Towards Reliable Computation of Large-Scale Market-Based Optimal Power Flow

Hongye Wang Robert J. Thomas
hw41@cornell.edu *rjt1@cornell.edu*
School of Electrical and Computer Engineering, Cornell University

Abstract

The deregulated electricity market calls for robust OPF tools that can provide (a) deterministic convergence, (b) accurate computation of a variety of nodal prices, (c) support of continuous costs as well as discrete bids and offers, (d) full active and reactive power flow modeling of large-scale systems, and (e) satisfactory worst-case performance that meets the real-time dispatching requirement. For historical reasons, most prior research on OPF has focused on performance issues, without much treatment of requirements (a)-(c). This paper discusses these new challenges and presents two new algorithms for reliable computation of large-scale market-based OPF: trust-region based augmented Lagrangian method (TRALM) and step-controlled primal-dual interior point method (SCIPM). The former is more theoretically rigid while the latter is more effective in practice. The new algorithms, along with several existing ones, are tested and compared using large-scale power system models.

1. Introduction

The optimal power flow (OPF) problem has been one of the most widely studied subjects in the power system community since Carpentier first published the concept in 1962 [1]. Over the years, researchers have examined various algorithmic techniques that seek to speed up the OPF computation for real-time usage. References [2]-[6] captured most of the work done in the 1970s and the 1980s, a time when several constrained optimization techniques such as the Lagrange multiplier method, penalty function method, and sequential quadratic programming, coupled with gradient method and Newton method for unconstrained optimization, emerged as the leading nonlinear programming (NLP) algorithms for solving AC OPF. In recent years, algorithms based on the primal-dual interior point method (PDIPM) have gained popularity [7]-[13].

Despite all of the advancements being made, the full AC OPF has not been widely adopted in real-time scheduling operations of large-scale power systems. Instead, system operators often use simplified OPF tools that are based on linear programming (LP) and decoupled (DC) system models [14]. Historically, this is mainly due to the lack of powerful computer hardware and efficient AC OPF algorithms. With the advent of low-cost parallel computers and the continued progress of silicon integrations, however, the speed has now become a secondary concern, after algorithm robustness. The remaining prevalent argument for favoring LP-based DC OPF over NLP-based AC OPF is that LP algorithms are deterministic and always yield solutions albeit not necessarily the desired ones, while NLP algorithms are less robust and experience slow convergence or even divergence under worst-case scenarios.

The emergence of deregulated electricity markets poses new challenges to the solution of the OPF problem. Unlike in the regulated system where the goal of computing the OPF is merely minimizing the system cost based on smooth pre-determined quadratic costs, OPF computation is now part of the core pricing mechanism for electricity trading in the deregulated market where discrete bids and offers are changing frequently [14]-[18]. In order to meet their legal obligations of providing timely market settlements and to ensure market fairness and efficiency, independent system operators (ISO) must adopt OPF tools that provide (a) deterministic convergence, (b) accurate computation of a variety of nodal prices (Lagrange multipliers), (c) support of continuous costs as well as discrete bids and offers (e.g. piecewise linear costs), (d) full active and reactive power flow modeling of large-scale systems, and (e) satisfactory worst-case performance that meets the real-time dispatching requirement. Most prior research on OPF has not given enough emphasis to requirements (a)-(c).

1530-1605/07 $20.00 © 2007 IEEE

In this paper, we look into these arising challenges and present two new algorithms for reliable computation of large-scale market-based OPF: trust-region based augmented Lagrangian method (TRALM) and step-controlled primal-dual interior point method (SCIPM). The former integrates the well-proven penalty and augmented Lagrangian method [24] with the trust-region unconstrained optimization technique [19]-[23] to achieve algorithm robustness. The latter amends the popular primal-dual interior point method with a step control procedure that deals with the irregularity of piecewise linear cost curves. Experimental results show that both algorithms (in contrast to some existing ones) yield precise Lagrange multipliers and handle large-scale systems and piecewise linear cost curves well. The TRALM algorithm is more theoretically rigid, but in terms of computational performance, the SCIPM algorithm is superior and more promising for real-time applications. Section 2 of the paper presents the AC OPF formulation, describes trigonometric smoothing of discrete cost curves, and defines metrics that will be used for later algorithm comparisons. The TRALM and SCIPM algorithms are introduced in Section 3 and Section 4, respectively. Section 5 shows the numerical results and compares the algorithms using power system models of several different sizes. The paper is then concluded in Section 6.

2. Problem formulation

The AC OPF problem is mathematically formulated as:

$$\min_{P,Q,V,\theta} \quad C(P,Q,V,\theta) \qquad (1.a)$$

$$\text{s.t.} \quad FA_i(P,V,\theta) = 0 \qquad (1.b)$$

$$FR_i(Q,V,\theta) = 0 \qquad (1.c)$$

$$P_j^{\min} \le P_j \le P_j^{\max} \qquad (1.d)$$

$$Q_j^{\min} \le Q_j \le Q_j^{\max} \qquad (1.e)$$

$$V_i^{\min} \le V_i \le V_i^{\max} \qquad (1.f)$$

$$|SF_k(V,\theta)|^2 \le (S_k^{\max})^2 \qquad (1.g)$$

$$|ST_k(V,\theta)|^2 \le (S_k^{\max})^2 \qquad (1.h)$$

$$i \in \{1,2,...,N_{bus}\} \qquad (1.i)$$

$$j \in \{1,2,...,N_{gen}\} \qquad (1.j)$$

$$k \in \{1,2,...,N_{line}\} \qquad (1.k)$$

where (1.a) is a generic objective function representing the system cost, (1.b)-(1.c) are the nodal real and reactive power balancing equations, (1.d)-(1.f) are the limits on power generations and bus voltages, and (1.g)-(1.h) are squared branch flow constraints. The bounds in inequality constraints (1.d)-(1.h) are typically supplied by unit commitment and security analysis tools in the real-time system operation. Without much impact on the behavior of algorithms, we use the default bounds that come with the sample power system data throughout our study. The changes of price-sensitive loads, transformer taps, and switching shunt capacitors are not modeled in this work but can be easily accommodated through introduction of additional control variables.

