New Versions of Interior Point Methods Applied to the Optimal Power Flow Problem

Edgardo D. Castronuovo, Jorge M. Campagnolo and Roberto Salgado

***Abstract*- Interior Point methods for Nonlinear Programming have been extensively used to solve the Optimal Power Flow problem. These optimization algorithms require the solution of a set of nonlinear equations to obtain the optimal solution of the power network equations. During the iterative process to solve these equations, the search for the optimum is based on the combination of two directions: the affine-scaling and the centralization. In this work, it is shown that the suitable combination of these directions can increase the potential of the optimization algorithm in terms of speed and reliability. Real size and test networks are used to assess the performance of the presented methods.**

***Keywords*- Interior Point Methods, Nonlinear Programming, Nonlinear Systems, Optimal Power Flow, Optimization Methods, Power Systems.**

I Introduction

The application of Primal-Dual versions of the Interior Point method (IPM) has been essential to obtain the solution of the nonlinear Optimal Power Flow (OPF) problem [1, 3, 5, 6, 8]. These type of methods combine the simplicity in the treatment of the inequality restrictions with an efficient computational performance. These characteristics have been subject of many theoretical researches. A review of the application of the IPM in the OPF can be found in [10, 11].

To solve optimization problems through the IPM for Nonlinear Programming (NLP) a perturbation parameter is introduced in the complementarily Karush-Kuhn-Tucker (KKT) condition [2]. In the present work, it is shown that several versions of the IPMs can be used to the calculation of this parameter. Depending on the approach adopted for this calculation, it is possible to decrease the number of iterations and to improve the convergence characteristics in both simples and complex problems. In this work, the performance of five algorithms of nonlinear OPF is analyzed, with basis on results obtained for test-systems and real size networks.

This work was supported in part by CEPEL and RECOPE.

E. C. Castronuovo is with CEPEL, Caixa Postal 68007, 21944-970 Rio de Janeiro, RJ, Brazil. (e-mail: edgardo@cepel.br).

J. M. Campagnolo and R. Salgado are with Universidade Federal de Santa Catarina, CTC/EEL/Labspot – Campus Universitário, Trindade - 88040-900 Florianópolis, SC, Brazil (e-mail {campagno, salgado}@labspot.ufsc.br)

II The Interior Point Methods for Nonlinear Programming

Suppose a general optimization problem represented by

$$\begin{aligned} &\text{Min.} && f(\underline{x}) && (1.a)\\ &\text{s. t.} && \underline{h}(\underline{x})=\underline{0} && (1.b)\\ & && \underline{g}(\underline{x}) \le \underline{0} && (1.c) \end{aligned}$$

where, $\underline{x}$ is the vector of optimization variables; $f(\underline{x})$ represents the performance index to be optimized; $\underline{h}(\underline{x})$ and $\underline{g}(\underline{x})$ are the equality and inequality constraints, respectively.

The application of the optimality conditions of KKT in the problem represented by (1) provides

$$\begin{aligned} &\nabla_x L(\underline{x},\underline{\lambda}_h,\underline{\lambda}_g) = \underline{0} && (2.a)\\ &\underline{h}(\underline{x}) = \underline{0} && (2.b)\\ &\underline{g}(\underline{x}) \le \underline{0} && (2.c)\\ &[\underline{\lambda}_g]\ \underline{g}(\underline{x}) = \underline{0} && (2.d)\\ &\underline{\lambda}_g \ge \underline{0} && (2.e) \end{aligned}$$

where, $\nabla_x L(\underline{x},\underline{\lambda}_h,\underline{\lambda}_g) = \nabla_x f(\underline{x}) + \nabla_x \underline{h}(\underline{x})\ \underline{\lambda}_h + \nabla_x \underline{g}(\underline{x})\ \underline{\lambda}_g$ is the vector of first derivatives of the Lagrangean function with respect to the variables $\underline{x}$; $[..]$ denotes a diagonal matrix with the variables considered; and $\underline{\lambda}_h$ and $\underline{\lambda}_g$ are the Lagrange multipliers of the equality and inequality constraints, respectively.

