基于原–对偶内点算法的 WLAV 状态估计

郭　伟　单渊达
（东南大学电气工程系　210096　南京）

摘　要　针对现有 WLAV 状态估计算法存在的问题，将原–对偶内点算法引入 WLAV 状态估计问题的求解过程。结合 WLAV 状态估计问题的特点，提出了罚参数修正及迭代限制方法，并利用稀疏矩阵技术进行计算。算例分析表明，该算法的数值稳定性好、迭代次数少、计算速度较快，性能优于传统方法。

关键词　WLAV 状态估计　线性规划　内点法

分类号　TM 732

0　引言

电力系统状态估计器是整个 EMS 的基础，自 Schweppe 将状态估计引入电力系统后，占主导地位的方法一直是加权最小二乘法（WLS），文献[1,2]在此基础上又有了大量的改进。与此同时，人们也在探求其它方法，目的是寻找对不良数据具有自动排除能力的方法。其中，加权最小绝对值（WLAV）状态估计在近年来研究较多。WLAV 解与 WLS 解有着本质的不同，对于一个具有 n 个状态量、m 个量测量（$m > n$）的系统，WLAV 解使得 m 个量测中的 n 个量测残差为零，剩余的 $m - n$ 个量测的残差不为零，而 WLS 解使得 m 个量测上的加权残差平方和达到最小。

WLAV 方法最早是由 Irving 及 Owen 引入电力系统状态估计，他们将 WLAV 转化为线性规划问题（LP）进行求解。Kotiuga 和 Vidyasagar 认为，WLAV 估计的本质是量测集的插值（interpolating），故 WLAV 估计器具有一定的排除粗差能力[3]。作为线性规划问题，人们运用了不同的方法[4,5]求解 WLAV 状态估计，其中文献[4]基于 Barrodale–Roberts 法，是一种较有效的方法；针对 WLAV 方法受“杠杆点”（leverage point）影响的问题，Abur 等又提出了对量测方程进行变换的方法[6]。上述各种方法收敛较慢，且编程也较繁琐。因此，尽管 WLAV 状态估计具有鲁棒性，其应用仍然受到限制。

自从 Karmarkar 于 1984 年提出具有多项式时间可解性的线性规划内点法以来，出现了大量的内点算法，其中两类主要的方法是仿射尺度法和路径跟踪法（或称原–对偶方法）[7,8]。大量的数值试验表明，路径跟踪法比仿射尺度法具有更好的数值稳定性及收敛速度。1994 年，Singh 及 Alvarado 将内点算法用于电力系统状态估计[9]，用壁垒函数法求解原问题，用仿射尺度法求解对偶问题，但文中未给出估计结果，仅从迭代次数上比较了两种方法，认为后一方法优于前者。文中方法实质上都是仿射尺度法，而且，对偶仿射尺度法的最大缺点是不能给出原变量的好的估计[8]。

针对上述方法存在的问题，本文将原–对偶算法引入 WLAV 状态估计，并结合 WLAV 状态估计问题的特点，提出了罚参数修正及迭代限制方法，同时利用稀疏矩阵方法中的符号运算技术进行计算，在保持原–对偶算法的优良特性的基础上，减少了迭代次数，加快了计算速度。

1　模型描述

电力系统状态估计问题的量测方程为：

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v} \tag{1}$$

其中　$\boldsymbol{z}$ 是 $m \times 1$ 量测向量；$\boldsymbol{h}(\boldsymbol{x})$ 是 $m \times 1$ 非线性量测函数向量；$\boldsymbol{v}$ 是 $m \times 1$ 量测误差向量；$\boldsymbol{x}$ 为 $n \times 1$ 状态向量；m，n 分别是量测量及状态量的个数。

对方程(1)在某运行点 $\boldsymbol{x}^0$ 进行一阶近似，得出线性化的量测方程：

$$\Delta \boldsymbol{z} = \boldsymbol{H}(\boldsymbol{x}^0) \cdot \Delta \boldsymbol{x} + \boldsymbol{v} \tag{2}$$

