

Research Article

The Extension of Auxiliary Problem Principle for Solving Dynamic Economic Dispatch Problem in Power Systems

Yaming Ren 

College of Mechanical and Control Engineering, Guilin University of Technology, Guilin, China

Correspondence should be addressed to Yaming Ren; renyaming1981@gmail.com

Received 25 February 2022; Revised 4 May 2022; Accepted 19 May 2022; Published 24 June 2022

Academic Editor: Ching-Feng Wen

Copyright © 2022 Yaming Ren. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we study the convex optimization problem with linear constraint, and the objective function is composed of m separable convex functions. Considering the special case where the objective function is composed of two separable convex functions, the auxiliary problem principle (APP) is an effective parallel distributed algorithm for solving the special case. Inspired by the principle of APP, a natural idea to solve separable convex optimization problem with $m \geq 3$ is to extend the method of APP, resulting in the APP-like algorithm. The convergence of the APP-like algorithm is not clear yet. In this paper, we give a sufficient condition for the convergence of the APP-like algorithm. Specifically, the APP algorithm is a special case of the APP-like algorithm when $m = 2$. However, simulation results show that the convergence efficiency of the APP-like algorithm is affected by the selection of penalty parameter. Therefore, we propose an improved APP-like algorithm in this paper. Simulation results show that the improved APP-like algorithm is robust to the selection of penalty parameter and that the convergence efficiency of the improved APP-like algorithm is better when compared with the APP-like algorithm.

1. Introduction

For the following convex problem with linear constrain,

$$\begin{aligned} \min \quad & f_1(x_1) + f_2(x_2), \\ \text{s.t} \quad & A_1x_1 + A_2x_2 = b, \\ & x_1 \in \Omega_1 \text{ and } x_2 \in \Omega_2, \end{aligned} \quad (1)$$

where $f_1: R^{m_1} \rightarrow R$ and $f_2: R^{m_2} \rightarrow R$ are convex function, $x_1 \in R^{m_1}$ and $x_2 \in R^{m_2}$ are closed convex sets, $A_1 \in R^{r \times m_1}$ and $A_2 \in R^{r \times m_2}$ are the given fixed matrices, and $b \in R^r$ is the given fixed vector.

Further analysis of problem (1) shows that the objective function of problem (1) is separable without intersecting variables. Then, a natural idea is whether a splitting algorithm can be adopted to solve problem (1). The alternating direction multiplier method (ADMM) is an effective distributed iteration strategy [1] for solving problem (1) and has been widely used to solve engineering problems [2, 3]. For problem (1), the corresponding ADMM iteration strategy can be expressed as follows [4]:

$$\begin{cases} x_1^{k+1} = \arg \min \left\{ f_1(x_1) + \frac{c}{2} \|A_1x_1 + A_2x_2^k - b\|^2 + \langle -\lambda^k, A_1x_1 \rangle \mid x_1 \in \Omega_1 \right\}, \\ x_2^{k+1} = \arg \min \left\{ f_2(x_2) + \frac{c}{2} \|A_1x_1^{k+1} + A_2x_2 - b\|^2 + \langle -\lambda^k, A_2x_2 \rangle \mid x_2 \in \Omega_2 \right\}, \\ \lambda^{k+1} = \lambda^k - c(A_1x_1^{k+1} + A_2x_2^{k+1} - b), \end{cases} \quad (2)$$

where $\lambda \in R^r$ represents the Lagrange multiplier, $c > 0$ represents penalty parameter, and $\langle \cdot, \cdot \rangle$ denotes the inner product, i.e., $\langle x, x \rangle = x^T x$.

In this paper, we further consider the general case of problem (1), the objective functions are composed of m separable convex functions ($m \geq 3$) with linear constraint.

$$\begin{aligned} \min \quad & \sum_{i=1}^m f_i(x_i), \\ \text{s.t.} \quad & \sum_{i=1}^m A_i x_i = b, \\ & x_i \in \Omega_i \quad i = 1, 2, \dots, m \end{aligned} \quad (3)$$

By comparing problem (1) with problem (3), an intuitive idea is to directly apply ADMM to solve problem (3) resulting in an ADMM-like iteration strategy is shown in the following equation:

$$\left\{ \begin{aligned} x_1^{k+1} &= \arg \min \left\{ f_1(x_1) + \frac{c}{2} \left\| A_1 x_1 + \sum_{j=2}^m A_j x_j^k - b \right\|^2 + \langle -\lambda^k, A_1 x_1 \rangle \mid x_1 \in \Omega_1 \right\}, \\ x_2^{k+1} &= \arg \min \left\{ f_2(x_2) + \frac{c}{2} \left\| A_1 x_1^{k+1} + A_2 x_2 + \sum_{j=3}^m A_j x_j^k - b \right\|^2 + \langle -\lambda^k, A_2 x_2 \rangle \mid x_2 \in \Omega_2 \right\}, \\ x_i^{k+1} &= \arg \min \left\{ f_i(x_i) + \frac{c}{2} \left\| \sum_{j=1}^{i-1} A_j x_j^{k+1} + A_i x_i^k + \sum_{j=i+1}^m A_j x_j^k - b \right\|^2 + \langle -\lambda^k, A_i x_i \rangle \mid x_i \in \Omega_i \right\}, \\ x_m^{k+1} &= \arg \min \left\{ f_m(x_m) + \frac{c}{2} \left\| \sum_{j=1}^{m-1} A_j x_j^{k+1} + A_m x_m^k - b \right\|^2 + \langle -\lambda^k, A_m x_m \rangle \mid x_m \in \Omega_m \right\}, \\ \lambda^{k+1} &= \lambda^k - c \left(\sum_{j=1}^m A_j x_j^{k+1} - b \right), \end{aligned} \right. \quad (4)$$

For problem (1), the ADMM-like method is the classical ADMM on the condition that $m = 2$. It is clear that the classical ADMM is convergent [1]. However, we cannot prove the convergence of ADMM-like iteration strategy when $m \geq 3$ [5]. To tackle the divergence of ADMM-like method, predictive correction ADMM-like iteration strategies ADM-G (ADM with Gaussian back substitution) [6] and ADBC (alternating direction-based contraction method) [7] are proposed. Specifically, for given $w^k = (x_1^k, x_1^k, \dots, x_m^k, \lambda^k)$, the prediction results \tilde{w}^k is generated by ADMM-like method (4) and the correction results w^{k+1} is generated by $w^{k+1} = w^k + \alpha(\tilde{w}^k - w^k)$, where α represents the correction step size (The expressions of parameter α are different in ADM-G and ADBC). The introduction of correction step ensures the convergence of the ADMM-like iterative strategy, but the correction step needs to calculate the inverse matrix, which greatly increases the complexity of the iterative strategy.