In today's market, only the cost of real power is being optimized and thus the objective function (1.a) degenerates into *C(P)*. When quadratic costs are used, the objective function takes the form of:

$$C(P) = \sum_j (c_j + b_j P_j + a_j P_j^2) \qquad (2)$$

Several existing second-order NLP algorithms such as the primal-dual interior point method handle this form of smooth objective function quite well. The electricity market, however, does not use quadratic costs because they do not cognitively match how the market participants want to trade in the real world. Instead, piecewise linear costs based on incremental offers and decremental bids are adopted for better pricing transparency and flexibility. In this case, assuming no startup cost, the objective function can be written as:

$$C(P)=\sum_j \{r_j^{m'}(P_j - d_j^{m'-1}) + \sum_m [r_j^m (d_j^m - d_j^{m-1})]\} \qquad (3)$$

$$m \in \{t \mid 1 \le t \le NB_j,\ d_j^t < P_j\}$$

where *r* represents an offer price, *d* is the power output at the breakpoint on a piecewise linear price curve ($d_0 \equiv 0$), *NB* is the number of power blocks offered to the market from a given generator, and m' is by definition the block index that satisfies $d_j^{m'-1} < P_j < d_j^{m'}$. Figure 1 illustrates an example price curve with four blocks.

As we will see later in Section 5, existing NLP-based OPF algorithms often break down when being applied to solve (1) with the objective function (3), especially in the case of dealing with large-scale systems. There are two main difficulties, i.e. the non-differentiability of price curves, and the lack of step length control to accommodate the abrupt rise and fall of derivatives around breakpoints. We will address the second issue when introducing the new algorithms.

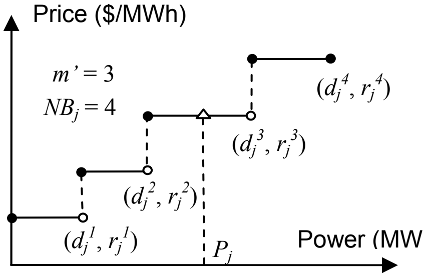

Figure 1. An example discrete price-power curve for the market-based OPF

To overcome the first one, we smooth the objective function using trigonometric functions. (Quadratic smoothing works equally well for second-order NLP algorithms. Trigonometric smoothing is used here because it achieves high-order continuity with no extra computational overhead.) Let the price *r* be a piecewise linear function of the real power *d* as shown in Figure 1, the smoothed price-power function *r'* and the corresponding objective function can be expressed as:

$$\begin{aligned} r'(d) \;&=\; r_+^m - r_-^m \cos[\pi(d - d_-^m)/(d_+^m - d_-^m)] \\ &\qquad if \;\; d_-^m < d < d_+^m , \\ &\quad r(d) \;\; otherwise \\ r_\pm^m \;&\equiv\; \frac{1}{2}(r^{m+1} \pm r^m) \\ d_\pm^m \;&\equiv\; d^m - \alpha(d^m - d^{m\pm1}) \\ m &\in \{t \,|\, 1 \le t < NB_j\} \end{aligned} \tag{4.a}$$

$$C(P) = \sum_j \int_0^{P_j} r_j^{'}(x)\, dx \tag{4.b}$$

where α is a positive number that controls the precision of the approximation. Figure 2 shows a smoothed price curve along with its unmodified counterpart.

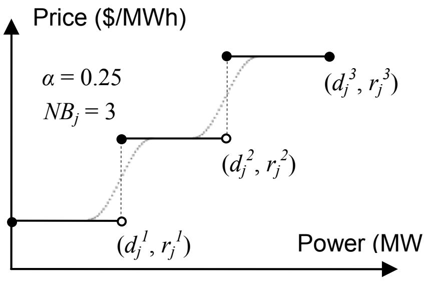

Figure 2. Trigonometric smoothing of a discrete price curve

The Lagrangian of (1) is:

$$\begin{aligned} L = {} & C(P,Q,V,\theta) + \lambda_{FA}^T\, FA(P,V,\theta) \\ & + \lambda_{FR}^T\, FR(Q,V,\theta) + \mu_{P+}^T\,(P - P^{\max} + Z_{P+}) \\ & + \mu_{P-}^T\,(P^{\min} - P + Z_{P-}) \\ & + \mu_{Q+}^T\,(Q - Q^{\max} + Z_{Q+}) \\ & + \mu_{Q-}^T\,(Q^{\min} - Q + Z_{Q-}) \\ & + \mu_{V+}^T\,(V - V^{\max} + Z_{V+}) \\ & + \mu_{V-}^T\,(V^{\min} - V + Z_{V-}) \\ & + \mu_{SF}^T\,(|\,SF(V,\theta)\,|^2 - (S^{\max})^2 + Z_{SF}) \\ & + \mu_{ST}^T\,(|\,ST(V,\theta)\,|^2 - (S^{\max})^2 + Z_{ST}) \end{aligned} \tag{5}$$

where *Z*'s are slack variables and λ and μ are the Lagrange multipliers that are used in the market to price various kinds of electricity transactions. For example, λ_{FA} is the vector of real-power nodal prices. Given a benchmark solution of the first-order and second-order Karush-Kuhn-Tucker (KKT) conditions of (5)

$$(C^*,\, P^*,\, Q^*,\, V^*,\, \theta^*,\, \lambda_{FA,FR}^*,\, \mu_{P\pm,Q\pm,V\pm,SF,ST}^*)$$

and a trial solution under study

$$(C,\, P,\, Q,\, V,\, \theta,\, \lambda_{FA,FR},\, \mu_{P\pm,Q\pm,V\pm,SF,ST}),$$

we define the following metrics to measure the accuracy of the trail solution:

$$\delta_C \equiv \max(\,|\,C - C^*\,| \big/ (1 + |\,C^*\,|\,)\,) \tag{6.a}$$

$$\begin{aligned} \delta_X \equiv \max(& |\,[(P^* + e)]^{-1}(P - P^*)|_\infty , \\ & |\,[(V^* + e)]^{-1}(V - V^*)|_\infty \;) \end{aligned} \tag{6.b}$$

$$\begin{aligned} \delta_\lambda \equiv \max(& |\,[(\lambda_{FA}^* + e)]^{-1}(\lambda_{FA} - \lambda_{FA}^*)|_\infty , \\ & |\,[(\lambda_{FR}^* + e)]^{-1}(\lambda_{FR} - \lambda_{FR}^*)|_\infty \;) \end{aligned} \tag{6.c}$$

$$\begin{aligned} \delta_\mu \equiv \max(& |\,[(\mu_{P\pm}^* + e)]^{-1}(\mu_{P\pm} - \mu_{P\pm}^*)|_\infty , \\ & |\,[(\mu_{Q\pm}^* + e)]^{-1}(\mu_{Q\pm} - \mu_{Q\pm}^*)|_\infty , \\ & |\,[(\mu_{V\pm}^* + e)]^{-1}(\mu_{V\pm} - \mu_{V\pm}^*)|_\infty , \\ & |\,[(\mu_{SF}^* + e)]^{-1}(\mu_{SF} - \mu_{SF}^*)|_\infty , \\ & |\,[(\mu_{ST}^* + e)]^{-1}(\mu_{ST} - \mu_{ST}^*)|_\infty \;) \end{aligned} \tag{6.d}$$

In (6), *e* is the unitary vector and the […] operator diagonalizes the enclosed vector. In a fair market, the

OPF must ensure that all of these δ values are small at the dispatch point.

3. Trust-region based augmented Lagrangian method

The augmented Lagrangian method (ALM) [24] solves a generic optimization problem

$$\begin{aligned} \min_{X} \quad & f(X) \\ \text{s.t.} \quad & H(X) = 0 \\ & G(X) \le 0 \end{aligned} \tag{7}$$

by converting it into a sequence of unconstrained optimization problems with penalty terms:

$$\begin{aligned} \min_{X} \; & L^{k}(X) \equiv f(X) + (\lambda^{k})^{T} H(X) \\ & + \frac{1}{2} H(X)^{T} [W^{k}] H(X) \\ & + \frac{1}{2U_{j}^{k}} \sum_{j=1}^{ni} \{(\max[0, \mu_{j}^{k} + U_{j}^{k} G_{j}(X)])^{2} - (\mu_{j}^{k})^{2}\} \end{aligned} \tag{8}$$

In (8), *ni* is the number of inequality constraints, λ^k and μ^k are trial Lagrange multipliers, and W^k and U^k are penalty parameters. In the so-called "multiplier method", λ^k, μ^k, W^k, and U^k are updated after each round of unconstrained optimization according to:

$$\lambda^{k+1} = \lambda^{k} + [W^{k}] H(X^{k})$$

$$\mu_{j}^{k+1} = \max\{\mu_{j}^{k} + U_{j}^{k} G_{j}(X^{k}),\, 0\}$$

$$W_{j}^{k+1} = \begin{Bmatrix} \beta_{W} W_{j}^{k} & if\, |H_{j}(X^{k})| > \gamma_{W}\, |H_{j}(X^{k})| \\ W_{j}^{k} & if\, |H_{j}(X^{k})| \le \gamma_{W}\, |H_{j}(X^{k})| \end{Bmatrix} \tag{9}$$

$$U_{j}^{k+1} = \begin{Bmatrix} \beta_{U} U_{j}^{k} & if\, G_{j}(X^{k}) > \gamma_{U}\, G_{j}(X^{k}) \\ U_{j}^{k} & if\, G_{j}(X^{k}) \le \gamma_{U}\, G_{j}(X^{k}) \end{Bmatrix}$$

where X^k is the solution of (8), $0 < \gamma_W$, $\gamma_U < 1$, and β_W, $\beta_U > 1$. Convergence is achieved when

$$\begin{aligned} & \| \nabla_{X} L_{X}(X^{k}) \| \le \varepsilon^{k} \\ & \| \lambda^{k+1} - \lambda^{k} \| / (1 + \| \lambda^{k} \|_{\infty}) \le \varepsilon_{\lambda} \\ & \| \mu^{k+1} - \mu^{k} \| / (1 + \| \mu^{k} \|_{\infty}) \le \varepsilon_{\mu} \end{aligned} \tag{10}$$

are satisfied. In (10), the ε's are the tolerance parameters and ε^k decreases to a near-zero value ε^{∞} as the sub-optimization number k increases. Combining with a suitable unconstrained optimization algorithm, the augmented Lagrangian method can solve large-scale nonlinear constrained optimization problems very reliably with accurate Lagrangian multipliers.

In the TRALM algorithm, we use a trust-region method to solve (8). Trust-region methods represent a category of globally convergent unconstrained optimization algorithms [19]-[23]. Compared to Newton's method, which was widely adopted in earlier OPF algorithms, trust-region methods are more robust in handling large-scale systems with indefinite starting points. Fundamentally, trust-region methods form and solve a sequence of simpler approximate optimization problems within "trust regions" (the neighborhoods where approximations remain valid) along the path that leads to the optimum.

```
Let $0 < \tau < \eta < 1, 0 < \gamma_1 < 1 < \gamma_2$,
    $\Delta_0 > 0$, and $X_0$ be given,
while $\| \nabla_X L(X_t) \| > \varepsilon$ do
    $\psi_t(S) \equiv \nabla_X L(X_t)^T S + \frac{1}{2} S^T \nabla_X^2 L(X_t) S$
    $S_t = \arg \min_{\|S\| \le \Delta_t} \psi_t(S)$
    $\rho_t = \frac{L(X_t + S_t) - L(X_t)}{\psi_t(S_t)}$
    if $\rho_t > \tau$, $X_{t+1} \leftarrow X_t + S_t$
    else $X_{t+1} \leftarrow X_t$
    end if
    if $\rho_t \le \tau$, $\Delta_{t+1} \leftarrow \gamma_1 \| S_t \|$
    else if $\rho_t > \eta$ and $\| S_t \| = \Delta_t$, $\Delta_{t+1} \leftarrow \gamma_2 \Delta_t$
    else $\Delta_{t+1} \leftarrow \Delta_t$
    end if
    $t \leftarrow t + 1$
end do
```

