fail	Fail
$c = 10^{-5}$	Fail	Fail	Fail	Fail
$c = 10^{-4}$	159	$9.93 \times 10^{-5}$	10.58	$0.718 \times 10^3$

**Table 4:** The results for test system 1 based on strategy 3

Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	39	$3.74 \times 10^{-5}$	1.80	$0.720 \times 10^3$
$c = 10^{-7}$	35	$7.26 \times 10^{-5}$	1.64	$0.719 \times 10^3$
$c = 10^{-6}$	30	$8.88 \times 10^{-5}$	1.36	$0.720 \times 10^3$
$c = 10^{-5}$	29	$7.88 \times 10^{-5}$	1.32	$0.720 \times 10^3$
$c = 10^{-4}$	25	$9.96 \times 10^{-5}$	1.21	$0.719 \times 10^3$

**Table 5:** The results for test system 1 based on strategy 4

Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	38	$9.55 \times 10^{-5}$	1.98	$7.21 \times 10^2$
$c = 10^{-7}$	37	$6.33 \times 10^{-5}$	1.92	$7.20 \times 10^2$
$c = 10^{-6}$	37	$5.74 \times 10^{-5}$	2.27	$7.21 \times 10^2$
$c = 10^{-5}$	28	$9.91 \times 10^{-5}$	1.43	$7.21 \times 10^2$
$c = 10^{-4}$	25	$5.69 \times 10^{-5}$	1.29	$7.20 \times 10^2$

**Table 6:** The results for test system 1 based on strategy 5

Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	102	$9.29 \times 10^{-5}$	5.40	$7.18 \times 10^2$
$c = 10^{-7}$	74	$8.22 \times 10^{-5}$	3.82	$7.18 \times 10^2$
$c = 10^{-6}$	81	$8.86 \times 10^{-5}$	4.09	$7.18 \times 10^2$
$c = 10^{-5}$	91	$8.61 \times 10^{-5}$	4.69	$7.18 \times 10^2$
$c = 10^{-4}$	63	$8.95 \times 10^{-5}$	3.48	$7.18 \times 10^2$

**Table 7:** The results for test system 1 based on strategy 6

Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	41	$5.76 \times 10^{-5}$	2.21	$7.18 \times 10^2$
$c = 10^{-7}$	38	$6.19 \times 10^{-5}$	1.88	$7.18 \times 10^2$
$c = 10^{-6}$	35	$2.58 \times 10^{-5}$	1.81	$7.18 \times 10^2$
$c = 10^{-5}$	32	$3.19 \times 10^{-5}$	1.67	$7.18 \times 10^2$
$c = 10^{-4}$	28	$6.17 \times 10^{-5}$	1.46	$7.18 \times 10^2$

**Table 8:** The results for test system 1 based on strategy 7

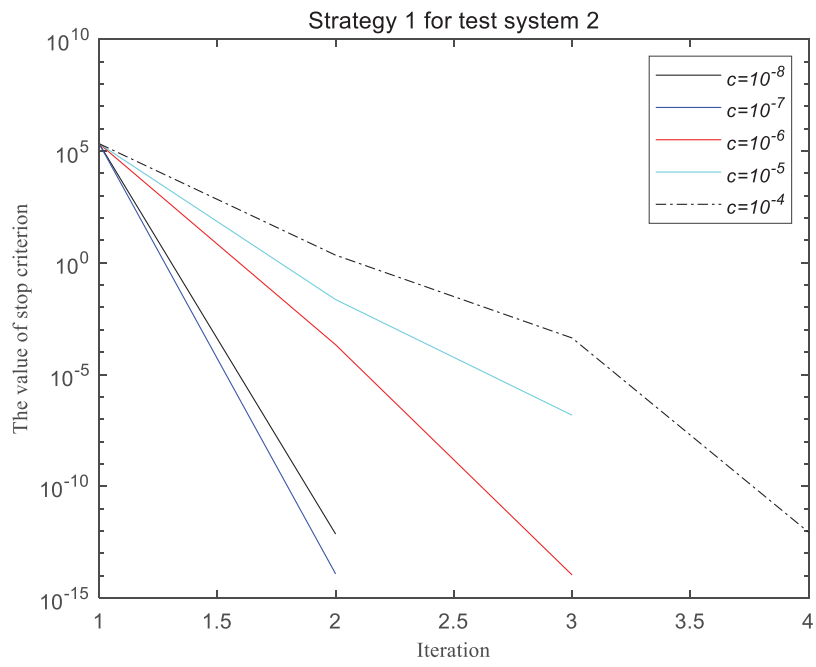
Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	Fail	Fail	Fail	Fail
$c = 10^{-7}$	Fail	Fail	Fail	Fail
$c = 10^{-6}$	Fail	Fail	Fail	Fail
$c = 10^{-5}$	Fail	Fail	Fail	Fail
$c = 10^{-4}$	160	$9.63 \times 10^{-5}$	9.50	$7.18 \times 10^2$

