

相容线性系统的行投影块迭代算法

张 燕^{1,2}, 杨 洋¹, 陆伟东¹

(1 南京审计学院 应用数学系, 江苏 南京 210029; 2 河海大学 理学院, 江苏 南京 210098)

[摘要] 探讨了一种行投影块迭代算法来求解大型相容线性系统. 该算法基于 Kaczmarz 算法, 主要思想是首先对系数矩阵 A 进行分块, 然后通过选取离当前迭代点距离最远的块来进行投影, 并将投影作为下一个迭代点. 数值结果显示, 行投影迭代算法对坏条件问题非常有效, 所提出的算法与经典的 Ciminio 算法相比, 收敛速度更快. 另外还提出一种新的对系数矩阵 A 分块的列分解策略, 该策略基于每块的列相关性估计而得出.

[关键词] 行投影块迭代算法, 列分解策略, 最远块

[中图分类号] O29 [文献标识码] A [文章编号] 1672-1292(2006)01-0047-05

Block-Iterative Algorithm with Row Projection for Consistent Linear System

ZHANG Yan^{1,2}, YANG Yang¹, LU Weidong¹

(1 Department of Applied Mathematics, Nanjing Audit University, Nanjing 210029, China

2 School of Science, Hohai University, Nanjing 210098, China)

Abstract In this paper we discuss a block-iterative algorithm with row projection for solving large consistent linear system. This algorithm is based on the Kaczmarz algorithm. The key idea is that the coefficient matrix A is firstly divided into blocks, then the current iterative point is projected onto the remotest block measured by the distance between the iterative point and the block, and the projection is taken as the next iterative point. Numerical simulations show that block-iterative algorithm with row projection is very efficient for solving ill-conditioned problems. Compared with the classical Ciminio algorithm, algorithm accelerates the convergence greatly. In addition we present a new column partition strategy based on the estimate of column-dependence of each block, to divide the coefficient matrix A .

Key words block-iterative algorithm with row projection, column partition strategy, the remotest block

0 引言

对于大型非对称线性系统, 直接法求解很不现实, 一般采用迭代法. 通常使用的迭代法, 如共轭梯度法、残量多项式法、Jacob 迭代法等, 往往只适用于特定的问题, 应用上有其局限性. 大部分的非对称解不是要求存储、计算和系数矩阵的特殊谱性质来保证收敛性, 就是要求对系数矩阵进行对称化. 存储和计算量会随着迭代步数惊人的增加, 对称化过程也会对系数矩阵带来潜在的丢失数据的影响. 避免这些问题的一个方法就是加速行投影迭代算法. 行投影迭代算法应用很广泛, 算法很简单, 但经典行投影算法, 如 Ciminio 算法、Kaczmarz 算法, 收敛速度很慢. 本文主要探讨了一种快速行投影迭代算法. 该方法基于 Kaczmarz 算法, 主要思想是首先对系数矩阵 A 进行分块, 然后通过选取最远的块来控制序列进行迭代.

1 行投影迭代算法

对于线性系统 $A^T x = b$, $A \in \mathbf{R}^{n \times m}$, 假设该系统是相容的且系数矩阵 A 为非零矩阵. 首先对 $A^T x = b$ 进行如下分块: $A = [A_1, A_2, \dots, A_q]$, $b^T = [b_1^T, b_2^T, \dots, b_q^T]$.

收稿日期: 2005-09-26
基金项目: 南京审计学院青年科研基金资助项目 (NSK2005/C08).
作者简介: 张 燕 (1978-), 女, 讲师, 主要从事应用数学方面的教学与研究. E-mail zhy8317@nau.edu.cn

其中 $A_i \in \mathbf{R}^{n \times m_i}$, $b_i \in \mathbf{R}^{m_i}$, $i = 1 \dots, q$ 且 $\sum_{i=1}^q m_i = m$. 同时假设矩阵 A 的每一块 A_i 均为列满秩, 即 $\text{rank}(A_i) = m_i$. 该假设可以通过第三部分讨论的分块算法得到保证.

定义 H_i 为下面的集合: $H_i \equiv \{x \in \mathbf{R}^n: A_i^T x = b_i\}$
其中 $i = 1 \ 2 \ \dots, \ q$ 因此任何 $x^* \in \bigcap_{i=1}^q H_i$ 均为线性系统 $A^T x = b$ 的解. 此外可以计算任意 $x \in \mathbf{R}^n$ 在 H_i 上的直交投影 $P_i(x)^{[1]}$: $P_i(x) = x + A_i(A_i^T A_i)^{-1}(b_i - A_i^T x)$.

