

参数线性规划问题的新型光滑精确罚函数神经网络^①

陈珊珊, 楼旭阳, 崔宝同

(江南大学 物联网工程学院 无锡 214122)

摘要: 针对不等式约束条件下, 目标函数和约束条件中含有参数的线性规划问题, 提出一种基于新型光滑精确罚函数的神经网络计算方法. 引入误差函数构造单位阶跃函数的近似函数, 给出一种更加精确地逼近于 L1 精确罚函数的光滑罚函数, 讨论了其基本性质; 利用所提光滑精确罚函数建立了求解参数线性规划问题的神经网络模型, 证明了该网络模型的稳定性和收敛性, 并给出了详细的算法步骤. 数值仿真验证了所提方法具有罚因子取值小、结构简单、计算精度高等优点.

关键词: 参数线性规划; L1 精确罚函数; 误差函数; 神经网络

Novel Smooth Exact Penalty Function Neural Networks for Parameter Linear Programming Problems

CHEN Shan-Shan, LOU Xu-Yang, CUI Bao-Tong

(School of IoT Engineering, Jiangnan University, Wuxi 214122, China)

Abstract: In view of solving linear programming problems with parameters both in objective function and constraints, a computational method based on novel smooth exact penalty function neural networks is proposed. First, the error function is introduced to constructing the approximate function of unit step function, which is used to give the smooth penalty function that more accurately approximates the L1 exact penalty function, and its basic properties are discussed. Second, the neural network model for parameter linear programming problems is constructed based on the proposed smooth exact penalty function and the stability and convergence of the neural networks are proved. Moreover, the specific calculation steps of our proposed neural network model for the optimization are given. Finally, a numerical example is given to illustrate that the proposed method possesses the smaller penalty factor, easier construction and higher accuracy.

Key words: parametric linear programming; L1 exact penalty function; error function; neural network

参数线性规划问题是运筹学的理论组成之一, 其诞生起源于对实际问题的需要, 其应用多半出现在我们的日常管理中. 例如, 单位货物运价变化的运输问题, 资源利用的数量变化问题^[1], 化学工业的实时优化模型^[2]等. 在这些流程管理或过程控制中, 决策者需要根据市场动态、生产过程的变化及时地调整解决方案. 为了确保受控对象的协调与精确, 达到最优运作和产出这一目标, 可以在线性规划问题的建模中加入一个在一定范围内变化的参数, 即形成了参数线性规划问题.

在求解参数线性规划问题的研究中, 单纯形表上作业法是最传统的计算方法之一^[3]. 然而, 随着当代控制科学与工业系统的发展, 受控对象的规模和复杂度不可同日而语. 单纯形法的计算步骤复杂而繁琐, 无法胜任实时计算的重任. 神经网络拥有大规模并行、分布式存储等优势, 无疑是计算复杂规划问题最主流的方法之一.

建立神经网络模型解决规划问题时, 利用罚函数法来构造能量函数和相应的微分动力学系统是十分有效的方法^[4]. 上世纪 60 年代末, Zangwill^[5]等首次提出

^①收稿时间:2014-02-27;收到修改稿时间:2014-03-28

精确罚函数的概念, 将约束规划问题转化成无约束规划问题, 其罚因子是固定常数. 然而, 精确罚函数的不可微性一直阻碍着最速下降法的使用, 从而使得神经网络模型的建立遇到困难. 近年来, 为了实现精确罚函数的光滑化, 国内外的学者都在努力寻找逼近精确罚函数的罚函数, 取得了诸多成果^[6-10]. 文献[6]详细地介绍了经典的改进法, 即在提出 L_2 精确罚函数的基础上, 通过对不可微项进行多种次方处理加以变型, 使之光滑. 在此基础上, 文献[7-9]通过引入光滑参数, 定义了连续可微分段函数, 对低阶精确罚函数进行了光滑化处理. 文献[10]在改进经典双参数光滑化模型的基础上, 提出一类逼近 L_1 精确罚函数的罚函数, 并给出了多种用于建立该模型的具体函数. 总的来说, 这些方法的构造比较复杂, 变化因子太多, 计算难度大.

