

Research Article

Predictor-Corrector Primal-Dual Interior Point Method for Solving Economic Dispatch Problems: A Postoptimization Analysis

**Antonio Roberto Balbo,¹ Márcio Augusto da Silva Souza,²
Edméa Cássia Baptista,¹ and Leonardo Nepomuceno³**

¹ *Department of Mathematics, Universidade Estadual Paulista (UNESP), 17033-360 Bauru, SP, Brazil*

² *Graduate Program in Electrical Engineering, Universidade Estadual Paulista (UNESP),
17033-360 Bauru, SP, Brazil*

³ *Department of Electrical Engineering, UNESP—Universidade Estadual Paulista,
17033-360 Bauru, SP, Brazil*

Correspondence should be addressed to Leonardo Nepomuceno, leo@feb.unesp.br

Received 9 November 2011; Accepted 3 April 2012

Academic Editor: Jianming Shi

Copyright © 2012 Antonio Roberto Balbo et al. 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.

This paper proposes a predictor-corrector primal-dual interior point method which introduces line search procedures (IPLS) in both the predictor and corrector steps. The Fibonacci search technique is used in the predictor step, while an Armijo line search is used in the corrector step. The method is developed for application to the economic dispatch (ED) problem studied in the field of power systems analysis. The theory of the method is examined for quadratic programming problems and involves the analysis of iterative schemes, computational implementation, and issues concerning the adaptation of the proposed algorithm to solve ED problems. Numerical results are presented, which demonstrate improvements and the efficiency of the IPLS method when compared to several other methods described in the literature. Finally, postoptimization analyses are performed for the solution of ED problems.

1. Introduction

Since its introduction in 1984, the projective transformation algorithm proposed by Karmarkar in [1] has proved to be a notable interior point method for solving linear programming problems (LPPs). This pioneer study caused an upheaval in research activities in this area. Among all the variations of Karmarkar's original algorithm, the first to attract the attention of researchers was the one that uses a simple affine transformation to replace Karmarkar's original and highly complex projective transformation, enabling work with

LPP in its standard form. The affine algorithm was first introduced in [2] by Dikin, a Soviet mathematician. Later, in 1986, the work was independently rediscovered by Barnes in [3] and by Vanderbei et al. in [4]. They proposed using the primal-affine algorithm to solve LPP in standard form and also presented proof of the algorithm's convergence. A similar algorithm, called dual-affine, was developed and implemented by Adler et al. [5] to solve the LPP in the form of inequality. Compared to the relatively cumbersome projective transformation, the implementation of the primal-affine and dual-affine algorithms was simpler because of its direct relationship with the LPP. These two algorithms produced promising results when applied to large problems [6], although the theoretical proof of polynomial time complexity was not obtained from the affine transformation. Megiddo and Shub's study in [7] indicated that the trajectory that leads to the optimal solution provided by affine algorithms depends on the initial solution. A poor initial solution, which is close to a viable domain vertex, could result in an investigation that covers all vertex problems.

Nevertheless, the polynomial time complexity of primal-affine and dual-affine algorithms can be re-established by incorporating a logarithmic barrier function into the objective function of the original LPP. The purpose of this procedure is to solve the problem pointed out by Megiddo, that is, to prevent an interior solution from becoming "trapped" at the border of the problem (possibly a vertex). This procedure also provides proof of the complexity of the method. The idea of using the logarithmic barrier function method for convex programming problems was developed by Fiacco and McCormick in [8, 9], based on the method proposed by Frisch in [10]. After the introduction of Karmarkar's algorithm in 1984, the logarithmic barrier function method was reconsidered to solve linear programming problems. Gill et al. used this method in [11] to develop a projected Newton barrier method and demonstrated its equivalence with Karmarkar's projective algorithm. The methods proposed, among others, by Ye in [12], Renegar in [13], Vaidya in [14], and Megiddo in [15], as well as those of a central trajectory—called path-following methods, which were proposed by Gonzaga in [16, 17] and Monteiro and Adler in [18], use the objective function augmented by the logarithmic barrier function. A third variation, the so-called primal-dual interior point algorithm, was introduced by Monteiro et al. in [19] and by Kojima et al. in [20]. This algorithm explores a potential primal-dual function, a variation of the logarithmic barrier function, called the potential function. The polynomial time complexity theory was successfully demonstrated by Kojima et al. in [20] and Monteiro et al. in [19] based on Megiddo's work, which provided a theoretical analysis for the logarithmic barrier method and proposed the primal-dual approach.

The predictor-corrector procedure initially defined by Mehrotra and Sun in [21] and implemented in by Lustig et al. in [22] explored variant directions of the primal-affine interior point methods in the predictor step. In the corrector step, the point previously obtained (in the predictor step) was "centralized" to exploit the potential function related to the logarithmic barrier function. This procedure significantly improved the performance of the primal-dual interior point methods.

This strategy was reviewed and modified by Wu et al. in [23] and successfully applied to solve optimal power flow problems. Wu used the logarithmic barrier penalization, the Newton method, and first-order approximations in the predictor step to determine search directions and the approximate solution. Second-order approximations were considered in the corrector step to refine solutions obtained in the predictor step.

The methods related to the primal-dual interior point methodology, especially those proposed by Kojima et al. in [20], Monteiro and Adler in [18], and Monteiro et al. in [19], which were broadly investigated by Fang and Puthenpura in [24], have been explored in this past decade to solve linear, nonlinear, and integer mathematical programming problems in several fields of research.

This paper proposes a primal-dual interior point method that uses the predictor-corrector strategy described by Lustig et al. in [22] and Wu et al. in [23] and incorporates line search procedures in both the predictor and corrector steps. Theoretical aspects as well as iterative schemes and computational implementation are investigated. A Fibonacci line search procedure is carried out in the predictor step and an Armijo line search is used in the corrector step to calculate the step sizes while taking into account constraints in the variables of the problem. The line search procedures adopted are aimed at improving the overall convergence of the proposed method for solving quadratic programming problems. The method is applied to solve the economic dispatch (ED) problem, a classical quadratic problem studied in the field of electrical power systems. The results obtained with the proposed method are compared with those obtained by several others described in the literature for solving ED problems, such as the primal-dual method described in [6], evolutionary algorithms found in [25–27], genetic and coevolutionary genetic algorithms described by Samed in [27], the cultural algorithm described by Rodrigues in [26], and a hybrid atavistic genetic algorithm given in [25]. This comparative investigation demonstrates the efficiency of the proposed IPLS method.

This paper is organized as follows: Section 2 presents the ED problem; Section 3 develops the theory of the proposed IPLS method and presents its algorithm; Section 4 describes a computational implementation of the method, applying it to solve ED problems. This section also includes a postoptimization analysis. Finally, conclusions are drawn in Section 5.

2. The Economic Dispatch Problem

The economic dispatch problem (ED) is defined as an optimal allocation process of electricity demands among available generating units, where operational constraints must be satisfied while minimizing generation costs. Happ reported in [28] that, by 1920, several engineers had become aware of the economic allocation problem. According to the aforementioned author, one of the first methods employed to solve the ED problem was to request power from the most efficient unit (the merit order loading method). This method was based on the following idea: the next incremental active power was to be supplied by the most efficient plant, until it reached its maximum operational point, and so on, successively. Although this method failed to minimize costs, it was employed until 1930, when the equal incremental cost criterion began to produce better results.

The idea behind the method of incremental costs is that the next incremental system load increase should be allocated to the unit with the lowest incremental cost, which is determined by measuring the derivative of the cost curve. Steinberg and Smith mathematically proved in [29] the equal incremental costs criterion, which was already being used empirically. Around 1931, this method had already become well established. In theory, the method described by Steinberg and Smith in [29] also ensures that if there are no active constraints at the point of optimal operation, the incremental costs of all units should be equal. This rule is still widely used by power system operators today.

2.1. Optimization Model for the Classical Economic Dispatch Problem

The economic dispatch (ED) problem is concerned with minimizing active power production costs while meeting the system demand and taking into account the operational limits of all generating units. The ED problem is mathematically described in

$$\begin{aligned} \text{Minimize } C_T &= \sum_{i=1}^n (C_i(P_i) = a_i P_i^2 + b_i P_i + c_i) \\ \text{subject to } \sum_{i=1}^n P_i &= D \\ P_i^{\min} &\leq P_i \leq P_i^{\max}; \quad i = 1, \dots, n, \end{aligned} \quad (2.1)$$

where C_T : total cost function for generating units, n : number of generating units, $C_i(P_i)$: cost of generation unit i (without considering the valve-point effect), a_i, b_i, c_i : coefficient of the cost function for generating unit i , P_i : power output of generating unit i , D : total power demand, P_i^{\min} : minimum power output limit for generating unit i , and P_i^{\max} : maximum power output limit for generating unit i .