其中　$\Delta \boldsymbol{z} = \boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}^0)$；$\Delta \boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}^0$；$\boldsymbol{H}(\boldsymbol{x}^0) = \dfrac{\partial \boldsymbol{h}(\boldsymbol{x}^0)}{\partial \boldsymbol{x}}$，为 $m \times n$ 量测雅可比矩阵。

WLAV 状态估计模型为：

$$\min \sum_{j=1}^{m} w_j |v_j| \tag{3}$$

1998-05-06 收稿，1998-06-16 改回。

其中 w_j 是权重向量 $\boldsymbol{w}(m\times 1)$ 的第 j 个分量；$\boldsymbol{v}=\Delta\boldsymbol{z}-\boldsymbol{H}(\boldsymbol{x}^0)\cdot\Delta\boldsymbol{x}$，是量测残差向量。

由以上分析，WLAV 状态估计可以描述成如下标准型线性规划原问题：

$$\min \boldsymbol{c}^{\mathrm{T}}\cdot[\Delta\boldsymbol{x}_\eta^{\mathrm{T}},\ \Delta\boldsymbol{x}_\rho^{\mathrm{T}},\ \boldsymbol{\eta}^{\mathrm{T}},\ \boldsymbol{\rho}^{\mathrm{T}}]^{\mathrm{T}} \tag{4}$$

$$\text{s.t.}\begin{cases}[\boldsymbol{H},\ -\boldsymbol{H},\ \boldsymbol{I},\ -\boldsymbol{I}]\begin{bmatrix}\Delta\boldsymbol{x}_\eta\\ \Delta\boldsymbol{x}_\rho\\ \boldsymbol{\eta}\\ \boldsymbol{\rho}\end{bmatrix}=\Delta\boldsymbol{z} & (5)\\ \Delta\boldsymbol{x}_\eta,\Delta\boldsymbol{x}_\rho,\boldsymbol{\eta},\boldsymbol{\rho}\geqslant\boldsymbol{0} & (6)\end{cases}$$

其中 $\boldsymbol{c}^{\mathrm{T}}=[\boldsymbol{0},\ \boldsymbol{0},\ \boldsymbol{w}^{\mathrm{T}},\ \boldsymbol{w}^{\mathrm{T}}]$；$\Delta\boldsymbol{x}_\eta,\Delta\boldsymbol{x}_\rho$ 为 $n\times 1$ 向量；$\boldsymbol{\eta}$ 与 $\boldsymbol{\rho}$ 为 $m\times 1$ 向量。

为了满足式(6)的非负性要求及式(2)的物理意义，应有 $\Delta\boldsymbol{x}_\eta-\Delta\boldsymbol{x}_\rho=\Delta\boldsymbol{x}$，$\boldsymbol{\eta}-\boldsymbol{\rho}=\boldsymbol{v}$，且 $\Delta x_\eta(i)$ 与 $\Delta x_\rho(i)$ 至少有一个为零，$\eta(i)$ 与 $\rho(i)$ 也至少有一个为零。

若记 $\boldsymbol{A}=[\boldsymbol{H},\ -\boldsymbol{H},\ \boldsymbol{I},\ -\boldsymbol{I}]$，$\boldsymbol{b}=\Delta\boldsymbol{z}$，$\boldsymbol{x}^{\mathrm{T}}=[\Delta\boldsymbol{x}_\eta^{\mathrm{T}},\ \Delta\boldsymbol{x}_\rho^{\mathrm{T}},\ \boldsymbol{\eta}^{\mathrm{T}},\ \boldsymbol{\rho}^{\mathrm{T}}]$，则式(4)～式(6)构成了标准的原线性规划问题。据此，不难得出对偶线性规划问题：

$$\max\ \Delta\boldsymbol{z}^{\mathrm{T}}\cdot\boldsymbol{\omega}$$

$$\text{s.t.}\quad \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\omega}+\boldsymbol{s}=\boldsymbol{c}$$