给定一个迭代点 x^k , 行投影迭代算法将 x^k 在所有 H_i 上的直交投影的线性组合 $\sum_{i=1}^q w_i P_i(x^k)$ 作为下一个迭代点, 其中 $\sum_{i=1}^q w_i = 1$ 经典的行投影迭代算法有 Cimmino 算法^[2]、Kaczmarz 算法^[3], 但其收敛速度很慢. 以下给出 Cimmino 算法:

初始化: 任意 $x^0 \in \mathbf{R}^n$
迭代步骤: $x^{k+1} = x^k + \lambda_k \sum_{i=1}^q w_i^k (P_i(x^k) - x^k)$, $k \leftarrow k + 1$

其中 λ_k 是松弛参数, $w^k = (w_i^k)_{i=1}^q$ 是变化的权向量且满足 $w_i^k \geq 0$ 及 $\sum_{i=1}^q w_i^k = 1$

对于经典的 Cimmino 算法, 权为常数. Aharoni 和 Censor^[4] 证明了如果对所有的 $k \geq 0$ 任意的 $\varepsilon > 0$ 有 $\varepsilon \leq \lambda_k \leq 2 - \varepsilon$ 且 $\sum_{i=1}^\infty w_i^k = +\infty$ 对 $i = 1 \ 2 \ \dots, \ q$ 成立, 那么由 Cimmino 算法生成的序列 $\{x^k\}$ 收敛于向量 x^* , x^* 满足 $A^T x^* = b$

2 最远块控制序列的块迭代算法

经典的 Cimmino 算法由于收敛速度较慢, 在实际应用中很少使用. 本节给出一个新的行投影块迭代算法, 最远块控制序列的块迭代算法. 该算法基于 kaczmarz 算法, 通过选取离当前迭代点距离最远的块来进行直交投影, 并将投影作为下一个迭代点, 从而加快了收敛速度. 首先给出该算法, 然后证明其收敛性.

假设当前迭代点为 x^k , 首先计算 x^k 在所有 H_i 上的直交投影 $P_i(x^k)$, 然后比较所有的 $\|P_i(x^k) - x^k\|_2$ 并求出其中最大者, $i = 1 \dots, \ q$ 即找出离 x^k 直交距离最远的块 H_j . 假设 $\|P_j(x^k) - x^k\|_2$ 最大, 即 $\|P_j(x^k) - x^k\|_2 = \max_{1 \leq i \leq q} \|P_i(x^k) - x^k\|_2$, 然后选取 x^k 在 H_j 上的直交投影 $P_j(x^k)$ 为下一个迭代点, 即 $x^{k+1} = P_j(x^k)$. 继续上面的过程, 得到一个序列 $\{x^k\}$. 该序列收敛于线性系统 $A^T x = b$ 的解 x^* . 该算法描述如下:

算法 1
初始化: 对矩阵 A 进行分块, 得到 $A = [A_1, A_2, \dots, A_q]$ 及相应的 $b^T = [b_1^T, b_2^T, \dots, b_q^T]$.
主要步骤: 给定初始点 $x^0 \in \mathbf{R}^n$, $0 \leq \epsilon < 1$ 令 $k = 0$
for $i = 1 \ 2 \ \dots, \ q$
 $Q_i = (A_i^T A_i)^{-1}$
while $(\|A^T x^k - b\|_2 > \epsilon)$
/
 $d = A_1 Q_1 (b_1 - A_1^T x^k)$
 for $j = 2 \ 3 \ \dots, \ q$
 if $\|A_j Q_j (b_j - A_j^T x^k)\|_2 > \|d\|_2$
 $\{d = A_j Q_j (b_j - A_j^T x^k)\}$
 $x^{k+1} = x^k + d$
 $k = k + 1$
/

为了证明算法 1 的收敛性, 我们引用 [5] 的一些结果.

令 $H = \bigcap_{i \in P} H_i$, $\phi(x) = \min_{i \in P} \{d(x, H_i)\}$, 其中 $P = \{1, 2, \dots, q\}$, $d(x, H_i)$ 表示为点 $x \in \mathbf{R}^n$ 到闭凸集 $H_i \subseteq \mathbf{R}^n$ 的欧基里德距离.

定义 1^[5] 如果对任意 $x^* \in H$, 当 $k \geq 0$ 时有 $\|x^{k+1} - x^*\|_2 \leq \|x^k - x^*\|_2$, 则序列 $\{x^k\}_{k \geq 0}$ 被称为 Fejér 单调的. 易知每一个 Fejér 单调序列有界.

