

文章编号: 1001-0920(2006)12-1369-05

基于增强型kick策略的LS算法求解一类聚类问题

罗家祥, 唐立新, 田志波

(东北大学 信息科学与工程学院, 沈阳 110004)

摘要: 提出一种新型的基于环交换邻域的迭代局部搜索算法(LS), 用于求解一类聚类问题。算法的主要特点是: 1) 基于环交换的邻域结构: 环交换邻域与传统的Swap和Insert邻域相比, 算法在一次迭代中允许多个点同时移动; 2) 针对聚类问题提出了增强型的kick移动策略: 根据每组内点的密度分布摄动聚类中心, 对给定的解重新聚类。实验结果表明, 基于环交换的迭代局部搜索算法对求解该类聚类问题是有效的。

关键词: 聚类问题; LS算法; 增强型kick策略; 环交换邻域

中图分类号: TP18 **文献标识码:** A

New Iterated Local Search with Reinforced Kick Strategy for Clustering Problems

LUO Jia-xiang, TANG Li-xin, TIAN Zhi-bo

(College of Information Science and Engineering, Northeastern University, Shenyang 110004, China. Correspondent: LUO Jia-xiang, E-mail: fbhlljx@yahoo.com.cn)

Abstract: A new iterated local search (LS) algorithm with cycle exchange neighborhood is developed to solve a class of clustering problems. The main characteristics of the algorithm are as follows: 1) Cycle exchange neighborhood is used where several points are allowed to move simultaneously in an iteration, which is different from traditional neighborhoods such as swap and insert where only at most two points are permitted to move simultaneously in each iteration; 2) a new "reinforced kick" is proposed for the clustering problem, in which for a given solution, the center point of each cluster is updated according to the point distribution in the cluster, and the remained points are re-grouped based on these centers. Computational experiments show the efficiency of the new LS algorithm with cycle exchange.

Key words: Clustering problem; LS algorithm; Reinforced kick strategy; Cycle exchange neighborhood

1 引言

成组技术^[1]是计算机集成制造系统(CIMS)的一个重要研究方向。它有利于缩减中间物流, 对于提高批处理类型生产的效率具有重要作用。成组技术在单元制造中的一个有效的应用是单元成组, 其基本问题是将机器和零件分组, 使得零件在机器上的移动达到最小。

单元成组问题可归结为一类聚类问题。该类聚类问题可用一个0-1变量的二次规划模型来描述^[2]。

Dikin证明了一般带线性约束的二次规划(QP)为NP完备问题^[3]。当把变量的范围限制为0-1整数时, 问题的难度增加, 也为NP完备问题^[4]。解决聚类问题的方法主要有: k means算法及其改进方法^[5]、遗传算法^[6]、禁忌搜索算法^[7]等。Sultan等^[8]的研究表明, 在所研究的聚类问题中, 禁忌搜索算法具有明显的优势。

本文提出了基于环交换邻域的迭代局部搜索(LS)算法, 用于求解该类聚类问题。LS算法的特

收稿日期: 2005-11-03; 修回日期: 2006-01-14

基金项目: 国家自然科学基金项目(60674084, 60274049); 国家杰出青年科学基金项目(70425003); 高等学校优秀青年教师教学科研奖励计划项目[2002]383。

作者简介: 罗家祥(1979-), 女, 四川自贡人, 博士生, 从事物流优化、智能优化的研究; 唐立新(1966-), 男, 黑龙江绥化人, 教授, 博士生导师, 从事物流优化、生产调度等研究。

点是: 当搜索陷入局部最优解时, 通过对当前局部最优解实施预定的随机移动, 从而获得新的初始解, 继续进行迭代. 由于新解继承了前期好解的部分特性, 避免了搜索的随机性. 这种方法最早用来求解 TSP 问题^[9], 近年来的研究^[10]表明, LS 算法对许多 NP 难题的求解具有良好的效果.

影响 LS 算法的主要因素有: 邻域的构造、局域搜索算法和随机移动 (kick) 策略. 邻域的好坏直接影响搜索方法的性能. 对于组合优化问题, 传统的邻域如交换和插入, 只允许一个点或两个点发生移动. 针对所研究问题和传统邻域的这一缺陷, 本文采用大规模的环交换^[11]邻域, 其特点是允许多个点同时发生移动.