**Table 9:** The results for test system 1 based on strategy 8

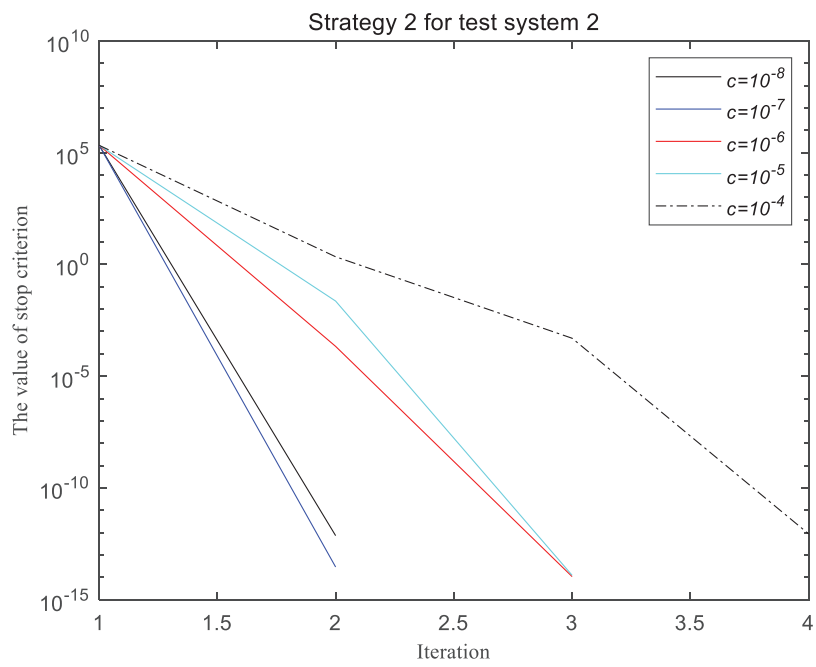
Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	43	$4.47 \times 10^{-5}$	2.15	$7.19 \times 10^2$
$c = 10^{-7}$	40	$7.34 \times 10^{-5}$	1.87	$7.18 \times 10^2$
$c = 10^{-6}$	36	$6.42 \times 10^{-5}$	1.73	$7.19 \times 10^2$
$c = 10^{-5}$	33	$4.58 \times 10^{-5}$	1.57	$7.19 \times 10^2$
$c = 10^{-4}$	30	$7.44 \times 10^{-5}$	1.43	$7.18 \times 10^2$

#### 4.2 Simulation Results for Test System 2

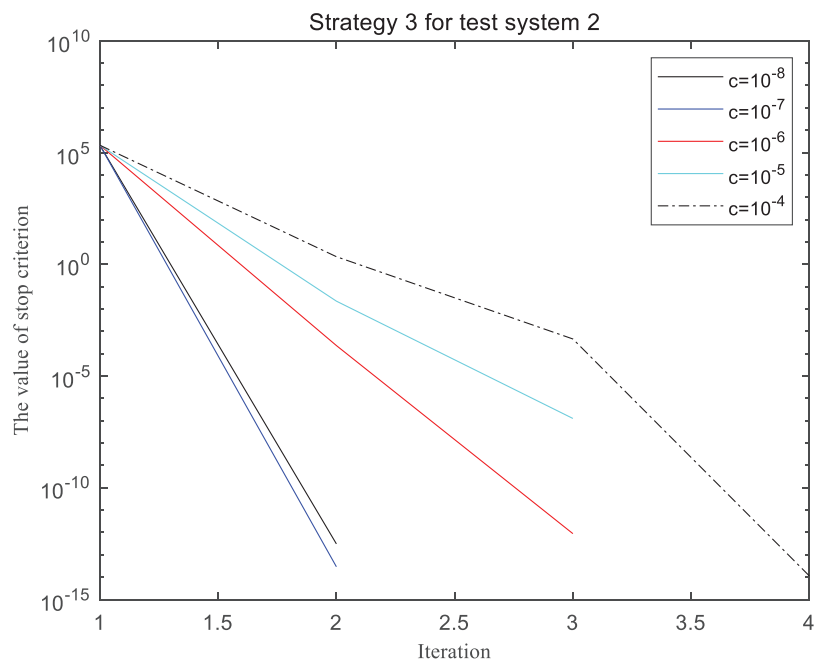
Figs. 10–17 illustrate the diverse trends of the stop criterion obtained by applying strategies 1–8 to Test System 2. In these figures, the horizontal axis denotes the number of iterations, and the vertical axis represents the numerical values of the stop criterion. Each figure within the series from Figs. 10–17 contains five distinct curves, which portray the fluctuating trends of the stop criterion values across various initial configurations of parameter  $c$ . The ancillary data for Test System 2, based on strategies 1–8, are sequentially presented in Tables 10–17, respectively. Each column within Tables 10–17 adheres to the same conventions as those in Table 2, providing a consistent framework for analysis and comparison.



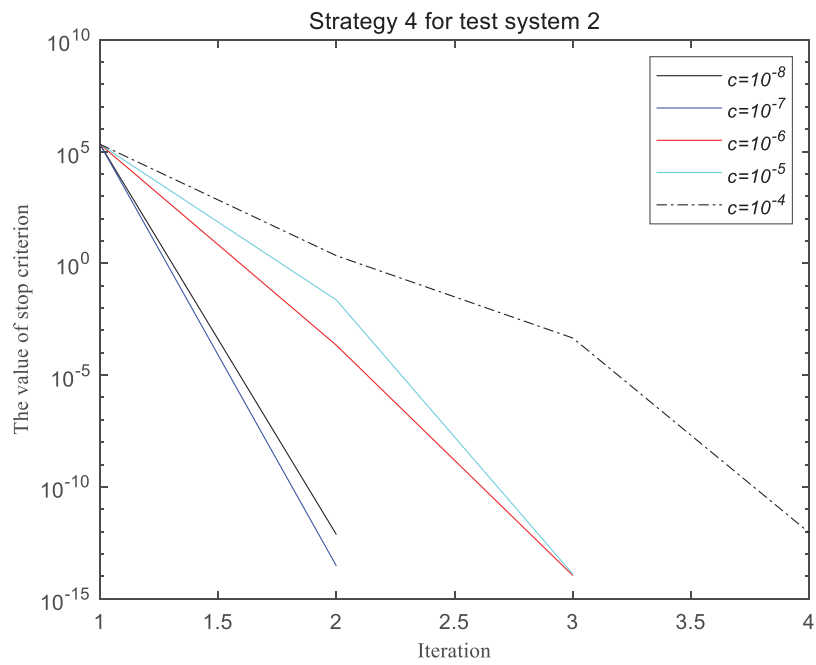
**Figure 10:** The curve of stop criterion for test system 2 based on strategy 1



