# 三维热传导方程的Krylov子空间方法并行分析

李丹丹, 程汤培, 王 群
(中国地质大学 (北京) 信息工程学院, 北京 100083)

摘 要: 热传导方程在地下水流动数值模拟、油藏数值模拟等工程计算中有着广泛应用, 其并行实现是加速问题求解速度、提高问题求解规模的重要手段, 因此热传导方程的并行求解具有重要意义。对 Krylov子空间方法中的 CG和 GMRES算法进行并行分析, 并对不同的预处理 CG算法作了比较。在 Linux集群系统上, 以三维热传导模型为例进行了数值实验。实验结果表明, CG算法比 GMRES算法更适合建立三维热传导模型的并行求解。此外, CG算法与 BJACOBI预条件子的整合在求解该热传导模型时, 其并行程序具有良好的加速比和效率。因此, 采用 BJACOBI预处理技术的 CG算法是一种较好的求解三维热传导模型的并行方案。

关键词: Krylov子空间方法; 线性方程组; 预条件子; 热传导方程; 共轭梯度算法; 广义极小残量
中图分类号: TP301.6 文献标志码: A 文章编号: 1001-3695(2010)04-1335-04
doi: 10.3969/j.issn.1001-3695.2010.04.035

# Analysis of parallel Krylov subspace method for three-dimensional heat equation

LI Dan-dan, CHENG Tang-pei, WANG Qun
(College of Information Engineering, China University of Geosciences (Beijing), Beijing 100083, China)

**Abstract** Heat equation has been widely used in engineering such as numerical simulation of groundwater flow, reservoir simulation and so on. The parallelism of heat equation is an important means of accelerating the simulation process and enhancing the modeling capabilities. This paper analyzed the parallelism of GMRES and CG algorithm included in Krylov subspace method, made a comparison with different preconditioned conjugate gradient methods. Numerical experiments on the three-dimensional heat equation were carried out on Linux clusters. The numerical results demonstrate that CG algorithm is more suitable than the GMRES algorithm for parallelizing the three-dimensional heat equation. The parallel program has a desirable speedup and efficiency when use CG algorithm integrating with BJACOBI preconditioner to solve the three-dimensional heat equation. So a better parallel solution to the three-dimensional heat equation is CG algorithm integrating with BJACOBI preconditioner.

**Key words** Krylov subspace method; linear equations; preconditioner; heat equation; CG; GMRES

## 0 引言

热传导方程, 简称抛物型方程, 它是根据热量守恒定律和傅里叶热传导实验定律导出。热传导方程在工程实践以及科学计算等领域中应用非常广泛。在科技高度发达的今天, 问题规模的不断扩大, 处理精度要求的不断提高, 使得热传导方程难以求解。借助偏微分方程的数值解法可以将热传导方程转换为稀疏线性代数方程组。大量的实践经验表明, 线性代数方程组的求解, 尤其是稀疏线性代数方程组的求解占整个计算量的 80%以上, 是整个问题计算的瓶颈[1]。因此, 热传导方程求解的核心即为稀疏线性代数方程组的求解。

线性代数方程组的数值解法总体上分为直接法和迭代法两大类。直接法在求解过程中带来大量非零元素, 增加了计算量和存储量, 且其不易并行, 不能满足求解大规模问题的需要, 因此通常使用迭代法[2]。迭代法是指在每个迭代中得到一个原问题的近似解, 并估计近似解与精确解之间的误差, 再基于这个误差构造下一步迭代, 从而改进当前的近似解。Krylov子空间方法是目前应用最广泛的迭代方法, 它具有存储量、计算量少且易于并行等优点, 非常适于大型稀疏线性方程组的并行求解[3]; 而且结合预条件子的 Krylov子空间方法是目前并行求解大型稀疏线性方程组的最主要方法, 因此对 Krylov子空间方法的并行分析是非常重要的。

## 1 三维热传导模型

考虑定义在三维规则区域 $\Omega=(0,M)\times(0,N)\times(0,I)$上的热传导方程:

$$\begin{cases}\dfrac{\partial^2 u}{\partial x^2}+\dfrac{\partial^2 u}{\partial y^2}+\dfrac{\partial^2 u}{\partial z^2}=\dfrac{\partial u}{\partial t} & (x,y,z)\in\Omega,\ t>0\\ u=u^0 & t=0\\ u=g & (x,y,z)\in\partial\Omega,\ t\geqslant 0\end{cases}$$

收稿日期: 2009-07-21; 修回日期: 2009-09-01

作者简介: 李丹丹(1988-), 女, 山西长治人, 硕士研究生, 主要研究方向为并行计算、地下水模拟(xingyunnancy@sina.com); 程汤培(1983-), 男, 博士研究生, 主要研究方向为并行计算、地下水数值模拟等; 王群(1955-), 男, 博导, 主要研究方向为大规模科学计算可视化、中间件技术.