传统的 kick 移动策略是预订好的随机移动, 缺点是没有导向性. 本文根据聚类问题的特点, 提出了增强型的 kick 移动策略, 即扰动当前解中各组的聚类中心, 再按聚类最小原则, 将其余的点聚类到各中心, 产生新的搜索初始解. 文中介绍了单元成组归结为聚类问题的二次规划数学模型, 阐述了采用增强型的 kick 移动策略和环交换邻域的 LS 算法解决聚类问题, 最后给出了实验结果和算法分析.

2 单元成组的二次规划模型

Kusiak 等^[11]提出了单元成组问题的二次规划模型. 以矩阵 $D = \{d_{ij}\} \in R^{n \times n}$ 表示零件在处理过程中的相似性, 目标是将这 n 个零件划分成 m 个零件簇, 使得组内任意两个零件之间的距离之和最小. 即

$$\text{Min} \sum_{k=1}^m \sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} x_{ik} x_{jk}, \quad k = 1, 2, \dots, m. \quad (1)$$

$$\text{s.t.} \quad x_{ik} = 1, \quad i = 1, 2, \dots, n, \quad k = 1 \quad (2)$$

$$x_{ik} = 0 \text{ or } 1, \quad i = 0, 1, \dots, n. \quad (3)$$

约束 (2) 表明每个零件只能划分到一个零件簇中; 约束 (3) 设定变量的取值, 表明零件 i 是否在零件簇 k 中. 目标函数是最小化零件簇中任意两个零件之间的距离之和. 若将零件视为聚类的点, 零件簇视为聚类的集合, 则问题转化为最小化集合中任意两点之间的距离之和的聚类问题.

3 带增强型 kick 移动策略和环交换邻域的迭代局部搜索算法

LS 算法用于解决多种优化问题. 当局域搜索陷入局部最优解时, 将当前局部最优解或历史最好解进行 kick 移动至一个新的初始解, 使得局域搜索继续进行. 这样可不损失已找到局部最优解或历史最好解的良好特性. 算法的搜索结果为搜索过程中得到的历史最好解. LS 算法的关键在于邻域的构造、局域搜索算法和 kick 移动策略.

3.1 环交换邻域

将环交换应用于成组技术, 可描述为零件或零件集合以环状的方式在零件簇中同时移动. 环交换最简单的情况是每个零件簇中最多有一个零件移动. 设零件集合 $A = \{1, 2, \dots, n\}$, 将 A 分为 m 个零件簇 $S = \{I^1, I^2, \dots, I^m\}$, $I^1 \cup I^2 \cup \dots \cup I^m = A, I^k \cap I^l = \emptyset, \forall k, l \in \{1, 2, \dots, m\}, k \neq l$. 零件簇 I 的费用为 $C(I) = \sum_{i,j \in I} d(i, j)$. 则整个聚类问题的目标函数为最小化 $\sum_k C(I^k), k = 1, 2, \dots, m$. S 的环交换邻域定义如下:

定义 1 设零件簇序列 $(I^{\sigma(1)}, I^{\sigma(2)}, \dots, I^{\sigma(r)})$. 其中: $I^{\sigma(k)}$ 表示序列中第 k 个位置的零件簇, $r = m$. 满足如下条件: 1) $I^{\sigma(k)} \cap I^{\sigma(l)} = \emptyset, k \neq l, k, l = 1, 2, \dots, r$; 2) $\forall I^k \in S, k = 1, 2, \dots, r$. 如果零件序列 (i^1, i^2, \dots, i^r) 作如下移动:

$$I^{\sigma(k)} = \begin{cases} I^{\sigma(k)} - \{i^k\} + \{i^{k-1}\}, & 2 \leq k \leq r; \\ I^{\sigma(k)} - \{i^k\} + \{i^r\}, & k = 1. \end{cases} \quad (4)$$

则得到新分类 $\{I^1, I^2, \dots, I^m\}$ 为 S 的环交换邻域的邻解. 零件所作的环状移动称为环交换. 可使目标函数减小的环交换称为负费用环交换. 环交换实例如图 1 所示.

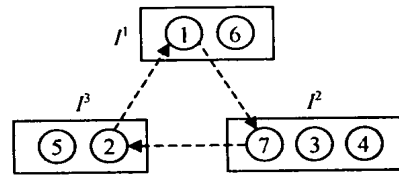


图 1 环交换实例

在图 1 中, 零件簇序列 (I^1, I^2, I^3) 中的零件 $(1, 7, 2)$ 作如下移动: 零件 1 从 I^1 移到 I^2 , 零件 7 从 I^2 移到 I^3 , 零件 2 从 I^3 移到 I^1 , 构成环状交换. 环交换后得到的新的零件分类 $\{I^1 = \{2, 6\}, I^2 = \{1, 3, 4\}, I^3 = \{5, 7\}\}$, 为原可行分类 S 的环交换邻域中的一个解, 而所有可通过环交换移动得到的解集为 S 的环交换邻域.