**Figure 11:** The curve of stop criterion for test system 2 based on strategy 2

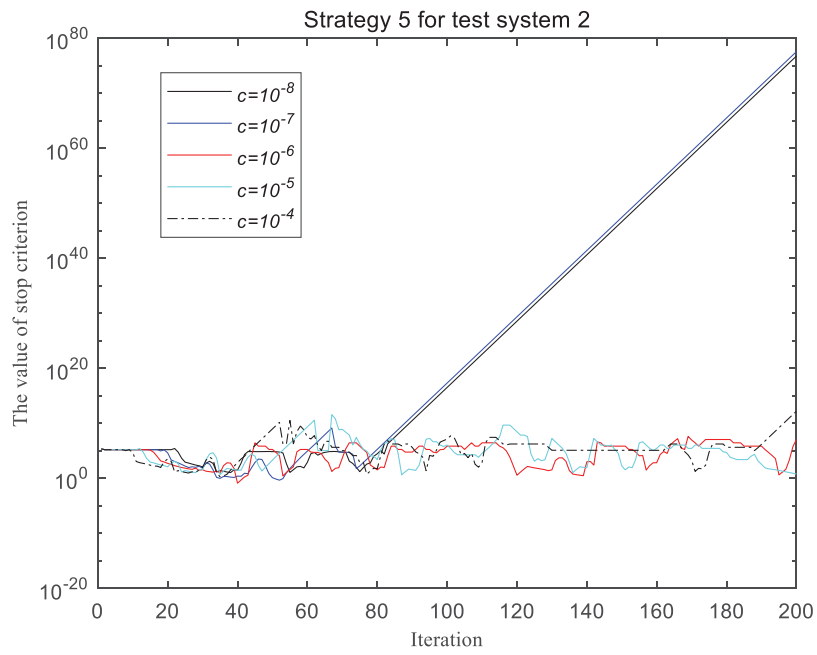


**Figure 12:** The curve of stop criterion for test system 2 based on strategy 3

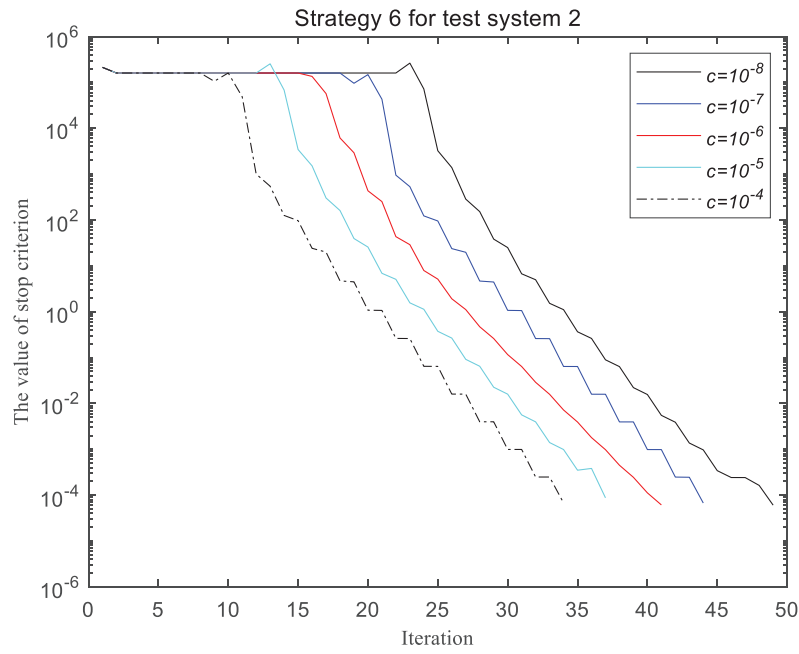


**Figure 13:** The curve of stop criterion for test system 2 based on strategy 4

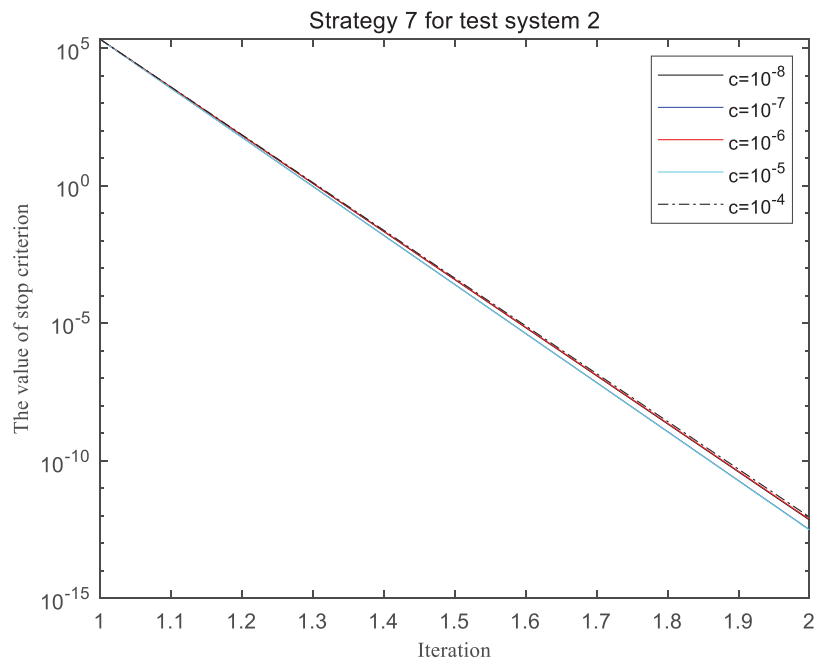




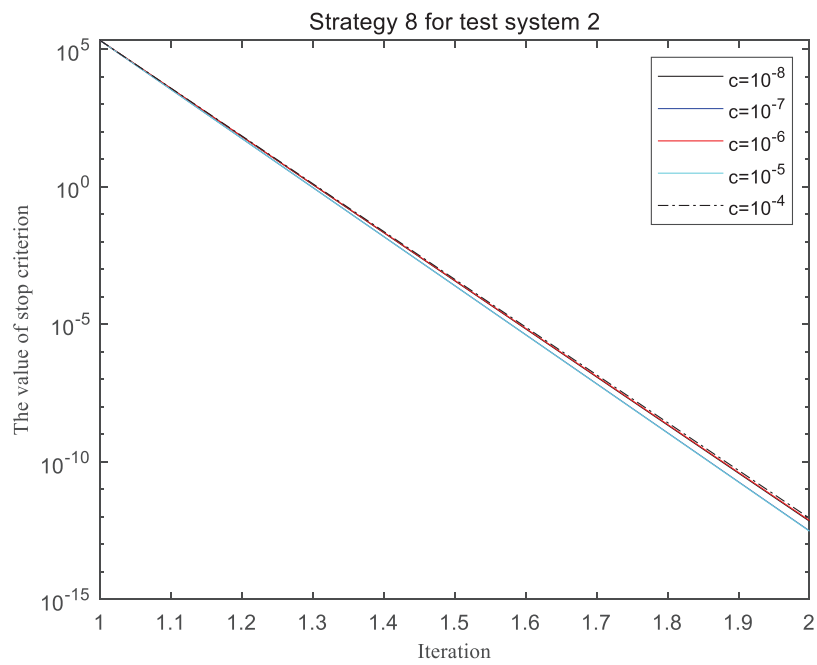
**Figure 14:** The curve of stop criterion for test system 2 based on strategy 5



**Figure 15:** The curve of stop criterion for test system 2 based on strategy 6



**Figure 16:** The curve of stop criterion for test system 2 based on strategy 7



**Figure 17:** The curve of stop criterion for test system 2 based on strategy 8

**Table 10:** The results for test system 2 based on strategy 1

Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	2	$3.08 \times 10^{-13}$	0.33	$1.03 \times 10^5$
$c = 10^{-7}$	2	$2.92 \times 10^{-14}$	0.30	$1.03 \times 10^5$
$c = 10^{-6}$	3	$8.86 \times 10^{-13}$	0.41	$1.03 \times 10^5$
$c = 10^{-5}$	3	$5.77 \times 10^{-7}$	0.41	$1.03 \times 10^5$
$c = 10^{-4}$	4	$1.23 \times 10^{-14}$	0.45	$1.03 \times 10^5$

**Table 11:** The results for test system 2 based on strategy 2

Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	2	$7.38 \times 10^{-13}$	0.29	$1.03 \times 10^5$