In order to obtain the solution of the optimization problem represented by (1) through the IPM, the modification of the KKT optimality conditions through the convergence process is required. Equations (2.c) are converted in equalities through the slack variables $\underline{s} > \underline{0}$, and the strict combinatory problem of the complementary equations (2.d) is perturbed through the parameter $\mu \ge 0$. This approach results in the following expressions:

$$\begin{aligned} &\nabla_x L(\underline{x},\underline{\lambda}_h,\underline{\lambda}_g) = \underline{0} && (3.a)\\ &\underline{h}(\underline{x}) = \underline{0} && (3.b)\\ &\underline{g}(\underline{x}) + \underline{s} = \underline{0} && (3.c)\\ &[\underline{\lambda}_g]\ \underline{s} - \mu\ \underline{e} = \underline{0} && (3.d)\\ &(\underline{s}, \underline{\lambda}_g, \mu) \ge \underline{0} && (3.e) \end{aligned}$$

where, μ is the *perturbation parameter*; and $\underline{e}$ is the unitary vector $[1,...,1]^t$.

The nonlinear equations (3.a) to (3.d) are iteratively solved by Newton's method in two steps. In the first, the variables

$\Delta\underline{x}$ and $\Delta\underline{\lambda}_h$ are determined through the solution of the reduced set of linear equations

$$\begin{bmatrix} H & J \\ J^t & 0 \end{bmatrix} * \begin{bmatrix} \Delta\underline{x} \\ \Delta\underline{\lambda}_h \end{bmatrix} = \begin{bmatrix} \underline{t_1} \\ \underline{t_2} \end{bmatrix} \quad (4)$$

where

$$H = \nabla_{xx} L + \nabla \underline{g}(\underline{x})^t \begin{bmatrix} \underline{\lambda_g} \\ \underline{s} \end{bmatrix} \nabla \underline{g}(\underline{x}) \quad (5.a)$$

$$J = \nabla \underline{h}(\underline{x}) \quad (5.b)$$

$$\underline{t_1} = -\underline{\nabla_x L}(\underline{x}, \underline{\lambda_h}, \underline{\lambda_g}) - \nabla \underline{g}(\underline{x})^t \left[\underline{s^{-1}}\right] \left(\mu \underline{e} + \left[\underline{\lambda_g}\right] \underline{g}(\underline{x})\right) \quad (5.c)$$

$$\underline{t_2} = -\underline{h}(\underline{x}) \quad (5.d)$$

$$\nabla_{xx} L = \nabla_{xx} f(x) + \nabla_{xx} h(x) + \nabla_{xx} g(x) \quad (5.e)$$

In the second step, the slack variables and corresponding multipliers are obtained by

$$\Delta\underline{s} = -\underline{g}(\underline{x}) - \underline{s} - \nabla \underline{g}(\underline{x})\ \Delta\underline{x} \quad (6.a)$$

$$\Delta\underline{\lambda}_g = -\underline{\lambda}_g + \left[\underline{s^{-1}}\right] * \left\{\mu \underline{e} - \left[\underline{\lambda}_g\right] \Delta\underline{s}\right\} \quad (6.b)$$

The solution of the optimization process in two steps is performed aiming at reducing the dimension of the linear system, and consequently, to decrease the number of operations in the factorization processes.

In order to assure the non-negativity of $\underline{s}$ and $\underline{\lambda}_g$ in (3.e) two step sizes are used to update the primal and the dual variables, respectively (7.a and 7.b).

$$\alpha_p = min\left[\min_{\Delta s_j < 0} \frac{s_j}{\left|\Delta s_j\right|}, 1.0\right] \quad (7.a)$$

$$\alpha_d = min\left[\min_{\Delta \lambda_{gj} < 0} \frac{\lambda_{gj}}{\left|\Delta \lambda_{gj}\right|}, 1.0\right] \quad (7.b)$$

The primal and dual variables are updated, according to (8).

$$\underline{x} = \underline{x} + 0.99995\ \alpha_p\ \Delta\underline{x} \quad (8.a)$$

$$\underline{s} = \underline{s} + 0.99995\ \alpha_p\ \Delta\underline{s} \quad (8.b)$$

$$\underline{\lambda}_h = \underline{\lambda}_h + 0.99995\ \alpha_d\ \Delta\underline{\lambda}_h \quad (8.c)$$

$$\underline{\lambda}_g = \underline{\lambda}_g + 0.99995\ \alpha_d\ \Delta\underline{\lambda}_g \quad (8.d)$$

where the factor *0.99995* assures that (3.e) is satisfied.