Thompson 等^[11]证明了搜索负费用环交换是 NP 难问题. 本文在缩减的环交换邻域内构造基于动态规划的环交换邻域搜索方法.

3.1.1 动态规划搜索负费用环交换

由于零件与点、零件簇与集合一一对应, 在以下描述中, 用点和集合的概念进行相关描述. 设在 S 的集合序列 $(I^{\sigma(1)}, I^{\sigma(2)}, \dots, I^{\sigma(r)})$ 中搜索负费用环. 为便于对动态规划的描述, 在序列中加入 $I^{\sigma(r+1)} = I^{\sigma(1)}$. 显然, 要想构成环, 必然要求动态规划搜索的首尾节点相同. 设初始点为 $s (s \in I^{\sigma(1)})$, 末节点为 t .

$= s(t \quad I^{\sigma(r+1)})$, 则动态规划方程如下:

$$\begin{cases} f_1^s(s) = 0, k = 1; \\ f_k^s(i^k) = C(I^{\sigma(k)} + \{s\} - \{i^k\}) - C(I^{\sigma(k)} + f_1^s(s), \\ \quad \forall i^k \in I^{\sigma(k)}, k = 2; \\ f_k^s(i^k) = \min_{i^{k-1} \in I^{\sigma(k-1)}} \{C(I^{\sigma(k)} + \{i^{k-1}\} - \{i^k\}) - \\ \quad C(I^{\sigma(k)} + f_{k-1}^s(i^{k-1}))\}, \\ \quad \forall i^k \in I^{\sigma(k)}, 3 \leq k \leq r; \\ f_k^s(t) = \min_{i^{k-1} \in I^{\sigma(k-1)}} \{C(I^{\sigma(k)} + \{i^{k-1}\} - \{t\}) - \\ \quad C(I^{\sigma(k)} + f_{k-1}^s(i^{k-1}))\}, \\ \quad k = r + 1. \end{cases} \quad (5)$$

显然, 若 $\min_{s \in I^{\sigma(1)}, t \in I^{\sigma(r+1)}, s=t} \{f_{r+1}^s(t)\} < 0$, 则经过回溯, 动态规划可搜索到削减的环交换邻域中最好的负费用环. 实施环交换后, 目标函数的改进量为环交换的负费用的绝对值.

3.1.2 基于动态规划的两种环交换邻域搜索策略

策略 1 上节所述的动态规划方法, 在遍历了集合 I^1 中的所有点后停止搜索. 与这种方法不同的是用动态规划方法找到一个负费用环后停止计算. 该方法称为基于集合的近似动态规划方法.

策略 2 上述两种搜索算法存在的缺点是不能保证所有点都参与环交换. 为此, 本文提出如下搜索方法: 分别以每个点为初始搜索点, 将包含该点的集合作为初始搜索集合 (搜索序列中的其他集合顺序随机生成), 用前述动态规划方法搜索负费用环. 该策略称为基于点的近似动态规划搜索策略.

3.1.3 加入虚拟零件的环交换邻域

对于当前解 s , 在每个集合中加入一个虚拟点. 这些虚拟点对集合费用不产生任何影响, 但在动态规划算法搜索邻域时, 作为一个真正的点参与环交换, 从而增大了动态规划或近似动态规划的搜索空间. 加入虚拟零件的环交换如图 2 所示, 其中 $\{-1\}$ 表示虚拟点.

加入虚拟点的环交换, 可使一些点插入到另一

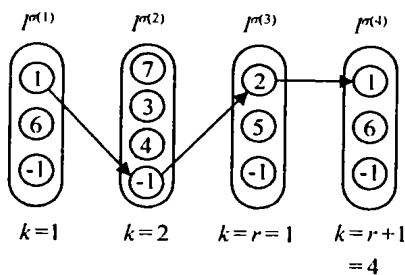


图 2 加入虚拟零件的环交换

个集合中, 从而避免了环交换中必须一对一的点置换, 增大了邻域搜索空间. 在下面的环交换邻域的搜索中, 每个集合都加入了虚拟点来扩大搜索空间.