ADM-G and ADBC both belong to the Gaussian Seidel iterative scheme. The Gauss-Seidel iterative scheme can use the latest iteration information in solving each sub-problem to obtain a better convergence rate. At the same time, due to the serial computation form of the Gauss-Seidel iterative scheme, the total time consumption of each iteration in the prediction step is equal to the sum of the time consumption of all sub-problems. According to the above description, the Gauss-Seidel iterative scheme requires a long computing time when the problem scale is large (in other words, when the value of m is large). Then, the parallel form is recommended when the problem scale is large. The auxiliary problem principle (APP), which was proposed by G. Cohen in 1980 [8], is an effective parallel distributed algorithm [9]. The iteration strategy of the auxiliary problem principle for solving (1) can be expressed as follows:

$$\begin{cases} x_1^{k+1} = \arg \min \left\{ f_1(x_1) + \frac{\beta}{2} \|A_1 x_1\|^2 - \beta \langle A_1 x_1, A_1 x_1^k \rangle + \langle -\lambda^k + c(A_1 x_1^k + A_2 x_2^k - b), A_1 x_1 \rangle \mid x_1 \in \Omega_1 \right\}, \\ x_2^{k+1} = \arg \min \left\{ f_2(x_2) + \frac{\beta}{2} \|A_2 x_2\|^2 - \beta \langle A_2 x_2, A_2 x_2^k \rangle + \langle -\lambda^k + c(A_1 x_1^k + A_2 x_2^k - b), A_2 x_2 \rangle \mid x_2 \in \Omega_2 \right\}, \\ \lambda^{k+1} = \lambda^k - c(A_1 x_1^{k+1} + A_2 x_2^{k+1} - b), \end{cases} \quad (5)$$

where $\beta > 2c$ is a sufficient condition for the convergence of auxiliary problem principle iterative strategy [10], other symbols have the same meaning as (4). Reference [11] further discussed APP and derive its $O(1/n)$ convergence

rate. In order to facilitate the analysis of the mathematical properties of the auxiliary problem principle, the equivalent form of the APP is given [11].

$$\begin{cases} x_1^{k+1} = \arg \min \left\{ f_1(x_1) + \frac{\beta-c}{2} \|A_1 x_1 - A_1 x_1^k\|^2 + \frac{c}{2} \|A_1 x_1 + A_2 x_2^k - b\|^2 + \langle -\lambda^k, A_1 x_1 \rangle \mid x_1 \in \Omega_1 \right\}, \\ x_2^{k+1} = \arg \min \left\{ f_2(x_2) + \frac{\beta-c}{2} \|A_2 x_2 - A_2 x_2^k\|^2 + \frac{c}{2} \|A_1 x_1^k + A_2 x_2 - b\|^2 + \langle -\lambda^k, A_2 x_2 \rangle \mid x_2 \in \Omega_2 \right\}, \\ \lambda^{k+1} = \lambda^k - c(A_1 x_1^{k+1} + A_2 x_2^{k+1} - b), \end{cases} \quad (6)$$

Inspired by the principle of APP, a natural idea for solving problem (3) is to extend the APP from problem (1) to problem (3), resulting in an APP-like algorithm as follows:

$$\begin{cases} x_1^{k+1} = \arg \min \left\{ f_1(x_1) + \frac{\beta-c}{2} \|A_1 x_1 - A_1 x_1^k\|^2 + \frac{c}{2} \left\| A_1 x_1 + \sum_{j=2}^m A_j x_j^k - b \right\|^2 + \langle -\lambda^k, A_1 x_1 \rangle \mid x_1 \in \Omega_1 \right\}, \\ x_2^{k+1} = \arg \min \left\{ f_2(x_2) + \frac{\beta-c}{2} \|A_2 x_2 - A_2 x_2^k\|^2 + \frac{c}{2} \left\| A_1 x_1^k + A_2 x_2 + \sum_{j=3}^m A_j x_j^k - b \right\|^2 + \langle -\lambda^k, A_2 x_2 \rangle \mid x_2 \in \Omega_2 \right\}, \\ x_i^{k+1} = \arg \min \left\{ f_i(x_i) + \frac{\beta-c}{2} \|A_i x_i - A_i x_i^k\|^2 + \frac{c}{2} \left\| \sum_{j=1}^{i-1} A_j x_j^k + A_i x_i^k + \sum_{j=i+1}^m A_j x_j^k - b \right\|^2 + \langle -\lambda^k, A_i x_i \rangle \mid x_i \in \Omega_i \right\}, \\ x_m^{k+1} = \arg \min \left\{ f_m(x_m) + \frac{\beta-c}{2} \|A_m x_m - A_m x_m^k\|^2 + \frac{c}{2} \left\| \sum_{j=1}^{m-1} A_j x_j^k + A_m x_m^k - b \right\|^2 + \langle -\lambda^k, A_m x_m \rangle \mid x_m \in \Omega_m \right\}, \\ \lambda^{k+1} = \lambda^k - c \left(\sum_{j=1}^m A_j x_j^{k+1} - b \right), \end{cases} \quad (7)$$

The convergence of APP-like algorithm (7) is not clear yet. In this paper, we prove that $\beta > mc$ is a sufficient condition for the convergence of APP-like algorithm.

Specifically, the APP algorithm is a special case of the APP-like algorithm when $m = 2$. This is the first innovation of this paper.

For APP algorithm, the penalty parameter c is a given positive number. For APP-like algorithm, the setting method of the penalty parameter c is same as that in APP algorithm. Simulation results show that the convergence efficiency of APP-like algorithm is affected by the selection of penalty parameter. Therefore, we propose an improved APP-like algorithm in this paper. Compared with APP-like algorithm, the improved APP-like algorithm is robust in terms of the selection of penalty parameter, and the convergence efficiency of the improved APP-like algorithm is better. This is the second innovation of this paper.

2. Preliminaries

For a general convex optimization problem is as follows:

$$\min\{f(x)|x \in \Omega\}, \quad (8)$$

where $\{f(x): R^n \rightarrow R\}$ is the convex function and Ω is the closed convex set. Assuming x^* is the optimal solution of the convex optimization problem (8), it can be shown that point x^* must be in the feasible region, and all feasible directions starting from point x^* are the ascending directions of the convex optimization problem (8). Supposing we use $\nabla f(x)$ to represent the first derivative of function $f(x)$, using $Sf(x)$ to represent all feasible directions of the function $f(x)$ at point x , and using $Sd(x)$ to represent all descending directions of the function $f(x)$ at point x :

$$Sf(x) = \{s \in R^n | s = x' - x, x' \in \Omega\}, \quad (9)$$

$$Sd(x) = \{s \in R^n | s^T \nabla f(x) < 0\}. \quad (10)$$

According to the definition (9 and 10), it can be known that the sufficient and necessary conditions for x^* to be the optimal solution of the convex optimization problem (8) can be expressed as follows:

$$x^* \in \Omega, \quad Sf(x^*) \cap Sd(x^*) = \emptyset. \quad (11)$$

According to the above description, it can be known that solving the convex optimization problem (8) is equivalent to solving the following variational inequality problem:

$$x \in \Omega, \quad (x' - x)^T \nabla f(x) \geq 0, \quad \forall x' \in \Omega. \quad (12)$$

It is clear that problem (8) contains no equality constraint condition, and problem (3) contains an equality constraint condition. Problem (3) can be easily transformed into a mathematical form of problem (8) by means of Lagrange function. The symbol λ in (13) represents the Lagrange multiplier. According to the above description, the variational inequality (VI) can be used to express the first-order optimality condition of the optimization problem (3) [12]. Let $W := \Omega_1 \times \Omega_2 \times \dots \times \Omega_m \times R^r$, solving (3) is equivalent to solving $w^* = (x_1^*, x_2^*, \dots, x_m^*, \lambda^*) \in W$, which satisfies following inequalities:

$$\begin{cases} f_1(x_1) - f_1(x_1^*) + (x_1 - x_1^*)(-A_1^T \lambda^*) \geq 0, \\ f_2(x_2) - f_2(x_2^*) + (x_2 - x_2^*)(-A_2^T \lambda^*) \geq 0, \\ f_m(x_m) - f_m(x_m^*) + (x_m - x_m^*)(-A_m^T \lambda^*) \geq 0, \quad \forall w \in W, \\ (\lambda - \lambda^*) \sum_{i=1}^m A_i x_i^* - b = 0, \end{cases} \quad (13)$$

and the compact form of (13) can be written as follows:

$$f(x) - f(x^*) + (w - w^*)F(w^*) \geq 0, \quad (14)$$

where

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}, \quad f(x) = \sum_{i=1}^m f_i(x_i), \quad w = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \\ \lambda \end{pmatrix}, \quad F(w) = \begin{pmatrix} -A_1^T \lambda \\ -A_2^T \lambda \\ \vdots \\ -A_m^T \lambda \\ \sum_{i=1}^m A_i x_i - b \end{pmatrix}, \quad (15)$$

and $F(w)$ is monotonic operator.

$$(F(u) - F(v))^T (u - v) \geq 0, \quad \forall u, v \in R^n. \quad (16)$$

Definition 1. For the operator F , if the operator F satisfies

Operator F is monotone operator [13].