$c = 10^{-7}$	2	$2.92 \times 10^{-14}$	0.29	$1.03 \times 10^5$
$c = 10^{-6}$	3	$1.08 \times 10^{-14}$	0.35	$1.03 \times 10^5$
$c = 10^{-5}$	3	$1.24 \times 10^{-14}$	0.35	$1.03 \times 10^5$
$c = 10^{-4}$	4	$8.88 \times 10^{-13}$	0.48	$1.03 \times 10^5$

**Table 12:** The results for test system 2 based on strategy 3

Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	2	$3.08 \times 10^{-13}$	0.40	$1.03 \times 10^5$
$c = 10^{-7}$	2	$2.92 \times 10^{-14}$	0.35	$1.03 \times 10^5$
$c = 10^{-6}$	3	$8.86 \times 10^{-13}$	0.46	$1.03 \times 10^5$
$c = 10^{-5}$	3	$1.26 \times 10^{-7}$	0.53	$1.03 \times 10^5$
$c = 10^{-4}$	4	$1.13 \times 10^{-14}$	0.47	$1.03 \times 10^5$

**Table 13:** The results for test system 2 based on strategy 4

Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	2	$7.38 \times 10^{-13}$	0.40	$1.03 \times 10^5$
$c = 10^{-7}$	2	$2.92 \times 10^{-14}$	0.34	$1.03 \times 10^5$
$c = 10^{-6}$	3	$1.081 \times 10^{-14}$	0.53	$1.03 \times 10^5$
$c = 10^{-5}$	3	$1.24 \times 10^{-14}$	0.50	$1.03 \times 10^5$
$c = 10^{-4}$	4	$8.88 \times 10^{-13}$	0.75	$1.03 \times 10^5$

**Table 14:** The results for test system 2 based on strategy 5

Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	Fail	Fail	Fail	Fail
$c = 10^{-7}$	Fail	Fail	Fail	Fail
$c = 10^{-6}$	Fail	Fail	Fail	Fail
$c = 10^{-5}$	Fail	Fail	Fail	Fail
$c = 10^{-4}$	Fail	Fail	Fail	Fail