3.2 局域搜索算法

根据上节提出的 3 种环交换邻域搜索策略, 局域搜索算法如下:

步骤 1: 搜索当前解 s 的环交换邻域, 得到解 S .

步骤 2: 若 S 优于 s , 则令 $s = S$, 重复邻域搜索; 否则, 转步骤 3.

步骤 3: 1) 对于动态规划搜索和搜索策略 1, 改变集合的动态规划搜索顺序, 返回步骤 1. 若连续 3 次迭代仍未改进历史最优解, 则局域搜索算法结束.

2) 对于搜索策略 2, 搜索下一个点. 若全部点搜索完毕, 则局域搜索算法结束.

3.3 kick 移动策略

LS 算法中的 kick 策略是对某局部最优解进行移动, 试图突破当前搜索空间, 使局域搜索算法逃出局部最优解, 从而提高算法的全局搜索能力. 然而, 当 kick 过大时, 得到的解具有随机性, 对局部最优解特性的继承性不强, 不能充分挖掘局部最优解的可改进能力; 当 kick 过小时, 搜索常会陷入相同的局部最优解, 逃出局优的能力不强.

本文根据 LS 的传统思想, 构造了一般的随机 kick 移动策略. 从两个未曾改变的集合中随机挑选两个零件实施 Swap 交换, 重复此动作, 直到最多只有一个未发生改变的集合为止.

随机 kick 移动策略的缺点是没有导向性, 经过移动后得到的解空间可能很差. 为弥补这一不足, 根据聚类问题的特点, 提出了增强型的 kick 移动策略. 对于当前局部最优解 $S = (I^1, I^2, \dots, I^m)$, 用 $d(i, j)$ 表示点 i 与 j 之间的距离, 构造的增强型 kick 策略如下:

1) 计算各集合 $I^k (k = 1, 2, \dots, m)$ 的中心

$$c_k = \operatorname{arg} \min_{i \in I^k} \min_{j \in I^k} d(i, j). \quad (6)$$

2) 根据最小距离原则, 其他点聚向中心点, 构成新的聚类.

各集合的中心点由当前解 s 计算得出, 因此必然继承了 s 的某些特性. 显然, s 越接近于最优解, 中心点改变的可能性越小, 新解对 s 的继承性越大. 同时对点进行重新聚类, 有可能改变当前的搜索空间.

4 实验结果和算法分析

上节介绍了局域搜索算法中采用 3 种不同的环交换邻域搜索策略, 并提出 2 种 kick 移动策略. 本文

通过实验计算对 kick 移动策略和环交换搜索策略进行选择, 目的在于比较算法中的可选择因素, 建立一种解决聚类问题的有效算法

本文的算法程序采用 C++ 编程, 并在 P4-2.4 PC 机上运行. 为验证算法的有效性, 针对所研究的聚类问题, 共产生 15 种规模的 150 组数据. 聚类的点数在 20~100 之间, 聚类的集合数在 3~5 之间, 点之间的距离服从 [1, 100] 均匀分布的随机数. 由于不知问题的下界或最优解, 本文用 c/c^* 衡量算法的性能, 其中 c^* 为几种算法中的最好结果. 显然, c/c^* 的值越接近于 1, 算法越好.

增强型 kick 移动策略与一般随机 kick 移动策略的比较结果如表 1 所示. 3 种不同的环交换搜索策略构成的局域搜索算法的比较结果如表 2 所示.

表 1 增强型 kick 移动策略与一般随机 kick 移动策略的比较

规模 $n \times m$	LS(增强型 kick 策略)		LS(随机 kick 策略)	
	算法性能	平均运行时间/s	算法性能	平均运行时间/s
12 × 3	1.023 0	0.41	1.000 0	0.54
12 × 5	1.008 2	0.27	1.000 0	0.33
30 × 5	1.006 6	5.57	1.019 2	7.70
40 × 5	1.003 6	22.80	1.019 2	21.39
30 × 10	1.034 3	2.10	1.048 5	2.79
40 × 10	1.007 9	6.25	1.067 9	7.82
50 × 10	1.004 9	16.14	1.054 8	13.30
40 × 15	1.007 4	3.37	1.071 6	5.17
50 × 30	1.030 2	2.17	1.084 8	2.97
100 × 40	1.015 2	14.39	1.155 1	18.40
平均	1.014 1		1.052 1	

从表 1 可以看出, 当采用相同的局域搜索算法时, 增强型 kick 移动策略的 LS 算法比一般 kick 移动策略的 LS 算法在较短的平均时间内获得了更好的计算结果, 说明增强型 kick 移动策略具有一定的

