

一种带有摄动的导向性蚁群算法

段 汐 杨 群 陈 兵 李媛祯

(南京航空航天大学计算机科学与技术学院 南京 210016)

摘 要 针对加入导向性局部搜索(Guided Local Search, GLS)的蚁群算法(Ant Colony Optimization, ACO)容易过早收敛的问题,提出一种带有摄动的导向性蚁群算法(Perturbation Guided Ant Colony Optimization, PGACO),该算法在当前解表现出过早收敛的趋势时,采用摄动(Perturbation)方式干扰解构建过程,使当前解移动到其邻域空间,从而产生一个新的可行解来避免算法过早收敛,提高算法求解的精度。实验结果表明,PGACO 能有效地改善过早收敛问题,获得更优的可行解和执行速度,同时具有更强的全局搜索能力,能进一步提高算法的性能。

关键词 蚁群算法,导向性局部搜索,摄动

中图分类号 TP181 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.12.032

Perturbation Guided Ant Colony Optimization

DUAN Xi YANG Qun CHEN Bing LI Yuan-zhen

(Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract The hybridizations of Ant Colony Optimization(ACO) with Guided Local Search(GLS) can be used to solve the problem that ACO is easily trapped in local optima. However, there is a problem that the algorithmic optima prematurely converges to suboptimal solutions. This paper presented a Perturbation Guided Ant Colony Optimization(PGACO) algorithm to avoid the problem. A proposed perturbation method is used to move current solution to a neighbor solution space to build new global optimal solutions when the algorithm is prone to premature convergence. The experimental results show that PGACO can effectively avoid a premature convergence of the algorithm to suboptimal solutions. PGACO can generate a better solution, simultaneously has a better global search capability.

Keywords Ant colony optimization, Guided local search, Perturbation

1 引言

蚁群算法(Ant Colony Optimization, ACO)由意大利学者 Marco Dorigo 于 1992 年提出^[1],是一种模拟自然界中蚂蚁集体寻径行为并广受关注的仿生算法。算法采用一种特殊的正反馈机制,即在实际路径搜索过程中,后续蚂蚁根据先出发蚂蚁于路径上留下的信息素浓度高低决定路径选择,因使用更短路径的蚂蚁将在路径上释放更多信息素,所以后续搜索能利用早期积累的信息不断地优化求解结果。该机制能对外界影响做出动态响应,具有很强的鲁棒性,但算法搜索时间长,且容易陷入局部最优解^[2]。

近年来,国内外学者针对蚁群算法容易陷入局部最优解的问题做了广泛深入的研究和改进。Hani 等^[3]在蚁群算法中加入导向性局部搜索,通过动态调整目标函数跳出局部最优解。Zhao 等^[4]将遗传算法的变异部分引入蚁群算法来避免算法陷入局部最优解。Marinakakis 等^[5]在蚁群算法中加入贪婪随机自适应搜索来解决陷入局部最优解的问题。其中,导向性局部搜索的方法因具有独特的目标函数修改机制,而

得到众多学者的关注,并逐渐应用于各种领域之中。

但是上述算法存在容易过早收敛的问题,极大地影响了其对解空间的探索和求解的精度。针对此缺陷,本文提出一种带有摄动的导向性蚁群算法,它利用导向性局部搜索在搜索过程中积累的导向信息形成一种新的摄动方式,当算法显示出过早收敛的趋势时,通过摄动干扰算法的收敛,提供一个不同于当前方法生成的可行解,有效提高了算法求解的精确度,增强了算法的全局搜索能力。

2 相关工作

导向性局部搜索(Guided Local Search, GLS)是一种成功用于组合优化问题求解的启发式智能搜索算法^[6]。与其它启发式算法类似,导向性局部搜索能作为一个通用算法框架,通过与其它算法结合来提高算法的鲁棒性和启发式搜索能力^[7]。该算法的主要思想是使用与搜索相关的信息来指导带有启发式特征的局部搜索向探索不充分的解空间移动,具有较高的抽象度和通用性。算法求解过程可根据问题灵活地改变,如改变惩罚因子修改规则、采用可变换特征代价等,使算

到稿日期:2013-06-25 返修日期:2013-08-16

段 汐(1989—),女,硕士生,主要研究方向为并行计算、进化算法、无线网络, E-mail: duanxi6@126.com; 杨 群(1971—),女,博士,副教授,主要研究方向为软件方法学、并行计算; 陈 兵(1970—),男,博士,教授,主要研究方向为计算机网络、无线网络、网络安全; 李媛祯(1990—),女,硕士生,主要研究方向为并行计算、群智能算法的优化和应用。