**Table 15:** The results for test system 2 based on strategy 6

Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	49	$6.06 \times 10^{-5}$	6.57	$1.18 \times 10^5$
$c = 10^{-7}$	44	$6.74 \times 10^{-5}$	5.50	$1.18 \times 10^5$
$c = 10^{-6}$	41	$6.13 \times 10^{-5}$	5.55	$1.18 \times 10^5$
$c = 10^{-5}$	37	$8.76 \times 10^{-5}$	4.61	$1.18 \times 10^5$
$c = 10^{-4}$	34	$6.79 \times 10^{-5}$	4.93	$1.18 \times 10^5$

**Table 16:** The results for test system 2 based on strategy 7

Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	2	$6.93 \times 10^{-13}$	0.28	$1.03 \times 10^5$
$c = 10^{-7}$	2	$3.08 \times 10^{-13}$	0.29	$1.03 \times 10^5$
$c = 10^{-6}$	2	$7.38 \times 10^{-13}$	0.29	$1.03 \times 10^5$
$c = 10^{-5}$	2	$3.00 \times 10^{-13}$	0.27	$1.03 \times 10^5$
$c = 10^{-4}$	2	$8.94 \times 10^{-13}$	0.28	$1.03 \times 10^5$

**Table 17:** The results for test system 2 based on strategy 8

Parameter	Iteration	Stop criterion	Program execution time (s)	The optimal value of the objective function
$c = 10^{-8}$	2	$6.93 \times 10^{-13}$	0.29	$1.03 \times 10^5$
$c = 10^{-7}$	2	$3.08 \times 10^{-13}$	0.28	$1.03 \times 10^5$
$c = 10^{-6}$	2	$7.38 \times 10^{-13}$	0.33	$1.03 \times 10^5$
$c = 10^{-5}$	2	$3.00 \times 10^{-13}$	0.30	$1.03 \times 10^5$
$c = 10^{-4}$	2	$8.94 \times 10^{-13}$	0.26	$1.03 \times 10^5$

### 4.3 Simulation Analysis

Strategies 1 and 2 are parallel and serial incarnations of the Proximal Point Algorithm (PPA), respectively, wherein the parameter  $c$  is held constant throughout the iterative process. The primary objective is to ascertain the impact of the parameter  $c$  on the convergence efficacy of the Parallel PPA (PPPA) and Serial PPA (SPPA) algorithms. The data presented in Figs. 2, 3, 9 and 10 elucidate that the value of parameter  $c$  exerts a substantial influence on the convergence efficacy of the PPPA and SPPA algorithms.

In practical applications, while an optimal value for parameter  $c$  undoubtedly exists, its precise determination a priori often presents a formidable challenge. Strategies 3 and 4, informed by trend theory, dynamically modulate parameter  $c$  in response to the evolving trends in equation coupling conditions. The aim is to endow the algorithms with enhanced convergence properties and mitigate the convergence impediments arising from suboptimal initial parameter selection. A comparison between Figs. 2 and 4 reveals that strategy 3 significantly attenuates the influence of the initial parameter  $c$  selection on the algorithm's convergence. In Fig. 2, convergence is achieved exclusively with  $c = 10^{-4}$  within the stipulated maximum iterations, whereas Fig. 4 demonstrates rapid convergence across all five prescribed initial  $c$  values. Although disparate initial parameter  $c$  selections in Fig. 4 result in varying iteration counts required for convergence, as anticipated, a greater number of iterations are generally necessitated to adjust or refine parameter  $c$  when its initial value deviates markedly from the optimal one. Notably, the optimal value for parameter  $c$  cannot be predetermined. A similar phenomenon is observed upon comparing Figs. 3 and 5.

The comparison between Figs. 10 and 12 for Test System 2 reveals a perfect alignment, satisfying the stop criteria within a mere 2–4 iterations. The prescribed initial parameter  $c$  value enabled strategy 3 to achieve an exemplary convergence rate, obviating the need for any further trend-based adjustments to parameter  $c$ . A comparison between Figs. 11 and 13 substantiates the same conclusion.

The PPPA algorithm and its refined version, PPPA-TT, pertain to the Jacobi iteration (parallel iteration), whereas the SPPA algorithm and its augmented variant, SPPA-TT, are categorized under Gauss-Seidel iteration (serial iteration). Theoretically, Gauss-Seidel iteration can capitalize on the most recent iterative information, conferring superior convergence attributes compared to the Jacobi iteration paradigm. However, the data proffered in Figs. 2–5 and 10–13 of this study indicate that both the serial and parallel PPA algorithms predicated on PPA theory, along with their respective enhanced versions, manifest nearly identical convergence characteristics. A plausible rationale is that the quadratic term introduced by PPA in the original objective function circumscribes the update amplitude in each iteration. To probe the impact of this quadratic term introduced by PPA on convergence properties, the Alternating Direction Method of Multipliers (ADMM) is introduced.

Strategies 5 and 6, under the premise of the ADMM algorithm, respectively utilize Residual Balancing Theory and trend theory for parameter adjustments, providing a seamless framework for comparing Residual Balancing Theory with trend theory. In the context of Test System 1, trend theory significantly outperforms Residual Balancing Theory in terms of convergence characteristics, as evidenced by the comparison between Figs. 6 and 7. Moreover, this advantage becomes markedly pronounced in Test System 2, with the contrast between Figs. 14 and 15 clearly demonstrating this superiority.

Strategies 4 and 6 are both serial algorithms that utilize trend theory to adjust parameters, providing a seamless framework for comparing the SPPA algorithm with the ADMM algorithm. In the context of Test System 1, the SPPA algorithm significantly outperforms the ADMM algorithm in terms of convergence characteristics, as evidenced by the comparison between Figs. 5 and 7. Concurrently, this advantage is also observed in Test System 2, with the contrast between Figs. 13 and 15 clearly demonstrating this superiority. This also confirms that trend theory can be seamlessly integrated into the ADMM.