表 2 几种环交换搜索策略的比较

规模 $n \times m$	LS- 动态规划		LS- 策略 1		LS- 策略 2	
	算法性能	平均运行时间/s	算法性能	平均运行时间/s	算法性能	平均运行时间/s
20 × 3	1.009 7	0.93	1.008 6	0.68	1.009 9	1.84
40 × 3	1.006 5	12.67	1.004 5	10.50	1.005 4	17.20
60 × 3	1.006 1	90.69	1.009 9	62.87	1.001 9	76.75
80 × 3	1.007 7	307.62	1.005 0	194.92	1.000 5	225.60
100 × 3	1.006 9	926.15	1.005 0	516.07	1.000 8	604.72
20 × 4	1.017 3	0.70	1.012 3	0.52	1.022 6	1.29
40 × 4	1.022 8	9.77	1.013 0	7.10	1.006 0	16.52
60 × 4	1.015 8	55.47	1.013 5	41.24	1.003 9	70.67
80 × 4	1.009 5	224.82	1.015 3	147.17	1.001 6	200.27
100 × 4	1.018 7	527.07	1.011 7	395.97	1.002 2	544.61
20 × 5	1.039 2	0.50	1.021 1	0.38	1.041 3	1.08
40 × 5	1.018 0	7.47	1.015 5	6.75	1.013 5	10.69
60 × 5	1.016 4	38.18	1.017 0	33.38	1.004 6	43.75
80 × 5	1.025 9	140.43	1.011 9	123.55	1.002 9	166.18
100 × 5	1.017 9	374.68	1.008 3	300.56	1.002 3	344.03
平均	1.015 9	181.14	1.011 5	122.78	1.008 0	155.01

优势. 从表 2 可以看出, 在相同条件下, 3 种搜索环交换搜索策略对 LS 算法的性能影响差别不大. 本文选择算法性能较好的基于点的近似动态规划搜索策略搜索负费用环.

表 3 LS 算法与禁忌搜索算法的性能比较

规模 $n \times m$	LS		Tabu	
	算法性能	平均运行时间/s	算法性能	平均运行时间/s
20 × 3	1.000 0	1.84	1.016 0	0.55
40 × 3	1.000 0	17.20	1.030 4	1.32
60 × 3	1.000 0	76.75	1.053 1	2.28
80 × 3	1.000 0	225.60	1.060 0	3.39
100 × 3	1.000 0	604.72	1.065 6	4.77
20 × 4	1.000 0	1.29	1.027 1	0.52
40 × 4	1.000 0	16.52	1.077 2	1.19
60 × 4	1.000 0	70.67	1.093 0	2.03
80 × 4	1.000 0	200.27	1.109 6	2.98
100 × 4	1.000 0	544.61	1.111 2	4.15
20 × 5	1.000 0	1.08	1.074 5	0.51
40 × 5	1.000 0	10.69	1.117 6	1.12
60 × 5	1.000 0	43.75	1.133 1	1.86
80 × 5	1.000 0	166.18	1.155 8	2.72
100 × 5	1.000 0	344.03	1.159 3	3.72
平均	1.000 0	155.01	1.085 6	2.21

LS 算法与禁忌搜索算法^[7] 的比较结果如表 3 所示. 禁忌搜索算法的参数设置与文献 [7] 中一致. 根据表 3 可得出以下结论:

- 1) 基于问题性质提出的增强性 kick 策略, 在一定程度上提高了 LS 算法在解决聚类问题时的全局搜索能力.
- 2) 尽管 LS 算法的运行时间较长, 但从算法的最终结果看, LS 算法比禁忌搜索算法提高了 7.88%, 说明环交换邻域的 LS 算法在解决聚类问题时具有一定的优越性.
- 3) 当每个集合中的点数增加时, 程序的运行时间明显增长, 如图 3 所示. 这是由于动态规划实质为隐枚举方法, 随着问题规模的增大, 消耗的时间会越来越长.