其中: $u = u(x, y, z, t)$为未知函数; $u^0 = u^0(x, y, z)$与 $g = g(x, y, z, t)$为已知函数, 分别定义在区域 Ω 的内部和边界上。

上述三维热传导方程采用有限差分法离散后可归结为大型稀疏线性代数方程组 $Ax = b$, 其中 A 为对称正定且对角占优的大型稀疏矩阵。具体的有限差分格式为时间上采用一阶向后隐式差分, 空间上采用七点格式的二阶中心差分。

## 2 Krylov 子空间方法并行分析

### 2.1 Krylov 子空间方法

Krylov子空间方法近二十年来发展非常迅速, 已占据了大部分的求解代数方程组的应用, 且表现出很好的应用前景。Krylov子空间方法的主要思想可理解为各迭代步递归地构造残差向量的过程。

设要求解的线性代数方程组为 $Ax = b$, 取 $K_m = K_m(A, r^0) = \mathrm{Span}(r^0, Ar^0, \cdots, A^{m-1}r^0)$。其中: $r^0 = b - Ax^0$, 且 $x^0$ 为初始解, 从 $x^0 + K_m$ 中寻找近似解 $x^m$, 使相应的残差向量与另某个子空间 $L_m$ 正交, 即 $r^m = b - Ax^m \perp L_m$[1], 则称 $k_m$ 为 Krylov子空间, 且上述方法称为 Krylov子空间方法。不同的 $L_m$ 可以导出不同的算法, 如共轭梯度(CG)、共轭梯度平方(CGS)及广义极小残量(GMRES)算法等。

Krylov子空间方法中两种较流行的算法为 CG 和 GMRES 算法。CG算法用于对称正定线性方程组的求解, 该算法稳定且解不易发散, 与预处理技术结合, 容易形成稳定、高效的线性方程求解器; 它具有的短迭代计算式, 保证了计算实现的高效性[2]。GMRES算法用于非对称线性方程组的求解, 它具有极小残差性, 但由于随迭代步数的增加, 计算量和存储量都呈线性增长, 从而不具有实现的高效性[1], 常用的 GMRES算法一般为其重新开始的版本 GMRES($m$)。GMRES($m$)算法的收敛性与 $m$ 值密切相关, $m$ 值估计太小, 则降低迭代法的收敛性; $m$ 值估计过高, 则会增加运算量与内存的需求量。但对 GMRES($m$)算法中 $m$ 的取值, 至今没有理论上的结果。

### 2.2 预处理技术

Krylov子空间方法的一个很大优势在于: 其可与预条件子结合使用[4]。一个好的预条件子可以加快迭代法的收敛速度[5], 也可以显著改善迭代法的健全性和可靠性[6], 因此预处理技术受到了广泛的关注, 如 ILU 预处理[7]、ASM 预处理[8]等。此外, 一些研究人员还采用双预处理技术以更好地提高迭代法的收敛性[9]。所谓预处理过程就是把线性系统 $Ax = b$ 变为等价系统 $P^{-1}Ax = P^{-1}b$。其中 P 是 A 在某种程度上的近似, 它应该至少满足以下两个条件: a) $P^{-1}y$ 易于求解; b) $\kappa(P^{-1}A) < \kappa(A)$。即要求 P 的选取应该使预处理后的线性系统比原线性系统更加易于迭代求解。其中 P 就称为预条件子。

预条件子主要包括三类: a)简单预条件子。它的构造方法非常简单, 主要包括 Jacob 预条件子和 SSOR 预条件子。b)完全分解预条件子。这类预条件子主要指不完全分解方法(ILU)和不完全 Cholesky方法(ICC)。c)解析启发式预条件子。其包括区域分解方法(DDM)、Schwarz方法和多水平方法等。