引理 1^[5] 设 $H_i \subset \mathbf{R}^n$ 为闭凸集, $i \in P$, $H = \bigcap_{i \in P} H_i$, $H \neq \emptyset$. 如果序列 $\{x^k\}$ 满足

(i) $\{x^k\}$ 对 H 是 Fejér 单调的, 且

(ii) $\lim_{k \rightarrow \infty} \phi(x^k) = 0$

则序列 $\{x^k\}$ 收敛于 $x^* \in H$.

定理 1 算法 1 生成的序列 $\{x^k\}$ 收敛于线性系统 $A^T x = b$ 的解 x^* .

证明 如果算法 1 生成的序列 $\{x^k\}$ 满足引理 1 的条件 (i) 和 (ii), 则 $\{x^k\}$ 收敛于线性系统 $A^T x = b$ 的

解 x^* . 为证明 (i), 考虑最优问题 $\min \|y - x^k\|_2$, $A_j^T(y - x^k) = b_j - A_j^T x^k$. 该问题最小范数解为 $\|d_j^k\|_2 = \|P_j(x^k) - x^k\|_2$. 根据 Kuhn-Tucker 最优条件 [6], 存在向量 $v_j^k \in \mathbf{R}^m$ 使得 $d_j^k = P_j(x^k) - x^k = A_j v_j^k$. 所以, $A_j^T A_j v_j^k = b_j - A_j^T x^k$. 两边乘以 $(v_j^k)^T$, 得到 $(v_j^k)^T A_j^T A_j v_j^k = (v_j^k)^T (b_j - A_j^T x^k)$.

因此, $\|d_j^k\|_2^2 = (v_j^k)^T (b_j - A_j^T x^k)$. 又因为 x^* 满足 $A_j^T x^* = b_j$, 从而有

$$\|d_j^k\|_2^2 = (v_j^k)^T A_j^T (x^* - x^k) = (d_j^k)^T (x^* - x^k) \quad (1)$$

由算法 1 知 $x^{k+1} = x^k + d^k$, 其中 $\|d^k\|_2 = \max_{1 \leq j \leq q} \|d_j^k\|_2$. 因此有, $\|x^{k+1} - x^*\|_2^2 = \|x^k + d^k - x^*\|_2^2 =$

$$\|x^k - x^*\|_2^2 - 2(d^k)^T (x^* - x^k) + \|d^k\|_2^2$$

由 (1) 得到: $\|x^{k+1} - x^*\|_2^2 = \|x^k - x^*\|_2^2 - \|d^k\|_2^2$. 即

$$\|x^{k+1} - x^*\|_2^2 = \|x^k - x^*\|_2^2 - \alpha_k, \quad \text{其中 } \alpha_k = \max_{1 \leq j \leq q} \|d_j^k\|_2^2 \quad (2)$$

以下证明算法 1 生成的序列 $\{x^k\}$ 满足引理 1 的条件 (ii).

由 (2) 知序列 $\{\|x^k - x^*\|_2\}$ 单调下降且有界. 有 $\{\|x^k - x^*\|_2\}$ 收敛于一实数. 因为 $\lim_{k \rightarrow \infty} \|x^{k+1} - x^*\|_2^2 = \lim_{k \rightarrow \infty} \|x^k - x^*\|_2^2 - \lim_{k \rightarrow \infty} \alpha_k$, 从而 $\lim_{k \rightarrow \infty} \alpha_k = 0$

又因为 $\phi(x^k) = \alpha_k$, 所以有 $\lim_{k \rightarrow \infty} \phi(x^k) = 0$

综上所述, 算法 1 生成的序列 $\{x^k\}$ 收敛于线性系统 $A^T x = b$ 的解 x^* .

3 系数矩阵 A 的分块

本节中给出一个对系数矩阵 A 的分块算法. 应用该算法, 每块最多有 μ 列, 且每块 $A_i \in \mathbf{R}^{n \times \mu_i}$ 均为满秩, 即每块的列是线性独立的.

假设正在生成一个块 A_i . 该块已包含矩阵 A 中的 j 列 ($j < \mu$), 且已知 A_i 的 $QR^{[7]}$ 分解 $A_i = Q_j R_j$. 令 r_{hh} 为 R_j 的位于 (h, h) 位置的元素. 块 A_i 的列相关性的估计 β_i 定义为 $\beta_i = \frac{d_{\max}}{d_{\min}}$, 其中 $d_{\max} = \max_{1 \leq h \leq j} \{r_{hh}\}$, $d_{\min} = \min_{1 \leq h \leq j} \{r_{hh}\}$.