Figure 3. Pseudo code for the trust-region method adopted in TRALM

The pseudo code for the trust-region method adopted in TRALM is shown Figure 3, where Δ_0 is set according to [22] and X_0 is the X^k solved in the previous TRALM iteration. When solving the OPF with discrete prices using TRALM, the difficulty resulting from the abrupt change of derivatives, as mentioned in Section 2, is successfully mitigated by the automatic trust region sizing procedure. Large disruptive trial steps crossing breakpoints on price curves would result in small ρ and hence get rejected and trigger the reduction of the trust-region size Δ.

Coleman et al proposed a two-dimensional trust-region method (which is now adopted in the MATLAB optimization toolbox) for solving large-scale optimization problems [21]-[22]. In their method, the trust region formed by $||S|| \le \Delta$ in Figure 3 is replaced with a two-dimensional region that spans the gradient direction and the direction generated by the modified PCG or Cholesky procedure. Our experiments show that, however, neither the PCG nor the Cholesky variation of this 2-D trust-region method is capable of solving large-scale OPF's.

To solve the sub-problem in Figure 3, we use the algorithm documented in [20], which is essentially a Newton's procedure applied to solve for α in:

$$\begin{aligned} \phi(\alpha) &\equiv \frac{1}{\Delta_t} - \frac{1}{\| S(\alpha) \|} = 0 \\ S(\alpha) &\equiv -(\nabla_X^2 L(X_t) + \alpha I)^{-1} \nabla_X L(X_t) \end{aligned} \tag{11}$$

where $\alpha \ge 0$ and $\nabla_X^2 L(X_t) + \alpha I$ is positive definite.

4. Step-controlled primal-dual interior point method

The primal-dual interior point method (PDIPM) and its many variations have become the algorithms of choice for solving OPF's over the past decade [7]-[13]. Given an optimization problem in the form of (7), PDIPM formulates the Lagrangian with barrier functions as:

$$\begin{aligned} L^\gamma(X,Z,\lambda,\mu) &\equiv f(X) + \lambda^T H(X) \\ &+ \mu^T (G(X)+Z) - \gamma \sum_{j=1}^{ni} \ln(Z_j) \end{aligned} \tag{12}$$

and uses the Newton's method to solve its first-order KKT conditions:

$$\begin{aligned} \nabla_X L^\gamma(X,Z,\lambda,\mu) &= 0 \\ H(X) &= 0 \\ G(X) + Z &= 0 \\ [\mu] Z - \gamma e &= 0 \end{aligned} \tag{13}$$

where Z, μ, and γ are strictly positive. Each Newton step involves variable substitutions and the solution of a reduced system of (13):

$$\begin{aligned} \Delta Z &= -G(X) - Z - \nabla G(X)^T \Delta X \\ \Delta \mu &= -\mu + [Z]^{-1} (\gamma e - [\mu] \Delta Z) \\ &\begin{bmatrix} M & \nabla H(X) \\ \nabla H(X)^T & 0 \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} -N \\ -H(X) \end{bmatrix} \\ M &\equiv \nabla_{XX}^2 L^\gamma(X,Z,\lambda,\mu) \\ &\quad + \nabla G(X)[\mu][Z]^{-1} \nabla G(X)^T \\ N &\equiv \nabla_X L^\gamma(X,Z,\lambda,\mu) \\ &\quad + \nabla G(X)[Z]^{-1}([\mu]G(X) + \gamma e) \end{aligned} \tag{14}$$

The variables (including γ) are updated according to:

$$\begin{aligned} \alpha_p &= \min(\min_{\Delta Z_j \prec 0} (Z_j / \Delta Z_j), 1) \\ \alpha_d &= \min(\min_{\Delta \mu_j \prec 0} (\mu_j / \Delta \mu_j), 1) \\ X &\leftarrow X + \xi \alpha_p \Delta X \\ Z &\leftarrow Z + \xi \alpha_p \Delta Z \\ \lambda &\leftarrow \lambda + \xi \alpha_d \Delta \lambda \\ \mu &\leftarrow \mu + \xi \alpha_d \Delta \mu \\ \gamma &\leftarrow \sigma \frac{\mu^T Z}{ni} \end{aligned} \tag{15}$$

where ξ and σ are constants that are typically set to 0.99995 and 0.1 respectively in experiments.

Although the PDIPM algorithm fit nicely with the traditional OPF that uses smooth polynomial cost curves, we cannot count on it to solve the market-based OPF in the form of (1) and (3) as demonstrated in the 118-bus OPF example shown in Figure 4. When