为了使光滑化的模型结构更加简单、计算效率更高, 本文采用文献[11]的思路, 引入一个常见的简单函数作为 L_1 精确罚函数中不可微项的近似. 不同的是, 该文献引入的是 sigmoid 函数, 本文引入了逼近效果更加明显的误差函数. 进一步, 利用最速下降原理建立了神经网络模型来求解参数线性规划问题. 最后, 算例仿真说明了该方法的可行性、有效性.

1 光滑精确罚函数的建立

考虑如下含有不等式约束的参数线性规划问题:

$$\begin{aligned} \min f(x, \lambda) & \quad (1) \\ \text{st } g_i(x, \lambda) \leq 0, \quad & i=1, 2, \dots, m \\ \lambda \in [\alpha, \beta] & \\ x \in R^n & \end{aligned}$$

其中, $f(x, \lambda)$ 和 $g_i(x, \lambda)$ ($i=1, 2, \dots, m$) 都是 R^n 上的线性可微凸函数, $\lambda > 0$ 为参数, α 和 β 皆为常数 ($\alpha < \beta$). 假设对每个 $\lambda \in [\alpha, \beta]$, 问题(1)都有可行解, 且可行集为 $X_0 = \{x \in X | g_i(x, \lambda) \leq 0, \lambda \in [\alpha, \beta], i=1, 2, \dots, m\}$.

利用 Zangwill^[5]所提的 L_1 精确罚函数, 我们将含有不等式约束的参数线性规划问题转化为无约束的参数线性规划问题, 即把问题(1)转化成问题(2):

$$\min P_0(x, \lambda, \rho) = f(x, \lambda) + \rho \sum_{i=1}^m (g_i(x, \lambda))^+ \quad (2)$$

其中, $\rho > 0$ 为罚因子, $(g_i(x, \lambda))^+ = \max\{g_i(x, \lambda), 0\}$ ($i=1, 2, \dots, m$), 并有如下结论^[12]: 若 L_1 精确罚函数对某些 $\rho_0 > 0$ 有极小点, 则对任一参数 $\rho > \rho_0$, L_1 精确罚函

数均有极小点, 并且若 ρ 的值充分大, 那么 $P_0(x, \lambda, \rho)$ 的极小点皆为可行的, 亦为最优的.

为了使不可微项 $(g_i(x, \lambda))^+$ 光滑化, 文献[11]提出如下(3)式表示:

$$(g_i(x, \lambda))^+ = \int_{-\infty}^{g_i(x, \lambda)} w(z) dz \quad (3)$$

其中, $w(z)$ 是由(4)式定义的单位阶跃函数:

$$w(z) = \begin{cases} 0, & z \leq 0 \\ 1, & z > 0 \end{cases} \quad (4)$$

并且引用 Sigmoid 函数:

$$s(x, \alpha) = \frac{1}{1 + e^{-\alpha x}}, \alpha > 0 \quad (5)$$

作为 $w(z)$ 的一个近似, 于是不可微项 $(g_i(x, \lambda))^+$ 可由 Sigmoid 函数的积分来近似, 即得到(6)式:

$$\begin{aligned} (g_i(x, \lambda))^+ & \approx Q_i(x, \lambda, \alpha) = \int_{-\infty}^{g_i(x, \lambda)} s(z, \alpha) dz \\ & = g_i(x, \lambda) + \frac{1}{\alpha} \log(1 + e^{-\alpha g_i(x, \lambda)}) \quad (6) \end{aligned}$$

事实证明, 这种方法在解决凸不等式组和互补问题方面具有良好的效果.