对于矩阵 A 中的一列 a_l 是否该加入块 A_i 的准则为: a_l 没有被分配给任何其他的块, 且 a_l 加入后 A_i 的列相关性估计 β_i 不超过 κ , κ 为预先设定一个正数.

为了决定一个新的列 a_l 是否该加入第 i 块, 令 $A_i = [A_i, a_l]$, 且根据已有的 A_i 的 QR 分解来得到 A_i 的 QR 分解, 这是个 QR 分解更新问题^[7]. 然后计算 A_i 的列相关性估计, 如果该估计仍然小于 κ , 则 a_l 加入块 A_i .

刚开始时 $j = 1$, A_i 只有一列, 因此 $A_i = Q_1 R_1$, 其中 $R_1 = [\|A_i\|_2, 0]^T$, $0 \in \mathbf{R}^{n-1}$, $Q_1 = I_n - 2vv^T$, $v = \frac{R_1 - A_i}{\|R_1 - A_i\|_2}$.

如果已知 A_i 包含系数矩阵 A 中的 j 列, $j < \mu$ 且 $A_i = Q_j R_j$, 则可以计算新的 QR 分解 $A_i = Q_{j+1} R_{j+1}$.

具体过程请参考 [7].

注意到 $Q_j^T A_i = Q_j^T [A_i, a_l] = [R_j, Q_j^T a_l]$ 除了最后一列外有上三角形状. 令 $\zeta = (Q_j^T a_l)(j+1:n) \in \mathbf{R}^{n-j}$, $Q = I_{n-j} - 2 \frac{(\zeta - v)(\zeta - v)^T}{\|\zeta - v\|_2^2}$. 其中 $v = [\| \zeta \|_2, 0]^T \in \mathbf{R}^{n-j}$, 则有 $Q\zeta = v$ 且 $\begin{bmatrix} I_j & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} R_j & \eta \\ 0 & \zeta \end{bmatrix} = \begin{bmatrix} R_j & \eta \\ 0 & 0 \end{bmatrix}$. 其中 $R_j = R_j(1:j, 1:j)$, $\eta = (Q_j^T a_l)(1:j) \in \mathbf{R}^j$.

因此, $Q_{j+1} = Q_j \begin{bmatrix} I_j & 0 \\ 0 & Q \end{bmatrix}$, $R_{j+1} = \begin{bmatrix} R_j & \eta \\ 0 & v \end{bmatrix}$.

令 $d_{\max} = \max\{d_{\max}, \| \zeta \|_2\}$, $d_{\min} = \min\{d_{\min}, \| \zeta \|_2\}$, 如果 $d_{\min} \neq 0$ 且 $\frac{d_{\max}}{d_{\min}} \leq \kappa$ 则列 a_l 加入块 A_i . 如果更新后的块 A_i 的列数仍然小于 μ 可以继续通过上面的过程来分析新的列是否该加入.