其中 $\boldsymbol{\omega}$ 为对偶向量；$\boldsymbol{s}\geqslant\boldsymbol{0}$，为松弛向量。

原–对偶算法基于对数障碍函数方法，能在保持解的原可行性和对偶可行性的基础上，沿一条原–对偶路径寻找到最优解。对上述 WLAV 状态估计原–对偶问题，为消除原问题中 $\boldsymbol{x}$ 的不等约束，应用对数壁垒函数技术，有：

$$\min\ \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}-\mu\sum_{j=1}^{2n+2m}\ln x_j \tag{7}$$

$$\text{s.t.}\begin{cases}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}\\ \boldsymbol{x}>\boldsymbol{0}\end{cases}$$

其中 μ 是壁垒参数(或罚参数)，为标量。

注意式(7)中有 $\sum_{j=1}^{2n+2m}\ln x_j$ 及 $\boldsymbol{x}>\boldsymbol{0}$，似与上面所述 $\boldsymbol{x}$ 中存在零元素相矛盾，在第 2 节中将说明，实际实现中，$\boldsymbol{x}$ 不为零，但不影响最终结果。由 Karush–Kuhn–Tucker 条件，有：

$$\begin{cases}\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b},\ \boldsymbol{x}>\boldsymbol{0} & \text{原可行性}\\ \boldsymbol{A}^{\mathrm{T}}\boldsymbol{\omega}+\boldsymbol{s}=\boldsymbol{c},\ \boldsymbol{s}>\boldsymbol{0} & \text{对偶可行性}\\ \boldsymbol{X}\boldsymbol{S}\boldsymbol{e}-\mu\boldsymbol{e}=\boldsymbol{0} & \text{补偿松弛}\end{cases} \tag{8}$$

其中 $\boldsymbol{X}$ 和 $\boldsymbol{S}$ 分别是以向量 $\boldsymbol{x}$ 和 $\boldsymbol{s}$ 的分量为对角元的对角阵；$\boldsymbol{e}$ 是全 1 向量。

运用牛顿法可以获得如下的转移方向：

$$\boldsymbol{A}\,\Delta\boldsymbol{x}=-(\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}) \tag{9}$$

$$\boldsymbol{A}^{\mathrm{T}}\,\Delta\boldsymbol{\omega}+\Delta\boldsymbol{s}=-(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\omega}+\boldsymbol{s}-\boldsymbol{c}) \tag{10}$$

$$\boldsymbol{S}\,\Delta\boldsymbol{x}+\boldsymbol{X}\,\Delta\boldsymbol{s}=-(\boldsymbol{X}\boldsymbol{S}\boldsymbol{e}-\mu\boldsymbol{e}) \tag{11}$$

令式(9)～式(11)的右端分别为向量 $\boldsymbol{t},\boldsymbol{u},\boldsymbol{\lambda}$，有：

$$\begin{bmatrix}\boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{A}^{\mathrm{T}} & \boldsymbol{I}\\ \boldsymbol{S} & \boldsymbol{0} & \boldsymbol{X}\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{x}\\ \Delta\boldsymbol{\omega}\\ \Delta\boldsymbol{s}\end{bmatrix}=\begin{bmatrix}\boldsymbol{t}\\ \boldsymbol{u}\\ \boldsymbol{\lambda}\end{bmatrix} \tag{12}$$

获得转移方向后，即可移动到新点，更新解向量：

$$\boldsymbol{x}^{k+1}=\boldsymbol{x}^k+\beta_{\mathrm{P}}\,\Delta\boldsymbol{x}^k \tag{13}$$

$$\boldsymbol{\omega}^{k+1}=\boldsymbol{\omega}^k+\beta_{\mathrm{D}}\,\Delta\boldsymbol{\omega}^k \tag{14}$$

$$\boldsymbol{s}^{k+1}=\boldsymbol{s}^k+\beta_{\mathrm{D}}\,\Delta\boldsymbol{s}^k \tag{15}$$