Strategies 3 and 8 are both parallel algorithms that employ trend theory to adjust parameters, offering a seamless framework for comparing the PPPA-TT with the APP-TT. In the context of Test System 1, the PPPA-TT slightly surpasses the APP-TT in terms of convergence characteristics, as indicated by the comparison between Figs. 4 and 9. Similarly, in Test System 2, the APP-TT slightly surpasses the PPPA-TT in convergence characteristics, as evident from the contrast between Figs. 12 and 17. Thus, the PPPA-TT and the APP-TT can be considered to have similar convergence characteristics. This also confirms that trend theory can be seamlessly integrated into the APP algorithm.

The comparison between Figs. 8 and 9 shows that Strategy 8 significantly reduces the impact of the initial parameter  $c$  selection on the convergence of the APP algorithm. In Fig. 8, convergence is achieved only with  $c = 10^{-4}$  within the specified maximum number of iterations, whereas Fig. 9 demonstrates rapid convergence across all five prescribed initial  $c$  values. This confirms that trend theory can be seamlessly integrated into the APP algorithm. The comparison between Figs. 16 and 17 for Test System 2 shows perfect alignment, meeting the stop criteria within just 2 iterations. The initial parameter  $c$  value did not trigger the execution of trend theory, with parameter  $c$  remaining unchanged throughout the iteration process.

Tables 2 through 9 present a detailed numerical record of the application effects of Strategies 1 to 8 on Test System 1, as depicted in Figs. 2 through 9, complemented by the optimal values of the objective function upon meeting the stop criterion. Within Tables 2–9, the objective function values are nearly identical across different strategies when the stop criterion is satisfied, thereby corroborating the reliability and accuracy of the algorithmic outcomes. The PPPA-TT and SPPA-TT methods proposed in this paper undoubtedly possess the most optimal convergence properties, characterized by the minimal number of iterations required.

Tables 10 through 17 offer a detailed numerical presentation of the outcomes for strategies 1 to 8 on Test System 2, as illustrated in Figs. 10 through 17, and include the optimal values of the objective function upon meeting the stop criterion. Regarding the optimal values, when the stop criterion is satisfied, strategies 1–4 and strategies 7–8 achieve consistent optimal values that are significantly superior to the optimal solutions obtained by strategies 5–6.

## 5 Conclusion

Multi-Area Economic Dispatch (MAED) problem is congruent with a linearly constrained separable convex optimization problem and can be readily transformed into a saddle point problem. Here, a novel parameter  $c$ , which serves as a penalty term and is predetermined, is introduced. Within the Proximal Point Algorithm (PPA) framework, the customization of the positive definite matrix  $G$  significantly streamlines the development of efficient parallel and serial iterative strategies specifically tailored for the saddle point problem. Contrary to the Alternating Direction Method of Multipliers (ADMM), which is inherently serial, the PPA provides flexibility, enabling adaptation to either parallel or serial iteration as necessitated by the context. The PPA and its enhanced versions incorporate a quadratic term into the objective function; this term, despite potentially retarding numerical updates, aids in stabilizing iterations in the proximity of the optimal solution, thereby achieving faster convergence compared to ADMM. Furthermore, the quadratic term in the PPA simplifies the proofs of convergence relative to ADMM. Moreover, empirical evidence from applications indicates that the convergence efficiency of the proximal point algorithm designed in this study is markedly contingent upon the parameter  $c$ . In a departure from traditional methods, inspired by trend theory, this study introduces modified, variable-parameter algorithms customized for parameter  $c$ . The elegance of trend theory, which only accounts for primal residuals, facilitates seamless integration into Parallel PPA (PPPA), Serial PPA (SPPA), and ADMM. In contrast, the primal-dual residual balancing method, impeded by its necessity for dual residual calculations, cannot be directly applied to other algorithms. Simulation results disclose that PPPA-TT, SPPA-TT, ADMM-TT and PPA-TT exhibit significant

robustness to the initial value of parameter  $c$  and demonstrate superior convergence characteristics when juxtaposed with ADMM-RB. Furthermore, the methodologies presented in this study offer substantial scope for further theoretical exploration. Specifically, this paper achieves positive definiteness of the matrix by constructing a diagonally dominant matrix, which is a sufficient but not necessary condition for ensuring the convergence of the algorithm. Future work can discuss new convergence conditions to enhance the convergence properties of the algorithm. Additionally, the trend theory adjustment strategy presented in this paper is a monotonic adjustment strategy. In future research, bidirectional adjustment strategies can be discussed.

**Acknowledgement:** The authors are grateful to the reviewers for their constructive feedback, which significantly enhanced the quality of this paper.

**Funding Statement:** This research was funded by Guangxi Science and Technology Base and Talent Special Project, grant number GuiKeAD20159077; Foundation of Guilin University of Technology, grant number GLUTQD2018001.

**Author Contributions:** The authors confirm contribution to the paper as follows: study conception and design: Yaming Ren; data collection, analysis and interpretation of results and draft manuscript preparation: Yaming Ren and Xing Deng. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The authors confirm that the data supporting the findings of this study are available within the article.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest to report regarding the present study.