III The Perturbation Parameter

Since the original KKT condition (2.d) must be satisfied at the optimal point, the parameter of perturbation μ must converge to zero during the iterations. The parameter μ can be computed through the primal-dual distance. For the *i-th* inequality constraint, this distance is defined by (2.d). The primal-dual distance can be expressed as a function of the slack variable $\underline{s}$, as adopted in the present work. The parameter μ is evaluated through the Primal-Dual Average Distance (*pdad*) from the current solution to the optimal point, measured in the complementary equations. This value is modified by a parameter σ, the so-called *parameter of the direction combination*, which defines the trajectory to the optimal solution. Therefore, μ is expressed as (9), for *niq* inequality constraints [1].

$$\mu = \sigma * pdad = \sigma * \frac{\underline{\lambda_g}^t \underline{s}}{niq} \quad (9)$$

In order to select the value of σ, two cases must be observed:

- $\sigma = 0$, which corresponds to the so-called *affine-scaling direction*. In this case, the optimal point is obtained through the non-perturbed solution of the KKT conditions;
- $\sigma = 1$, which results in the so-called *centralization direction*. In this case, a perturbed set of nonlinear equations is solved. Therefore, a non-optimal solution is found, with a primal-dual distance equal to the initial value of μ.

The value of σ between these limits represents a linear combination of the affine-scaling and centralization directions [3]. To decrease the value of μ during the iterative process it is necessary that $0 < \sigma < 1$, as suggested in [2].

The *Central Path* is defined as the curve that satisfies (3), for pre-specified values of μ. At each iteration, it is possible to evaluate the proximity to this path. The primal-dual Euclidean distance to the central trajectory in the *k*-iteration is defined as:

$$\delta(\underline{\lambda_g}^k, \underline{s}^k, \mu^k) = \left\| \frac{\left[\underline{\lambda_g}^k\right] * \underline{s}^k}{\mu^k} - \underline{e} \right\| \quad (10)$$

IV Versions of the IPM for NLP

A. Conventional Primal-Dual IPM.

In this approach, the parameter σ assumes a constant value (generally close to 0.1) during the iterative process. This choice defines a search direction in which 90% aims the path to the optimal point and 10% the centralization of the trajectory.

B. Successive Directions algorithm

For this algorithm [3, 4], the distance to the central path is evaluated at each iteration and compared to a pre-specified value. If the distance is smaller than this value, the affine-scale direction is used. Otherwise, the centralization direction is adopted.

C. Centralized IPM

At each iteration of this algorithm [3, 4], the proximity to the central path is assessed. If this proximity satisfies the tolerance, the search is performed in a direction similar to that of the conventional algorithm, with $0 < \sigma < 1$. Otherwise, the centralization direction is adopted.

D. Predictor-Corrector Primal-Dual IPM.

At each iteration of the PC-PDIPM two steps are performed, which are described as follows.

Prediction step: At this step, the primal-dual distances corresponding to the following iteration are estimated, considering $\sigma = 0$. Thus, (4), (6) and (7) must be solved. At this stage, the slack primal ($\underline{s}^{as}$) and dual ($\underline{\lambda}_g^{as}$) variables are obtained from (11).

$$\underline{s}^{as} = \underline{s} + 0.99995\ \alpha_p\ \Delta\underline{s} \quad (11.a)$$
$$\underline{\lambda}_g^{as} = \underline{\lambda}_g + 0.99995\ \alpha_d\ \Delta\underline{\lambda}_g \quad (11.b)$$

Correction Step: in which a second-order expansion in Taylor series of (3.d) is performed. This provides

$$[\underline{s}]\ \Delta\underline{\lambda}_g + [\underline{\lambda}_g]\ \Delta\underline{s} + [\Delta\underline{s}]\ \Delta\underline{\lambda}_g = -[\underline{\lambda}_g]\ \underline{s} + \mu\underline{e} \quad (12)$$