3. Convergence Analysis of the APP-Like Algorithm

Lemma 1. Let sequence $\{w^k\}$ is generated by APP-like algorithm (7). If $\beta > mc$, then we obtain:

$$\lim_{k \rightarrow \infty} \|A_i x_i^{k+1} - A_i x_i^k\| = 0, \quad (17)$$

$$\lim_{k \rightarrow \infty} A_i x_i^{k+1} = A_i x_i^*, \quad i = 1, 2, \dots, m,$$

$$\lim_{k \rightarrow \infty} \|\lambda^{k+1} - \lambda^k\| = 0, \quad (18)$$

$$\lim_{k \rightarrow \infty} \lambda^{k+1} = \lambda^*.$$

Proof of Lemma 1. According to the description in section 2, solving the sub-problems in (7) is equivalent to solving x_i^{k+1} , which satisfies

$$f_1(x_1) - f_1(x_1^{k+1}) + (x_1 - x_1^{k+1})^T \cdot \left\{ (\beta - c)A_1^T(A_1 x_1^{k+1} - A_1 x_1^k) + cA_1^T \left(A_1 x_1^{k+1} + \sum_{j=2}^m A_j x_j^k - b \right) - A_1^T \lambda^k \right\} \geq 0, \quad \forall x_1 \in \Omega_1, \quad (19)$$

$$f_i(x_i) - f_i(x_i^{k+1}) + (x_i - x_i^{k+1})^T \cdot \left\{ (\beta - c)A_i^T(A_i x_i^{k+1} - A_i x_i^k) + cA_i^T \left(\sum_{j=1}^{i-1} A_j x_j^k + A_i x_i^{k+1} + \sum_{j=i+1}^m A_j x_j^k - b \right) - A_i^T \lambda^k \right\} \geq 0, \quad \forall x_i \in \Omega_i, \quad 2 \leq i \leq m-1. \quad (20)$$

$$f_m(x_m) - f_m(x_m^{k+1}) + (x_m - x_m^{k+1})^T \cdot \left\{ (\beta - c)A_m^T(A_m x_m^{k+1} - A_m x_m^k) + cA_m^T \left(\sum_{j=1}^{m-1} A_j x_j^k + A_m x_m^{k+1} - b \right) - A_m^T \lambda^k \right\} \geq 0, \quad \forall x_m \in \Omega_m. \quad (21)$$

Considering

$$\lambda^{k+1} = \lambda^k - c \left(\sum_{i=1}^m A_i x_i^{k+1} - b \right). \quad (22)$$

Combing (19)–(22), we obtain

$$f(x) - f(x^{k+1}) + (w - w^{k+1})^T \{F(w^{k+1}) + M(w^{k+1} - w^k)\} \geq 0, \quad \forall w \in W, \quad (23)$$

where

$$M = \begin{pmatrix} (\beta - c)A_1^T A_1 & -cA_1^T A_2 & \cdots & -cA_1^T A_m & 0 \\ -cA_2^T A_1 & (\beta - c)A_2^T A_2 & \cdots & -cA_2^T A_m & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -cA_m^T A_1 & -cA_m^T A_2 & \cdots & (\beta - c)A_m^T A_m & 0 \\ 0 & \cdots & 0 & 0 & \frac{1}{c}I_r \end{pmatrix}. \quad (24)$$

By setting $w = w^*$ in (23), we obtain

$$(w^* - w^{k+1})^T M(w^{k+1} - w^k) \geq f(x^{k+1}) - f(x^*) + (w^{k+1} - w^*)^T F(w^{k+1}). \tag{25}$$

Mapping F is monotone, so we have

$$(w^{k+1} - w^*)^T F(w^{k+1}) \geq (w^{k+1} - w^*)^T F(w^*). \tag{26}$$

By combining (14) and (26), we obtain

$$f(x^{k+1}) - f(x^*) + (w^{k+1} - w^*)^T F(w^{k+1}) \geq 0. \tag{27}$$

By combining (25)–(27), we obtain

$$\begin{aligned} & (w^* - w^{k+1})^T M(w^{k+1} - w^k) \geq 0 \\ \Rightarrow & (w^* - w^k + w^k - w^{k+1})^T M(w^{k+1} - w^k) \geq 0 \\ \Rightarrow & (w^* - w^k)^T M(w^{k+1} - w^k) \geq (w^k - w^{k+1})^T M(w^k - w^{k+1}). \end{aligned} \tag{28}$$

By using (28), we obtain

$$\begin{aligned} \|w^k - w^*\|_M^2 - \|w^{k+1} - w^*\|_M^2 &= \|w^k - w^*\|_M^2 - \|w^k - w^* - (w^k - w^{k+1})\|_M^2 \\ &= 2(w^* - w^k)^T M(w^{k+1} - w^k) - \|w^k - w^{k+1}\|_M^2 \geq 2\|w^k - w^{k+1}\|_M^2 - \|w^k - w^{k+1}\|_M^2 \\ &= \|w^k - w^{k+1}\|_M^2. \end{aligned} \tag{29}$$

And (29) can be rewritten as follows:

$$\|v^k - v^*\|_G^2 - \|v^{k+1} - v^*\|_G^2 \geq \|v^k - v^{k+1}\|_G^2, \tag{30}$$

where

$$v = \begin{pmatrix} A_1 x_1 \\ A_2 x_2 \\ \vdots \\ A_m x_m \\ \lambda \end{pmatrix},$$

$$G = \begin{pmatrix} (\beta - c) & -c & \cdots & -c & 0 \\ -c & (\beta - c) & \cdots & -c & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -c & -c & \cdots & (\beta - c) & 0 \\ 0 & \cdots & 0 & 0 & \frac{1}{c} I_r \end{pmatrix}. \tag{31}$$

Considering

$$\|v^k - v^{k+1}\|_G^2 = \sum_{i=1}^m \sum_{j=i+1}^m \|(A_i x_i^k - A_i x_i^{k+1}) - (A_j x_j^k - A_j x_j^{k+1})\|_{cI_r}^2 + \sum_{i=1}^m (\beta - mc) \|A_i x_i^k - A_i x_i^{k+1}\|_{I_r}^2 + \|\lambda^{k+1} - \lambda^k\|_{\frac{1}{c} I_r}^2 \geq 0. \tag{32}$$

For (32), it is clear that if $\beta > mc$, $\|v^k - v^{k+1}\|_G^2 = 0$ is true only and only if $v^k = v^{k+1}$. Therefore, we can say that matrix G is a symmetric positive matrix when $\beta > mc$ and (30) is Fejér monotone [14]. Then, we can obtain

$$\begin{aligned} \lim_{k \rightarrow \infty} \|A_i x_i^{k+1} - A_i x_i^k\| &= 0, \\ \lim_{k \rightarrow \infty} A_i x_i^{k+1} &= A_i x_i^*, \quad i = 1, 2, \dots, m, \end{aligned} \tag{33}$$

$$\lim_{k \rightarrow \infty} \|\lambda^{k+1} - \lambda^k\| = 0 \blacklozenge \lim_{k \rightarrow \infty} \lambda^{k+1} = \lambda^*. \quad (34)$$

Based on above discussion, the proof of Lemma 1 is completed.