块分解算法描述如下:
算法 2 令 $\Gamma = 1, 2, \dots, m$, $\Gamma_s = \{k \mid \text{矩阵 } A \text{ 中的列 } a_l \text{ 已经分配给某块}\}$
初始化: Set $i = 1$, $\Gamma_s = \emptyset$

```
While  $\Gamma_s \neq \Gamma$  do
    令  $\Gamma_c = \Gamma \setminus \Gamma_s$ ,  $j = 1$ ,  $\Gamma_i = \emptyset$ 
    选取  $l \in \Gamma_c$ 
    令  $A_i = [a_l]$ ,  $R_1 = [\| a_l \|_2, 0]^T$ 
    计算  $v = \frac{R_1 - A_i}{\| R_1 - A_i \|_2}$ ,  $Q_1 = I_n - 2vv^T$ 
    令  $d_{\max} = \| a_l \|_2$ ,  $d_{\min} = \| a_l \|_2$ 
     $\Gamma_c \leftarrow \Gamma_c \setminus \{l\}$ ,  $\Gamma_s \leftarrow \Gamma_s \cup \{l\}$ ,  $\Gamma_i \leftarrow \Gamma_i \cup \{l\}$ 
While  $(j < \mu \text{ 且 } \Gamma_c \neq \emptyset)$  do
    选取  $l \in \Gamma_c$ 
    令  $\alpha = (Q_j^T a_l)(j+1:n)$ ,  $v = [\| \alpha \|_2, 0]^T \in \mathbf{R}^{n-j}$ 
    计算  $Q = I_{n-j} - 2 \frac{(\alpha - v)(\alpha - v)^T}{\| \alpha - v \|_2^2}$ 
     $Q_{j+1} = Q_j \begin{bmatrix} I_j & 0 \\ 0 & Q \end{bmatrix}$ 
     $t_{\max} \leftarrow \max\{d_{\max}, \| \alpha \|_2\}$ ,  $t_{\min} \leftarrow \min\{d_{\min}, \| \alpha \|_2\}$ 
    if  $t_{\min} \neq 0$  且  $\frac{t_{\max}}{t_{\min}} \leq \kappa$  then
        更新  $A_i \leftarrow [A_i, a_l]$ 
         $\Gamma_c \leftarrow \Gamma_c \setminus \{l\}$ ,  $\Gamma_s \leftarrow \Gamma_s \cup \{l\}$ ,  $\Gamma_i \leftarrow \Gamma_i \cup \{l\}$ 
         $j \leftarrow j + 1$ 
         $d_{\max} \leftarrow t_{\max}$ ,  $d_{\min} \leftarrow t_{\min}$ 
    else
         $\Gamma_c \leftarrow \Gamma_c \setminus \{l\}$ 
    end
end
end
i ← i + 1
End
```

4 数值结果

下面应用算法 1 来求解随机生成的相容线性系统, 并且和经典 Cimm ino 算法作比较. 数值结果见表 1

~ 表 3 所有的数值测试均用 M atlab6.5 完成. 有关数值测试的说明如下:

测试问题: 求解线性相容系统 $A^T x = b, A \in \mathbf{R}^{n \times m}$. 系统矩阵 A 分 3 种情况由 M atlab 随机生成:

(1) $A = \text{rand}(n, m)$, 如表 1 所示.

(2) $B = \text{rand}(m/2, m), A = \begin{bmatrix} B \\ B + 10^{-9} \times \text{rand}(m/2, m) \end{bmatrix}$, 如表 2 所示.

(3) $B = \text{rand}(m/2, m), A = \begin{bmatrix} B \\ B \end{bmatrix}$, 如表 3 所示.

其中 $\text{rand}(n, m)$ 表示随即生成 n 行 m 列的矩阵. 解设置为 $x^* = (1, 1, \dots, 1)^T$, 从而生成 $b = A^T x^*$.

算法中的参数设置: 初始点 $x^0 = (0, 0, \dots, 0)^T$, C i m m i n o 算法中, $\lambda_k = 1, \omega_i^k = \frac{1}{q}$.

表 1 好条件线性系统下的迭代步数和 CPU 时间比较

m	n	μ	$K(A)$	Ite-Num	$\ A^T x^k - b\ _2$	CPU T i n e / s	算法
100	80	50	101.8438	170	9.737e-007	0.6710	RA
				251	9.981e-007	0.9620	CM
200	150	50	146.1775	683	9.624e-007	7.5510	RA
				895	9.979e-007	9.7740	CM
300	250	50	320.4410	4054	9.677e-007	102.2170	RA
				6808	9.964e-007	177.4050	CM
400	350	50	514.6338	10215	9.971e-007	467.0820	RA
				16868	9.430e-007	775.9650	CM
800	700	50	690.41	24723	9.9478e-007	2778.52	RA
				34768	9.9976e-007	4734.19	CM

注: RA 为算法 1, CM 为经典 C i m m i n o 算法; μ 为块 A_i 的最多列数; Ite-Num 为迭代步数; $K(A)$ 为矩阵 A 的条件数.

表 2 坏条件线性系统下的迭代步数和 CPU 时间比较

m	n	μ	$K(A)$	Ite-Num	$\ A^T x^k - b\ _2$	CPU T i n e / s	算法
100	100	50	2.695e+12	1	0.066e-007	0.0100	RA
				15	4.685e-007	0.0710	CM
200	200	50	2.544e+12	101	8.081e-007	1.3620	RA
				126	9.483e-007	1.7420	CM
300	300	50	1.179e+13	201	8.931e-007	5.8180	RA
				285	9.753e-007	8.6720	CM
400	400	50	2.953e+13	259	9.877e-007	12.9490	RA
				380	9.868e-007	18.9970	CM
800	800	50	1.620e+14	686	9.948e-007	106.52	RA
				1033	9.9118e-007	170.94	CM