Since the product $[\Delta\underline{s}]\Delta\underline{\lambda}_g$ is nonlinear, the PC-PDIPM uses the information of the prediction stage to estimate this term. Then, (12) is modified to

$$[\underline{s}]\ \Delta\underline{\lambda}_g + [\underline{\lambda}_g]\ \Delta\underline{s} = -[\underline{\lambda}_g]\ \underline{s} + \mu\underline{e} - [\Delta\underline{s}^{as}]\ \Delta\underline{\lambda}_g^{as} \quad (13)$$

The changes expressed in (13) are reflected in both the right side of (4) and the evaluation of $\Delta\lambda_g$. Then, in the correction stage (5.c) and (6.b) are substituted by (14.a) and (14.b), respectively.

$$\underline{t_1} = -\nabla_x L(\underline{x}, \underline{\lambda}_h, \underline{\lambda}_g) + \\ -\nabla\underline{g}(\underline{x})^t [\underline{s}^{-1}]\ (\mu\underline{e} + [\underline{\lambda}_g]\underline{g}(\underline{x}) - \Delta\underline{s}^{as}\Delta\underline{\lambda}_g^{as}) \quad (14.a)$$

$$\Delta\underline{\lambda}_g = -\underline{\lambda}_g + [\underline{s}^{-1}] * \{\mu\underline{e} - [\underline{\lambda}_g]\Delta\underline{s} - \Delta\underline{s}^{as}\Delta\underline{\lambda}_g^{as}\} \quad (14.b)$$

The parameter σ is dynamically estimated through (15).

$$\sigma = \left(\frac{(\underline{\lambda}_g^{as})^t\ \underline{s}^{as}}{\underline{\lambda}_g^t\ \underline{s}} \right)^3 \quad (15)$$

where the products $(\underline{\lambda}_g^t\ \underline{s})$ and $\{(\underline{\lambda}^{as}{}_g)^t\ \underline{s}^{as}\}$ represent the primal-dual distances in the current iteration and the predicted primal-dual distance if the affine-scaling direction is used, respectively.

Equation (15) dynamically estimates the need for the centralization in the trajectory to the optimal solution. If the primal-dual distances decrease, when the affine-scaling direction is considered, this direction is predominantly used at the correction stage ($\sigma \rightarrow 0$). Otherwise, the correction stage privileges the centralization direction ($\sigma \rightarrow 1$).

E. Largest Step Path Following algorithm

In this algorithm [7, 8], the trajectory to the optimal solution is located at a pre-specified distance of the central path. At each iteration, the value of σ satisfying this adjacency condition is computed.

From any initial point, the algorithm seeks a trajectory located at a pre-specified distance ε (defined by the user) of the central path. Thus, in the *(k+1)-th* iteration, the equality (16) should be satisfied.

$$\delta(\underline{\lambda}_g^{k+1}, \underline{s}^{k+1}, \mu^{k+1}) = \left\| \frac{(\underline{\lambda}_g^{k+1})^t * \underline{s}^{k+1}}{\sigma * pdad} - \underline{e} \right\| = \varepsilon \quad (16)$$

The values of the σ that combine the affine-scaling and centralization directions to satisfy (16) are sought. The substitution of (8) in (16) and the reordering of the resulting expression provides (17) [3].

$$\delta(\underline{\lambda}_g^{k+1}, \underline{s}^{k+1}, \mu^{k+1}) = \left\| \frac{(\Delta\underline{\lambda}_g)^t * \Delta\underline{s}}{\sigma * pdad} \right\| = \varepsilon \quad (17)$$

The search direction is a linear combination of the affine-scaling and centralization directions [8]. Therefore, (17) can be expressed as a function of the parameter of the direction combination, as in (18).

$$\delta(\underline{\lambda}_g^{k+1}, \underline{s}^{k+1}, \mu^{k+1}) = \varepsilon = \\ \left\| \frac{(\sigma * \Delta\underline{\lambda}_g^{\,c} + (1-\sigma) * \Delta\underline{\lambda}_g^{as})^t * (\sigma * \Delta\underline{s}^{\,c} + (1-\sigma) * \Delta\underline{s}^{as})}{\sigma * pdad} \right\| \quad (18)$$

For a pre-specified value of ε, the only variable unknown of (18) is σ. If the Euclidean norm is used, the value of σ is obtained from the solution of a fourth order equation. In the present work, the Bisection method [7, 9] is applied to solve this equation and the largest root in the (0, 1) interval is used. If there is no value of σ in this interval, $\sigma = 0.1$ is adopted.