### 2.3 PCG和 PGMRES($m$)的主要计算

CG和 GMRES算法是 Krylov子空间方法中两种非常有效的算法, 它们主要涉及向量内积、矩阵向量相乘两种操作, 易于并行, 因此相应的预处理迭代法一直是科学研究中的热点问题[5,7,10]。下面以 PCG 和 PGMRES($m$)算法为例, 说明并行 Krylov子空间方法的主要计算。

#### 2.3.1 预处理共轭梯度法(PCG)

PCG 算法的主要计算在于迭代循环, 具体的计算步骤如下:

a)选取 $x^0$ 为初始值, 计算 $r^0 = f - Ax^0$, $p^0 = z^0 = P^{-1}r^0$。

b) for $k = 0, 1, \cdots$直到收敛, 计算(迭代循环—并行):

$$w^k = Ap^k \tag{1}$$

$$\alpha_k = \frac{(z^k, r^k)}{(p^k, w^k)} \tag{2}$$

$$x^{k+1} = x^k + \alpha_k p^k \tag{3}$$

$$r^{k+1} = r^k - \alpha_k w^k \tag{4}$$

$$Pz^{k+1} = r^{k+1} \tag{5}$$

$$\beta_{k+1} = \frac{(z^{k+1}, r^{k+1})}{(z^k, r^k)} \tag{6}$$

$$p^{k+1} = z^{k+1} + \beta_{k+1}p^k \tag{7}$$

#### 2.3.2 预处理广义极小残量法(PGMRES($m$))

GMRES算法的主要计算在于 Arnoldi迭代循环, PGMRES($m$)并行算法可描述如下:

a)初始化(并行)。选择 $x_0$, 计算 $r = b - Ax_0$, 求解 $Pw = r$ 并计算 $v_1 = w / \| w \|_2$。

b) Amoldi迭代(并行)。迭代 $m$ 步, $j = 1, 2, \cdots, m$。具体的 Amoldi算法如下:

$$\text{for } i = 1, \cdots, m$$

$$Pw = Av_i$$

$$\text{for } k = 1, \cdots, i \tag{8}$$

$$h_{k\,i} = (w, v_k)$$

$$w = w - h_{k\,i}v_k \tag{9}$$

$$\text{end } k$$

$$h_{i+1,\,i} = \| w \|_2$$

$$v_{j+1} = \frac{W}{h_{i+1,\,i}} \tag{10}$$

$$\text{end } i \tag{11}$$

c)计算近似解(并行) $x^m = x^0 + Q_{n,m}y_m$。其中: $y_m$ 极小化 $\| \beta\xi - H_{m,m}y \|_2$; $Q_{n,k}$ 是 $n \times k$ 矩阵, 矩阵的列向量由正交基向量 $v_1, v_2 \cdots, v_k$ 构成, $H_{mm}$ 是 Hessenberg矩阵。

d)判断。计算 $r^m = b - Ax^m$, 若满足条件, 则停止, 否则令 $x^0 = x^m$, 返回 a)重新计算。

### 2.4 PCG与 PGMRES($m$)算法的并行化

从上面两种预处理迭代法的求解过程可以看出, Krylov子空间方法的主要计算包括四部分:

a)矩阵向量相乘(如 PCG算法中式(1)与 PGMRES($m$)算法中式(8)的计算)。

b)向量内积(如 PCG算法中式(2)(6)与 PGMRES($m$)算法中式(9)的计算)。

c)求解预处理方程组 (如 PCG 算法中式 ( 5)与 PGMRES ( *m*)算法中步骤 a)的计算 ), P 是 A 的一个近似。

d)向量更新 (如 PCG算法中式 ( 3) ( 7)与 PGMRES( *m* )算法中式 ( 10) ( 11)的计算 )。

在并行求解的过程中, 矩阵向量相乘与向量内积这两部分的操作都是分散在各个处理机上并行执行的。对于分布式内存的并行计算机, 各个处理单元都拥有自己独立的局部存储器, 各处理机之间通过消息传递来交换信息。因此 Krylov子空间方法中主要的通信有:

a)计算矩阵向量相乘时, 处理机间需要通过传递消息来交换数据, 但由于矩阵的带状稀疏性, 通常只涉及到相邻处理机之间的通信。

b)在计算向量内积时, 向量先被分散到各个处理机中, 然后各处理机并行地计算局部内积; 最后各处理机把各个子内积从自身传到某个处理机中进行求和。在传递子内积的过程中就必然会涉及到数据的通信。

c)在对向量进行更新时, 需要从某一处理器广播参数 ( PCG算法中需要广播的参数是 $a_k$ 和 $\beta_{k+1}$, PGMRES( *m* )算法中需要广播的参数是 $h_{k,i}$和 $h_{i+1,i}$)到其他处理器时的通信等。