其中 原步长 $\beta_{\mathrm{P}}=\dfrac{1}{\max\{1,\ -(\Delta x^k)_j/\alpha x_j^k\}}$；对偶步长 $\beta_{\mathrm{D}}=\dfrac{1}{\max\{1,\ -(\Delta s^k)_j/\alpha s_j^k\}}$；$\alpha<1$，本文取 0.999；上标 k 表示第 k 次迭代。

2 实现

原–对偶算法的主要计算在于转移方向的获得，即式(12)的求解。对式(12)，可以用两种解法。

第 1 种解法是由式(10)得：

$$\Delta\boldsymbol{s}=\boldsymbol{u}-\boldsymbol{A}^{\mathrm{T}}\,\Delta\boldsymbol{\omega}$$

代入式(11)，经整理有：

$$\boldsymbol{X}^{-1}\boldsymbol{S}\,\Delta\boldsymbol{x}-\boldsymbol{A}^{\mathrm{T}}\,\Delta\boldsymbol{\omega}=\boldsymbol{X}^{-1}\boldsymbol{\lambda}-\boldsymbol{u} \tag{16}$$

由式(16)及式(9)得：

$$\begin{bmatrix}\boldsymbol{X}^{-1}\boldsymbol{S} & -\boldsymbol{A}^{\mathrm{T}}\\ -\boldsymbol{A} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{x}\\ \Delta\boldsymbol{\omega}\end{bmatrix}=\begin{bmatrix}\boldsymbol{X}^{-1}\boldsymbol{\lambda}-\boldsymbol{u}\\ -\boldsymbol{t}\end{bmatrix} \tag{17}$$

注意到式(17)中左端第 1 项为对称阵，可用改进 Cholesky 进行因子化分解。而 $\boldsymbol{X}^{-1}\boldsymbol{S}$ 为对角阵，又使分解进一步简化，具体参见文献[10]。这种方法的缺点在于因子分解后左端第 1 项的右下角产生大量的注入。

第 2 种解法[8]是将式(10)两边同乘 $\boldsymbol{A}\boldsymbol{X}\boldsymbol{S}^{-1}$，得：

$$\boldsymbol{A}\boldsymbol{X}\boldsymbol{S}^{-1}\boldsymbol{A}^{\mathrm{T}}\,\Delta\boldsymbol{\omega}=\boldsymbol{A}\boldsymbol{X}\boldsymbol{S}^{-1}\boldsymbol{u}-\boldsymbol{A}\boldsymbol{X}\boldsymbol{S}^{-1}\,\Delta\boldsymbol{s} \tag{18}$$

记 $\boldsymbol{X}^{-1}\boldsymbol{\lambda}=\mu\boldsymbol{X}^{-1}\boldsymbol{e}-\boldsymbol{S}\boldsymbol{e}=\boldsymbol{p}$

将式(11)两边同乘 $\boldsymbol{A}\boldsymbol{X}\boldsymbol{S}^{-1}\boldsymbol{X}^{-1}$，并将式(9)代入，整理后有：

$$\boldsymbol{A}\boldsymbol{X}\boldsymbol{S}^{-1}\,\Delta\boldsymbol{s}=\boldsymbol{A}\boldsymbol{X}\boldsymbol{S}^{-1}\boldsymbol{p}-\boldsymbol{t} \tag{19}$$

将式(19)代入式(18)有：

$$(\boldsymbol{A}\boldsymbol{X}\boldsymbol{S}^{-1}\boldsymbol{A}^{\mathrm{T}})\,\Delta\boldsymbol{\omega}=\boldsymbol{A}\boldsymbol{X}\boldsymbol{S}^{-1}(\boldsymbol{u}-\boldsymbol{p})+\boldsymbol{t} \tag{20}$$