The objective function in (2.1) may also represent the so-called valve-point effects, as in [28], which are associated with the opening of pressure valves at some specific operating points. In such cases, $C_i(P_i)$ is mathematically described as in

$$C_i(P_i) = a_i P_i^2 + b_i P_i + c_i + \left| e_i \sin \left(f_i \left(P_i^{\min} - P_i \right) \right) \right|. \quad (2.2)$$

The cost function $C_i(P_i)$ described in (2.2) is continuous but not differentiable. Although it is more representative, function (2.2) makes ED a much more complex problem to solve due to its characteristics of nondifferentiability.

The literature describes several different methodologies to solve ED problems. In those studies, the methodology associated with evolutionary algorithms stands out, especially when issues related to nondifferentiability (such as those described in (2.2)) are involved. Evolutionary algorithms have been used to solve ED problems because they are able to find optimal solutions even when the objective function and/or constraints are not continuous or non-differentiable. Numerical problems related to evolutionary algorithms involve the inability to verify the optimality conditions associated with the solutions obtained and also the computational effort necessary to obtain the solutions, especially for large systems.

Optimal solutions to ED problems have also been investigated through traditional nonlinear programming methods, including interior point methods [6, 23, 30]. This paper proposes a predictor-corrector primal-dual interior point method for solving ED problems, which incorporates line search procedures in both the predictor and corrector steps. The results presented here demonstrate that the proposed method improves the solution of ED problems. The following section examines the application of the primal-dual interior point method to solve quadratic programming problems.

3. Solution Technique for the Quadratic Programming Problem

Quadratic programming problems (QPPs) represent a special class of nonlinear programming in which the objective function is quadratic and the constraints are linear [31]. The problem is expressed mathematically by

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}x^T Qx + c^T x \\ \text{subject to} \quad & Ax = b \\ & l \leq x \leq u, \end{aligned} \tag{3.1}$$

where $A \in \mathfrak{R}^{m \times n}$ so that $\text{rank}(A) = n$, $b \in \mathfrak{R}^m$, $x \in \mathfrak{R}^n$, $c \in \mathfrak{R}^n$, $u \in \mathfrak{R}^n$, $l \in \mathfrak{R}^n$, and $Q \in \mathfrak{R}^{n \times n}$ is a diagonal matrix. Problem (3.1) is equivalent to (3.2), where $z \in \mathfrak{R}^n$ is a slack variable and $r \in \mathfrak{R}^n$ is a surplus variable:

$$\begin{aligned} \text{Minimize} \quad & \frac{1}{2}x^T Qx + c^T x \\ \text{subject to} \quad & Ax = b \\ & x + z = u; \quad z \geq 0 \\ & x - r = l; \quad r \geq 0. \end{aligned} \tag{3.2}$$

For $\mu > 0$, it is possible to incorporate a logarithmic barrier function to the objective function of (3.2) and eliminate inequality constraints. This procedure results in the following nonlinear optimization problem:

$$\begin{aligned} \text{Minimize} \quad & F_\mu(x, r, z) = \frac{1}{2}x^T Qx + c^T x - \mu \sum_{j=1}^n \ln r_j - \mu \sum_{j=1}^n \ln z_j \\ \text{subject to} \quad & Ax = b \\ & x + z = u \\ & x - r = l. \end{aligned} \tag{3.3}$$

The Lagrangian function related to (3.3) is expressed by

$$\begin{aligned} L_\mu(x, r, z, w, s, y) = & \frac{1}{2}x^T Qx + c^T x + w^T (b - Ax) + s^T (l - x + r) + y^T (x + z - u) \\ & - \mu \sum_{j=1}^n \ln s_j - \mu \sum_{j=1}^n \ln y_j, \end{aligned} \tag{3.4}$$

where the dual variables associated with the three equality constraints in (3.3) are, respectively, $w \in \mathfrak{R}^m$, $y \in \mathfrak{R}^n$, $s \in \mathfrak{R}^n$. The optimal Karush-Kuhn-Tucker (KKT) conditions for problem (3.1) are depicted in

$$\begin{aligned}
\frac{\partial L_\mu}{\partial x} = 0 &\iff -Qx - c + A^T w + s - y = 0, \\
\frac{\partial L_\mu}{\partial w} = 0 &\iff Ax = b, \\
\frac{\partial L_\mu}{\partial y} = 0 &\iff x + z = u, \\
\frac{\partial L_\mu}{\partial s} = 0 &\iff x - r = l, \\
\frac{\partial L_\mu}{\partial z} = 0 &\iff ZYe - \mu e = 0, \\
\frac{\partial L_\mu}{\partial r} = 0 &\iff RSe - \mu e = 0,
\end{aligned} \tag{3.5}$$

where R, Z, S and Y are diagonal matrices whose diagonal elements are r_i, z_i, s_i , and y_i ; $i = 1, \dots, n$, respectively; $e = (1, \dots, 1)^T$; μ is a dual metric or adjustment parameter for the curve defined by the central trajectory (path-following parameter). The set Ω^0 given in (3.6) is defined to simplify notation. This set describes interior points for problem (3.2) and its corresponding dual problem:

$$\begin{aligned}
\Omega^0 = \{ &(x, w, z, r, y, s) \perp -Qx + A^T w + s - y = c, \\
&Ax = b, x + z = u, x - r = l, (z, r, y, s) > 0 \}.
\end{aligned} \tag{3.6}$$

Equations (3.4), (3.5), and (3.6) are considered in the following sections to perform the analysis of important issues concerning the proposed IPLS method, such as search directions, step sizes, stopping criteria, and updating of the barrier parameter.

3.1. Search Directions

In this section, the search directions used in the proposed IPLS method are investigated. In Section 3.2, the search directions for the predictor step are calculated while the search directions for the corrector step are determined in Section 3.3. The proposed strategy is a variant of the approach developed by Wu et al. in [23].

3.2. Search Directions—Predictor Step

Let us suppose that, in an iteration k , a point h^k satisfies the primal and dual feasibility conditions expressed by (3.5). In this case, the definition of the new point h^{k+1} depends solely on the calculation of a search direction and on a step size in such a direction. Disregarding the step size in the iteration $k + 1$, the new point h^{k+1} is defined by the following equation:

$$h^{k+1} = \begin{pmatrix} x^{k+1} \\ w^{k+1} \\ z^{k+1} \\ r^{k+1} \\ y^{k+1} \\ s^{k+1} \end{pmatrix} = \begin{pmatrix} x^k + d_x^k \\ w^k + d_w^k \\ z^k + d_z^k \\ r^k + d_r^k \\ y^k + d_y^k \\ s^k + d_s^k \end{pmatrix}. \quad (3.7)$$

Therefore, it is necessary to determine the direction of the movement d^k to obtain the new point h^{k+1} . Following the steps in the Newton method applied to the nonlinear system (3.5), d^k can be obtained by solving for the system (3.8), which is equivalent to (3.9):

$$J(h^{(k)})d^{(k)} = -F(h^{(k)}), \quad (3.8)$$

$$\begin{pmatrix} A & 0 & 0 & 0 & 0 & 0 \\ -Q & A^T & 0 & 0 & -I & I \\ I & 0 & I & 0 & 0 & 0 \\ I & 0 & 0 & -I & 0 & 0 \\ 0 & 0 & Y & 0 & Z & 0 \\ 0 & 0 & 0 & S & 0 & R \end{pmatrix} \begin{pmatrix} d_x^k \\ d_w^k \\ d_z^k \\ d_r^k \\ d_y^k \\ d_s^k \end{pmatrix} = \begin{pmatrix} t^k \\ g^k \\ f^k \\ o^k \\ q^k \\ v^k \end{pmatrix}, \quad (3.9)$$

where the residuals of (3.9) for the predictor step are expressed as in

$$\begin{aligned} t^k &= b - Ax^k; & g^k &= Qx^k + c - A^T w^k - s^k + y^k; & f^k &= u - x^k - z^k; \\ o^k &= l - x^k + r^k; & q^k &= \mu_k e - Z_k Y_k e; & v^k &= \mu_k e - R_k S_k e; & e &= (1, 1, \dots, 1)^T. \end{aligned} \quad (3.10)$$