根据上述思路, 我们引入误差函数 $f_e(x)$ 构造一类新函数 $H(x, k)$, 作为单位阶跃函数的近似函数. 误差函数的定义如(7)式所示, 是定义域 R 上的单调递增奇函数, 其值域为 $(-1, 1)$. 具体函数图像如图 1 所示:

$$f_e(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\eta^2} d\eta \quad (7)$$

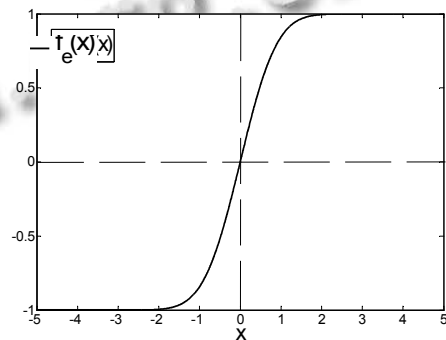


图 1 误差函数 $f_e(x)$ 随 x 变化曲线

以 $f_e(x)=0$ 为对称轴, 将函数 $f_e(x)$ 的纵坐标收缩为原来的 0.5 倍, 横坐标保持不变. 然后, 再以 $x=0$ 为轴, 整体沿着纵坐标增大的方向垂直平移 0.5 个单位, 即得到函数 $H(x)$. 将函数 $f_e(x)$ 的调节参数 k 考虑在内, 可构造一类新的函数 $H(x, k)$.

函数 $H(x, k)$ 的定义式可表示为(8)式所示:

$$H(x,k) = \frac{1}{2} + \frac{1}{2} f_e(x,k), k > 0 \quad (8)$$

其中, k 为常数. 此时, 函数 $H(x,k)$ 的定义域为 R , 值域为 $(0,1)$. 显然, 在定义域内, 对任意给定的一点, 总能在单位阶跃函数中找到与其一一对应的点.

对于(8)式来说, 注意到其有下列基本性质:

i. $\lim_{x \rightarrow -\infty} H(x,k) = 0, \lim_{x \rightarrow 0} H(x,k) = 1/2, \lim_{x \rightarrow +\infty} H(x,k) = 1.$

ii. 对任何正整数 $k, H(x,k)$ 是 k 次连续可微的. 且有: $\frac{dH(x,k)}{dx} = \frac{ke^{-k^2x^2}}{\sqrt{\pi}}, \frac{d^2H(x,k)}{dx^2} = -\frac{2k^3xe^{-k^2x^2}}{\sqrt{\pi}}.$

iii. $H(x,k)$ 在 R 上是单调递增函数.

图 2 分别给出了当 $\alpha = k = 1$ 和 $\alpha = k = 5$ 时, $w(x), s(x,\alpha), H(x,k)$ 随 x 变化的曲线.

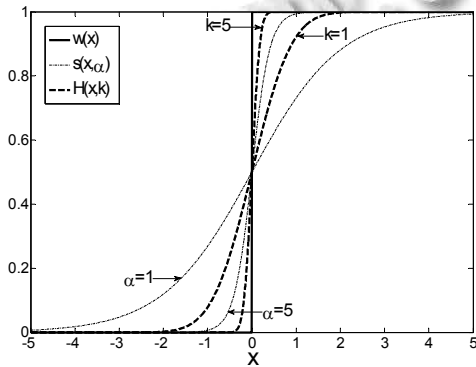


图 2 $w(x), s(x,\alpha), H(x,k)$ 随 x 变化曲线

显然, 在相同常数下, $H(x,k)$ 逼近 $w(x)$ 的效果比文献[11]中的 $s(x,\alpha)$ 更加明显, 且常数越大, 近似程度越高.