通过式(20)求得 $\Delta\boldsymbol{\omega}$ 后，相应有：

$$\Delta\boldsymbol{s}=\boldsymbol{u}-\boldsymbol{A}^{\mathrm{T}}\,\Delta\boldsymbol{\omega} \tag{21}$$

$$\Delta\boldsymbol{x}=\boldsymbol{X}\boldsymbol{S}^{-1}[\boldsymbol{p}-\Delta\boldsymbol{s}] \tag{22}$$

尽管 $\boldsymbol{A}=[\boldsymbol{H},\ -\boldsymbol{H},\ \boldsymbol{I},\ -\boldsymbol{I}]$ 是 $m\times(2n+2m)$ 矩阵，但在实际实现时只需存储 $m\times n$ 的 $\boldsymbol{H}$ 阵。同样，对 $(2n+2m)\times 1$ 的向量 $\boldsymbol{c}$，也只需存储 $m\times 1$

的向量 $\boldsymbol{w}$，这样就减少了对内存的需求。

对于式(20)的求解，除了利用稀疏矩阵方法中常用的模拟定序及符号分解技术外，还可根据左端中 $\boldsymbol{AXS}^{-1}\boldsymbol{A}^{\mathrm{T}}$ 的特点进行计算。若记 $\boldsymbol{XS}^{-1}=\mathrm{diag}[\boldsymbol{D}_1,\boldsymbol{D}_2,\boldsymbol{D}_3,\boldsymbol{D}_4]$，其中 $\boldsymbol{D}_1,\boldsymbol{D}_2,\boldsymbol{D}_3,\boldsymbol{D}_4$ 分别是 $n\times n,n\times n,m\times m,m\times m$ 的对角阵，则有：

$$\boldsymbol{AXS}^{-1}\boldsymbol{A}^{\mathrm{T}}=\boldsymbol{HD}_1\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{HD}_2\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{D}_3+\boldsymbol{D}_4 \quad (23)$$

从式(23)可以看出，$\boldsymbol{AXS}^{-1}\boldsymbol{A}^{\mathrm{T}}$ 的计算较为方便。因雅可比矩阵 $\boldsymbol{H}$ 是稀疏阵，且在每个 SE(state estimation) 迭代中保持恒定，本文利用稀疏矩阵技术，对 $\boldsymbol{AXS}^{-1}\boldsymbol{A}^{\mathrm{T}}$ 也进行了符号计算，这样连同求解式(20)时的模拟定序及符号分解技术，去除了每次数值运算中的查找，从而大大提高了算法的计算速度。

另外，用原-对偶算法求解 WLAV 状态估计问题时，对于 WLAV 状态估计模型中 $\Delta x_\eta(i)$ 与 $\Delta x_\rho(i)$ 至少有一个为零、$\eta(i)$ 与 $\rho(i)$ 也至少有一个为零的要求，实际上在每步迭代中无法满足，只能在迭代中按实际值进行计算。但是 $\Delta\boldsymbol{x}_\eta-\Delta\boldsymbol{x}_\rho=\Delta\boldsymbol{x}$，$\boldsymbol{\eta}-\boldsymbol{\rho}=\boldsymbol{v}$ 的要求仍然满足，因此并不影响最终结果。

本文的原-对偶算法对启动没有严格要求，可从任意点启动，这是该算法的一大优点。事实上，对 WLAV 状态估计问题，原问题的初始可行解可以通过式(2)方便地获得，而对偶问题的初始可行解的获得则较困难。

对罚参数 μ，由于在最优点 μ 必须为零，从而满足补偿松弛条件，因此 μ 在迭代中需逐步减少，文献[8]用“对偶间隙”的方法，按 $\mu=\boldsymbol{x}^{\mathrm{T}}\boldsymbol{s}/(2n+2m)$ 选取。由于前述 WLAV 状态估计问题的特点，按此方法选取并不理想。在我们的实现中，每一步迭代时，μ 的选取按固定的比例下降。一个简单而有效的方法是：μ 开始为 1，其后按 1/10 的比例缩小。