The directions $d_x^k, d_w^k, d_z^k, d_r^k, d_y^k, d_s^k$ to be determined in the predictor step use the residuals defined in (3.10), resulting in (3.11)–(3.16), as follows:

$$Ad_x^k = t^k, \quad (3.11)$$

$$-Qd_x^k + A^T d_w^k + d_s^k - d_y^k = g^k, \quad (3.12)$$

$$d_x^k + d_z^k = f^k, \quad (3.13)$$

$$d_x^k - d_r^k = o^k, \quad (3.14)$$

$$Y_k d_z^k + Z_k d_y^k = q^k, \quad (3.15)$$

$$S_k d_r^k + R_k d_s^k = v^k. \quad (3.16)$$

From (3.13) and (3.14), we obtain (3.17) and (3.18), respectively:

$$d_z^k = -d_x^k + f^k, \quad (3.17)$$

$$d_r^k = d_x^k - o^k. \quad (3.18)$$

Isolating d_y^k and d_s^k in (3.15) and (3.16), respectively, leads to (3.19) and (3.20), as follows:

$$d_y^k = Z_k^{-1} (q^k - Y_k d_z^k), \quad (3.19)$$

$$d_s^k = R_k^{-1} (v^k - S_k d_r^k). \quad (3.20)$$

Combining the results found in (3.17), (3.18), (3.19), and (3.20) with those given in (3.11)–(3.16), and considering (3.21), yields the directions (3.22) and (3.23):

$$\theta = (R_k^{-1} S_k + Z_k^{-1} Y_k + Q)^{-1}, \quad (3.21)$$

$$d_w^k = (A\theta A^T)^{-1} [A\theta (g^k + p^k) + t^k], \quad (3.22)$$

$$d_x^k = \theta (A^T d_w^k - g^k - p^k), \quad (3.23)$$

where

$$p^k = R_k^{-1} (S_k o^k - v^k) + Z_k^{-1} (q^k - Y_k f^k). \quad (3.24)$$

Note that, after calculating (3.23), the remaining components of the direction vector d_z^k, d_r^k, d_y^k , and d_s^k in (3.17), (3.18), (3.19), and (3.20), respectively, are easily calculated. Since matrix $A\theta A^T$ is symmetrical and positive definite in (3.22) (considering that Q is symmetrical and positive definite), d_w^k can be determined by using the Cholesky decomposition.

3.3. Search Directions—Corrector Step

Analogously to the procedure carried out in Section 3.2, this section describes the calculation of the search direction for the corrector step \tilde{d}^k , which is obtained by solving the linear system (3.25), as follows:

$$J(h^k)\tilde{d}^k = -\tilde{F}(h^k), \quad (3.25)$$

where $\tilde{F}(h^k)$ is obtained by considering second-order approximations in the residuals (3.10) (calculated in the predictor step), so that (3.25) is equivalent to

$$\begin{pmatrix} A & 0 & 0 & 0 & 0 & 0 \\ -Q & A^T & 0 & 0 & -I & I \\ I & 0 & I & 0 & 0 & 0 \\ I & 0 & 0 & -I & 0 & 0 \\ 0 & 0 & Y & 0 & Z & 0 \\ 0 & 0 & 0 & S & 0 & R \end{pmatrix} \begin{pmatrix} \tilde{d}_x^k \\ \tilde{d}_w^k \\ \tilde{d}_z^k \\ \tilde{d}_r^k \\ \tilde{d}_y^k \\ \tilde{d}_s^k \end{pmatrix} = \begin{pmatrix} \tilde{t}^k \\ \tilde{g}^k \\ \tilde{f}^k \\ \tilde{o}^k \\ \tilde{q}^k \\ \tilde{v}^k \end{pmatrix}, \quad (3.26)$$

where

$$\begin{aligned} \tilde{t}^k &= b - Ax^k; & \tilde{g}^k &= Qx^k + c - A^T w^k - s^k + y^k; & \tilde{f}^k &= u - x^k - z^k; \\ \tilde{o}^k &= l - x^k + r^k; & \tilde{q}^k &= \mu_k e - Z_k Y_k e - D_z^k D_y^k e; & \tilde{v}^k &= \mu_k e - R_k S_k e - D_x^k D_s^k e, \end{aligned} \quad (3.27)$$

D_x^k, D_z^k, D_y^k and D_s^k are diagonal matrices whose diagonal components are $(d_x^k)_i, (d_z^k)_i, (d_y^k)_i$ and $(d_s^k)_i, i = 1, \dots, n$, respectively.

The calculation of the residuals $t^k, g^k, f^k, o^k, q^k, v^k$, described in (3.10), and of $\tilde{t}^k, \tilde{g}^k, \tilde{f}^k, \tilde{o}^k, \tilde{q}^k, \tilde{v}^k$, described in (3.27), basically distinguishes the predictor and corrector steps in the proposed method. It is important to note that the corrector step procedure uses direction values d_x^k, d_z^k, d_y^k , and d_s^k , which have already been calculated in the predictor step, to redefine residuals \tilde{q}^k and \tilde{v}^k in (3.27). Using (3.25) and (3.26) and following the same steps

taken to determine the directions of the predictor step, as seen in Section 3.2, the components of the direction vector \tilde{d}^k can be calculated using the following:

$$\tilde{d}_w^k = (A\theta A^T)^{-1} [A\theta(g^k + \tilde{p}^k) + \tilde{t}^k], \quad (3.28)$$

$$\tilde{d}_x^k = \theta(A^T \tilde{d}_w^k - g^k - \tilde{p}^k), \quad (3.29)$$

$$\tilde{d}_z^k = -\tilde{d}_x^k + \tilde{f}^k, \quad (3.30)$$

$$\tilde{d}_r^k = \tilde{d}_x^k + \tilde{\sigma}^k, \quad (3.31)$$

$$\tilde{d}_y^k = Z_k^{-1}(\tilde{q}^k - Y_k \tilde{d}_z^k), \quad (3.32)$$

$$\tilde{d}_s^k = R_k^{-1}(\tilde{v}^k - S_k \tilde{d}_r^k), \quad (3.33)$$

where

$$\tilde{p}^k = R_k^{-1}(S_k \tilde{\sigma}^k - \tilde{v}^k) + Z_k^{-1}(\tilde{q}^k - Y_k \tilde{f}^k), \quad (3.34)$$

and θ is defined in (3.21).

3.4. Step Size

After calculating the search directions for the predictor and corrector steps, it is possible to move to a new point $(x^{k+1}, w^{k+1}, z^{k+1}, r^{k+1}, y^{k+1}, s^{k+1})$, while ensuring that $s^{k+1} > 0$, $z^{k+1} > 0$, $y^{k+1} > 0$ and $r^{k+1} > 0$. In order to ensure nonnegativity constraints over the slack variables, the step to be taken in each direction in both the predictor and corrector steps must be controlled. The basics of this procedure are described in [31] and are discussed below.

3.4.1. Predictor Step Size

Considering the variables defined in the predictor step, the step size for primal and dual variables is calculated as described below:

$$x^{k+1} = x^k + \alpha_{P_k} d_x^k, \quad (3.35)$$

$$r^{k+1} = r^k + \alpha_{P_k} d_r^k, \quad (3.36)$$

$$z^{k+1} = z^k + \alpha_{P_k} d_z^k, \quad (3.37)$$

$$w^{k+1} = w^k + \alpha_{D_k} d_w^k, \quad (3.38)$$

$$y^{k+1} = y^k + \alpha_{D_k} d_y^k, \quad (3.39)$$

$$s^{k+1} = s^k + \alpha_{D_k} d_s^k. \quad (3.40)$$

The step size α_{P_k} for the primal variables is calculated through

$$\alpha_{P_k} = \text{Min}\{\alpha_P, \alpha_Q, \alpha_{LS}\}, \quad (3.41)$$

where, for $0 < \alpha < 1$, the step sizes $\alpha_P, \alpha_Q, \alpha_{LS}$ are determined as follows.