According to the description of Lemma 1, it is clear that if all $A_i \blacklozenge i = 1, 2, \dots, m$ are full column rank matrices, then we can obtain $\lim_{k \rightarrow \infty} x_i^{k+1} = x_i^*$ and $\lim_{k \rightarrow \infty} \lambda^{k+1} = \lambda^*$. In other words, the sequence $\{w^k\}$ generated by APP-like algorithm (7) approximates the optimal solution of problem (3) on the

premise that all $A_i \blacklozenge i = 1, 2, \dots, m$ are full column rank matrices and $\beta > mc$. However, in solving practical problems, we cannot guarantee that all $A_i \blacklozenge i = 1, 2, \dots, m$ are full column rank matrices. For the general matrix $A_i \blacklozenge i = 1, 2, \dots, m$, Lemma 2 is given. \square

Lemma 2. *Let sequence $\{w^k\}$ is generated by APP-like algorithm (7). If problem (3) is convex, continuous, differentiable, and satisfies the following equation:*

$$\lim_{k \rightarrow \infty} \|A_i x_i^{k+1} - A_i x_i^k\| = 0, \quad \lim_{k \rightarrow \infty} \|\lambda^{k+1} - \lambda^k\| = \lim_{k \rightarrow \infty} \left\| c \left(\sum_{i=1}^m A_i x_i^{k+1} - b \right) \right\| = 0, \quad (35)$$

then, the sequence $\{w^k\}$ converges to the optimal solution w^* of problem (3).

Proof of Lemma 2. For convex optimization problem, according to the Karloch-Kuhn-Tucker condition (KKT condition), it is very convenient to determine whether the current iteration point is the optimal solution of the convex optimization problem. For problem (3), if the current iteration point $w \in W$ satisfies the following equality constraints:

$$\nabla f_i(x_i) - A_i^T \lambda = 0, \quad i = 1, 2, \dots, m, \quad (36)$$

$$\sum_{i=1}^m A_i x_i - b = 0 \blacklozenge x_i \in \Omega_i, \quad (37)$$

then the current iteration point w is the optimal solution of the optimization problem (3).

The sequence $\{w^k\}$ is generated by APP-like algorithm (7). It is clear that $x_1^{k+1} \in \Omega_1$ satisfy the following equality:

$$\begin{aligned} \nabla f_1(x_1^{k+1}) + (\beta - c)A_1^T(A_1 x_1^{k+1} - A_1 x_1^k) + cA_1^T \left(A_i x_i^{k+1} + \sum_{j=2}^m A_j x_j^k - b \right) - A_1^T \lambda^k &= 0 \\ \Rightarrow \nabla f_1(x_1^{k+1}) - A_1^T \lambda^{k+1} + (\beta - c)A_1^T(A_1 x_1^{k+1} - A_1 x_1^k) + cA_1^T \left(A_i x_i^{k+1} + \sum_{j=2}^m A_j x_j^k - b \right) - A_1^T(\lambda^k - \lambda^{k+1}) &= 0 \\ \Rightarrow \nabla f_1(x_1^{k+1}) - A_1^T \lambda^{k+1} = A_1^T(\lambda^k - \lambda^{k+1}) - (\beta - c)A_1^T(A_1 x_1^{k+1} - A_1 x_1^k) - cA_1^T \left(A_i x_i^{k+1} + \sum_{j=2}^m A_j x_j^k - b \right) & \quad (38) \\ = cA_1^T \left(\sum_{j=1}^m A_j x_j^{k+1} - b \right) - (\beta - c)A_1^T(A_1 x_1^{k+1} - A_1 x_1^k) - cA_1^T \left(\sum_{j=1}^m A_j x_j^{k+1} - b + \sum_{j=2}^m (A_j x_j^k - A_j x_j^{k+1}) \right) \\ = -(\beta - c)A_1^T(A_1 x_1^{k+1} - A_1 x_1^k) - cA_1^T \sum_{j=2}^m (A_j x_j^k - A_j x_j^{k+1}). \end{aligned}$$

Similarly, we can obtain

$$\begin{aligned} \nabla f_i(x_i^{k+1}) - A_i^T \lambda^{k+1} &= -(\beta - c)A_i^T(A_i x_i^{k+1} - A_i x_i^k) \\ &\quad - cA_i^T \left(\sum_{j=1}^{i-1} (A_j x_j^k - A_j x_j^{k+1}) + \sum_{j=i+1}^m (A_j x_j^k - A_j x_j^{k+1}) \right) \blacklozenge x_i \in \Omega_i \blacklozenge 2 \leq i \leq m-1, \end{aligned} \quad (39)$$

and

$$\nabla f_m(x_m^{k+1}) - A_m^T \lambda^{k+1} = -(\beta - c)A_m^T(A_m x_m^{k+1} - A_m x_m^k) - cA_m^T \sum_{j=1}^{m-1} (A_j x_j^k - A_j x_j^{k+1}) \diamond \quad x_m \in \Omega_m. \quad (40)$$

According to the description of Lemma 1, it is clear that,

$$\lim_{k \rightarrow \infty} \|Ax_i^{k+1} - Ax_i^k\| = 0, \quad \lim_{k \rightarrow \infty} \|\lambda^{k+1} - \lambda^k\| = \lim_{k \rightarrow \infty} \left\| c \left(\sum_{i=1}^m A_i x_i^{k+1} - b \right) \right\| = 0. \quad (41)$$

By combining (38)–(41), we can obtain

$$\lim_{k \rightarrow \infty} \nabla f_i(x_i^{k+1}) - A_i^T \lambda^{k+1} = 0, \quad i = 1, 2, \dots, m, \quad (42)$$

$$\lim_{k \rightarrow \infty} \sum_{i=1}^m A_i x_i^{k+1} - b = 0 \diamond \quad x_i \in \Omega_i. \quad (43)$$

Therefore, we can say that sequence $\{w^k\}$ generated by APP-like algorithm (7) converges to the optimal solution w^* of problem (3) when problem (3) is convex, continuous, differentiable, and satisfies (35).

Therefore, the proof of Lemma 2 is completed. \square

4. The Improved APP-Like Algorithm and Convergence Analysis

In Section 3, we prove the convergence of the proposed APP-like algorithm. In this section, we will discuss the parameter selection of the proposed APP-like algorithm and give the improved APP-like algorithm. In addition, we will prove the convergence of the improved APP-like algorithm.

4.1. The Improved APP-Like Algorithm. For the proposed APP-like algorithm, the objective function contains $c/2 \|A_1 x_1 + \sum_{j=2}^m A_j x_j^k - b\|^2$, we can say that $c/2 \|A_1 x_1 + \sum_{j=2}^m A_j x_j^k - b\|^2$ represents the penalty function for the equality constraint $\sum_{i=1}^m A_i x_i = b$ of the original optimization problem. The concept of penalty functions reminds us of the concept of exterior penalty function. To be specific, the solution of the exterior penalty function tends to be optimal solution as the penalty parameter tends to infinity. However, when the penalty parameter tends to infinity, it is easy to

cause the ill of exterior penalty functions. Fortunately, for the proposed APP-like algorithm, the objective function contains not only the penalty function but also the Lagrange function. This method of combining the punishment function and the Lagrange function is called augmented Lagrange function method, and the obvious advantage is that the penalty parameter tends to infinity is not required as the convergence condition of the proposed APP-like algorithm.