Small modifications in the conventional version of the IPM are required to implement these combinations of the directions. The last two versions require an additional backward-forward substitution at each iteration, with upper cpu-time by iteration. However, the ability of control the path to the convergence can reduce the total of computational effort, as verified in Results section.

V RESULTS

Aiming at assessing the performance of the IPM versions, an OPF computer program was implemented. The function to be minimized is the active power generation cost, expressed as a quadratic function of the *ng* active power generation Pg_i; the equality constraints $\underline{h}(\underline{x})$ are the power balance equations; the inequality constraints $\underline{g}(\underline{x})$ are the limits in the active and reactive power generation and the bus voltage magnitude bounds; and the optimization variables $\underline{x}$ are the magnitude and angle of the bus voltages and active power generation.

The optimization algorithms were implemented in Fortran 77. The IEEE 30 and 118 buses test systems and two real networks, equivalent to the South-Southeastern Brazilian (SSB) power system (with 352 and 750 buses), were used. Table 1 shows the main characteristics of these systems.

TABLE 1
CHARACTERISTICS OF THE TEST SYSTEMS

	IEEE 30	IEEE 118	SSB 352	SSB 750
Branches	41	179	385	952
Generators	6	34	30	87
Order of (4)	119	471	1407	2999

A. Conventional IPM

Table 2 shows the number of iterations to the convergence required for the conventional IPM, with $\sigma = 0.1$. The values of the perturbation parameter at each iteration are shown in Fig. 1, for SSB 352 system.

TABLE 2
CONVENTIONAL IPM - CONVERGENCE – NUMBER OF ITERATIONS

	IEEE 30	IEEE 118	SSB 352	SSB 750
Iterations	10	14	14	25

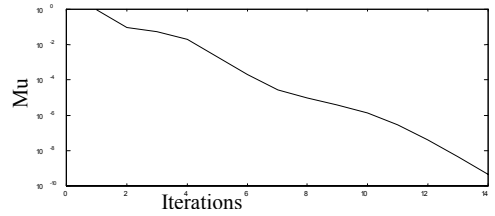

Fig. 1: Conventional IPM - Parameter of Perturbation – SSB 352

From Fig. 1, it is noted that the parameter of perturbation decreases almost linearly in a logarithmic scale, approximately an order of magnitude for iteration. This result shows the ability of the IPMs in reducing the perturbation in order to satisfy the original KKT conditions. Fig. 2 shown the distances between the trajectory and the central path at each iteration, as defined in (10).

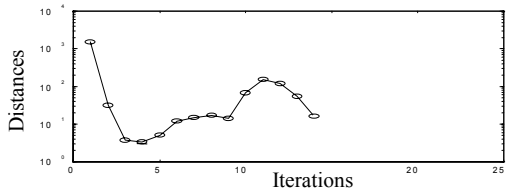

Fig. 2: Conventional IPM - Distances between Trajectory and Central Path – SSB 352

From Fig. 2, the search direction of the conventional IPM does not keep a constant distance to the central path. During some iterations this distance is up to 200 times the value of μ. In reality, this version does not consider the distance to the central path in its formulation, using an σ empiric and constant during the iterations.

B. Successive Directions algorithm

Table 3 shows the number of iterations for the convergence requested for the Successive Directions algorithm. The maximum distance between the central and trajectory path corresponds to the value of the primal-dual average distance (*pdad*) at each iteration.

TABLE 3
SUCCESSIVE DIRECTIONS IPM - CONVERGENCE – NUMBER OF ITERATIONS

	IEEE 30	IEEE 118	SSB 352	SSB 750
Iterations	14	19	19	37

From Tables 2 and 3, it is observed that the Successive Directions algorithm requests a total number of iterations to the convergence greater than that corresponding to the Conventional version (average 28%). However the proximity between the search trajectory and the central path adopted in this example is significant. Larger tolerances can decrease the number of iterations of the Successive Directions algorithm, making its performance similar to that of the conventional version and reducing the centralization of the trajectory. For this reason, a compromise should be adopted between these targets. In Fig. 3, the proximity between the convergence and central path is showed.