- (i) The step size for primal variables α_P is obtained without violating the nonnegativity requirements of the primal variables:

$$\alpha_P = \text{Min}\left\{-\frac{\alpha z_i^k}{d_{z_i}^k}, -\frac{\alpha r_i^k}{d_{r_i}^k} \text{ such that } d_{z_i}^k, d_{r_i}^k < 0\right\}. \quad (3.42)$$

- (ii) The step size α_Q is the maximum step size possible without increasing the objective value:

$$\alpha_Q = -\frac{\left[(d_x^k)^t (Qx^k + c)\right]}{\left[(d_x^k)^t Q d_x^k\right]}. \quad (3.43)$$

- (iii) The step size α_{LS} is determined as in (3.44) from the Fibonacci line search strategy, which is briefly summarized as follows:

$$\alpha_{LS} = \text{Min}_{\alpha_k} \left\{ F_\mu \left(\hat{x}^{k+1}(\alpha_k) \right) \right\} = \text{Min}_{\alpha_k} \left\{ F_\mu \left(\hat{x}^k + \alpha_k d_{\hat{x}}^k \right) \right\}, \quad (3.44)$$

where F_μ is defined in (3.3).

For notation simplicity, in the summary of the Fibonacci search, the following identities are defined: $\hat{x}^{k+1}(\alpha_{LS}) = (x^{k+1}(\alpha_{LS}), r^{k+1}(\alpha_{LS}), z^{k+1}(\alpha_{LS}))^T = (x^k + \alpha_{LS} d_x^k, r^k + \alpha_{LS} d_r^k, z^k + \alpha_{LS} d_z^k)^T$, and $d_{\hat{x}}^k = (d_x^k, d_r^k, d_z^k)$; so that d_x^k is defined in (3.23), d_r^k is defined in (3.18), and d_z^k is defined in (3.17). Only primal variables are considered in the Fibonacci search algorithm used by the IPLS method. Starting at a point $\hat{x}^k = (x^k, r^k, z^k)^T$, the algorithm searches a new point \hat{x}^{k+1} in direction $d_{\hat{x}}^k$ using the function F_μ defined in (3.3). The Fibonacci method calculates α_{LS} so that the minimization of F_μ is ensured. α_{LS} is determined considering the Fibonacci sequence. The initial value of α_{LS} is set taking into account the interval of uncertainty $[0, 1]$.

The step size α_{D_k} for the dual variables (3.38)–(3.40) is calculated through (3.45) for $0 < \alpha < 1$:

$$\alpha_{D_k} = \text{Min}\left\{1, -\frac{\alpha s_i^k}{d_{s_i}^k}, -\frac{\alpha y_i^k}{d_{y_i}^k} \text{ such that } d_{s_i}^k, d_{y_i}^k < 0\right\}. \quad (3.45)$$

3.4.2. Corrector Step Size

In the corrector step, the calculation of step size is defined analogously, considering the variables from the following

$$x^{k+1} = x^k + \beta_{P_k} \tilde{d}_x^k, \quad (3.46)$$

$$r^{k+1} = r^k + \beta_{P_k} \tilde{d}_r^k, \quad (3.47)$$

$$z^{k+1} = z^k + \beta_{P_k} \tilde{d}_z^k, \quad (3.48)$$

$$w^{k+1} = w^k + \beta_{D_k} \tilde{d}_w^k, \quad (3.49)$$

$$y^{k+1} = y^k + \beta_{D_k} \tilde{d}_y^k, \quad (3.50)$$

$$s^{k+1} = s^k + \beta_{D_k} \tilde{d}_y^k, \quad (3.51)$$

where

$$\beta_{P_k} = \text{Min}\{\beta_P, \beta_Q, \beta_{LS}\}, \quad (3.52)$$

while the step sizes $\beta_P, \beta_Q, \beta_{LS}$ are determined as follows.

- (i) The step size for primal variables β_P is obtained without violating the nonnegativity requirements of the primal variables:

$$\beta_P = \text{Min}\left\{1, -\frac{\alpha z_i^k}{\tilde{d}_{z_i}^k}, -\frac{\alpha r_i^k}{\tilde{d}_{r_i}^k} \text{ such that } \tilde{d}_{z_i}^k, \tilde{d}_{r_i}^k < 0\right\}. \quad (3.53)$$

- (ii) The step size β_Q is the maximum step size possible without increasing the objective value:

$$\beta_Q = -\frac{\left[\left(\tilde{d}_x^k\right)^t (Qx^k + c)\right]}{\left[\left(\tilde{d}_x^k\right)^t Qd_x^k\right]}. \quad (3.54)$$

- (iii) The step size β_{LS} in (3.55) is determined from the Armijo line search:

$$\beta_{LS} = \beta_k, \quad (3.55)$$

so that

$$F_\mu(\tilde{x}^{k+1}(\beta_k)) \leq F_\mu(\tilde{x}^k) + \beta_k \nabla F_\mu(\tilde{x}^k)^T \tilde{d}_{\tilde{x}}^k, \quad (3.56)$$

where $\hat{x}^{k+1}(\beta_k) = (x^{k+1}(\beta_k), r^{k+1}(\beta_k), z^{k+1}(\beta_k))^T = (x^k + \beta_k \tilde{d}_x^k, r^k + \beta_k \tilde{d}_r^k, z^k + \beta_k \tilde{d}_z^k)^T$; $\tilde{d}_x^k = (\tilde{d}_x^k, \tilde{d}_r^k, \tilde{d}_z^k)$; so that $\tilde{d}_x^k, \tilde{d}_r^k, \tilde{d}_z^k$ are defined in (3.29)–(3.31), respectively. In the proposed IPLS method, the Armijo search is also performed in the direction defined by primal variables. Starting at a point $\hat{x}^k = (x^k, r^k, z^k)$, the method searches for a new point \hat{x}^{k+1} , in direction \tilde{d}_x^k , so that the function F_μ defined in (3.3) decreases. This search calculates β_{LS} , thereby ensuring the reduction of F_μ . To prevent oscillations in the iterative process, the initial choice for β_k should not be too high, and to prevent the process from stopping prematurely, it should not be too low. Therefore, this value is generally adjusted to $\beta_0 = 1$. If (3.44) is not satisfied, β_k is updated using the following sequence:

$$\beta_{k+1} = \frac{\beta_k}{\delta}, \quad k = 0, 1, \dots, n; \text{ with } \delta > 1, \text{ usually set to } \delta = 2. \quad (3.57)$$

The following equation is used for the analysis of (3.56):

$$\nabla F_\mu(\hat{x}^k) = (\nabla F_\mu(x^k), \nabla F_\mu(r^k), \nabla F_\mu(z^k))^T = (Qx^k + c, -\mu_k R_k^{-1}e, -\mu_k Z_k^{-1}e)^T \quad (3.58)$$

with μ_k determined as in Section 3.6.

The step size for dual variables β_{D_k} is calculated by (3.59), without violating the nonnegativity requirements of the dual variables:

$$\beta_{D_k} = \text{Min} \left\{ 1, -\frac{\alpha s_i^k}{\tilde{d}_{s_i}^k}, -\frac{\alpha y_i^k}{\tilde{d}_{y_i}^k} \text{ such that } \tilde{d}_{s_i}^k, \tilde{d}_{y_i}^k < 0 \right\}. \quad (3.59)$$

The general principle used here to calculate α_{P_k} and β_{P_k} is to choose a step size that reduces the quadratic objective by a maximum amount without violating the nonnegativity requirements of the primal variables.

The basic idea for defining the line searches in both the predictor and corrector steps is to use a more accurate search for the predictor step (which uses first order approximation to calculate the residuals) and a simpler search, albeit more robust, for the corrector step (which uses second-order approximation to calculate the residuals). Therefore, the Fibonacci search is used in the predictor step, since it is more accurate, and provides the minimum value for the objective function in the predefined direction; the Armijo search is used in the corrector step because it is simpler and more robust.

3.5. Stopping Rules

Interior point algorithms do not find exact solutions for linear or quadratic programming problems. Therefore, stopping rules are needed to decide when the solution obtained in a current iteration is sufficiently close to the optimal solution. In this study, the stopping rules are based on [32].