According to the description of (30), the sequence $\{v^k\}$ generated by the proposed APP-like algorithm gradually approaches the optimal solution and the penalty functions $c/2 \|A_1 x_1 + \sum_{j=2}^m A_j x_j^k - b\|^2$ gradually approach zero, which means the convergence efficiency of the proposed APP-like algorithm becomes progressively worse when the current iteration point approximates the optimal solution. Inspired by the method of exterior penalty function, we propose the iterative strategy of penalty parameter as follows:

$$c_{k+1} = \begin{cases} 2 * c_k, & k < k_{\text{stop}}, \\ c_k, & k \geq k_{\text{stop}}, \end{cases} \quad (44)$$

where k_{stop} is an integer greater than one. According to the description of (44) and considering the initial penalty parameter $c_0 > 0$, it can be known that the sequence $\{c_k\}$ has upper and lower bounds, which means the iterative strategy of penalty parameter will not cause the ill-conditioned Hessian matrix of objective function.

For APP-like algorithm with $m \geq 3$, the objective function contains $(\beta - c)/2 \|A_i x_i - A_i x_i^k\|^2$ and the parameter β is set to be $\beta > mc$ in order to ensure the convergence of APP-like algorithm. Based on the above discussion, we can give the improved the APP-like algorithm as follows:

$$\left\{ \begin{aligned}
 x_1^{k+1} &= \arg \min \left\{ f_1(x_1) + \frac{\beta_k - c_k}{2} \|A_1 x_1 - A_1 x_1^k\|^2 + \frac{c_k}{2} \left\| A_1 x_1 + \sum_{j=2}^m A_j x_j^k - b \right\|^2 + \langle -\lambda^k, A_1 x_1 \rangle \mid x_1 \in \Omega_1 \right\}, \\
 x_2^{k+1} &= \arg \min \left\{ f_2(x_2) + \frac{\beta_k - c_k}{2} \|A_2 x_2 - A_2 x_2^k\|^2 + \frac{c_k}{2} \left\| A_1 x_1^k + A_2 x_2 + \sum_{j=3}^m A_j x_j^k - b \right\|^2 + \langle -\lambda^k, A_2 x_2 \rangle \mid x_2 \in \Omega_2 \right\}, \\
 x_i^{k+1} &= \arg \min \left\{ f_i(x_i) + \frac{\beta_k - c_k}{2} \|A_i x_i - A_i x_i^k\|^2 + \frac{c_k}{2} \left\| \sum_{j=1}^{i-1} A_j x_j^k + A_i x_i^k + \sum_{j=i+1}^m A_j x_j^k - b \right\|^2 + \langle -\lambda^k, A_i x_i \rangle \mid x_i \in \Omega_i \right\}, \\
 x_m^{k+1} &= \arg \min \left\{ f_m(x_m) + \frac{\beta_k - c_k}{2} \|A_m x_m - A_m x_m^k\|^2 + \frac{c_k}{2} \left\| \sum_{j=1}^{m-1} A_j x_j^k + A_m x_m^k - b \right\|^2 + \langle -\lambda^k, A_m x_m \rangle \mid x_m \in \Omega_m \right\}, \\
 \lambda^{k+1} &= \lambda^k - c_k \left(\sum_{j=1}^m A_j x_j^{k+1} - b \right), \\
 c_{k+1} &= 2 * c_k, \quad k < k_{\text{stop}}, \\
 c_{k+1} &= c_k, \quad k \geq k_{\text{stop}}, \\
 \beta_{k+1} &= (m + \delta)c_{k+1}, \quad \delta > 0.
 \end{aligned} \right. \tag{45}$$

4.2. Convergence Analysis of the Improved APP-Like Algorithm

Lemma 3. Let sequence $\{w^k\}$ is generated by improved APP-like algorithm (45). If problem (3) is convex, continuous, and differentiable, then sequence $\{w^k\}$ converges to the optimal solution w^* of problem (3).

Proof of Lemma 3

For given w^k , w^{k+1} is generated by the improved APP-like algorithm. Based on the description of the improved APP-like algorithm, it is clear that $c_k = c_0 \times 2^{k_{\text{stop}}}$ and $\beta_k = c_k \times 2^{k_{\text{stop}}} (m + \delta)$ when $k \geq k_{\text{stop}}$. According to the proof of Lemma1, if $k \geq k_{\text{stop}}$, then we can get

$$\|v^k - v^*\|_M^2 - \|v^{k+1} - v^*\|_M^2 \geq \|v^k - v^{k+1}\|_M^2, \tag{46}$$

where

$$v = \begin{pmatrix} A_1 x_1 \\ A_2 x_2 \\ \vdots \\ A_m x_m \\ \lambda \end{pmatrix}, \tag{47}$$

$$M = \begin{pmatrix} (\beta_k - c_k) & -c_k & \cdots & -c_k & 0 \\ -c_k & (\beta_k - c_k) & \cdots & -c_k & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -c_k & -c_k & \cdots & (\beta_k - c_k) & 0 \\ 0 & \cdots & 0 & 0 & \frac{1}{c_k} I_r \end{pmatrix}, \quad c^k = c^0 \times 2^{k_{\text{stop}}}, \beta^k = c^0 \times 2^{k_{\text{stop}}} (m + \delta).$$

If $k \geq k_{stop}$, the matrix M is a fixed symmetric positive definite matrix. Similar to the analysis of (30), we can obtain (46) Fejér monotone. Then, we can obtain

$$\lim_{k \rightarrow \infty} \|A_i x_i^{k+1} - A_i x_i^k\| = 0, \quad \lim_{k \rightarrow \infty} A_i x_i^{k+1} = A_i x_i^*, \quad i = 1, 2, \dots, m, \quad (48)$$

$$\lim_{k \rightarrow \infty} \|\lambda^{k+1} - \lambda^k\| = 0 \blacklozenge \lim_{k \rightarrow \infty} \lambda^{k+1} = \lambda^*. \quad (49)$$

Based on the description of Lemma 2, it is clear that sequence $\{w^k\}$ converges to the optimal solution w^* of problem (3) if it is convex, continuous, and differentiable.

Therefore, the proof of Lemma 3 is completed. \square

5. Application

In this section, we firstly give the mathematical model of the power system dynamic economic dispatching problem (DEDP) and point out that the mathematical model for DEDP can be easily transformed into the multiseparable convex problem with linear constraints studied in this paper. After that, we use the proposed APP-like algorithm and improved APP-like algorithm to solve DEDP and analyze the corresponding simulation results. We coded by the MATLAB 2018b, and all codes were implemented on Dell T5820 Server with Xeon- W-2102 CPU at 2.9 GHz and 16G of memory.