法更符合求解需要。

近年来,导向性局部搜索因简洁紧凑、应用效果明显而逐渐被学者关注并引入蚁群算法求解之中。例如,Hani等^[3,8]在蚁群算法中加入导向性局部搜索来处理设备布局问题和产业规划问题。Chehade等^[9]使用混合的导向性局部搜索蚁群算法来处理装配生产线的设计问题。邢立宁等^[10]使用导向性局部搜索蚁群算法处理卫星地面站系统任务调度问题。

虽然加入导向性局部搜索的蚁群算法^[3,8-10]确实能提高算法求解的效果,生成较蚁群算法更优的可行解,但是由于导向性局部搜索每次只修改效用值较高的特征解所对应的惩罚因子,以致惩罚因子的改变对目标函数的影响减小。若目标函数在一段时间内持续选择同一个解为最优解,则算法收敛,因此目标函数的选择是算法收敛的关键。在逼近最优解的过程中,很可能出现一个被当前目标函数选中的次优解,惩罚因子的修改不再能改变目标函数在一段时间内的选择,以致该次优解不停地被目标函数选择为当前最优解,导致算法过早收敛。

基于上述阐述,本文提出带有摄动的导向性蚁群算法来处理导向性局部搜索蚁群算法容易过早收敛的问题。

本文第3节主要阐述导向性局部搜索蚁群算法的相关内容;第4节引入摄动的概念,提出一种带有摄动的导向性蚁群算法;第5节介绍实验及数据分析;最后是结束语。

3 导向性局部搜索蚁群算法

导向性局部搜索蚁群算法在蚁群算法中使用惩罚因子,通过惩罚因子动态地调整目标函数,改变下一次迭代中最优解的选取过程,从而改进算法陷入局部最优解的不足,是一种有效的启发式算法。

导向性局部搜索的一个主要特征是迭代使用局部搜索,同时综合搜索空间的信息来指导局部搜索过程向可能产生最优解的区域移动^[11]。

加入导向性局部搜索后的蚁群算法目标函数定义如下:

$$h(s) = g(s) + \lambda \cdot \sum_{r=1}^M p_r \cdot I_r(s) \quad (1)$$

其中, $g(s)$ 表示蚁群算法初始目标函数, p_r 表示路径 s 中路段 r 上附加的惩罚因子, M 是路段的个数, λ 是惩罚因子影响力参数, $I_r(s)$ 是一个指示函数,表示当路段 r 存在于路径 s 中时其值为1,反之为0。

定义1 GN_{\max} 为局部搜索停滞次数上限,当局部搜索停滞次数超过该值时,认为算法陷入一个局部最优解。

当算法陷入一个局部最优解 s' 时,算法使用效用值函数计算每一个路段的效用值,效用值较高的路段将受惩罚,惩罚因子 p_r 随之增加。该函数定义如下,其中, c_r 代表路段 r 的代价:

$$util(s', r) = I_r(s') \cdot \frac{c_r}{1 + p_r} \quad (2)$$

由上文可知,在蚁群算法中加入导向性局部搜索,能有效避免蚁群算法容易陷入局部最优解的问题。但仍存在过早收敛至非最优解的缺陷,解质量有待进一步提高,针对该缺陷,本文提出一种带有摄动的导向性蚁群算法,该算法在导向性局部搜索蚁群算法的基础上引入摄动,结合惩罚因子共同作用于下次迭代的节点选择过程,能有效避免过早收敛至非最优解,从而提高算法的全局搜索能力。

4 带有摄动的导向性蚁群算法 PGACO

4.1 摄动

摄动(Perturbation)是一类移动解至其邻域的方法,又称扰动。它能简单而有效地产生一个不同于当前方法的新解,因而被越来越多的学者应用于各个领域。

现有摄动方法^[12-14]多采用随机产生法或影响信息素更新的过程来产生新解,没有充分利用构建解过程中正反馈的启发式特性,导致算法在搜索效果和搜索质量上都有很大的损失,不利于提高蚁群算法的效率。因此,本文提出一种带有摄动的导向性蚁群算法(Perturbation Guided Ant Colony Optimization, PGACO),此算法设计一种新的摄动方法来构建解,不但可以更为充分地利用蚁群算法的正反馈特性,而且因将启发式信息与惩罚因子相关联共同作用于蚂蚁构建解的过程,从而能准确地体现路径长短和惩罚因子在全局寻优中的作用,为探索更多解空间和寻求最优解提供了好的基础,有效提高了算法的全局搜索能力。