处理机间的通信将会影响程序的并行性能。为了减少并行计算过程中的通信量, 可以将不需要通信的计算与通信重叠进行, 从而屏蔽通信时间。以 PCG 算法为例, 可以将上一次迭代中的向量更新操作 $x^k = x^{k-1} + \alpha_{k-1} p^{k-1}$延迟到式 ( 2), 与内积计算同时进行, 以屏蔽内积 $(z^k, r^k)$和 $(p^k, w^k)$所带来的通信开销。

## 3　数值实验

实验的硬件环境是由百兆以太网相连的四个刀片组成的并行 Linux集群, 每个刀片上的配置为 Xeon 3. 0( 2M cache) CPU, 2 GB容量的 DDR2主存。

实验 1　GMRES( *m* )算法的收敛性与 *m* 值的选取密不可分, *m* 值选取适当, 可以显著加快算法的收敛速率。在该实验中, 选取 $m = 5 \rightarrow 40$, CG 和 GMRES( *m* )算法求解网格规模为 $100 \times 100 \times 96$的三维热传导模型的计算时间、残差及迭代次数如表 1所示。

表 1　各迭代法下计算时间、残差及迭代次数

| 迭代方法 | 计算时间 /s | 残差 | 迭代次数 |
|---|---|---|---|
| CG | 282. 671 | 0. 056 527 | 126 |
| GMRES( 5) | 523. 592 | 0. 147 695 | 269 |
| GMRES( 10) | 445. 541 | 0. 113 225 | 180 |
| GMRES( 15) | 513. 194 | 0. 121 192 | 155 |
| GMRES( 20) | 580. 808 | 0. 115 214 | 143 |
| GMRES( 25) | 596. 540 | 0. 099 124 | 137 |
| GMRES( 30) | 720. 586 | 0. 094 784 | 134 |
| GMRES( 35) | 762. 075 | 0. 004 580 | 130 |
| GMRES( 40) | 842. 456 | 0. 104 377 | 127 |

从表 1中的数据可以看出, 随着 *m* 值的增大, GMRES( *m* )算法的迭代次数减少, 即迭代法的收敛速度提高; 但由于 *m* 值的增大, 使得 GMRES( *m* )算法的计算量以及内存需求量增加, 整个热传导模型的求解时间不一定是 *m* 的减函数。从表中的数据也可以看出, 当 *m* 的值为 10时, GMRES( *m* )算法的求解时间最短, 但其计算时间仍比 CG算法的求解时间长。

对于大型问题的并行计算, 有两个问题是最被关注的: 一个是对给定的问题规模、计算时间的长短; 另一个是对给定的问题规模及处理机数, 该并行程序是否具有较好的加速比。因此测试在不同处理机数下, 分别采用 GMRES( 10)算法和 CG算法求解网格规模为 $100 \times 100 \times 96$的三维热传导模型的加速比, 如图 1所示。

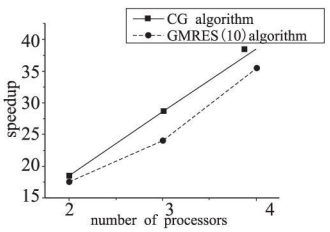


图1　GMRES(10)算法和CG算法在不同处理机数下的加速比

从图 1中可以看出, 在给定问题规模和处理机数时, CG算法有更好的加速比。因此, 无论从计算时间还是从并行性能来看, CG算法都比 GMRES算法更适合本文热传导模型的并行求解。

实验 2　表 1中的数据显示, 单纯依靠迭代法来求解大型线性方程组所需要的计算时间较长。预条件子可以加快迭代法的收敛速度, 但预条件子的加速效果与求解问题所形成的线性方程组密不可分。因此, 为了更好地提高迭代法收敛性, 将 CG算法与下列预条件子进行整合: 不使用预条件子; BJACOBI 预条件子; ASM 预条件子。问题规模为 $100 \times 100 \times 96$的三维热传导模型采用上述三种求解方案后的部分测试结果如表 2所示。