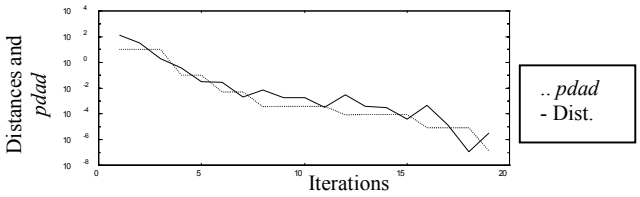

Fig. 3: Successive Directions IPM – Primal Dual Average Distance and Proximity between Trajectory and Central Path – SSB 352

In this algorithm, similarly to the case of the Centralized Interior Point algorithm, the primal-dual average distance (*pdad*) should be used (instead of μ). The use of $\sigma = 0$ in the centralization steps results in $\mu = 0$, which implies in difficulties to observe the performance of these methods.

The steps in the centralization direction do not produce appreciable decrease in the value of *pdad*. This effect is the reason for the non-continuous variation in the decrease of the value of the *pdad*, shown in Fig. 3. This figure shows the ability of the Successive Directions algorithm to approximate both the trajectory and the central path. The distances between these directions decrease as the value of the *pdad* is reduced, resulting in an overlap between these curves.

C. Centralized IPM

Table 4 presents the number of iterations for the convergence of the Centralized IPM. The maximum distance between the central and trajectory path corresponds to the value of the primal-dual average distance (*pdad*) at each iteration.

TABLE 4
CENTRALIZED IPM - CONVERGENCE – NUMBER OF ITERATIONS

	IEEE 30	IEEE 118	SSB 352	SSB 750
Iterations	13	17	19	35

From this table, the number of iterations required for the convergence of the centralized IPM algorithm is generally smaller than that corresponding to the Successive Directions algorithm (Table 3). The reduction obtained by the centralized IPM algorithm is due to the percent of centralization (10 or 100%) taken during the iterative process. However, the centralization can require more iterations than those of the conventional method (Table 2). The condition of neighborhood between the trajectory and the central path imply in requirements, which result in a larger number of iterations even for relatively simple problems.

The Centralized Interior Point can be convenient in problems in which the optimization variables have small intervals of variation. For example, the active power loss can be minimized by fixing the active power injection in every generation bus, except for the reference bus, and minimizing the active power injection in this bus. Table 5 shows the number of iterations for convergence of this problem.

TABLE 5
CENTRALIZED AND CONVENTIONAL IPMS - LOSS MINIMIZATION PROBLEM
CONVERGENCE – NUMBER OF ITERATIONS

	IEEE 30	IEEE 118	SSB 352	SSB 750
Conventional IPM	11	10	10	8
Centralized IPM (prox. = *pdad*)	15	12	10	11
Centralized IPM (prox. = 3 * *pdad*)	14	9	8	7

Table 5 presents the number of iterations for convergence of the Centralized IPM with two proximity levels between the search direction and the central path. For a strong centralization in the search direction, it is generally requested some more iterations (between 0 and 30%) than those required by the conventional IPM version. However, small increments in the approximation (3 times the value of *pdad*) allow to reduce the number of iterations to the convergence for the largest systems (between 10 and 20%), with respect to those of the conventional IPM version.

D. Predictor-Corrector Primal-Dual IPM

Table 6 shows the number of iterations for the convergence requested by P-C PDIPM and the active power cost minimization.

TABLE 6
P-C PDIPM - CONVERGENCE – NUMBER OF ITERATIONS

	IEEE 30	IEEE 118	SSB 352	SSB 750
Iterations	7	10	11	18

In this case, the number of iterations is 25% to 30% smaller than that corresponding to the conventional IPM version (Table 2). This reduction is due to both, the more accurate representation of the complementarily equations and the dynamic determination of the parameter of directions combination (σ). The values of σ during the iterative process are shown in Fig. 4.

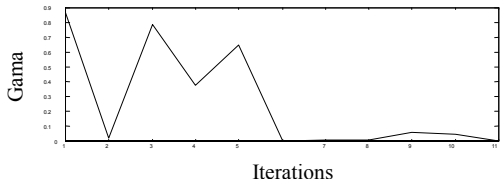

Fig. 4: P-C PDIPM - σ Values - SSB 352

In all cases previously analyzed the parameter σ could assume a constant value or could change to two pre-specified values. In case of the P-C PDIPM, the value of σ can assume any value in the interval (0, 1). In Fig. 4, the non-continuos variation of σ during the iterative process can be noted. At the beginning of the iterative process, a large variation in the values of σ is observed, with high centralization indexes. The algorithm uses small values of σ in last iterations, giving priority to the convergence of the process through the affine-scaling direction.