4.2 PGACO

4.2.1 PGACO 摄动

PGACO的摄动方法如下:

$$j_k = \begin{cases} \arg \max_{l \in N_i^k} \{\tau_{il} \cdot [\eta_{il}]^{d \cdot \beta}\}, & d = e^p, \text{ if } q \leq q_0 \\ J_i^k, & \text{ else} \end{cases} \quad (3)$$

其中, j_k 表示蚂蚁 k 选择的下一节点, p_r 表示路段 r 上的惩罚因子,特别地, r 表示由节点 i 到节点 l 的路段, N_i^k 表示蚂蚁 k 在节点 i 上可选的邻域范围, τ_{il} 代表节点 i 到节点 l 上的信息素, $\eta_{il} = 1/d_{il}$ 代表节点 i 到节点 l 上的启发式信息, d_{il} 代表节点 i 到节点 l 的路径长度, β 代表启发式信息的影响力因子, q 是一个随机变量, q_0 表示一个预先给定的阈值, J_i^k 表示蚂蚁 k 的可选择邻域中与节点 i 最邻近的节点。

蚂蚁构建路径时维护一个禁忌表存放当前已经访问过的城市节点。在选择下一节点前先获得一个随机值 q ,若 $q \leq q_0$,则选择节点时选择邻域中不存在于禁忌表且使 $\tau_{il} \cdot [\eta_{il}]^{d \cdot \beta}$ 值最大的节点;若 $q > q_0$,则选择邻域中不存在于禁忌表中且最近的节点。

4.2.2 过早收敛次数计算

定义2 CS_{\max} 为能够容忍的最大无更优可行解的次数,使用一个计数器 CS 存储无更优可行解的持续迭代次数。当本次迭代中 CS 等于 CS_{\max} 时,使用式(3)作为状态转移函数选择下次迭代的路径节点。 CS 计算规则如式(4)所示。

$$CS(t+1) = \begin{cases} 0, & \text{if current solution is better than last iteration's} \\ \min(CS(t)+1, CS_{\max}), & \text{else} \end{cases} \quad (4)$$

4.2.3 PGACO 算法过程

带有摄动的导向性蚁群算法过程如下:

Step 1 初始化参数,城市个数 N ,蚂蚁数量 M ,迭代次数上限 NC_{\max} ,无更优可行解次数最大值 CS_{\max} ,局部搜索停滞次数上限 GN_{\max} 。随机分配 M 只蚂蚁的初始位置并添加到禁忌表中。

Step 2 如果 CS 不等于 CS_{\max} ,则根据如下路径选择规则^[1]选择节点,否则根据式(3)选择路径节点。各蚂蚁逐步选

择节点形成完整路径。

路径选择规则如下：

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, & \text{if } j \in N_i^k \\ 0, & \text{else} \end{cases} \quad (5)$$

其中, p_{ij}^k 代表位于节点 i 的蚂蚁 k 选择节点 j 的概率, α 代表信息素的影响力因子。

Step 3 根据式(1)从 M 只蚂蚁构建的路径中选择最优路径, 该路径为局部最优解。将其与全局最优解比较, 更新全局最优解, 根据如下信息素更新规则^[1]更新信息素矩阵:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}^{best} \quad (6)$$