5.1. The Mathematical Model of the Dynamic Economic Dispatching Problem for Power System. The mathematical model of dynamic economic dispatching problem for power system can be expressed as follows:

$$\begin{aligned} \min \quad & F = \sum_{h=1}^H \sum_{i=1}^N f(P_{i,h}) = \sum_{h=1}^H \sum_{i=1}^N (a_i P_{i,h}^2 + b_i P_{i,h} + c_i), \\ \text{s.t} \quad & \sum_{i=1}^N P_{i,h} = P_{D,h}, \quad h = 1, 2, \dots, H, \\ & P_{i,\min} \leq P_{i,h} \leq P_{i,\max}, \quad h = 1, 2, \dots, H, \\ & P_{i,h} - P_{i,h-1} \leq r_u^i, \quad h = 2, \dots, H, \\ & P_{i,h-1} - P_{i,h} \leq r_d^i, \quad h = 2, \dots, H, \end{aligned} \quad (50)$$

where F represents the objective function, N represents the total number of power generation unit, H represents the total dispatching period of power generation unit, $P_{i,h}$ represents the output of the i th power generation unit in the h th dispatching period, a_i , b_i , and c_i are given parameters in objective function, $P_{D,h}$ represents the system load in the h th dispatching period, and r_u^i and r_d^i are the upper and lower ramp rate limits for the i th generation unit and are given.

For problem (50), the equivalent mathematical model can be rewritten as follows:

$$\begin{aligned} \min \quad & \sum_{i=1}^H F_i(x_i), \\ \text{s.t} \quad & \sum_{i=1}^N A_i x_i = b, \\ & x_i \in \Omega_i \quad i = 1, 2, \dots, N, \end{aligned} \quad (51)$$

where

$$A_i = I_{N \times N}, \quad (52)$$

$$b = [P_{D,1}, P_{D,2}, \dots, P_{D,H}]^T, \quad (53)$$

$$x_i = [P_{i,1}, P_{i,2}, \dots, P_{i,H}]^T, \quad i = 1, 2, \dots, N, \quad (54)$$

$$\begin{aligned} \Omega_i = \{ & x_i | P_{i,\min} \leq P_{i,h} \leq P_{i,\max}, \\ & h = 1, 2, \dots, H; P_{i,h} - P_{i,h-1} \\ & \leq r_u^i, P_{i,h-1} - P_{i,h} \leq r_d^i, \quad h = 2, \dots, H \}, \end{aligned} \quad (55)$$

$$F_i(x_i) = \sum_{h=1}^H (a_i P_{i,h}^2 + b_i P_{i,h} + c_i). \quad (56)$$

It is clear that A_i in (52) is full column rank matrices. Therefore, we can directly use the APP-like algorithm and improved APP-like algorithm to solve problem (51) in the form of distributed parallel.

5.2. Test System. In this section, a test system [15] containing ten generation units is presented to verify the validity and correctness of the proposed algorithms. The data refer to generation units are shown in Table 1. In addition, the demand of the ten-unit test system was divided into 24 intervals in reference [15]. Reference [16] employs the demand data in 1–8 intervals as the data refer to system demand. In this paper, we also employ the demand data in 1–8 intervals as in reference [16] and the data refer to system demand are shown in Table 2. For the test system, according to the description in Section 5.1, the original optimization problem can be decomposed into ten sub-optimization problems. The information shown in Table 1 shows that the output upper and lower limits, for the 10th generation unit, are both 55. Therefore, in the actual simulation process, we can directly set the output of the 10th generation unit to be 55, and only solve nine sub-optimization problems.

TABLE 1: The data refer to generation units [15].

Unit	P max (MW)	P min (MW)	a (\$/MW2h)	b (\$/MWh)	c (\$/h)	Ur (MW/hour)	Dr (MW/hour)
1	470	150	0.00043	21.6	958.2	80	80
2	460	135	0.00063	21.05	1313.6	80	80
3	340	73	0.00039	20.81	604.07	80	80
4	300	60	0.0007	23.9	471.6	50	50
5	243	73	0.00079	21.620	480.29	50	50
6	160	57	0.00056	17.87	601.75	50	50
7	130	20	0.00211	16.51	502.7	30	30
8	120	47	0.0048	23.23	639.4	30	30
9	80	20	0.10908	19.58	455.6	30	30
10	55	55	0.00951	22.54	692.4	30	30

TABLE 2: The data refer to system demand [16].

Period	1	2	3	4	5	6	7	8
Load (MW)	1036	1110	1258	1406	1480	1628	1702	1776

5.3. Simulation Analysis

5.3.1. APP-Like Algorithm for Solving Test System. In this section, we use APP-like algorithm to solve the test system and observe the influence of parameter on the convergence efficiency of APP-like algorithm. We set the maximum number of iterations as 50. For the iterative strategy of penalty parameter shown in (44), we set $k_{\text{stop}} = 30$ and $\delta = 0.1$. The convergence condition is set to $\max(|w^{k+1} - w^k|) < 0.0001$. If the convergence condition is not satisfied when the number of iterations is greater than 50, then we can say that the corresponding algorithm fail to convergence with maximum iterations.

The simulation results based on the APP-like algorithm for the test system are shown in Table 3. In order to verify the influence of different penalty parameter settings on the convergence efficiency of the APP-like algorithm, we designed five different penalty parameter settings named Case1–Case5, respectively, which are shown in the first column in Table 3. After that, the second column represents the initial parameter setting of the APP-like algorithm, the third column represents the number of iterations when convergence conditions are met, and the fourth column represents the value of $\max(|w^{k+1} - w^k|)$ when convergence conditions is met or the number of iterations reaches the maximum iterations. In addition, the word “Fail” in Table 3 represents the corresponding algorithm fail to convergence with maximum iterations. The curve of $\max(|w^{k+1} - w^k|)$ changing with the number of iterations is shown in Figure 1.

According to the data in Table 3 and the curve in Figure 1, it can be known that the APP-like algorithm meets the convergence condition only under case5. By comparison, the APP-like algorithm fails to converge or stops convergence prematurely under case1–case4. For the further analysis of Figure 1 and Table 3, we find that the value of $\max(|w^{k+1} - w^k|)$ gradually approaches the convergence condition with the increase of the penalty parameter. Let’s recall the iterative formula (7) of APP-like algorithm. Taking the i th sub-optimization problem as an example, there are two terms $(\beta - c)/2 \|A_i x_i - A_i x_i^k\|^2$ and $c/2 \|\sum_{j=1}^{i-1} A_j x_j^k + A_i x_i^k - b\|^2$

related to the penalty parameter in the objective function. $c/2 \|\sum_{j=1}^{i-1} A_j x_j^k + A_i x_i^k - b\|^2$ represents the punishment for the equality constraint. For case1–case4 in Table 3 and Figure 2, at the initial stage of iteration, it is obvious that $\max(|w^{k+1} - w^k|)$ is large, so $c/2 \|\sum_{j=1}^{i-1} A_j x_j^k + A_i x_i^k - b\|^2$ can play a good punishment role. However, as $\max(|w^{k+1} - w^k|)$ approaches zero gradually, the penalty effect of $c/2 \|\sum_{j=1}^{i-1} A_j x_j^k + A_i x_i^k - b\|^2$ decreases gradually when the penalty parameter is set smaller. This explains the reason why case1–case4 in Table 3 and Figure 2 fail to convergence.

The existence of $(\beta - c)/2 \|A_i x_i - A_i x_i^k\|^2$ is to ensure the convergence of the APP-like algorithm. However, we must realize that the existence of $(\beta - c)/2 \|A_i x_i - A_i x_i^k\|^2$ also prevents the update of the current iteration point. Therefore, we expect $(\beta - c)/2 \|A_i x_i - A_i x_i^k\|^2$ to be as small as possible under the premise of ensuring the convergence of the APP-like algorithm. In fact, for the APP-like algorithm, the value of penalty parameters and β/c is given. That is to say that the APP-like algorithm ignores the role of $(\beta - c)/2 \|A_i x_i - A_i x_i^k\|^2$ in the iterative process on the premise of ensuring algorithm convergence.