表 3 极坏条件线性系统下的迭代步数和 CPU 时间比较

m	n	μ	$K(A)$	Ite-Num	$\ A^T x^k - b\ _2$	CPU T i n e / s	算法
100	100	50	0.0466e+20	1	0.075e-007	0	RA
				15	4.668e-007	0.0700	CM
200	200	50	1.1872e+20	91	8.199e-007	1.2420	RA
				135	9.619e-007	1.8730	CM
300	300	50	0.1018e+20	177	8.604e-007	5.5680	RA
				258	9.661e-007	7.8010	CM
400	400	50	0.1895e+20	284	9.991e-007	14.1200	RA
				388	9.873e-007	20.2090	CM
800	800	50	1.7187e+19	631	9.887e-007	103.38	RA
				939	9.881e-007	150.47	CM

数值结果显示, 行投影迭代算法对求解坏条件数问题很有效. 和经典 C i m m i n o 算法相比, 算法 1 有效地加快了收敛性.

(下转第 66 页)

- [12] HECTOR POMARES, IGNACIO ROJAS, JESÚS GONZÁLEZ, et al. Structure identification in complete rule-based fuzzy systems[J]. IEEE Transactions on Fuzzy Systems, 2002, 10(3): 349–359.
- [13] JANETTE C. CANO, PATRICIA A. Nava. A fuzzy method for automatic generation of membership function using fuzzy relations from training examples[C] // Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS-FLINT 2002). New Orleans, 2002: 158–162.
- [14] 李洪兴. 变论域自适应模糊控制器[J]. 中国科学(E 辑), 1999, 29(1): 32–42.
LI Hongxing. Self-adaptive fuzzy controller with changing universe of discourse[J]. Science in China (Series E), 1999, 29(1): 32–42 (in Chinese).
- [15] SUDKAMPAT, HAMMELL R J. Interpolation, completion and learning fuzzy rules[J]. IEEE Trans Syst Man Cybern, 1994, 24: 332–342.
- [16] 边肇祺. 模式识别[M]. 北京: 清华大学出版社, 2000: 234–241.
BIAN Zhaoqi. Pattern Recognition[M]. Beijing: Tsinghua University Press, 2000: 234–241. (in Chinese).
- [17] 高新波. 模糊聚类分析及其应用[M]. 西安: 西安电子科技大学出版社, 2004.
GAO Xinbo. Fuzzy Cluster Analysis and Its Applications[M]. Xi'an: Xidian University Press, 2004. (in Chinese).
- [18] KENNEDY J, EBERHART R. Particle swarm optimization[C] // Proceedings of the IEEE International Conference on Neural Networks 1995: 1942–1948.
- [19] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory[C] // Proceedings of the 6th International Symposium on Micro Machine and Human Science Piscataway, Nagoya: IEEE Service Center, 1995: 39–43.
- [20] 刘靖明, 韩丽川, 侯立文. 一种新的聚类算法——粒子群聚类算法[J]. 计算机工程与应用, 2005(20): 183–185.
LIU Jingming, HAN Lichuan, HOU Liven. A new cluster algorithm——particle swarm cluster algorithm[J]. Computer Engineering and Applications, 2005(20): 183–185 (in Chinese).

[责任编辑: 刘 健]

(上接第 51 页)

[参考文献] (References)

- [1] SCOLNICK H, ECHEBEST N, GUARDARUCCI M T, et al. A class of optimized row projection methods for solving large non-symmetric linear systems[J]. Appl Numer Math, 2002, 41(4): 499–513.
- [2] CMMINO G. Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari[J]. Ricerca Sci II 1938, 9: 326–333.
- [3] LIU CHANGWEN. An acceleration scheme for row projection methods[J]. Jour Comp Appl Math, 1995, 57: 363–391.
- [4] AHARONIR, CENSOR Y. Block-iterative projection methods for parallel computation of solutions to convex feasibility problems[J]. Linear Algebra Appl 1989, 120: 165–175.
- [5] GUBN LG, POLYAK B T, RAK E V. The method of projections for finding the common point of convex sets[J]. USSR Comput Math Phys 1967, 7: 1–24.
- [6] FLETCHER R. Practical Methods of Optimization[M]. 2nd ed. New York: Wiley, 1987: 199–202.
- [7] GOLUB G H, VAN LOAN C F. Matrix Computations[M]. 3rd ed. Baltimore: Johns Hopkins University Press, 1996: 223–606.

[责任编辑: 刘 健]