In Fig. 5, the distances between both the search trajectory and the central path are shown for the SSB 352 system.

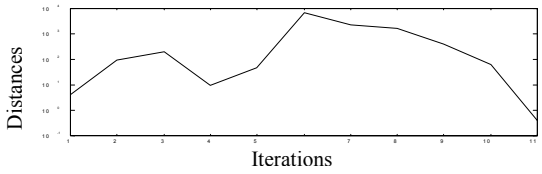

Fig. 5: P-C PDIPM – Proximity between Trajectory and Central Path – SSB 352

The P-C PDIPM algorithm does not consider the distance of the trajectory to the central path in neither predictor nor corrector steps. Consequently, the distances between the search trajectory and the central path change at each iteration, reaching large values (up to 9,000 times the *pdad* value in the iteration), as shown in the Fig. 5.

The P-C PDIPM requires a number of iterations to the convergence lower than that of the conventional IPM version, at the expenses of increasing the distances between the search trajectory and the central path.

E. Largest Step Path Following Algorithm

Table 7 presents the number of iterations for the convergence of the Largest Step Path Following algorithm. In this case, the proximity between the trajectory and central path is 3 times of value of *pdad* at each iteration.

TABLE 7
LARGEST STEP PATH FOLLOWING ALGORITHM
ACTIVE POWER COST MINIMIZATION - CONVERGENCE
NUMBER OF ITERATIONS

	IEEE 30	IEEE 118	SSB 352	SSB 750
Iterations	10	18	18	29

In the active power cost minimization, the solution of the optimization problem through Largest Step Path Following algorithm (Table 7) requests more iterations (in average, 11%) than the conventional IPM method (Table 2).

Table 8 shows the number of iterations for convergence for the problem of active power loss minimization, assuming the same proximity factor used in Table 7.

TABLE 8
LARGEST STEP PATH FOLLOWING IPM
LOSS MINIMIZATION PROBLEM - CONVERGENCE – NUMBER OF ITERATIONS

	IEEE 30	IEEE 118	SSB 352	SSB 750
Largest Step Path Following	10	8	8	6

The modification of the region of the feasible solutions emphasizes the efficiency of the Largest Step Path Following algorithm to search for the optimal solution in a trajectory close to the central path. From Tables 5 and 8, it can be noted that the application of this algorithm results in a reduction of 25% in the number of iterations (in average) with respect to the conventional IPM method. These reductions are generally greater than those relative to the Centralized IPM (Table 5). In Fig. 6, the distances to the central path for both Largest Step Path Following and Conventional algorithms, are shown.

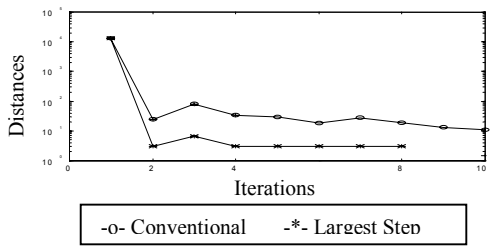

Fig. 6: Largest Step Path Following and Conventional IPMs – Proximity between Trajectory and Central Path – SSB 352

From Fig. 6, it can be observed the ability of the Largest Step Path Following algorithm to preserve a pre-specified proximity between the search trajectory and the central path. In iterations 1 and 3 this proximity condition is not satisfied and then the value $\sigma = 0.1$ is adopted. In the minimization of active power loss, a trajectory relatively close to the central path results in a decrease of the number of iterations.

VI CONCLUSIONS

Interior Point methods can be very efficient in the resolution of real power systems optimization problems. The new versions of this method can improve its performance. Only small modifications in the conventional algorithm are necessary to implement these IPM versions. The Predictor-Corrector Primal-Dual algorithm, the Centralized IPM and Largest Step Path Following algorithm reduce the computational effort requested for the solution of the OPF. The selection of the algorithm to be used in each application depends on the features of the problem to be solved.