### Abbreviations/Nomenclature

MAED	Multi-area economic dispatch
ADMM	Alternating Direction Method of Multipliers
APP	Auxiliary Problem Principle
PPA	Proximal point algorithm
PPPA	Parallel distributed algorithm based on the PPA
SPPA	Serial distributed algorithm based on the PPA
PPPA-TT	Parallel distributed algorithm based on PPA principle and trend theory
SPPA-TT	Serial distributed algorithm based on PPA principle and trend theory
ADMM-RB	Serial distributed algorithm based on ADMM and residual balancing theory
ADMM-TT	Serial distributed algorithm based on ADMM and trend theory
APP-TT	Parallel distributed algorithm based on APP and trend theory
$\lambda$	Lagrangian multiplier
$c$	Penalty factor
$P_i$	Active output of $i$ -th unit
$P_{i,max}$	The upper limit of $i$ -th unit
$P_{i,min}$	The lower limit of $i$ -th unit
$load_i$	The system load for area $i$
$a_i, b_i$ and $c_i$	The cost coefficient for the $i$ -th unit
$N_i$	The number of units in area $i$
$f(x_1)$ and $g(x_2)$	The generation cost of areas 1 and 2
$T_{12}$	The maximum transfer power flow between area 1 and area 2
$P_{b1}$ and $P_{b2}$	Transfer power flow between area 1 and area 2



## References

1. Visutarrom T, Chiang T. Economic dispatch using metaheuristics: algorithms, problems, and solutions. *Appl Soft Comput.* 2024;150(4):110891. doi:10.1016/j.asoc.2023.110891.
2. Dong J, Wang H, Yang J, Gao L, Wang K, Zhou X. Low carbon economic dispatch of integrated energy system considering power supply reliability and integrated demand response. *Comput Model Eng Sci.* 2022;132(1):319–40. doi:10.32604/cmesci.2022.020394.
3. Liu J, Dai Z, Bo R, Meng F, Ou M. Optimal economic dispatch policy for prosumer with energy storage considering self-consumption demand. *Comput Ind Eng.* 2023;176(1):108853. doi:10.1016/j.cie.2022.108853.
4. Zhu J, Mo X, Xia Y, Guo Y, Chen J, Liu M. Fully-decentralized optimal power flow of multi-area power systems based on parallel dual dynamic programming. *IEEE Trans Power Syst.* 2022;37(2):927–41. doi:10.1109/TPWRS.2021.3098812.
5. Espaa TA, Vassiliadis VS. Higher-order interior point methods for convex nonlinear programming. *Comput Chem Eng.* 2024;180(1):108475. doi:10.1016/j.compchemeng.2023.108475.
6. Zhang C, Jian J, Yang L. Regularised primal-dual interior-point method for dynamic optimal power flow with block-angular structures. *IET Generat, Trans Distribut.* 2020;14(9):1694–704. doi:10.1049/iet-gtd.2019.1528.
7. Lu W, Liu M, Lin S, Li L. Fully decentralized optimal power flow of multi-area interconnected power systems based on distributed interior point method. *IEEE Trans Power Syst.* 2018;33(1):901–10. doi:10.1109/TPWRS.2017.2694860.
8. Ye H, Zhu J, Chen J, Wang Z, Zhuo Y, Liu H, et al. Quadratic taylor expansion-based approximate dynamic programming for fully decentralized AC-OPF of multi-area power systems. *IEEE Trans Power Syst.* 2023;38(5):4940–9. doi:10.1109/TPWRS.2022.3217871.
9. Wang Y, Xiong G, Xu S, Suganthan PN. Large-scale power system multi-area economic dispatch considering valve point effects with comprehensive learning differential evolution. *Swarm Evol Comput.* 2024;89(4):101620. doi:10.1016/j.swevo.2024.101620.
10. Alghamdi AS, Zohdy MA, Aldoihi S. Enhancing renewable energy integration: a gaussian-bare-bones levy cheetah optimization approach to optimal power flow in electrical networks. *Comput Model Eng Sci.* 2024;140(2):1339–70. doi:10.32604/cmesci.2024.048839.
11. Paramguru J, Kumar Barik S, Kumar Barisal A, Dhiman GH, Jhaveri R, Alkahtani M, et al. Addressing economic dispatch problem with multiple fuels using oscillatory particle swarm optimization. *Comput Mater Contin.* 2021;69(3):2863–82. doi:10.32604/cmcc.2021.016002.
12. Ali A, Shah A, Keerio MU, Mugheri NH, Abbas G, Touti E, et al. Multi-objective security constrained unit commitment via hybrid evolutionary algorithms. *IEEE Access.* 2024;12:6698–718. doi:10.1109/ACCESS.2024.3351710.
13. Zhao P, Zhang Y, Hua Q, Li H, Wen Z. Bio-inspired optimal dispatching of wind power consumption considering multi-time scale demand response and high-energy load participation. *Comput Model Eng Sci.* 2023;134(2):957–79. doi:10.32604/cmesci.2022.021783.
14. Girishkumar G, Ganesan S, Jayakumar N, Subramanian S. Black widow optimization for multi area economic emission dispatch. *Intell Automat Soft Comput.* 2023;35(1):609–25. doi:10.32604/iasc.2023.027514.
15. Yin L, Sun Z. Distributed multi-objective grey wolf optimizer for distributed multi-objective economic dispatch of multi-area interconnected power systems. *Appl Soft Comput.* 2022;117(6):108345. doi:10.1016/j.asoc.2021.108345.
16. Sakthivel VP, Goh HH, Srikrishna S, Sathya PD, Abdul Rahim SK. Multi-objective squirrel search algorithm for multi-area economic environmental dispatch with multiple fuels and valve point effects. *IEEE Access.* 2021;9:3988–4007. doi:10.1109/ACCESS.2020.3046257.
17. Chen J, Marrani HI. An efficient new hybrid ICA-PSO approach for solving large scale non-convex multi area economic dispatch problems. *J Electr Eng Technol.* 2020;15(3):1127–45. doi:10.1007/s42835-020-00416-7.
18. Makahleh FM, Amer AA, Manasrah A, Attar HAA, Solyman A, Ahmadi Kamarposhti M. Optimal management of energy storage systems for peak shaving in a smart grid. *Comput Mater Contin.* 2023;75(2):3317–37. doi:10.32604/cmcc.2023.035690.
19. Ellahi M, Abbas G. A hybrid metaheuristic approach for the solution of renewables-incorporated economic dispatch problems. *IEEE Access.* 2020;8:127608–21. doi:10.1109/ACCESS.2020.3008570.