表 2　不同预处理 CG算法在不同处理机数下的计算时间、残差及迭代次数

| 预条件子 | 进程数 | 计算时间 /s | 残差 | 迭代次数 |
|---|---|---|---|---|
| NONE | 1 | 282. 671 | 0. 056 527 | 126 |
| | 2 | 152. 308 | 0. 056 527 | 129 |
| | 3 | 99. 689 | 0. 056 527 | 131 |
| ASM | 1 | 109. 475 | 0. 029 653 | 41 |
| | 2 | 59. 979 | 0. 034 819 | 41 |
| | 3 | 40. 948 | 0. 037 166 | 41 |
| BJACOBI | 1 | 103. 546 | 0. 029 653 | 41 |
| | 2 | 55. 669 | 0. 038 980 | 44 |
| | 3 | 38. 144 | 0. 044 313 | 46 |

表 2中的数据显示, 两种类型的预条件子均可以很好地提高迭代法的收敛性, 减少迭代次数, 缩短计算时间。但整合了 BJACOBI预条件子的 CG算法的计算时间最短。分析其原因在于, 一方面 BJACOBI预条件子易于执行, 且适合于求解矩阵最大元素位于对角线上的线性方程组; 另一方面在于 BJACOBI 预条件子构造方法最为简单, 它是由系数矩阵的对角矩阵给出, 在构造的过程中无须引入非零元素, 因此 BJACOBI预条件子无须矩阵以外的额外存储空间, 对内存的需求很低。

由表 2中的数据还可以看出, 随着处理机数的增加, 迭代次数也有增多的趋势。这是因为问题规模一定时, 参与计算的处理机数增加, 使得每个处理机所分配的问题区域减小。但随着求解区域的变小, 预条件子加速收敛的作用将弱化, 从而使得求解线性方程组的迭代次数增加。

结合实验 1和 2的测试结果,可以得出一个重要的结论,求解本文三维热传导模型较好的并行求解方案为 CG算法和 BJACOBI预条件子的整合。

实验 3 将 CG算法与 BJACOBI预条件子进行整合,依次在处理机数为 1、2、3、4的 Linux集群上,对网格规模为 100×100×96的三维热传导模型进行测试。部分测试结果如表 3所示。

表 3 不同处理机数下的计算时间、加速比及效率

| 处理机数 | 计算时间 /s | 加速比 | 效率 /% |
|---|---|---|---|
| 1 | 103.546 | 1 | 100 |
| 2 | 55.669 | 1.86 | 93.00 |
| 3 | 38.144 | 2.78 | 92.67 |
| 4 | 30.454 | 3.40 | 90.00 |

表 3中的数据显示,对于四个节点组成的并行计算系统,加速比是处理机数的增函数,而效率是处理机数的减函数。其原因在于问题规模确定时,串行运行时间 $T_s$ 为定值,并行计算时间 $T_p$ 随处理机数 $p$ 的增加而缩短,而并行总开销 $T_0$ 随处理机数 $p$ 的增加而增加。由加速比的定义表达式 $S=T_s/T_p$ 以及效率的等价表达式 $E=pT_s/P(T_0+T_s)=T_s(T_0+T_s)=1/(1+T_0/T_s)$ 可以推出,问题规模确定时,加速比随处理机数的增加而增加,效率随处理机数的增加而降低。

实验 4 在处理机数为 2的 Linux集群上对网格规模从 100×100×36增大至 100×100×96的三维热传导模型进行测试,该实验的部分测试结果如表 4所示。

表 4 不同网格规模下计算时间及效率

| 网格规模 | 计算时间 /s | 效率 /% |
|---|---|---|
| 100×100×36 | 20.261 | 88.00 |
| 100×100×48 | 26.424 | 89.50 |
| 100×100×60 | 33.282 | 91.00 |
| 100×100×72 | 41.156 | 91.50 |
| 100×100×84 | 48.735 | 92.50 |
| 100×100×96 | 55.669 | 93.00 |

从表 4可以看出,并行计算的效率是问题规模的增函数。这是因为,一般而言问题规模增加时,计算时间的增长要比通信时间的增长快,这意味着并行开销与串行运行时间的比值即 $T_0/T_s$ 会随着问题规模的增加而减少。由效率的表达式 $E=pT_s/p(T_0+T_s)=T_s(T_0+T_s)=1/(1+T_0/T_s)$ ($T_s$ 表示指定问题规模的串行运行时间,$T_0$ 表示系统的总并行开销,$P$ 表示处理机的个数)推导可知,并行效率会随着问题规模的增加而增加。