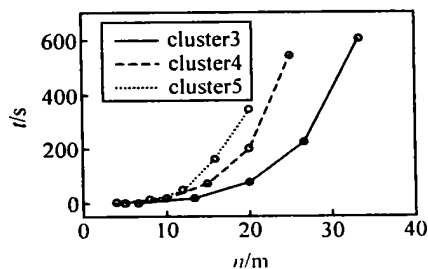


图 3 运行时间随规模的变化

5 结 论

本文研究一类聚类问题, 该问题已被证明为 NP 难问题. 文中提出用基于环交换的邻域迭代局部搜

索算法求解该问题 根据聚类问题的特点, 提出了增强型的kick 策略 实验结果表明, 采用环交换邻域和增强型kick 策略的LS 算法, 能在实际可容忍的时间内获得满意的聚类

参考文献(References)

- [1] Kusiak A, Cheng C H. Group Technology: A Analysis of Selected Models and Algorithms[A]. *Design, A nalysis and Control of M anufacturing Cells* [C]. Atlanta: American Society of Mechanical Engineers, 1991: 99-114
- [2] Chu C H. Cluster Analysis in Manufacturing Cellular Formation[J]. *Int J of M anagement Science*, 1989, 17(3): 289-295
- [3] Sahni S, Gonzales T. P-complete Approximation problems[J]. *J of American S tatistic A ssociation*, 1976, 23(2): 555-565
- [4] Turki A, Yajima Y, Enkawa T. A New Composite Algorithm for Clustering Problems[J]. *Int T ransfer Operatinal Research*, 1996, 3(2): 197-206
- [5] Chan E Y, Ching W K, Ng M K, et al An Optimization Algorithm for Clustering Using Weighted Dissimilarity Measures[J]. *Pattern Recognition*, 2004, 37(5): 943-952
- [6] Cowgill M C, Harvey R J, Watson L T. A Genetic Algorithm Approach to Cluster Analysis[J]. *Computers and M athematics w ith Applications*, 1999, 37(7): 99-108
- [7] Sultan K S. A Tabu Search Approach to the Clustering Problem [J]. *Patter Recognition*, 1995, 28(9): 1443-1454
- [8] Sultan K S, Maroof K M. Computational Experience on Four Algorithms for the Hard Clustering Problem [J]. *Pattern Pecognition Letters*, 1996, 17(3): 295-308
- [9] Baum E B. Towards Practical "Neural" Computation for Combinatorial Optimization Problems: Neural Networks for Computing[A]. *Proc A IP Conf* [C]. New York, 1986: 53-58
- [10] Lourenco H R, Martin O, St üzle T. A Beginner's Introduction to Iterated Local Search[A]. *M IC '2001-4th M etaheuristics Int Conf* [C]. Porto, 2001: 1-6
- [11] Thompson P M, Orlin J B. *The Theory of Cyclic T ransfers*[R]. Cambridge: Operations Research Center, MIT, 1989
- [2] Richalet J. Model Predictive Heuristic Control: Applications to Industrial Processes[J]. *A utomatica*, 1978, 14(5): 413-428
- [3] Rouhani R, Mehra R K. Model Algorithmic Control (MAC): Basic Theoretical Properties[J]. *A utomatica*, 1982, 18(4): 401-414
- [4] Culter C R, Ramaker B L. Dynamic Matrix Control: A Computer Control Algorithm [A]. *Proc of Joint A utomatic Control Conf* [C]. San Francisco, 1980
- [5] Clarke D W, Mohtadi C, Tuffs P S. Generalized Predictive Control[J]. *A utomatica*, 1987, 23(1): 137-160
- [6] Ahmed M S. Neural-net-based Direct Adaptive Control for a Class of Nonlinear Plant[J]. *IEEE T rans on Neural Networks*, 2000, 45(1): 671-677.
- [7] Haddad W M, Hayakawa T, Chellaboina V, et al Robust Adaptive Control for Nonlinear Uncertain Systems[J]. *A utomatica*, 2003, 39(1): 551-556
- [8] 王伟. 广义预测控制理论及其应用[M]. 北京: 科学出版社, 1998
(Wang W. *Theory and Application of Generalized Predictive Control*[M]. Beijing: Science Press, 1998)
- [9] Clarke D W, Mohtadi C, Tuffs P S. Properties of Generalized Predictive Control [J]. *A utomatica*, 1989, 25(6): 859-875
- [10] 席裕庚. 预测控制[M]. 北京: 国防工业出版社, 1993
(Xi Y G. *Predictive Control*[M]. Beijing: National Defense Industry Press, 1993)
- [11] Rani K Y, Unbehauen H. Study of Predictive Controller Tuning Methods [J]. *A utomatica*, 1997, 33(12): 2243-2248
- [12] Norgaard M, Ravn O, Poulsen N K, et al *Neural Networks for Modelling and Control of Dynamic Systems*[M]. London: Springer-Verlag, 2000

(上接第1368 页)