WLAV 状态估计问题的求解必须迭代获得，每个 SE 迭代求解一个 LP 问题，重新线性化后转入下一 SE 迭代。而重新线性化的后果将导致下一次线性规划问题的目标函数增大。因此，采取迭代限制十分必要。迭代限制后，可能在线性规划未达到最优时退出一个 LP 迭代而转入下一 SE 迭代，但这样可以使重新线性化后，目标函数不会产生大的回升，有利于 WLAV 状态估计达到渐近收敛。文献[9]采用按迭代次数进行限制的方法，每个 LP 迭代次数固定，我们认为并不合理。本文提出按最优性条件的松紧程度进行限制的方法，即随着 SE 迭代的进行，线性规划的最优性条件由松逐渐变为严格，算例表明本文的方法较好。

3 算例分析

对所提出的方法在 IEEE-14 节点及 IEEE-30 节点系统进行了仿真计算。所用数据如表 1。其中量测数据均是由潮流计算结果加上一定的随机量测误差进行模拟得出，原始系统数据参见文献[11]。采用不同迭代限制策略的仿真结果见表 2。

表 1 量测构成
Table 1 Measurement configuration

系统	潮流量测数	注入量测数	电压量测数
IEEE-14	34	14	5
IEEE-30	78	20	10

表 2 不同迭代限制时的收敛效果
Table 2 Effects of different iteration limits

SE迭代次数	14 节点 LP 迭代次数			30 节点 LP 迭代次数		
	不限	限 1	限 2	不限	限 1	限 2
1	9	3	3	10	3	3
2	8	3	5	8	3	5
3	8	9	5	8	9	5
4	8	9	6		8	
5		9	6			

注：不限指无迭代次数限制；限 1 指按迭代次数进行限制，前两次迭代限制为 3 次；限 2 指按最优性条件松紧程度进行限制。

表 2 中，按最优性条件松紧程度限制的具体方法为：随着 SE 迭代的进行，对偶间隙限制逐渐减小。由表中的结果可以看出，有限制情况下的收敛次数优于无限制的情况，而按最优性条件松紧程度进行限制又优于按迭代次数进行限制。

表 3 为 IEEE-14 节点系统，运用本文方法按最优性条件松紧程度进行限制和运用文献[4]中的解耦方法分别得到的 LP 迭代次数比较，可见本文方法的收敛性明显优于文献[4]中的解耦法。当按最优性条件松紧程度进行限制(限 2)时，在奔腾 166 PC 机上所用计算时间如表 4 所示。

表 3 两种方法的比较
Table 3 Comparison between two methods

SE 迭代次数	文献[4]方法 LP 迭代次数		本文方法(限 2) LP 迭代次数
	有功	无功	
1	21	25	3
2	17	14	5
3	9	4	5
4	1	0	6
5	0	0	6

表 4 CPU 时间
Table 4 CPU time

s

系统	预处理(第 1 步)	实际计算(第 2 步)	总时间
IEEE-14	0.11	0.11	0.22
IEEE-30	0.61	0.16	0.77

表 4 中,预处理包含了模拟定序、符号分解及对 $\boldsymbol{AXS}^{-1}\boldsymbol{A}^{\mathrm{T}}$ 阵的符号运算。这一部分所需的时间较长,但对于给定的网络及量测结构,第 1 步的预处理只进行一次,其后对不同的量测数据只需进行第 2 步的计算。由表中可以看出,算法的运算速度较快。