## 4 结束语

随着科学技术的迅猛发展,求解问题规模愈加庞大,导致热传导方程的求解非常困难,因而大规模热传导方程求解已成为科学研究和工程计算等领域的一个难题。本文通过对 Krylov子空间方法中的 CG算法和 GMRES($m$)算法进行并行分析,并对不同预处理的共轭梯度法进行比较。在数值实验的基础上,找到了一种适合基于有限差分法建立的三维热传导模型的并行求解方案,且该并行程序具有良好的并行性能。热传导方程的并行实现,使得大规模热传导方程的求解不再是科学上的难题,从而使其更好地应用于工程实践以及科学计算领域。

参考文献:


[1] 李晓梅,吴建平. Krylov子空间方法及其并行计算[J]. 计算机科学, 2005, 32(1): 19-20.

[2] 李晓梅,迟利华. 并行求解大型稀疏线性方程组的研究概况[J]. 指挥技术学院学报, 1999, 10(3): 1-8.

[3] 杭旭登. 偏微分方程迭代并行解法与网格优化方法[D]. 成都: 中国工程物理研究院, 2003.

[4] OLIVERIA S, DENG Yuan-hua. Preconditioned Krylov subpace methods for transport equations[J]. Progress in Nuclear Energy, 1998, 33(1-2): 155-174.

[5] SUNDAR S, BHAGAVAN B K, SASTRIK S. Comparison of Lanczos and CGS solvers for solving numerical heat transfer problems[J]. Computers & Mathematics with Applications, 1999, 37(8): 107-117.

[6] ZHANG JUN. Preconditioned Krylov subspace methods for solving nonsymmetric matrices from CFD applications[J]. Computer Methods in Applied Mechanics and Engineering, 2000, 189(3): 825-840.

[7] MITTAL R C, AL-KURDI A H. An efficient method for constructing an ILU preconditioner for solving large sparse nonsymmetric linear systems by the GMRES method[J]. Computers & Mathematics with Applications, 2003, 45(10-11): 1757-1772.

[8] SARKISM, SZYLD D B. Optimal left and right additive Schwarz preconditioning for minimal residual methods with Euclidean and energy norms[J]. Computer Methods in Applied Mechanics and Engineering, 2007, 196(8): 1612-1621.

[9] LIN H W, CHEN L D. Application of the Krylov subspace methods to numerical heat transfer[J]. Numerical Heat Transfer, 1996, 30(3): 249-270.

[10] LALOS A S, BERBERIDIS K. An efficient conjugate gradient method in the frequency domain: application to channel equalization[J]. Signal Processing, 2008, 88(1): 99-116.


(上接第 1310页)


[2] 周立美,张立卫,贺素香. 求解非线性互补问题的微分方程方法[J]. 运筹学学报, 2005, 9(3): 10-16.

[3] 刘水霞,陈国庆. $P_0$-函数箱约束变分不等式的正则半光滑牛顿法[J]. 高等学校计算数学学报, 2006, 28(2): 111-121.

[4] 屈彪,王长珏,张树霞. 一种求解非线性互补问题的方法及其收敛性[J]. 计算数学, 2006, 28(3): 247-258.

[5] STORN R, PRICE K. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces[R]. Berkeley: University of California, 2006.

[6] LAMPINEN J. A bibliography of differential evolution algorithm[EB/OL]. (2002-10-14). http://www.lut.fi/~jlampine/debiblio.htm.

[7] LIN Y C, HWANG K S, WANG Feng-sheng. Co-evolutionary hybrid differential evolution for mixed-integer optimization problems[J]. Engineering Optimization, 2001, 33(6): 663-682.

[8] CHENG S L, HWANG C. Optimal approximation of linear systems by a differential evolution algorithm[J]. IEEE Trans on Systems, Man and Cybernetics, Part A, 2001, 31(6): 698-707.

[9] 赵晓颖,刘志国,姜凤利. 求解一类不可微优化问题的极大熵微粒群混合算法[J]. 江西师范大学学报, 2007, 31(2): 193-196.

[10] 刘淳安. 非线性规划问题的极大熵多目标粒子群算法[J]. 计算机工程与设计, 2008, 29(4): 914-916.

[11] 周丽美. 依赖凝聚函数求解一类非线性互补问题的一种微分方程解法[J]. 数学的实践与认识, 2006, 36(2): 238-243.