随后, 用(8)式替换(5)式, 代入(6)式求积分, 得到(9)式:

$$\begin{aligned} (g_i(x,\lambda))^+ &\approx Q_2(x,\lambda,k) \\ &= \int_{-\infty}^{g_i(x,\lambda)} H(z,k) dz \\ &= \frac{1}{2} \int_{-\infty}^x (1 + \operatorname{erf}(g_i^{-1}(x,\lambda),k)) dg_i^{-1}(x,\lambda) \end{aligned} \quad (9)$$

其中, $g_i^{-1}(x,\lambda)$ 是 $g_i(x,\lambda)$ 的反函数. 进一步, 利用不可微项 $(g_i(x,\lambda))^+$ 的近似项 $Q_2(x,\lambda,k)$, 可以将问题(2)转化成问题(10):

$$\min P(x,\lambda,\rho,k) = f(x,\lambda) + \rho \sum_{i=1}^m Q_2(x,\lambda,k) \quad (10)$$

显然, 对于任意的 $\rho > 0$, 有

$$\lim_{k \rightarrow +\infty} P(x,\lambda,\rho,k) = P_0(x,\lambda,\rho).$$

2 神经网络模型及其稳定性

$$\dot{x} = -\frac{\partial f(x,\lambda)}{\partial x} - \rho \sum_{i=1}^m H(g_i(x,\lambda),k) \frac{\partial g_i(x,\lambda)}{\partial x} \quad (11)$$

其中,

$$\begin{aligned} \frac{\partial f(x,\lambda)}{\partial x} &= \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_m} \right)^T, \\ \frac{\partial g_i(x,\lambda)}{\partial x} &= \left(\frac{\partial g_i}{\partial x_1}, \frac{\partial g_i}{\partial x_2}, \dots, \frac{\partial g_i}{\partial x_m} \right)^T. \end{aligned}$$

定理 1. 若对于任意给定的 ρ , 问题(10)都有一个最优解, 并且(11)式总能找到一个状态变量 x 使得其在 x^* 处是渐近稳定的, 那么该神经网络模型的平衡点即是问题(10)的最优解.

证明: 问题(10)取得最优解的必要条件为:

$$\frac{\partial P(x,\lambda,\rho,k)}{\partial x} = \frac{\partial f(x,\lambda)}{\partial x} + \rho \sum_{i=1}^m \frac{\partial Q_2(x,\lambda,k)}{\partial x} = 0 \quad (12)$$

即

$$-\frac{\partial f(x,\lambda)}{\partial x} - \rho \sum_{i=1}^m \frac{\partial Q_2(x,\lambda,k)}{\partial x} = 0 \quad (13)$$

对于每个 $\lambda \in [\alpha, \beta]$, (11)式是渐近稳定的, 则平衡点 x^* 满足(13)式. 因此, (11)式的平衡点即是问题(10)的最优解. 证毕.

定理 2. 如果问题(10)中的 $P(x,\lambda,\rho,k)$ 有下界, 那么神经网络模型(11)式的状态变化使函数 $P(x,\lambda,\rho,k)$ 的值减小, 最终将到达 $P(x,\lambda,\rho,k)$ 的零梯度点并停留于该点.

证明: 定义 $P(x,\lambda,\rho,k)$ 为神经网络模型(11)式的一个能量函数 $E(x,\lambda,\rho,k)$, 则:

$$\begin{aligned} \frac{dE(x,\lambda,\rho,k)}{dt} &= \frac{\partial f^T(x,\lambda)}{\partial x} \dot{x} + \rho \sum_{i=1}^m \frac{\partial Q_2(x,\lambda,k)}{\partial x} \dot{x} \\ &= \left(\frac{\partial f^T(x,\lambda)}{\partial x} + \rho \sum_{i=1}^m \frac{\partial Q_2(x,\lambda,k)}{\partial x} \right) \dot{x} \\ &= \dot{x}^T (-\dot{x} - \dot{x}) \dot{x} \leq 0 \end{aligned} \quad (14)$$

由(14)式可知, 神经网络模型(11)式是稳定的. 证毕.