最后,给出按最优性条件松紧程度进行限制时,对 IEEE-30 母线系统的估计结果(见表 5)。

表 5 IEEE-30 母线系统估计结果
Table 5 State estimation results of IEEE-30 bus system

母线号	母线电压		母线号	母线电压	
	幅值[1)]	相角/(°)		幅值[1)]	相角/(°)
1	1.0500	0.0000	16	1.0478	−9.7881
2	1.0339	−2.7352	17	1.0461	−10.1200
3	1.0309	−4.6934	18	1.0321	−10.7989
4	1.0258	−5.6184	19	1.0309	−10.9722
5	1.0059	−8.9827	20	1.0356	−10.7789
6	1.0216	−6.4559	21	1.0406	−10.4261
7	1.0074	−8.0126	22	1.0411	−10.4191
8	1.0232	−6.4521	23	1.0316	−10.6265
9	1.0584	−8.1079	24	1.0294	−10.8681
10	1.0529	−9.9677	25	1.0301	−10.7430
11	1.0915	−6.2600	26	1.0127	−11.1511
12	1.0565	−9.1771	27	1.0391	−10.4111
13	1.0884	−7.9987	28	1.0181	−6.8195
14	1.0430	−10.0688	29	1.0196	−11.6043
15	1.0395	−10.1894	30	1.0083	−12.4562

注:1)为标幺值。

4 结论

本文针对现有 WLAV 状态估计算法存在的问题,将原-对偶内点算法引入 WLAV 状态估计,并结合 WLAV 状态估计问题的特点,提出了罚参数修正及迭代限制方法,同时利用稀疏矩阵方法中的符号运算技术进行计算。算例表明,所提出的算法在保持了原-对偶内点方法的数值稳定性及收敛性的基础上,进一步减少了迭代次数,在计算速度上也取得了良好的效果,性能优于传统方法。

参 考 文 献

1 于尔铿.电力系统状态估计.北京:水利电力出版社,1985

2 Holten L, Gjelsvik A, Aam S, et al. Comparison of Different Methods for State Estimation. IEEE Trans on PWRS, 1988, 3(4): 1798~1806

3 Kotiuga W W, Vidyasagar M. Bad Data Rejection Properties of Weighted Least Absolute Value Techniques Applied to Static State Estimation. IEEE Trans on PAS, 1982, 101(4):844~853

4 Abur A, Celik M K. A Fast Algorithm for the Weighted Least Absolute Value State Estimation. IEEE Trans on PWRS, 1991, 6(1):1~8

5 El-Keib A A, Singh H. Fast Linear Programming State Estimation Using the Dual Formation. IEEE Trans on PWRS, 1992, 7(2):620~628

6 Celik M K, Abur A. Use of Scaling in WLAV Estimation of Power System States. IEEE Trans on PWRS, 1992, 7(2):684~692

7 Monteiro R, Adler I. Interior Path Following Primal-Dual Algorithms (Part Ⅰ): Linear Programming. Mathematical Programming, 1989, 44:27~42

8 方述成,普拉普森 S.线性优化及其扩展——理论与算法.北京:科学出版社,1994

9 Singh H, Alvarado F L. Weighted Least Absolute Value State Estimation Using Interior Methods. IEEE Trans on PWRS, 1994, 9(3):1478~1484

10 Wei H, Sasaki H, Yokoyama R. An Application of Interior Point Quadratic Programming Algorithm to Power System Optimization Problems. IEEE Trans on PWRS, 1996, 11(1):260~266

11 张伯明,陈寿孙.高等电力网络分析.北京:清华大学出版社,1996

郭 伟,男,1970 年生,博士研究生,研究方向为电力系统状态估计。

单渊达,男,1930 年生,教授,博士生导师,研究方向为电力系统稳定控制、电力系统规划及 EMS。

WEIGHTED LEAST ABSOLUTE VALUE STATE ESTIMATION BASED ON PRIMAL-DUAL INTERIOR POINT ALGORITHM

Guo Wei, Shan Yuanda (Southeast University, 210096, Nanjing, China)

Abstract In view of the questions existing in the current WLAV state estimation algorithms, a primal-dual interior point algorithm is used to solve the WLAV state estimation problem. The ways about how to update the penalty parameter and to limit the iteration counts are presented based on the features of WLAV state estimation problem. Sparse matrix technique is also used in the presented algorithm. The numerical results show that the algorithm has the features, namely good robustness, less iteration counts and fast calculation. The performance of the algorithm is superior to the traditional methods.

Keywords WLAV state estimation linear programming interior point method