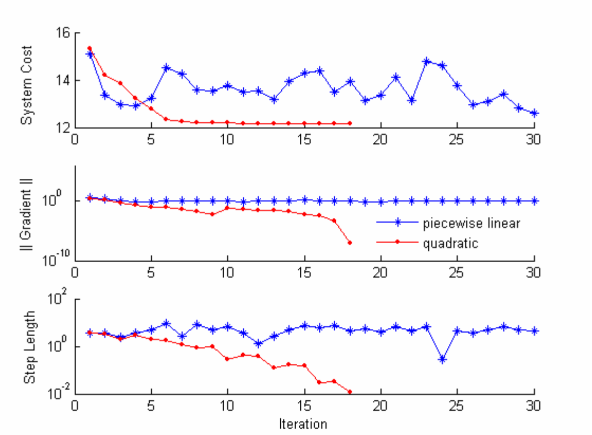

Figure 4. The progressions of PDIPM iterations in solving the 118-bus OPF with quadratic costs and with 3-block piecewise linear costs

```
Let $0 < \kappa < 1, 0 < \eta << 1, 0 < \varepsilon << 1,$
    $X_0, Z_0, \lambda_0, \mu_0$, and $\gamma_0$ be given,
$L_t \equiv L^{\gamma_t}(X_t, Z_t, \lambda_t, \mu_t)$
$\psi_t(\Delta X_t) \equiv (\nabla_X L_t)^T \Delta X_t + \frac{1}{2}(\Delta X_t)^T(\nabla_X^2 L_t)\Delta X_t$
$\rho_t(\Delta X_t) \equiv \frac{L^{\gamma_t}(X_t + \Delta X_t, Z_t, \lambda_t, \mu_t) - L_t}{\psi_t(\Delta X_t)}$
$fc_t \equiv \frac{\max(\max_{1 \le j \le ni}[G_j(X_t)], \| H(X_t) \|_\infty)}{1 + \max(\| \lambda_t \|_\infty, \| \mu_t \|_\infty)}$
$gc_t \equiv \frac{\| \nabla_X L_t \|_\infty}{1 + \max(\| \lambda_t \|_\infty, \| \mu_t \|_\infty)}$
$cc_t \equiv \frac{\mu_t^T Z_t}{1 + \| X_t \|_\infty}$
$oc_t \equiv \frac{\| f(X_t) - f(X_{t-1}) \|}{1 + \| f(X_{t-1}) \|}$
$scipm \leftarrow false$
while $\forall cond \in (fc_t, gc_t, cc_t, oc_t) > \varepsilon$ do
  compute $(\Delta X_t, \Delta Z_t, \Delta\lambda_t, \Delta\mu_t)$ according to (14)
  if $scipm = true$
    while $\rho_t(\Delta X_t) < 1 - \eta$ or $\rho_t(\Delta X_t) > 1 + \eta$ do
      $\Delta X_t \leftarrow \kappa \Delta X_t$
    end do
  end if
  compute $(X_{t+1}, Z_{t+1}, \lambda_{t+1}, \mu_{t+1}, \gamma_{t+1})$ from
    $(X_t, Z_t, \lambda_t, \mu_t, \gamma_t)$ and
    $(\Delta X_t, \Delta Z_t, \Delta\lambda_t, \Delta\mu_t)$ according to (15)
  if $fcond_{t+1} \ge fcond_t$ and $gcond_{t+1} \ge gcond_t$
    $scipm \leftarrow true$
  end if
  $t \leftarrow t + 1$
end do
```

Figure 5. Pseudo code for the step-controlled primal-dual interior point method

dealing with piecewise linear cost curves, the gradient and Hessian variables used in (13) and (14) change drastically from iteration to iteration due to abrupt price changes at the breakpoints. This behavior causes a loss of the strong descending property of Newton steps.

The SCIPM algorithm shown in Figure 5 overcomes this difficulty by monitoring the accuracy of the quadratic approximation of the Lagrangian during the OPF computation and shortening the step length if any sudden change of derivative coupled with a large PDIPM step results in an inaccurate approximation. Empirically, it is more efficient to start applying such step control procedure after the normal PDIPM step fails to improve the gradient condition or the feasibility condition. Although Figure 5 uses PDIPM as the baseline algorithm, we shall point out that the same step control concept applies to other interior point methods as well. Figure 6 shows the positive result of applying SCIMP to solve the same 118-bus market-based OPF as used in Figure 4. With step adjustments, SCIPM is able to reduce the system cost and the gradients continuously and eventually converges to the optimal solution.

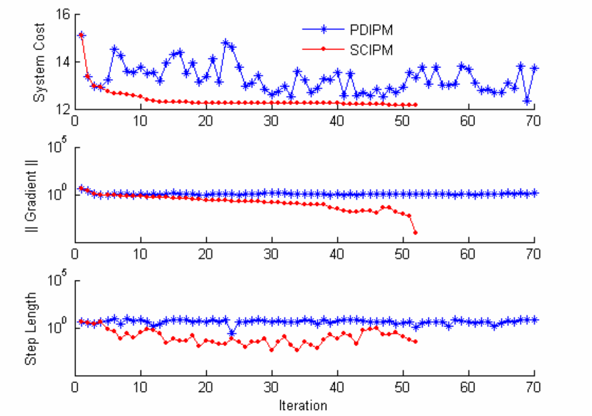

Figure 6. A comparison of PDIPM and SCIPM in solving the 118-bus OPF with 3-block piecewise linear costs

Alternatively, the difficulty with discontinuous price curves can be addressed by changing the OPF formulation to use a separate control variable for each power block in (1). However, in this case, both memory footprint and computation time will increase along with the number of variables.

5. Numerical results

We tested our OPF algorithms using several power system models whose characteristics are summarized in Table 1. The tests were run on a PC with Intel 3.3GHz P4 processor (2MB L2 cache), 2GB memory, and Linux 2.6.9 kernel. All optimization programs and the underlying linear algebra functions, except the LU and Cholesky factorization modules, were developed in house and compiled using the GCC 3.4.4 compiler. LU and Cholesky factorizations were done using the UFsparse package [25]. The parameters used in TRALM, SCIPM, and the cost-curve smoothing, unless stated otherwise, were set according to Table 2. All experiments in this paper use flat starting points, i.e. unit voltages, zero phase angles, and generator outputs at the midpoints between maximum generations and minimum generations. Offer prices were randomly generated and ranged from \$50/MWh to \$100/MWh. The unit of cost was set to \$10,000 in (1).