5.3.2. Improved APP-Like Algorithm for Solving Test System.

In this section, we use improved APP-like algorithm to solve the test system and observe the influence of parameter on convergence efficiency of improved APP-like algorithm. In order to comparison with the results of APP algorithm, the initial parameter settings of improved APP-like algorithm are the same as those of the APP algorithm. The simulation results based on improved APP-like algorithm for the test system are shown in Table 4. The columns in Table 4 have the same meaning as the columns in Table 3. The curve of $\max(|w^{k+1} - w^k|)$ based on improved APP-like algorithm changing with iterations is shown in Figure 2.

According to the information shown in Table 4 and Figure 2, we must admit that the trends of these curves of $\max(|w^{k+1} - w^k|)$ are not entirely consistent. However, the curves of $\max(|w^{k+1} - w^k|)$ with different initial penalty

TABLE 3: The simulation results based on APP-like algorithm for test system.

Case	Parameter	Iteration	$\max(w^{k+1} - w^k)$
Case1	$c = 1, \beta = 9.1c$	Fail	0.3987
Case2	$c = 10, \beta = 9.1c$	Fail	0.0589
Case3	$c = 100, \beta = 9.1c$	Fail	0.0061
Case4	$c = 1000, \beta = 9.1c$	Fail	6.1481×10^{-4}
Case5	$c = 10000, \beta = 9.1c$	22	6.2349×10^{-5}

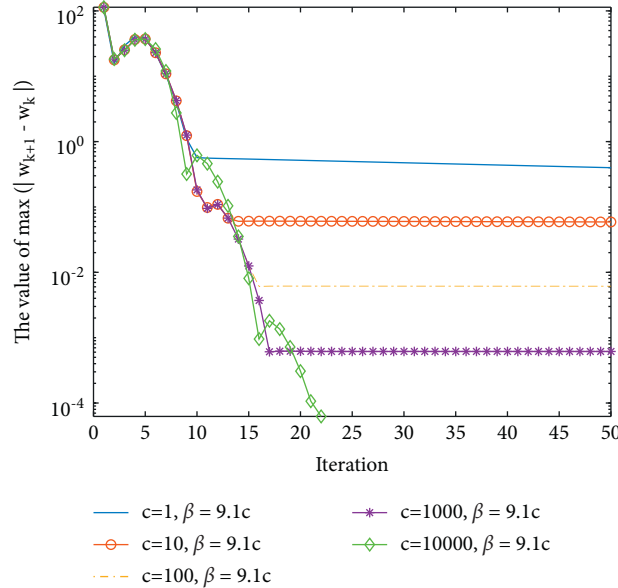


FIGURE 1: The curve of $\max(\text{abs}(w^{k+1} - w^k))$ based on the APP-like algorithm.

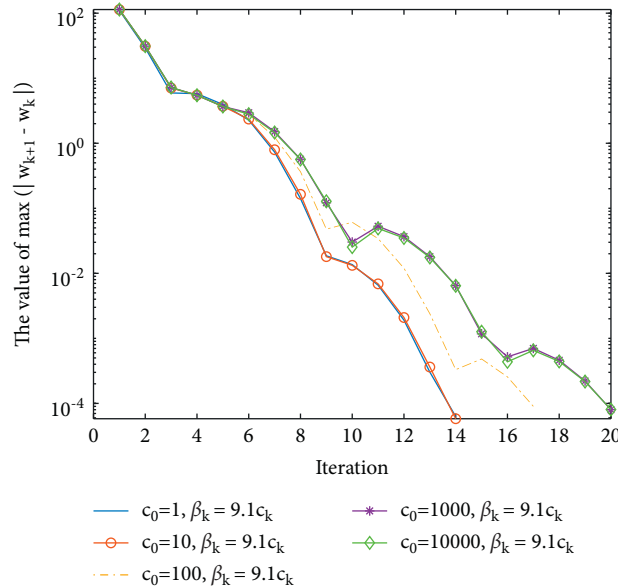


FIGURE 2: The curve of $\max(\text{abs}(w_{k+1} - w_k))$ based on improved APP-like algorithm.

parameter values maintain similar trends, and can quickly meet convergence condition. That is to say, for improved APP-like algorithm, the selection of penalty parameters is robust to a certain extent.

The difference between APP-like algorithm and improved APP-like algorithm lies in the updating of penalty parameters. For APP-like algorithm, the penalty parameter is given in advance and remains unchanged in the iterative

TABLE 4: The simulation results based on improved APP-like algorithm for test system.

Case	Parameter	Iteration	$\max(w^{k+1} - w^k)$
Case1	$c_0 = 1, \beta_k = 9.1c_k$	14	6.2929×10^{-5}
Case2	$c_0 = 10, \beta_k = 9.1c_k$	14	5.8110×10^{-5}
Case3	$c_0 = 100, \beta_k = 9.1c_k$	17	9.0122×10^{-5}
Case4	$c_0 = 1000, \beta_k = 9.1c_k$	20	8.0008×10^{-5}
Case5	$c_0 = 10000, \beta_k = 9.1c_k$	20	8.0579×10^{-5}

TABLE 5: The simulation results based on improved APP-like algorithm with different β .

Case	Parameter	Iteration	$\max(w^{k+1} - w^k)$
Case1	$c_0 = 10, \beta_k = 9.1c_k$	14	6.2929×10^{-5}
Case2	$c_0 = 100, \beta_k = 20c_k$	26	8.9700×10^{-5}
Case3	$c_0 = 1000, \beta_k = 30c_k$	Fail	1.6000×10^{-3}
Case4	$c_0 = 1, \beta_k = 40c_k$	Fail	7.0000×10^{-2}

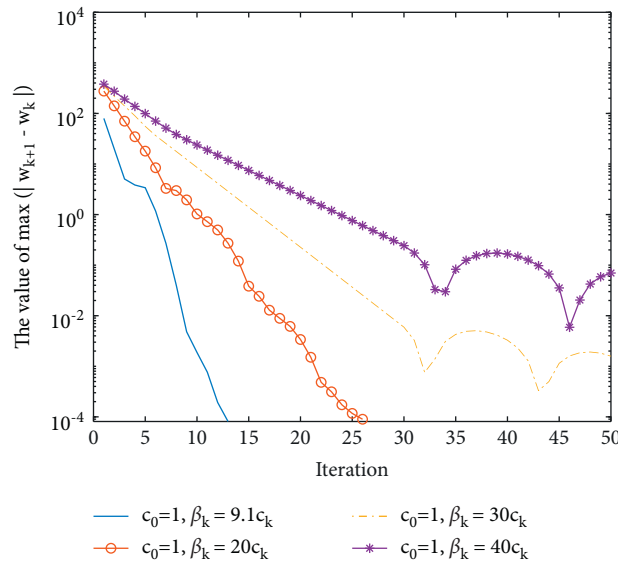


FIGURE 3: The curve of $\max(|w^{k+1} - w^k|)$ based on improved APP-like algorithm with different β .

process. For improved APP-like algorithm, the initial value of penalty parameters is given, but it is updated (The iterative strategy of penalty parameter can be found in (44)) in the iterative process.

According to the analysis of APP-like algorithm in section 5.3.1, it can be found that APP-like algorithm has achieved better convergence effect by setting a larger penalty parameter value to ensure that $c/2\|\sum_{j=1}^{i-1} A_j x_j^k + A_i x_i^k + \sum_{j=i+1}^m A_j x_j^k - b\|^2$ plays an effective punishment role. Comparatively speaking, improved APP-like algorithm has achieved better convergence effect by updating the penalty parameter to ensure that $c/2\|\sum_{j=1}^{i-1} A_j x_j^k + A_i x_i^k + \sum_{j=i+1}^m A_j x_j^k - b\|^2$ plays an effective punishment role. Both methods can achieve better convergence characteristics, and we cannot say which algorithm is better.