Many algorithms consider that a good approximate solution is the one that presents sufficiently small values for primal and dual residuals t^k, u^k , and also for the dual metric μ_k . Nevertheless, it is possible to use relative values for the metrics t^k, u^k , and μ_k in order to

reduce scaling problems, as described in [32]. Typical stopping rules are shown in (3.60)–(3.65), as follows:

(i) primal feasibility:

$$\frac{\|t^k\|}{\|b\| + 1} \leq \varepsilon_1 \quad (\text{predictor step}), \quad (3.60)$$

$$\frac{\|\tilde{t}^k\|}{\|b\| + 1} \leq \varepsilon_1 \quad (\text{corrector step}); \quad (3.61)$$

(ii) dual feasibility:

$$\frac{\|u^k\|}{\|Qx^k + c\| + 1} \leq \varepsilon_2 \quad (\text{predictor step}), \quad (3.62)$$

$$\frac{\|\tilde{u}^k\|}{\|Qx^k + c\| + 1} \leq \varepsilon_2 \quad (\text{corrector step}); \quad (3.63)$$

(iii) complementary slackness:

$$\|v^k\| \leq \varepsilon_3, \quad \|q^k\| \leq \varepsilon_3 \quad (\text{predictor step}) \quad (3.64)$$

$$\|\tilde{v}^k\| \leq \varepsilon_3, \quad \|\tilde{q}^k\| \leq \varepsilon_3 \quad (\text{corrector step}), \quad (3.65)$$

where ε_1 , ε_2 , and ε_3 are sufficiently small positive numbers. Eventually, other criteria can be adopted according to the specific characteristics of each problem, as can be seen in [24, 32].

3.6. Update of the Barrier Parameter

According to [32], the barrier parameter is updated using an inner product that involves the primal variables r^k and z^k , and the dual variables s^k and y^k , respectively, as shown in the following equation

$$\mu_k^1 = \frac{(r^k)^T s^k}{n}; \quad \mu_k^2 = \frac{(z^k)^T y^k}{n}, \quad (3.66)$$

so that μ_k is calculated using the following equation

$$\mu_k = \text{Min} \left\{ \sigma \mu_k^1, \sigma \mu_k^2 \right\} \text{ for a constant } 0 < \sigma < 1, \quad (3.67)$$

where the parameter σ is used to accelerate the convergence of the iterative process. This procedure for updating the barrier parameter proposed by Wright in [32] helps in the theoretical convergence proof and also in the complexity analysis of primal-dual methods.

3.7. Algorithm for the Proposed IPLS Method

Step 1 (initialisation). Adjust $k = 0$. Choose an arbitrary point: $(x^0; w^0; z^0, y^0, r^0, y^0, s^0) \in \Omega^0$ and choose $\varepsilon_1, \varepsilon_2$, and ε_3 as sufficiently small positive numbers.

Step 2 (intermediate calculations—predictor). Calculate: $t^k; g^k; f^k; o^k; q^k; v^k$ using (3.10), μ_k using (3.67) and matrix θ using (3.21).

Step 3 (finding directions of translation—predictor). Calculate search directions $d_x^k, d_w^k, d_z^k, d_r^k, d_y^k$, and d_s^k for the predictor step using (3.17)–(3.24).

Step 4 (computing step size—predictor). Calculate Fibonacci step sizes α_{P_k} and α_{D_k} using (3.41)–(3.44).

Step 5 (moving to a new solution—predictor). Update $x^{k+1}; w^{k+1}; s^{k+1}; y^{k+1}; z^{k+1}; r^{k+1}$ obtained from the predictor step, according to (3.35)–(3.44).

Step 6 (checking for optimality—predictor). If the criteria defined in (3.60), (3.62), and (3.64) are satisfied, stop. The solution is optimal. Otherwise, go on to the following step.

Step 7 (intermediate calculations—corrector). Calculate: $\tilde{t}^k; \tilde{g}^k; \tilde{f}^k; \tilde{o}^k; \tilde{q}^k; \tilde{v}^k$ using (3.27), μ_k using (3.67) and matrix θ using (3.21).

Step 8 (finding directions of translation—corrector). Calculate $\tilde{d}_x^k, \tilde{d}_w^k, \tilde{d}_z^k, \tilde{d}_r^k, \tilde{d}_y^k$, and \tilde{d}_s^k for the corrector step using (3.28)–(3.33).

Step 9 (computing step size—corrector). Calculate the Armijo step size β_{P_k} and β_{D_k} using (3.52)–(3.58).

Step 10 (moving to a new solution—corrector). Update $x^{k+1}; w^{k+1}; s^{k+1}; y^{k+1}; z^{k+1}; r^{k+1}$ using (3.46)–(3.48).

Step 11 (checking for optimality—corrector). If the criteria defined in (3.61), (3.63), and (3.65) are satisfied, stop. The solution is optimal. Otherwise, go on to the next step.

Step 12. Adjust $k = k + 1$ and return to Step 2.

The predictor Steps 2 through Step 6 are held in odd iterations, while the corrector Steps 7 through Step 11 are performed in even iterations. The next section describes numerical simulations involving the application of the proposed method to ED problems.

4. Application of the Proposed Algorithm to Solve ED Problems

In this section, the proposed method is applied to solve three power systems with 3, 6, and 13 generators, respectively. Tables 1, 3, and 5 show the data related to the generating units of the power systems. These data were extracted from [26, 27]. The tables list the coefficients of the cost function for each generating unit, as described in (2.1), and also minimum and maximum power output limits.

Tables 2 and 3 present the results of the application of the proposed method to solve the power system with 3 generators, while Tables 5 and 6 list the results for the system with

Table 1: Characteristics of the system with 3 generators.

Unit	P_i^{\min} (MW)	P_i^{\max} (MW)	a_i (\$/MW ²)	b_i (\$/MW)	c_i (\$)
1	100	600	0,001562	7,92	561
2	50	200	0,004820	7,97	78
3	100	400	0,001940	7,85	310

6 generators, and Tables 8 and 9 depict the results for the system with 13 generators. These tables compare the solutions obtained by the proposed IPLS method against those calculated by the following methods: the predictor-corrector primal-dual (PCPD) method described in [6], the hybrid genetic algorithm (HGA), the coevolutionary genetic algorithm (COEGA), the hybrid atavistic genetic algorithm (HAGA) and the cultural algorithm (CA). The solutions determined by the HGA and COEGA methods are available in [27], while the solutions obtained with the CA method are described in [26], and those obtained with HAGA method are given in [25], but only for the system comprising 13 generators.

The comparison between the IPLS method and the evolutionary approaches cited above were introduced in this section because these are traditional methods used in power system to solve ED problems, especially when a more general nondifferentiable objective function (as shown in (2.2)) is used. However, it is important to highlight that these evolutionary approaches are heuristic procedures, which provide only approximate solutions to ED problems. When ED is formulated as a quadratic problem, it can be solved by means of exact methods, such as the IPLS and PCPD method.

Therefore, to better evaluate the performance of proposed IPLS method, this method has been compared to the PCPD method in Table 10. The results in Table 10 have the main purpose to show the reduction in the computational effort when the IPLS method is compared to the PCPD method. Both methods were implemented using Borland Pascal 7.0 programming language.

4.1. Power System with 3 Generators

The main characteristics of the system containing 3 generators are described in Table 1. Parameters a_i, b_i, c_i stand for the coefficients of the cost functions for the generators, while P_i^{\min}, P_i^{\max} represent minimum and maximum power output capabilities, respectively, for generating unit i .

The following values are adopted to initialise the method:

$$x^0 = (450, 100, 300); \quad w^0 = (0, 0, 0); \quad y^0 = (0, 0, 0). \quad (4.1)$$

The parameters related to the system's total demand are set at $D = 850$ MW and the active power losses are neglected. The values adopted for $\varepsilon_1, \varepsilon_2,$ and ε_3 are 10^{-8} .

Table 2 shows the active power output calculated by the methods. The results for the HAGA algorithm are not presented in this case study. Table 3 compares optimal values for the objective functions obtained by each method. From the results presented in this table, it is clear that the dispatches calculated by all the types of genetic algorithms cannot reach the global optimum dispatch attained by the interior point methods PCPD and IPLS. As Table 3 indicates, the cost calculated by the IPLS and PCPD methods is lower than that obtained by

Table 2: Comparison of the power generation outputs obtained for the system with 3 generators.

Power output	HGA	COEGA	PCPD	IPLS
P_1 (MW)	470,8421	344,7295	393,1698	393,1698
P_2 (MW)	109,4012	193,9445	122,2264	122,2264
P_3 (MW)	269,7567	311,3260	334,6038	334,6038
ΣP_i (MW)	850,0000	850,0000	850,000	850,000

Table 3: Characteristics of the system with 3 generators.

Results	Generation costs (\$)
HGA	8.212,73
COEGA	8.223,86
PCPD	8.194,36
IPLS	8.194,36

Table 4: Characteristics of the system with 6 generators.