运用欧拉方法的思想, 用所提神经网络模型计算参数线性规划问题的主要方法是, 将参数区间 $[\alpha, \beta]$ 均分成 r 个小区间, 用神经网络模型(11)计算每个 λ_j 处目标函数的最优值, 此时 $\lambda_j = \alpha + (\beta - \alpha)j/r (j = 0, 1, \dots, r)$. 详细算法步骤及部分

Matlab 程序设计如下:

- Step1 设置初始点 x_0 , 令 $dx/dt=0$;
- Step2 设循环次数为 n , 令 $dt=1/n$;
- Step3 设 ρ 为罚因子, 且 $\rho>0$;
- Step4 设 k 为常数, 且 $k>0$;
- Step5 将参数区间 $[\alpha, \beta]$ 均分成 r 个, 每个区间的间隔是 $(\beta-\alpha)/r$;
- Step6 进入循环, 开始逐个计算每个 λ 处, 问题(1)的最优值:

```

for j = 1:r+1
    for i = 1:n
        F = -\frac{\partial f(x, \lambda)}{\partial x} - \rho \sum_{i=1}^m H(g_i(x, \lambda), k) \frac{\partial g_i(x, \lambda)}{\partial x}
        dx = Fdt;
        dx = max(x + dx, 0) - x;
    end;
    X(j) = x; X(j)即为\lambda的最优解
end
    
```

3 数值仿真

考虑如下参数线性规划问题^[13]:

$$\begin{aligned}
 \min \quad & (\lambda^2 - \lambda - 3)x_1 + (\lambda^2 + 2\lambda - 6)x_2 \\
 \text{s.t.} \quad & x_1 \leq 4 + \lambda^2 \\
 & 3x_1 + 2x_2 \leq 18 - 2\lambda^2 \\
 & x_1, x_2 \geq 0 \\
 & \lambda \in [0, 3]
 \end{aligned}$$

该问题的理论解为:

$$\begin{aligned}
 \forall \lambda \in [0, 1.3], \quad & x^*(\lambda) = \begin{bmatrix} 0 \\ 9 - \lambda^2 \end{bmatrix}, \\
 z^*(\lambda) &= -\lambda^4 - 2\lambda^3 + 15\lambda^2 + 18\lambda - 54, \\
 \forall \lambda \in [1.3, 2.3], \quad & x^*(\lambda) = \begin{bmatrix} 6 - \frac{2}{3}\lambda^2 \\ 0 \end{bmatrix}, \\
 z^*(\lambda) &= -\frac{2}{3}(\lambda^4 - \lambda^3) + 8\lambda^2 - 6\lambda - 18, \\
 \forall \lambda \in [2.3, 3], \quad & x^*(\lambda) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad z^*(\lambda) = 0.
 \end{aligned}$$

下面用本文所提方法求解此参数线性规划问题:

首先, 利用 L_1 新型光滑精确罚函数 $P(x, \lambda, \rho, k)$, 将该问题转化成无约束的参数线性规划问题:

$$\min P(x, \lambda, \rho, k) = -x_1^2 - 3x_2^2 + \rho \sum_{i=1}^4 H(g_i(x, \lambda), k) \frac{\partial g_i(x, \lambda)}{\partial x}$$

其中,

$$\begin{aligned}
 g_1(x_1, x_2, \lambda) &= x_1 - 4 - \lambda^2 \leq 0 \\
 g_2(x_1, x_2, \lambda) &= 3x_1 + 2x_2 + 2\lambda^2 - 18 \leq 0 \\
 g_3(x_1, x_2, \lambda) &= -x_1 \leq 0 \\
 g_4(x_1, x_2, \lambda) &= -x_2 \leq 0
 \end{aligned}$$

其次, 根据所提出的微分动力学方程(11)式, 建立如下神经网络模型:

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = - \begin{bmatrix} \lambda^2 - \lambda - 3 \\ \lambda^2 + 2\lambda - 6 \end{bmatrix} - \rho \begin{bmatrix} \sum_{i=1}^4 H(g_i(x, \lambda), k) \frac{\partial g_i(x, \lambda)}{\partial x_1} \\ \sum_{i=1}^4 H(g_i(x, \lambda), k) \frac{\partial g_i(x, \lambda)}{\partial x_2} \end{bmatrix}$$

最后, 在定义域内取任一初始值 $x_0 = [0, 6]^T$, 设循环计算次数 $n=200$, 罚因子 $\rho=4$, 区间分段数 $r=49$, 此时 $\lambda_j = 0 + (3-0)j/49 (j=0, 1, \dots, 49)$, 调节参数 $k=1000$, 代入模型进行计算. 用 MATLAB 进行数值仿真, 图 3-5 显示了在区间 $[0, 3]$ 中, 所提神经网络模型分别算出的 50 个 λ 处参数线性规划问题的最优值.

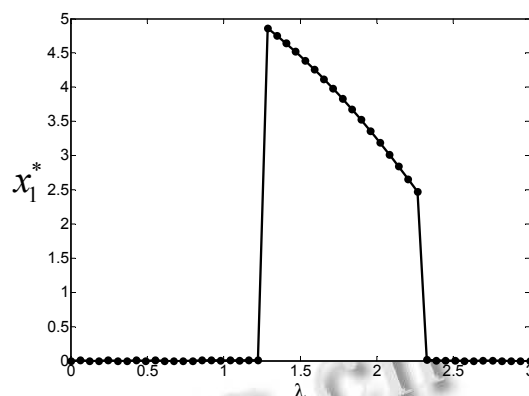


图 3 x_1^* 随 λ 变化的关系曲线

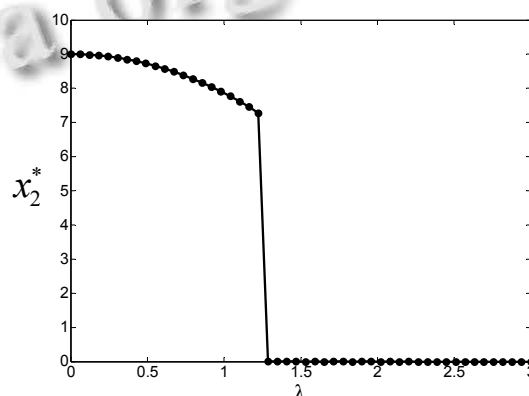


图 4 x_2^* 随 λ 变化的关系曲线

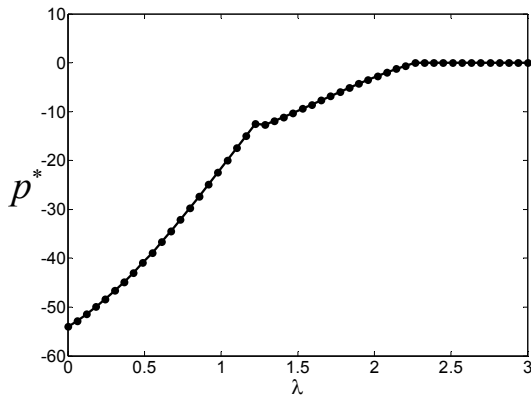


图 5 $P(x, \lambda)^*$ 随 λ 变化的关系曲线

设置相同的初始值, 运用文献[11]中的方法对该例进行仿真计算, 并与本文所提方法的计算结果进行了对比。

表 1 显示了 $\lambda \in [1, 2]$ 的 16 个点的计算结果对比情况。

表 1 两个模型计算结果对比

λ_j	本文解		文献[11]解		理论解	
	x_1	x_2	x_1	x_2	x_1	x_2
1.041	0.016	7.749	0.038	7.710	0.000	7.917
1.102	0.024	7.589	0.055	7.534	0.000	7.786
1.163	0.034	7.416	0.077	7.341	0.000	7.647
1.224	0.058	7.214	0.111	7.123	0.000	7.501
1.286	0.072	7.057	0.132	7.137	0.000	7.347
1.347	4.732	0.036	4.688	0.090	4.791	0.000
1.408	4.621	0.028	4.586	0.067	4.678	0.000
1.469	4.503	0.022	4.474	0.051	4.561	0.000
1.531	4.380	0.017	4.355	0.040	4.438	0.000
1.592	4.250	0.013	4.229	0.030	4.311	0.000
1.653	4.116	0.009	4.097	0.021	4.178	0.000
1.714	3.975	0.005	3.959	0.012	4.041	0.000
1.776	3.830	0.002	3.816	0.004	3.898	0.000
1.837	3.678	0.000	3.665	0.000	3.751	0.000
1.898	3.520	0.000	3.505	0.000	3.599	0.000
1.959	3.356	0.000	3.340	0.000	3.441	0.000

通过对比, 可以看出所提方法在计算方面的精度更高, 计算效果优势明显。

4 结论