Table 1. Summary of power system models used in the study

Buses	Generators	Branches	Total Load (MW)
30	6	41	189
57	7	80	1,250
118	54	186	4,242
300	69	411	23,525
2383	327	2896	24,558
2935	956	7028	394,794

Table 2. Values of parameters used in TRALM, SCIPM, and trigonometric smoothing of piecewise linear cost curves

<table>
<tr><td colspan="8">NB = 3, α = 0.04</td></tr>
<tr><td colspan="4">TRALM</td><td colspan="4">SCIPM</td></tr>
<tr><td>ε_λ</td><td>5e-3</td><td>τ</td><td>0.25</td><td>κ</td><td>0.5</td><td>Z_0</td><td>1.0</td></tr>
<tr><td>ε_μ</td><td>1e-1</td><td>η</td><td>0.75</td><td>η</td><td>0.1</td><td>λ_0</td><td>0.0</td></tr>
<tr><td>ε^0</td><td>2e0</td><td>γ_1</td><td>0.1</td><td>ε</td><td>1e-4</td><td>μ_0</td><td>1.0</td></tr>
<tr><td>ε^∞</td><td>1e-2</td><td>γ_2</td><td>2.0</td><td>X_0</td><td>flat</td><td>γ_0</td><td>1.0</td></tr>
<tr><td colspan="2">$\beta_{W,U}$ = 3</td><td colspan="2">$\gamma_{W,U}$ = 0.33</td><td colspan="2">$\xi = 0.99995$</td><td>σ</td><td>0.1</td></tr>
</table>

At the time of writing, we are not able to find any production-quality OPF tool to compare our new algorithms with. (In fact, the lack of robust commercial AC OPF tool has been a major hurdle for the market to institute advanced options of trading ancillary services such as reactive power [28].) Instead, the TRALM and SCIPM algorithms are compared with our own implementation of PDIPM and the MINOS used in MATPOWER [26]-[27] below, with regards to convergence, performance, accuracy, and scalability.

5.1. Convergence and performance

Table 3 lists the execution times of solving OPF's of six different sizes using the four algorithms. TRALM and SCIPM converged in all cases, while MINOS failed to solve large-scale OPF's and PDIPM failed to solve some market-based OPF's with piecewise linear cost curves. Even when PDIPM did succeed, it usually took much longer to converge. In terms of performance, SCIPM is better than TRALM in execution time and therefore more suitable for real-time applications.

Table 3. A comparison of execution times (sec.) of four OPF algorithms

<table>
<tr><td colspan="5">Solving OPF with quadratic costs</td></tr>
<tr><td>System</td><td>MINOS</td><td>PDIPM</td><td>TRALM</td><td>SCIPM</td></tr>
<tr><td>30-bus</td><td>0.06</td><td>0.05</td><td>0.20</td><td>0.06</td></tr>
<tr><td>57-bus</td><td>0.09</td><td>0.08</td><td>0.40</td><td>0.10</td></tr>
<tr><td>118-bus</td><td>1.2</td><td>0.3</td><td>2.3</td><td>0.4</td></tr>
<tr><td>300-bus</td><td>6.3</td><td>0.8</td><td>4.8</td><td>1.0</td></tr>
<tr><td>2383-bus</td><td>FAIL</td><td>12</td><td>168</td><td>14</td></tr>
<tr><td>2935-bus</td><td>FAIL</td><td>20</td><td>680</td><td>23</td></tr>
<tr><td colspan="5">Solving OPF with piecewise linear costs</td></tr>
<tr><td>System</td><td>MINOS</td><td>PDIPM</td><td>TRALM</td><td>SCIPM</td></tr>
<tr><td>30-bus</td><td>0.06</td><td>0.44</td><td>0.34</td><td>0.16</td></tr>
<tr><td>57-bus</td><td>0.09</td><td>3.70</td><td>0.40</td><td>0.20</td></tr>
<tr><td>118-bus</td><td>0.9</td><td>10.4</td><td>3.7</td><td>1.3</td></tr>
<tr><td>300-bus</td><td>3.8</td><td>FAIL</td><td>6.7</td><td>3.4</td></tr>
<tr><td>2383-bus</td><td>FAIL</td><td>51</td><td>202</td><td>32</td></tr>
<tr><td>2935-bus</td><td>FAIL</td><td>FAIL</td><td>1011</td><td>140</td></tr>
</table>

One theoretical pitfall of the SCIPM algorithm, like that of the PDIPM algorithm, is that it does not guarantee global convergence. In [12], the authors published some non-converging results of solving OPF's using PDIPM and proposed an algorithm that attempted to improve the convergence at the cost of performance through adjustments of the Hessian matrices. Our preliminary investigation showed that the problem formulation in [12], which treats all constraints as inequality constraints, does bring numerical difficulties to PDIPM when solving large-scale OPF's. We, however, were not able to regenerate the non-converging results in our experiments using the formulation in (1). SCIPM, as well as IPM in the context of solving OPF's with quadratic costs, consistently converges to the desired OPF solutions. By checking the second-order KKT condition, we were able to verify the convergence of IPM and SCIPM for all OPF cases whose load levels range from 50% of the base load to the maximum levels listed in Table 4.

These encouraging experimental results may imply that the region of attraction around the stationary point for our particular nonlinear system is large enough to counter occasional ill-defined Newton steps.

Table 4. Maximum load levels (relative to base loads) that each algorithm can accommodate, assuming uniform load scaling

System	MINOS	PDIPM	TRALM	SCIPM
30-bus	1.07	1.07	1.07	1.07
57-bus	1.54	1.54	1.54	1.54
118-bus	2.29	2.29	2.29	2.29
300-bus	1.29	1.29	1.29	1.29
2383-bus	N/A	1.02	1.02	1.02
2935-bus	N/A	1.23	1.23	1.23

5.2. Accuracy

Table 5 lists the result of a cross examination of the OPF solutions generated by SCIPM, TRALM, and MINOS. As defined in Section 2, δ_C, δ_X, δ_λ, and δ_μ measure the deviation of a trial OPF solution from the reference one. The small values reported in Table 5 indicate that SCIPM and TRALM are valid methods for computing large-scale OPF's.