Unit	P_i^{Min} (MW)	P_i^{Max} (MW)	a_i (\$/MW ²)	b_i (\$/MW)	c_i (\$)
1	10	125	0,15247	38,53973	756,79886
2	10	150	0,10587	46,15916	451,3251
3	35	225	0,02803	40,39655	1049,9977
4	35	210	0,03546	38,30533	1243,5311
5	130	325	0,02111	36,32782	1658,5696
6	125	315	0,01799	38,24041	1356,6592

the HGA and COEGA methods. As already discussed, this is an expected result, since the evolutionary approaches provide only approximate solution to the problem.

4.2. Power System with 6 Generators

Table 4 describes the main characteristics of the system containing 6 generators.

As in the previous case study, the parameters a_i, b_i, c_i stand for the coefficients of the cost functions for the generating unit I , while $P_i^{\text{min}}, P_i^{\text{max}}$ represent minimum and maximum power output capabilities, respectively, for the generating unit i .

The following values are adopted to initialise the method:

$$x^0 = (20, 30, 75, 75, 145, 155); \quad w^0 = (0, 0, 0, 0, 0, 0); \quad y^0 = (1, 1, 1, 1, 1, 1). \quad (4.2)$$

The parameters related to the system's total demand are set at $D = 500$ MW and the active power losses are neglected. The values adopted for $\varepsilon_1, \varepsilon_2$, and ε_3 are 10^{-8} .

Table 5 shows the active power dispatch calculated by the methods. The solutions for COEGA and HAGA algorithms were not presented by their authors. Table 6 compares optimal values for the objective function obtained by each method. Again, as expected, the costs calculated by the IPLS and PCPD methods are lower than those obtained by the HGA method.

Table 5: Comparison of the power generation outputs obtained for the system with 6 generators.

Generated power	HGA	PCPD	IPLS
P_1 (MW)	20,1367	17,36596	17,36596
P_2 (MW)	14,8645	10,00000	10,00000
P_3 (MW)	72,4008	61,340667	61,340667
P_4 (MW)	72,4008	77,97487	77,97487
P_5 (MW)	180,0617	177,81828	177,81828
P_6 (MW)	139,5865	155,500216	155,500216
ΣP_i (MW)	500,0000	500,0000	500,0000

Table 6: Values of the objective function for the system with 6 generators.

Results	Generation cost (\$)
HGA	27.037,29
PCPD	27.003,50
IPLS	27.003,50

Table 7: Characteristics of the system with 13 generators.

Unit	P_i^{Min} (MW)	P_i^{Max} (MW)	a_i (\$/MW ²)	b_i (\$/MW)	c_i (\$)
1	0	680	0,00028	8,1	550
2	0	360	0,00056	8,1	309
3	0	360	0,00056	8,1	307
4	60	180	0,00324	7,74	240
5	60	180	0,00324	7,74	240
6	60	180	0,00324	7,74	240
7	60	180	0,00324	7,74	240
8	60	180	0,00324	7,74	240
9	60	180	0,00324	7,74	240
10	40	120	0,00284	8,6	126
11	40	120	0,00284	8,6	126
12	55	120	0,00284	8,6	126
13	55	120	0,00284	8,6	126

4.3. Power System with 13 Generators

Table 7 describes the main characteristics of the system containing 13 generators.

The parameters in this table are analogous to those described in the preceding case studies. The following values are adopted to initialise the method:

$$\begin{aligned}
 x^0 &= (660, 320, 330, 150, 140, 155, 165, 150, 140, 70, 80, 75, 85); \\
 w^0 &= (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0); \\
 y^0 &= (30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30).
 \end{aligned} \tag{4.3}$$

The parameters related to the system's total demand are set at $D = 2520$ MW and the active power losses are neglected. The values adopted for ε_1 , ε_2 , and ε_3 are 10^{-8} .

Table 8: Comparison of the power generation outputs obtained for the system with 13 generators.

Power output	HGA	COEGA	CA	PCPD	IPLS
P_1	651,145	735,626	679,2551	680,0000	680,0000
P_2	319,982	337,495	359,8672	360,0000	360,0000
P_3	320,463	292,625	357,2368	360,0000	360,0000
P_4	137,776	146,713	154,8137	155,0000	155,0000
P_5	156,688	177,346	158,0946	155,0000	155,0000
P_6	147,007	131,552	155,8520	155,0000	155,0000
P_7	159,165	154,197	146,1697	155,0000	155,0000
P_8	145,378	159,550	146,8364	155,0000	155,0000
P_9	151,551	167,339	168,7979	155,0000	155,0000
P_{10}	82,259	60,677	40,0181	40,0000	40,0000
P_{11}	86,320	74,681	40,0000	40,0000	40,0000
P_{12}	82,893	56,537	55,0175	55,0000	55,0000
P_{13}	79,368	25,655	55,0488	55,0000	55,0000
ΣP_i (MW)	2.520,00	2.520,00	2.520,00	2.520,00	2.520,00

Table 9: Values of the objective function for the system with 13 generators.

Results	Generation cost (\$)
HGA	24.111,69
COEGA	24.072,03
HAGA	24.052,34
CA	24.052,10
PCPD	24.050,08
IPLS	24.050,08

Table 8 lists the power generation outputs obtained by the methods. The values of the dispatch calculated by the HAGA are not given by [25], who provided only the optimal value for the objective function. Once more, as Table 9 indicates, the total cost calculated by the PCPD and IPLS methods is lower than that calculated by all the others, although CA and HAGA approaches get close to the optimal solution point.

4.4. Performance of the Interior Point Methods Tested

This section evaluates the computational effort of the interior point methods tested here (PCPD and IPLS), measuring it in terms of the number of iterations required to obtain the optimal solution. To this end, Table 10 shows the number of iterations obtained by the solution algorithms of each method for the previously studied systems. It is important to highlight that the same parameter settings were used for the two methods.

Computational tests were carried out in an Intel Corel Quad Q9550, with 3.5 GB of RAM memory, in order to calculate and compare the CPU times between the proposed IPLS method and the PCPD method. For such a purpose, we utilized the specific unit GET TIME from Borland Pascal 7.0. The precision of this unit is up to milliseconds. The computational times calculated by this unit for the power systems tested (which include the systems with 3, 6, and 13 generators) were all null for both, the proposed method, and the PCPD method. This occurred due to the efficiency of the machine processor and also due to the efficiency of

Table 10: Number of iterations for the PCPD and IPLS methods.

Simulated system	PCPD	IPLS
3 generating units	26	12
6 generating units	22	15
13 generating units	23	17

both methods tested. Therefore, the efficiency of the methods is compared only in terms of number of iterations, as described in Table 10.

The basic difference between the PCPD and IPLS algorithms is that line searches are incorporated in the corrector and predictor steps of the IPLS method. Therefore, these results demonstrate that the introduction of line searches in the IPLS method greatly reduces the number of iterations required to solve ED problems. Since line searches represent only a minor additional computational effort in the overall process of finding a solution, the computational effort for solving ED problems is significantly improved by the proposed IPLS method.

4.5. Postoptimization Analysis: Incremental Costs

This section conducts a postoptimization analysis to evaluate the optimality conditions of the solutions obtained in the previous section for the systems in question. The purpose of this analysis is to evaluate the complete picture concerning optimal solutions for ED problems. The classical ED problem (2.1) can be rewritten as follows:

$$\begin{aligned}
 &\text{Minimize } C_T \\
 &\text{subject to } \sum_{i=1}^n P_i = D \\
 &P_i \in S,
 \end{aligned} \tag{4.4}$$

where $S = \{P_i \in \mathfrak{R} \mid P_i - P_i^{\min} \geq 0 \text{ and } P_i^{\max} - P_i \geq 0, i = 1, \dots, n\}$, and $C_T = \sum_{i=1}^n (C_i(P_i) = a_i P_i^2 + b_i P_i + c_i)$.

Problem (4.4) is subsequently used for the postoptimization analysis of the solutions obtained by the systems studied in the previous section.