Further studies are necessary in order to evaluate the performance of the algorithms with other objective functions.

VII REFERENCES

[1] GRANVILLE, Sergio. Optimal Reactive Dispatch through Interior Point Methods. *IEEE Transactions on Power Systems*, Piscataway, NJ, USA, v. 9, n. 1. Feb. 1994.

[2] EL-BAKRY, A.S.; TAPIA, R.A.; TSUCHIDA, T.; ZHANG, Y. On the Formulation and Theory of the Newton Interior-Point Method for Nonlinear Programming. *Journal of Optimization Theory and Applications*. USA, v. 89, n. 3. June 1996.

[3] CASTRONUOVO, E. D. *On the Application of Interior Points Methods to Optimal Power Flow, with High Performance Computation Techniques (in Portuguese)*. PhD. Thesis. Electrical Engineering Depth., Federal University of Santa Catarina, Brazil. May, 2001.

[4] VILLELA P., M. A. *A Study of Central Path Algorithms in Linear Monotonic Complementarily Problems (in Portuguese)*. MSc. Thesis. Mathematical Depth., Federal University of Santa Catarina, Brazil. Dec, 1997.

[5] MEHROTRA, S. On the Implementation of a Primal-Dual Interior Point Method. *SIAM J. Optimization*. v. 2, n. 4. Nov., 1992. p. 576-601.

[6] WU, Y.C.; DEBS, A. S.; MARSTEN, R. E. A Direct Nonlinear Predictor-Corrector Primal-Dual Interior Point Algorithm for Optimal Power Flows. *IEEE Transactions on Power Systems,* Piscataway, NJ, USA, v.9, n. 2. May, 1994. p.876-883.

[7] GONZAGA, C.C.; BONNANS, J.F. Fast Convergence of the Simplified Largest Step Path Following Algorithm. *Mathematical Programming*. USA, v.76, 1996. p. 95-115

[8] CASTRONUOVO, E.D.; CAMPAGNOLO, J.M.; SALGADO, R. A Largest-Step Central-Path Algorithm Applied to the Optimal Power Flow Problem. *Control and Automation Journal.* Brazil, v. 12, Dec. 2000.

[9] BARBEAU, E.J. *Polynomials*. New York, USA, Springer-Verlag New York Inc., 1989. 455p.

[10] QUINTANA, V.H., TORRES, G.L. and MEDINA-PALOMO J. Interior-Point Methods and Their Applications to Power Systems: A Classification of Publications and Software Codes. *IEEE Transactions on Power Systems,* Piscataway, NJ, USA, v.15, n. 1. Feb, 2000. p.170-176.

[11] HUNEAULT, M. and GALIANA, F.D. A Survey of the Optimal Power Flow Literature. *IEEE Transaction on Power Systems,* Piscataway, NJ, USA, v. 6, n. 2, May 1991.

VIII BIOGRAPHIES

Roberto Salgado received the B.Sc. degree in Electrical Engineering (1976), from Universidade Federal do Pará (Brazil), the M.Sc. degree (1981) from Universidade Federal de Santa Catarina (Brazil) and the Ph.D. degree (1989) from the University of Manchester Institute of Science and Technology (UK). He is lecturer in the Depth. of Electrical Eng. of the Universidade Federal de Santa Catarina (Brazil). His current interests are on Optimization applied to Electrical Eng., Voltage Stability and Power Systems Control.

Jorge M. Campagnolo received his B.Sc. (1979) from the Federal University of Rio Grande do Sul, Brazil. He received his M.Sc. (1984) from the Federal University of Santa Catarina and his D.Sc. (1994) from Federal University of Rio de Janeiro, Brazil. He jointed the Depth. of Electrical Engineering of Federal University of Santa Catarina where he is now an Associate Professor. Mr. Campagnolo general research interest is in the area of computer methods for power systems analysis and control.

Edgardo D. Castronuovo received the B.Sc. degree (1995) in Electrical Engineering from National University of La Plata (Argentina) and the M.Sc. degree (1997) and Ph.D. degree (2001) from Universidade Federal de Santa Catarina (Brazil). Currently, he is a researcher of the Brazilian Federal Center in Electrical Studies (CEPEL). His interests are on Optimization methods, High Performance Computation Techniques and Deregulation of the Electrical Energy Systems.