Table 5. Deviations of OPF solutions computed by different algorithms

	δ_C	δ_X	δ_λ	δ_μ
300-quad-SM*	1.7e-7	3.0e-6	1.2e-6	5.4e-7
300-pwln-SM	5.2e-4	6.4e-5	3.3e-4	9.8e-8
300-quad-TM	1.2e-6	2.1e-5	3.3e-6	0
300-pwln-TM	5.2e-4	5.9e-5	3.3e-4	0
2935-quad-ST	1.1e-6	3.0e-6	1.8e-6	5.4e-9
2935-pwln-ST	1.9e-8	6.6e-6	1.1e-5	2.8e-8

* Interpretation of label N-T-AB: N – power system model; T – cost curve type (quadratic or piecewise linear); A – trial algorithm (S for SCIPM and T for TRALM); B – reference algorithm (M for MINOS and T for TRALM).

The parameter α used in (4) has an impact on the accuracy of the OPF solution. As shown in Table 6, smaller α's yield more accurate solutions. In practice, 0.04 should be small enough to ensure satisfactory results.

Table 6. Accuracies of several OPF solutions computed by SCIPM with different α values (2935-bus system, reference α is 0.01)

α	δ_C	δ_X	δ_λ	δ_μ
0.2	1.6e-3	5.9e-4	6.2e-3	1.3e-8
0.1	7.4e-4	5.1e-5	3.4e-3	1.6e-8
0.08	5.6e-4	8.6e-5	3.6e-3	2.3e-8
0.06	3.8e-4	5.3e-5	3.3e-3	1.1e-8
0.04	2.0e-4	1.0e-4	8.8e-6	2.6e-8
0.02	4.2e-5	3.4e-5	1.5e-5	9.2e-9

5.3. Scalability

The execution time of solving an OPF depends both on the number of iterations taken and on the computational complexity of one single iteration. Assuming a constant transmission network density and a constant fill-in ratio for sparse matrix factorization, the complexities of one iteration of TRALM and one iteration of SCIPM are both $O(N_{bus})$, although their underlying coefficients are quite different.

Table 7. Numbers of iterations taken to solve OPF's of different sizes

Solving OPF with quadratic costs				
System	MINOS	PDIPM	TRALM	SCIPM
30-bus	350	12	134	12
57-bus	179	10	146	10
118-bus	1579	17	441	17
300-bus	3654	19	420	19
2383-bus	FAIL	33	1834	33
2935-bus	FAIL	30	2842	30
Solving OPF with piecewise linear costs				
System	MINOS	PDIPM	TRALM	SCIPM
30-bus	163	114	221	30
57-bus	184	496	171	20
118-bus	1190	606	698	52
300-bus	2202	FAIL	544	62
2383-bus	FAIL	157	2193	72
2935-bus	FAIL	FAIL	4310	177

Table 7 shows the numbers of iterations taken by each algorithm to solve the OPF's of different sizes. Although the exact relationship between the system size and the number of iterations is unclear, we can identify some general trends. First of all, the number of iterations generally rises as the system size increases.

Secondly, the pace of increase in iteration numbers for SCIPM is much slower than that of MINOS and TRALM and implies that SCIPM is more scalable and therefore suits large-scale systems better. Thirdly, for SCIPM, the number of iterations required to solve the 2935-bus market-based OPF is abnormally high, indicating that other system details (such as number of generators) also have a impact on the algorithm complexity. Neglecting those system-specific characteristics, the overall OPF complexity can be approximated by $O(N_{bus}^{1+\varepsilon})$, where ε is a small number that falls between 0.1 and 0.2 in the case of SCIPM.

Table 8. Numbers of iterations and execution times taken by SCIPM to solve the 2935-bus OPF with different *NB's*

NB	Its.	Time (sec.)	NB	Its.	Time (sec.)
1	32	21	6	219	174
2	157	94	7	244	194
3	177	140	8	259	208
4	196	155	9	289	231
5	198	158	10	282	226

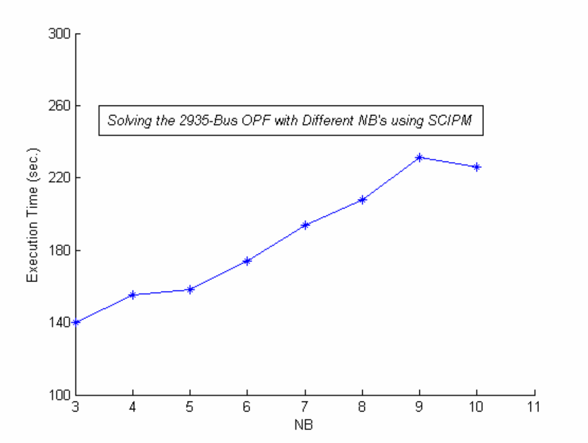

Figure 7. Illustration of the near-linear relationship between execution time and *NB* in the example of solving the 2935-bus OPF using SCIPM

The number of blocks contained in each discrete offer or bid curve also has an impact on the performance of second-order NLP-based OPF algorithms. Table 8 lists the computation time and number of iterations taken by SCIPM to solve the 2935-bus OPF with different *NB* values. Notice that computing an OPF solution becomes much simpler when *NB* is set to 1, which implies uniformly priced bids and offers. Therefore, when comparing the effectiveness of different algorithms in the context of market-based OPF's, we need to rule out this special case. Figure 7 shows the near-linear relationship between *NB* and the execution time. When designing market rules, the ISO should balance its customer's desire for flexibility and the real-time computational requirements to decide the maximum *NB* that can be allowed in the biddings.

6. Conclusions

In this paper, we discussed new computational challenges posed by the deregulated electricity market and proposed two new OPF algorithms, TRALM and SCIPM, to address them. Numerical studies showed that, for our OPF formulation, both algorithms are reliable and better than some existing algorithms in solving large-scale OPF's with discrete price curves. SCIPM is particularly good for real-time applications due to its efficiency. We argued that future OPF research should give more emphasis to algorithm robustness and accuracy, in light of the changing roles of OPF's in market-oriented applications. We plan to discuss alternative OPF formulations and how to integrate the new algorithms into the security-constrained OPF in separate papers in the near future.