4.6. The Lagrangian Function and KKT Conditions

Considering ED as redefined in (4.4), one finds the following associated Lagrangian function:

$$L(P_i, w, s_i, y_i) = C_T(P_i) + w \left(D - \sum_{i=1}^n P_i \right) + \sum_{i=1}^n s_i (P_i^{\min} - P_i) + \sum_{i=1}^n y_i (P_i - P_i^{\max}), \tag{4.5}$$

where $w \in \mathfrak{R}$, $s_i \in \mathfrak{R}$, and $y_i \in \mathfrak{R}$, $i = 1, \dots, n$, are the associated Lagrange multipliers. The KKT conditions associated with the Lagrangian function (4.5) are expressed through

$$D - \sum_{i=1}^n P_i = 0, \quad (4.6)$$

$$P_i - P_i^{\min} \geq 0; \quad s_i (P_i - P_i^{\min}) = 0; \quad i = 1, \dots, n, \quad (4.7)$$

$$P_i^{\max} - P_i \geq 0; \quad y_i (P_i^{\max} - P_i) = 0; \quad i = 1, \dots, n, \quad (4.8)$$

$$\lambda_i(P_i) - w - s_i + y_i = 0; \quad i = 1, \dots, n, \quad (4.9)$$

where

$$\lambda_i(P_i) = \frac{\partial C_i(P_i)}{\partial P_i} = 2a_i P_i + b_i; \quad i = 1, \dots, n. \quad (4.10)$$

In the literature on power systems, the Lagrange multiplier w is called energy price (i.e., the price of 1 MWh), while $\lambda_i(P_i)$ is called incremental or marginal cost of generating unit i . Using expression (4.9), the energy price can be calculated by

$$w = \lambda_i(P_i) - s_i + y_i; \quad i = 1, \dots, n. \quad (4.11)$$

4.7. Energy Prices and Incremental Costs Using KKT Conditions

If the optimal solution provided by the IPLS method satisfies the KKT conditions, then there are four possible combinations for analysing prices and incremental costs, which are examined below.

4.7.1. No Active Constraint

In this case there is no active constraint in the optimal solution of problem (4.4), so that $s_i^* = 0$, $y_i^* = 0$; $i = 1, \dots, n$. Thus, from (4.11), one reaches

$$w = \lambda_i^*(P_i^*); \quad i = 1, \dots, n, \quad (4.12)$$

that is, the energy price is equal to the incremental (marginal) costs for all the generating units. This situation corresponds to the rule commonly used by power system operators, which states that all marginal costs are equal. It is important to emphasize that this rule is valid only for this case and should, therefore, be used cautiously.

4.7.2. Active Constraints for Maximum Power Output Limit

In this case, some generating units are dispatched in their maximum power output capabilities, so that the constraints associated with maximum power output capabilities are

active in the optimal solution of (4.4). To analyse this case, let the set Ω_{\max} be defined with the indices of the generating units that have been optimally dispatched in their maximum power output. In this situation, $s_i^* = 0$ for $i = 1, \dots, n$, $y_j^* > 0$ for $j \in \Omega_{\max}$, and $y_j^* = 0$ for $j \notin \Omega_{\max}$. Starting from (4.11), one reaches

$$\begin{aligned} w &= \lambda_j^*(P_j^*) + y_j^*; & j \in \Omega_{\max}, \\ w &= \lambda_j^*(P_j^*); & j \notin \Omega_{\max}. \end{aligned} \quad (4.13)$$

Based on (4.13), it is possible to show that the marginal costs for the group of generating units that belong to Ω_{\max} are lower than the marginal costs for the group that does not belong to Ω_{\max} , since $y_j^* > 0$; $j \in \Omega_{\max}$.

4.7.3. Active Constraints for Minimum Power Output Limit

In this case, some generating units are dispatched in their minimum power output capabilities, so that the constraints associated with minimum power output capabilities are active in the optimal solution of (4.4). To analyse this case, let the set Ω_{\min} be defined with the indices of the generating units that have been optimally dispatched in their minimum power output. In this situation, $y_i^* = 0$ for $i = 1, \dots, n$, $s_j^* > 0$ for $j \in \Omega_{\min}$, and $s_j^* = 0$ for $j \notin \Omega_{\min}$. Thus, using (4.11), one reaches

$$\begin{aligned} w &= \lambda_j^*(P_j^*) - s_j^*; & j \in \Omega_{\min}, \\ w &= \lambda_j^*(P_j^*); & j \notin \Omega_{\min}. \end{aligned} \quad (4.14)$$

Based on (4.14), it can be shown that the marginal costs for the group of generating units that belong to Ω_{\min} are higher than the marginal costs for the group that does not belong to Ω_{\min} , since $s_j^* > 0$; $j \in \Omega_{\min}$.

4.7.4. Active Constraints for Both Minimum and Maximum Power Output Capabilities

In this case, some generating units are dispatched in their upper limits, while others are dispatched in their lower limits, or at some other operational point between their upper and lower limits. Obviously, the same unit cannot simultaneously assume lower and upper limits. The comments made in the preceding sections concerning marginal costs for the generating units that are dispatched in their lower or upper limits also apply to this case.

4.8. Results of Incremental Analyses for the Systems under Study

4.8.1. Case 1: Power System with 3 Generating Units

Table 2 describes the optimal dispatch, P_i^* , $i = 1, \dots, n$, calculated by the IPLS method for the system with 3 generating units. As this table indicates, no generating unit has reached its

Table 11: Incremental analysis for the 3-generator system.

Unit	λ_i^*	s_i^*	y_i^*	w^*
1	9,14826	0	0	9,14826
2	9,14826	0	0	
3	9,14826	0	0	

Table 12: Incremental analysis of the 6-generator system.

Unit	λ_i^*	s_i^*	y_i^*	w
1	43,8449	0	0	43,8449
2	48,2765	4,43164	0	
3	43,8449	0	0	
4	43,8449	0	0	
5	43,8449	0	0	
6	43,8449	0	0	

lower or upper limit, so there is no active constraint in the optimal solution. This situation coincides with the one described in Section 4.7.1. The energy price, marginal costs, and Lagrange multipliers are shown in Table 11 for this case. As described in Section 4.7.1, the values obtained by the IPLS method for s_i^* and y_i^* ; $i = 1, \dots, n$ are all zero. Also, the energy price is equivalent to the marginal costs, which are all equal, as expected. These results are in accordance with the theory described in Section 4.7.1.

4.8.2. Case 2: Power System with 6 Generating Units

Table 5 shows the optimal dispatch, P_i^* , $i = 1, \dots, n$, calculated by the IPLS method for the system with 6 generating units. As can be seen in this table, the generating unit 2 has reached its lower limit, so the optimal solution has one active constraint. This situation coincides with the one described in Section 4.7.3. Table 12 lists the energy price, marginal costs, and Lagrange multipliers calculated by the IPLS method for this case. As described in Section 4.7.3, the values obtained by the IPLS method for s_i^* and y_i^* ; $i = 1, \dots, n$ are all zero except for the value of s_2^* , which is associated with the constraint for lower output capabilities at unit 2. Except for unit 2, all the marginal costs are equal to the energy price. These results are in accordance with the theory described Section 4.7.3 and also with (4.14).

4.8.3. Case 3: Power System with 13 Generating Units

Table 8 shows the optimal dispatch, P_i^* , $i = 1, \dots, n$, calculated by the IPLS method for the system with 13 generating units. Table 13 lists the energy price, marginal costs, and Lagrange multipliers calculated by the IPLS method. As the latter table indicates, generated units 1, 2, and 3 have reached their upper limits, and generated units 10, 11, 12, and 13 have reached their lower limits, indicating that there are 6 active constraints in the optimal solution calculated by the IPLS method. This situation coincides with the one described in Section 4.7.4. As described previously, the values obtained by the IPLS method for s_i^* and y_i^* ; $i = 1, \dots, n$ are zero except for the values of y_1^* , y_2^* , y_3^* , which are associated with upper output capabilities, and also for s_{10}^* , s_{11}^* , s_{12}^* , s_{13}^* , which are associated with lower output

Table 13: Incremental analysis for the 13-generator system.

Unit	λ_i^*	s_i^*	y_i^*	w
1	8,4807	0	0,2635	8,7443
2	8,5031	0	0,2411	
3	8,5031	0	0,2411	
4	8,7443	0	0	
5	8,7443	0	0	
6	8,7443	0	0	
7	8,7443	0	0	
8	8,7443	0	0	
9	8,7443	0	0	
10	8,8272	0,0828	0	
11	8,8272	0,0828	0	
12	8,9124	0,1680	0	
13	8,9124	0,1680	0	

capabilities. The marginal costs for all the remaining units (4, 5, 6, 7, 8, and 9) are equal to the energy price. These results are in accordance with the theory described in Sections 4.7.2 and 4.7.3 and also with (4.13) and (4.14).

As expected, in all the cases analysed in Tables 11, 12, and 13, the value of the generation cost $w = \lambda_i^* - s_i^* + y_i^*$ was determined univocally, which proves the equal incremental cost criterion [29].