20. Habib S, Kamarposhti MA, Shokouhandeh H, Colak I, Barhoumi E. Economic dispatch optimization considering operation cost and environmental constraints using the HBMO method. *Energy Rep.* 2023;10(3):1718–25. doi:10.1016/j.egy.2023.08.032.
21. Da Silva MV, Deusdado N, Antao AN. Lower and upper bound limit analysis via the alternating direction method of multipliers. *Comput Geotech.* 2020;124(2):103571. doi:10.1016/j.compgeo.2020.103571.
22. Ni R, Pan Z, Gao X. Robust multi-robot trajectory optimization using alternating direction method of multiplier. *IEEE Robot Autom Lett.* 2022;7(3):5950–7. doi:10.1109/LRA.2022.3159848.
23. Anonymous. On augmented lagrangian methods of multipliers and alternating direction methods of multipliers for matrix optimization problems. *Nonlin Funct Anal Appl.* 2022;27(4):869–79. doi:10.22771/nfaa.2022.27.04.13.
24. Oulmelk A, Afraites L, Hadri A, Zaky MA, Hendy AS. Alternating direction multiplier method to estimate an unknown source term in the time-fractional diffusion equation. *Comput Math Appl.* 2024;156(1):195–206. doi:10.1016/j.camwa.2023.12.027.
25. Yang J, Yang X, Zhou Z, Liu Z, Fan M. Graph matching based point correspondence with alternating direction method of multipliers. *Neurocomputing.* 2021;462(4):344–52. doi:10.1016/j.neucom.2021.08.002.
26. Rabago JFT, Hadri A, Afraites L, Hendy AS, Zaky MA. A robust alternating direction method of multipliers numerical scheme for solving geometric inverse problems in a shape optimization setting. *Comput Math Appl.* 2024;175(1):19–32. doi:10.1016/j.camwa.2024.08.034.
27. Liu J, Zhou H, Chen L, Ouyang T. Alternating direction method of multiplier for solving electromagnetic inverse scattering problems. *Int J Microw Wirel Technol.* 2020;12(8):790–6. doi:10.1017/S175907872000015X.
28. Wang Z, Ding T, Mu C, Huang Y, Yang M, Yang Y, et al. An admm-based power system partitioned black-start and parallel restoration method considering high-penetrated renewable energy. *Int J Electr Power Energy Syst.* 2024;155(1):109532. doi:10.1016/j.ijepes.2023.109532.
29. Khojasteh M, Faria P, Vale Z. A distributed robust ADMM-based model for the energy management in local energy communities. *Sustain Energy Grids Netw.* 2023;36(1):101136. doi:10.1016/j.segan.2023.101136.
30. Mak TWK, Chatzos M, Tanneau M, Van Hentenryck P. Learning regionally decentralized ac optimal power flows with admm. *IEEE Trans Smart Grid.* 2023;14(6):4863–76. doi:10.1109/TSG.2023.3251292.
31. Cohen G. Auxiliary problem principle and decomposition of optimization problems. *J Optim Theory Appl.* 1980;32(3):277–305. doi:10.1007/BF00934554.
32. Chakrabarti S, Khajeh H, Nudell TR, Hesamzadeh MR, Baldick R. Transmission investment coordination using MILP lagrange dual decomposition and auxiliary problem principle. *IEEE Trans Energy Markets Policy Regulat.* 2024;2(1):52–62. doi:10.1109/TEMPR.2023.3323944.
33. Du H, Lin T, Li Q, Fu X, Xu X. Decentralized optimal power flow based on auxiliary problem principle with an adaptive core. *Energy Rep.* 2022;8:755–65. doi:10.1016/j.egy.2022.08.152.
34. Di Fazio AR, Risi C, Russo M, De Santis M. Decentralized voltage optimization based on the auxiliary problem principle in distribution networks with ders. *Appl Sci.* 2021;11(10):4509. doi:10.3390/app11104509.
35. Espaas TA, Vassiliadis VS. An interior point framework employing higher-order derivatives of central path-like trajectories: application to convex quadratic programming. *Comput Chem Eng.* 2022;158(1):107638. doi:10.1016/j.compchemeng.2021.107638.
36. Wang H, Chen X. A fast alternating direction method of multipliers algorithm for big data applications. *IEEE Access.* 2020;8:20607–15. doi:10.1109/ACCESS.2020.2967843.
37. He BS, Shen Y. On the convergence rate of customized proximal point algorithm for convex optimization and saddle-point problem. *Sci Sin Math.* 2012;42(5):515–25. doi:10.1360/012011-1049.
38. He BS, Yang H, Wang SL. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *J Optim Theory Appl.* 2000;106(2):337–56. doi:10.1023/A:1004603514434.
39. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundat Trends Mach Learn.* 2010;3(1):1–122. doi:10.1561/22000000016.
40. Sharifian Y, Abdi H. Solving multi-area economic dispatch problem using hybrid exchange market algorithm with grasshopper optimization algorithm. *Energy.* 2023;267(3):126550. doi:10.1016/j.energy.2022.126550.
41. Kumar S, Kumar V, Katal N, Singh SK, Sharma S, Singh P. Multiarea economic dispatch using evolutionary algorithms. *Math Probl Eng.* 2021;2021(7):3577087. doi:10.1155/2021/3577087.