7. Acknowledgement

This work was supported in part by the National Science Foundation under Award 0532744. The authors would like to thank R. D. Zimmerman for his constructive feedbacks.

8. References

[1] J. Carpentier, “Contribution a l’´etude du dispatching economique,” Bulletin de la Soci´et´e Fran¸caise des Electriciena, vol. 3, pp. 431–447, Aug. 1962.

[2] O. Alsac and B. Stott, “Optimal Load Flow with Steady-State Security,” *IEEE Trans. Power Apparatus and Systems*, vol. 93, pp. 745-751, May/June 1974.

[3] B. Stott, O. Alsac, and A. J. Monticelli, “Security Analysis and Optimization,” in *Proc. of the IEEE, Special Issue on Computers in Power System Operations*, vol. 75, pp. 1623-1644, Dec. 1987.

[4] O. Alsac, J. Bright, M. Prais, and B. Stott, “Further Developments in LP-Based Optimal Power Flow,” *IEEE Trans. Power Systems*, vol. 5, pp. 697-711, Aug. 1990.

[5] J. Carpentier and P. Bornard, “Towards an Integrated Secure Optimal Operation of Power Systems,” in *Proc. 1991 IEE Int. Conf. on Advances in Power System Control, Operation and Management*, vol. 1, pp. 1-16.

[6] M. Huneault and F. D. Galiana, “A Survey of the Optimal Power Flow Literature,” *IEEE Trans. Power Systems*, vol. 6, pp. 762-770, May 1991.

[7] S. Granville, “Optimal Reactive Dispatch through Interior Point Method,” *IEEE Trans. Power Systems*, vol. 9, pp. 136-146, Feb. 1994.

[8] Y. Wu, A. S. Debs, and R. E. Marsten, “A Direct Nonlinear Predictor-Corrector Primal-Dual Interior Point Algorithm for Optimal Power Flows,” *IEEE Trans. Power Systems*, vol. 9, pp. 876-882, May 1994.

[9] G. L. Torres and V. H. Quintana, “Optimal Power Flow by a Nonlinear Complementarity Method,” *IEEE Trans. Power Systems*, vol. 15, pp. 1028-1033, Aug. 2000.

[10] G. L. Torres and V. H. Quintana, “On a Nonlinear Multiple-Centrality-Corrections Interior-Point Method for Optimal Power Flow,” *IEEE Trans. Power Systems*, vol. 16, pp. 222-228, May 2000.

[11] E. D. Castronuovo, J. M. Campagnolo, and Roberto Salgoda, “On the Application of High Performance Computation Techniques to Nonlinear Interior Point Methods,” *IEEE Trans. Power Systems*, vol. 16, pp. 325-331, Aug. 2001.

[12] R. A. Jabr, A. H. Coonick, and B. J. Cory, “A Primal-Dual Interior Point Method for Optimal Power Flow Dispatching,” *IEEE Trans. Power Systems*, vol. 17, pp. 654-662, Aug. 2001.

[13] W. Qiu, A. J. Flueck, and F. Tu, “A New Parallel Algorithm for Security Constrained Optimal Power Flow with a Nonlinear Interior Point Method,” in *Proc. 2005 IEEE Power Engineering Society General Meeting*, pp. 2422-2428.

[14] *PJM Manual 11: Scheduling Operations*. [Online]. Available: http://www.pjm.com/contributions/pjm-manuals/manuals.html

[15] *PJM Manual 06: Financial Transmission Rights*. [Online]. Available: http://www.pjm.com/contributions/pjm-manuals/manuals.html

[16] *PJM Manual 12: Dispatching Operations*. [Online]. Available: http://www.pjm.com/contributions/pjm-manuals/manuals.html

[17] C. E. Murillo-Sanchez, “On the Integration of Unit Commitment and Optimal Power Flow,” Ph.D. dissertation, School of Electrical and Computer Engineering, Cornell University, 2000

[18] J. Chen, J. S. Thorp, and T. D. Mount, “Coordinated Interchange Scheduling and Opportunity Cost Payment: A Market Proposal to Seams Issues,” in *Proc. 2004 37th Hawaii Int. Conf. on System Sciences*, pp. 51-60.

[19] D. Gay, “Computing Optimal Locally Constrained Steps,” *SIAM J. Sci. Stat. Computing*, vol. 2, pp. 186-197, June 1981.

[20] J. J. More and D. C. Sorensen, “Computing a Trust Region Step,” *SIAM J. Sci. Stat. Computing*, vol. 4, pp. 553-572, Sept. 1983.

[21] T. F. Coleman and Y. Li, “An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds,” *SIAM J. Optimization*, vol. 6, pp. 418-445, May 1996.

[22] M. A. Branch, T. F. Coleman, and Y. Li, “A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems,” *SIAM J. Sci. Computing*, vol. 21, pp. 1-23, Aug. 1999.

[23] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999, pp. 95-97.

[24] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999, pp. 397-426.

[25] T. A. Davis, *UFsparse*, ver. 2.0.0.beta, [Online]. Available: http://www.cise.ufl.edu/research/sparse/UFsparse

[26] R. D. Zimmerman, C. E. Murillo-Sánchez, and D. Gan, “MATPOWER 3.0.0 User’s Manual,” [Online]. Available: http://www.pserc.cornell.edu/matpower

[27] B. D. Murtagh, M. A. Saunders, “MINOS 5.5 User’s Guide,” *Stanford University Systems Optimization Laboratory Technical Report SOL83-20R*.

[28] Federal Energy Regulatory Commission, “Principles for Efficient and Reliable Reactive Power Supply and Consumption”*,* *FERC Staff Reports*, Docket No. AD05-1-000, pp. 161-162, Feb. 2005. [Online]. Available: http://www.ferc.gov/legal/staff-reports.asp