本文对求解参数线性规划问题的 L_1 精确罚函数神经网络方法进行了研究, 创造性地提出引用误差函数作为单位阶跃函数的近似函数, 用于替换原方法中的

不可微项, 从而增强了用最速下降法建立神经网络模型的合理性及严谨性。所提出的网络模型结构简单, 罚因子可以取一个很小的常数, 求解精确度高, 大大提高了计算效率。最后, 数值实例说明了所提网络模型的可行性和有效性。

参考文献

- 1 申芸, 吕咏梅, 周永权. 用进化策略和神经网络求解参数线性规划. 计算机应用与软件, 2009, 26(2): 39-41.
- 2 Wang ZQ, Shao ZJ, Fang Xueyi, Chen Weifeng, Wan JN. A modified mnemonic enhancement optimization method for solving parametric nonlinear programming problems. 49th IEEE Conference on Decision and Control. 2010. 2210-2214.
- 3 刁在筠, 刘桂真, 宿洁, 马建华. 运筹学(第 3 版). 北京: 高等教育出版社, 2007.
- 4 孟志青, 胡奇英, 杨晓琪. 基于精确罚函数的一类广义非线性神经网络模型. 自动化学报, 2003, 29(5): 755-760.
- 5 Zangwill WI. Nonlinear programming via penalty function. Management Science, 1967, 13(5): 344-358.
- 6 Xu XS, Meng ZQ, Sun JW, Huang LG, Shen R. A second-order smooth penalty function algorithm for constrained optimization problems. Computational Optimization and Application, 2013, 55(1): 155-172.
- 7 徐新生, 孟志青. 低阶精确罚函数的一种二阶光滑逼近. 系统科学与数学, 2013, 33(5): 555-567.
- 8 Xu XS, Meng ZQ, Sun JW, Shen R. A penalty function method based on smoothing lower order penalty function. Journal of Computational and Applied Mathematics, 2011, 235(14): 4047-4058.
- 9 Jiang M, Shen R, Xu XS, Meng ZQ. Second-Order Smoothing Objective Penalty Function for Constrained Optimization Problems. Numerical Functional Analysis and Optimization, 2014, 35(3): 294-309.
- 10 连淑君, 刘炳状, 张连生. 一类逼近 L_1 精确罚函数的罚函数. 应用数学学报, 2007, 30(6): 961-971.
- 11 张菊亮, 章祥荪. 不等式约束最优化的非光滑精确罚函数的一个光滑近似. 系统科学与数学, 2000, 20(4): 499-505.
- 12 连淑君. 近似逼近 L_1 精确罚函数的罚函数. 应用数学, 2010, 23(2): 363-369.
- 13 Effati S, Jafarzadeh M. A new nonlinear neural network for solving a class of constrained parametric optimization problems. Applied Mathematics and Computation, 2007, 186(1): 814-819.