The analysed results indicate that the dispatches P_i^* , $i = 1, \dots, n$ calculated, respectively, in Tables 2, 5, and 8 and the results for λ_i^* , w , s_i^* , and y_i^* determined, respectively, in Tables 11, 12, and 13, satisfy the complementary slackness conditions set forth in (4.7) and (4.8).

5. Conclusions

This paper proposes a predictor-corrector primal-dual interior point method which introduces line search procedures (IPLS) in both the predictor and corrector steps. The Fibonacci search is used in the predictor step, while an Armijo search is used in the corrector step. The method is developed for application to the economic dispatch (ED) problem, which is an example of a quadratic programming problem studied in the field of power systems analysis. ED problems have already been solved through primal-dual interior point methods, although the strategy adopted here to solve the problem has not yet been tested.

The proposed algorithm is applied to solve ED problems in case studies involving systems with 3, 6, and 13 generating units. The results provided by the proposed IPLS method are compared to those provided by several other methods described in the literature, such as the predictor-corrector primal-dual (PCPD) interior point method, hybrid genetic algorithm, coevolutionary genetic algorithm, hybrid atavistic genetic algorithm, and cultural algorithm. The dispatches calculated by all the types of genetic algorithms could not reach the global optimal dispatch attained by the interior point methods PCPD and IPLS.

The computational effort of the interior point methods tested (PCPD and IPLS) was evaluated and measured in terms of the number of iterations required to find the optimal solution. The results indicate that the introduction of line searches in the IPLS method

considerably reduces the number of iterations required for solving ED problems. Line searches pose represent only a minor additional computational effort in the overall process of finding a solution; hence, the computational effort for solving ED problems is also greatly improved by the proposed IPLS method.

A theory for performing a postoptimization analysis was also analysed. This theory highlights the relation among energy prices, incremental generation costs, and other Lagrange multipliers. This mathematical relation is used to validate optimal solutions for ED problems calculated by the IPLS method.

Further research could involve the representation of the so-called valve point loading in the objective function of ED problems. In that case, the nondifferentiability of the objective function should be treated appropriately.

Acknowledgment

The authors are grateful for the financial support of FAPESP and CNPq (Brazil).

References

- [1] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, no. 4, pp. 373–395, 1984.
- [2] I. I. Dikin, "Iterative solution of problems of linear and quadratic programming," *Doklady Akademii Nauk SSSR*, vol. 174, pp. 747–748, 1967 (Russian).
- [3] E. R. Barnes, "A variation on Karmarkar's algorithm for solving linear programming problems," *Mathematical Programming*, vol. 36, no. 2, pp. 174–182, 1986.
- [4] R. J. Vanderbei, M. S. Meketon, and B. A. Freedman, "A modification of Karmarkar's linear programming algorithm," *Algorithmica*, vol. 1, no. 4, pp. 395–407, 1986.
- [5] I. Adler, M. G. C. Resende, G. Veiga, and N. Karmarkar, "An implementation of Karmarkar's algorithm for linear programming," *Mathematical Programming*, vol. 44, no. 1–3, pp. 297–335, 1989.
- [6] A. R. Balbo, M. A. S. Souza, and E. C. Baptista, "Métodos primal-dual de pontos interiores aplicados à resolução de problemas de despacho econômico: sobre a influência da solução inicial," in *Proceedings of the Simpósio Brasileiro de Pesquisa Operacional, João Pessoa—Pb. Anais do XL SBPO (XL SBPO '08)*, pp. 2074–2085, ILTC, Rio de Janeiro, RJ, Brazil, 2008.
- [7] N. Megiddo and M. Shub, "Boundary behavior of interior point algorithms in linear programming," *Mathematics of Operations Research*, vol. 14, no. 1, pp. 97–146, 1989.
- [8] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley & Sons, New York, NY, USA, 1968.
- [9] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming*, vol. 4 of *Classics in Applied Mathematics*, SIAM, Philadelphia, Pa, USA, 2nd edition, 1990.
- [10] K. R. Frisch, *The Logarithmic Potential Method of Convex Programming*, University Institute of Economics, Oslo, Norway, 1955.
- [11] P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin, and M. H. Wright, "On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method," *Mathematical Programming*, vol. 36, no. 2, pp. 183–209, 1986.
- [12] Y. Ye, "An $O(n^3L)$ potential reduction algorithm for linear programming," in *Contemporary Mathematics*, vol. 114, pp. 91–107, American Mathematical Society, Providence, RI, USA, 1990.
- [13] J. Renegar, "A polynomial-time algorithm, based on Newton's method, for linear programming," *Mathematical Programming*, vol. 40, no. 1, pp. 59–93, 1988.
- [14] P. M. Vaidya, "An algorithm for linear programming which requires $O(((m+n)n^2 + (m+n)^{1.5}n)L)$ arithmetic operations," *Mathematical Programming*, vol. 47, no. 2, pp. 175–201, 1990.
- [15] N. Megiddo, "On the complexity of linear programming," in *Advances in Economic Theory*, T. Bewley, Ed., pp. 225–268, Cambridge University Press, Cambridge, UK, 1989.

- [16] C. C. Gonzaga, "An algorithm for solving linear programming problems in $O(n^3L)$ operations," in *Progress in Mathematical Programming: Interior-Point and Related Methods*, N. Megiddo, Ed., pp. 1–28, Springer, New York, NY, USA, 1989.
- [17] C. C. Gonzaga, "Polynomial affine algorithms for linear programming," *Mathematical Programming*, vol. 49, no. 1, pp. 7–21, 1990.
- [18] R. D. C. Monteiro and I. Adler, "Interior path following primal-dual algorithms. I. Linear programming," *Mathematical Programming*, vol. 44, no. 1, pp. 27–41, 1989.
- [19] R. D. C. Monteiro, I. Adler, and M. G. C. Resende, "A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension," *Mathematics of Operations Research*, vol. 15, no. 2, pp. 191–214, 1990.
- [20] M. Kojima, S. Mizuno, and A. Yoshise, "A primal-dual interior point algorithm for linear programming," in *Progress in mathematical programming: Interior-Point and Related Methods*, N. Megiddo, Ed., pp. 29–47, Springer, New York, NY, USA, 1989.
- [21] S. Mehrotra and J. Sun, "An algorithm for convex quadratic programming that requires $O(n^{3.5}L)$ arithmetic operations," *Mathematics of Operations Research*, vol. 15, no. 2, pp. 342–363, 1990.
- [22] I. J. Lustig, R. E. Marsten, and D. F. Shanno, "On implementing Mehrotra's predictor-corrector interior-point method for linear programming," *SIAM Journal on Optimization*, vol. 2, no. 3, pp. 435–449, 1992.
- [23] Y. C. Wu, A. S. Debs, and R. E. Marsten, "Direct nonlinear predictor-corrector primal-dual interior point algorithm for optimal power flows," *IEEE Transactions on Power Systems*, vol. 9, no. 2, pp. 876–883, 1994.
- [24] S. C. Fang and S. Puthenpura, *Linear Optimization and Extensions: Theory and Algorithms*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.
- [25] J. O. Kim, D. J. Shin, J. N. Park, and C. Singh, "Atavistic genetic algorithm for economic dispatch with valve point effect," *Electric Power Systems Research*, vol. 62, no. 3, pp. 201–207, 2002.
- [26] N. M. Rodrigues, *Um algoritmo cultural para problemas de despacho de energia elétrica*, Dissertação de Mestrado, Universidade Estadual de Maringá, 2007.
- [27] M. M. A. Samed, *Um algoritmo genético Híbrido co-evolutivo para resolver problemas de despacho*, Tese de Doutorado, UEM, Departamento de Engenharia Química, 2004.
- [28] H. H. Happ, "Optimal power dispatch—a comprehensive survey," *IEEE Transactions on Power Apparatus and Systems*, vol. 96, no. 3, pp. 841–854, 1977.
- [29] M. J. C. Steinberg and T. H. Smith, *Economic Loading of Power Plants and Electric Systems*, MacGraw-Hill, 1943.
- [30] A. R. L. Oliveira, S. Soares, and L. Nepomuceno, "Optimal active power dispatch combining network flow and interior point approaches," *IEEE Transactions on Power Systems*, vol. 18, no. 4, pp. 1235–1240, 2003.
- [31] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Non Linear Programming-Theory and Algorithm*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1993.
- [32] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, Pa, USA, 1997.