Let's turn our attention to $(\beta - c)/2\|A_i x_i - A_i x_i^k\|^2$. APP-like algorithm ignores the role of $(\beta - c)/2\|A_i x_i - A_i x_i^k\|^2$ because the parameters β and c are given in advance. In fact, we expect $(\beta - c)/2\|A_i x_i - A_i x_i^k\|^2$ to be as small as possible under the premise of ensuring the convergence of the algorithm.

According to the information shown in Table 4 and Figure 2, obviously, the smaller the initial value of penalty parameter is set, the better the convergence characteristic of improved APP-like algorithm will be. According to the description of iteration strategy (44) for penalty parameters, it can be known that penalty parameter has upper and lower bounds in the iterative process. For the improved APP-like algorithm, if we give a small initial value of the penalty parameter, the value of $(\beta_k - c_k)/2\|A_i x_i - A_i x_i^k\|^2$ will be smaller throughout the iteration. This is why, in Figure 2, the smaller the initial value of penalty parameter is, the better the convergence efficiency is.

In the above part, we analyze the influence of penalty parameter on the convergence efficiency of the improved APP-like algorithm. If we further analyze $(\beta_k - c_k)/2\|A_i x_i - A_i x_i^k\|^2$, we will find that the value of $(\beta_k - c_k)/2\|A_i x_i - A_i x_i^k\|^2$ is related not only to the punishment parameter but also to the parameter β_k . The simulation results based on APP-like algorithm for the test system with different β_k/c_k are shown in Table 5. The corresponding curve of

$\max(|w^{k+1} - w^k|)$ is shown in Figure 3. According to the information shown in Table 5 and Figure 3, it is clear that, for the same initialization penalty parameter, the smaller β_k/c_k is, the better the convergence efficiency of APP algorithm is under the premise of ensuring the convergence of improved APP-like algorithm. The simulation results shown in Table 5 and Figure 3 are in agreement with our theoretical analysis of $(\beta_k - c_k)/2\|A_i x_i - A_i x_i^k\|^2$.

According to the theoretical analysis in Section 4 and the simulation results in Section 5, we suggest using the improved APP-like algorithm. Compared with the APP-like algorithm, the penalty parameter selection of improved APP-like algorithm is robust and the convergence efficiency of improved APP-like algorithm is better. Our suggestions for the initial parameters of improved APP-like algorithm are shown as follows:

- (1) It is better to select a smaller initial value of the penalty parameter, and 1 or less is suggested as the initial value of the penalty parameter.
- (2) On the premise of ensuring convergence, it is recommended to select parameter β_k as small as possible, and $\beta_k = (m + 0.1)c_k$ is suggested, where m presents the objective functions, which is composed of m separable convex functions.

6. Conclusion

In this paper, we study the convex optimization problem with linear constraint and the objective function is composed of m separable convex functions. According to the properties of the objective function, it is natural to consider a distributed iterative strategy. ADM-G and ADPC are efficient algorithms to solve the optimization problem in the form of the Gauss-Seidel iterative scheme. Due to the serial computation form of Gauss-Seidel iterative scheme, ADM-G and ADPC needs more computing time when the problem scale is large (in other words, when the value of m is large). Then, a natural idea is to solve optimization problem in parallel. The auxiliary problem principle (APP) is an effective parallel distributed algorithm for solving separable convex optimization problem with $m = 2$. Inspired by the principle of APP, a natural idea is to extend the method of APP, resulting in an APP-like algorithm. Therefore, we firstly prove that $\beta > mc$ is a sufficient condition for the convergence of APP-like algorithm. However, simulation results show the convergence efficiency of APP-like algorithm is sensitive to the selection of penalty parameter. In order to overcome the deficiency of APP-like algorithm, we proposed the improved APP-like algorithm with variable penalty parameter. Compared with APP-like algorithm, the improved APP-like algorithm is robust in terms of the selection of penalty parameter, and the convergence efficiency of improved APP-like algorithm is better.

Data Availability

The data in this paper are generated by MATLAB, and the MATLAB codes are available any time if needed from author Yaming Ren (renyaming1981@gmail.com).

Conflicts of Interest

There are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Guangxi Science and Technology Base and Talent Special Project (grant number: GuiKeAD20159077) and the Foundation of the Guilin University of Technology (grant number: GLUTQD2018001).

References

- [1] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.
- [2] J. Xu, H. Sun, and C. J. Dent, "ADMM-based distributed OPF problem meets stochastic communication delay," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5046–5056, 2019.
- [3] D. Phan and J. Kalagnanam, "Some efficient optimization methods for solving the security-constrained optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 863–872, 2014.
- [4] Y. Rens, S. Fei and H. Wei, "The self-adaptive alternating direction method for the multiarea economic dispatch problem," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 24, pp. 4611–4622, 2016.
- [5] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Mathematical Programming*, vol. 155, no. 1–2, pp. 57–79, 2016.
- [6] B. HeM. Tao and X. Yuan, "Alternating direction method with Gaussian back substitution for separable convex programming," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 313–340, 2012.
- [7] B. S. He, M. Tao, M. H. Xu, and X. M. Yuan, "An alternating direction-based contraction method for linearly constrained separable convex programming problems," *Optimization*, vol. 62, no. 4, pp. 573–596, 2013.
- [8] G. Cohen, "Auxiliary problem principle and decomposition of optimization problems," *Journal of Optimization Theory and Applications*, vol. 32, no. 3, pp. 277–305, 1980.
- [9] A. Khalatbarisoltani, M. Kandidayeni, L. Boulon, and X. Hu, "Power allocation strategy based on decentralized convex optimization in modular fuel cell systems for vehicular applications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14563–14574, 2020.
- [10] C. Beltran and F. J. Heredia, "Unit commitment by augmented Lagrangian relaxation: testing two decomposition approaches," *Journal of Optimization Theory and Applications*, vol. 112, no. 2, pp. 295–314, 2002.
- [11] Y. Ren and Z. Chen, "On the $O(1/n)$ convergence rate of the auxiliary problem principle for separable convex programming and its application to the power systems multi-area economic dispatch problem," *Mathematical and Computational Applications*, vol. 21, no. 3, pp. 35–11, 2016.
- [12] B. H. M. Tao and X. Yuan, "A splitting method for separable convex programming," *IMA Journal of Numerical Analysis*, vol. 35, no. 1, pp. 394–426, 2015.
- [13] H. E. Bingsingh and S. Yuan, "On the convergence rate of customized proximal point algorithm for convex optimization

- and saddle-point problem,” *Scientia Sinica Mathematica*, vol. 42, no. 5, pp. 515–525, 2012.
- [14] Y. Shen, “Some first-order algorithms for structured optimizations,” Nanjing University, Nanjing, China, Ph.D, 2012.
- [15] P. Attaviriyanupap, H. Kita, E. Tanaka, and J. Hasegawa, “A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function,” *IEEE Power Engineering Review*, vol. 22, no. 4, pp. 77–416, 2002.
- [16] Y. Ren, “Some properties for adjustable entropy function method and its application to power systems dynamic economic dispatch with wind integrated,” *Journal of Electrical and Computer Engineering*, vol. 2021, 12 pages, Article ID 2497209, 2021.