其中, ρ 是信息素的蒸发率, $\Delta\tau_{ij}^{best} = 1/C^{best}$, C^{best} 是至今最优路径长。

Step 4 若当前局部最优解与上次迭代的局部最优解相同, 则局部搜索停滞次数加 1; 否则为 0。根据式(4)更新无更优可行解次数 CS 。

Step 5 若局部搜索停滞次数等于 GN_{max} , 根据式(2)计算效用值并更新惩罚因子矩阵。

Step 6 若当前迭代次数 t 小于迭代次数上限 NC_{max} , 则转 Step 2; 否则输出最优解, 终止程序退出。

5 实验与结果分析

5.1 实验环境及参数设置

本文采用 TSPLIB 中对称 TSP 问题的部分实例进行实验, 实例名称的数字代表了实例的规模。实验软件环境为 VC++ .NET。本文中 ACO 主要参数设置参照文献[2], GLS 主要参数设置参照文献[11], $M=100, \alpha=1, \beta=2, \rho=0.2, \lambda=5, NC_{max}=1000, GN_{max}=10, CS_{max}=20, q_0=0.98$ 。其中, M 为蚂蚁个数, α 为信息素影响因子, β 为启发式信息影响因子, ρ 为信息素蒸发率, λ 为目标函数中惩罚因子的影响力参数, q_0 为式(3)中预先设置的随机概率。 NC_{max} 为迭代次数上限, GN_{max} 为局部搜索停滞次数上限, CS_{max} 为无更优可行解次数最大值。信息素大小限制在 $[\tau_{min}, \tau_{max}]$ 之间, 其中, 设 $\tau_{max} = \frac{1}{\rho \cdot C^{best}}, \tau_{min} = \frac{\tau_{max} \cdot (1 - \sqrt[0.05]{0.05})}{avg \cdot \sqrt[0.05]{0.05}}$, C^{best} 为最优路径长度的估计值, avg 为蚂蚁构建解时遍历的城市平均数, 信息素初始值等于信息素上限。

5.2 实验与结果分析

为了更为直观地观察本文带有摄动的导向性蚁群算法 PGACO 的执行效果, 选用最大最小蚁群算法 MMAS 以及文献[3]中用导向性局部搜索改进了的蚁群算法 ACO_GLS 共同进行比较, 实验数据取 7 组不同规模的问题实例, 重复执行 20 次后的平均值。计算结果如表 1 所列, 时间单位: 秒。

由表 1 可以看出, ACO_GLS 算法使用 GLS 处理 ACO 容易陷入局部最优解, 求解质量较 MMAS 更好, 但结合后的算法总执行时间较长, 影响求解效率。本文算法 PGACO 在 ACO_GLS 的基础上引入摄动优化求解结果, 减少了所花费的时间, 不论实例规模如何变化, 都能生成更优的可行解, 使求解质量进一步提高, 且有效改善了加入 GLS 后带来的容易过早收敛至次优解的问题。虽然 PGACO 收敛到最优解的时间比 ACO_GLS 长, 但 PGACO 收敛的平均迭代次数明显多于 ACO_GLS, 从总执行时间上看, PGACO 执行时间比 ACO_GLS 快, 时效性较好, 算法执行速度更快。

表 1 不同实例规模下的算法求解结果

实例名	算法	最优解	平均解	平均收敛次数	平均收敛时间	总时间
kroA100	PGACO	21288.6	21306.8	848.9	66.7	77.9
	ACO_GLS	21303.2	21417.6	48.9	6.2	82.0
	MMAS	21337.1	21489.2	74.3	5.4	71.4
d198	PGACO	15783.0	15798.2	430.6	34.8	85.1
	ACO_GLS	15802.6	15849.3	98.3	15.9	97.2
	MMAS	15832.5	15865.6	124.3	11.7	79.1
lin318	PGACO	42039.6	42091.3	753.0	140.1	187.4
	ACO_GLS	42128.0	42230.4	89.2	36.7	228.8
	MMAS	42228.9	42450.7	135.1	24.8	169.4
pcb442	PGACO	50785.3	50832.8	721.5	284.6	392.6
	ACO_GLS	50896.1	50934.5	98.4	46.7	434.7
	MMAS	50952.7	50973.1	110.6	40.6	358.5
rat783	PGACO	8845.9	8872.8	902.4	674.0	724.8
	ACO_GLS	9055.6	9132.7	128.6	113.8	817.3
	MMAS	9173.3	9269.5	142.2	105.2	711.7
dl291	PGACO	50851.3	50904.9	522.7	938.8	1796.2
	ACO_GLS	51037.5	51084.2	185.4	361.9	1952.6
	MMAS	51398.6	51504.7	250.3	431.8	1722.4
fl1577	PGACO	22292.7	22385.4	896.7	1843.4	2035.3
	ACO_GLS	22558.9	22642.1	317.3	760.1	2395.7
	MMAS	23092.1	23125.4	388.6	752.7	1937.4

蚂蚁数量增加则算法的全局搜索能力加强, 但同时也带来额外计算开销, 当蚂蚁数量等于城市规模时, 算法求解的结果较好^[2]。因此, 为了更为明确地看出算法的收敛情况, 选择城市规模等于蚂蚁数量的实例 kroA100 分别进行实验, 取 3 种算法的一次执行过程。

图 1 所示为使用 PGACO、ACO_GLS 和 MMAS 求解问题实例 kroA100 的进化过程。对比可见, ACO_GLS 收敛最快, 约 60 次迭代即收敛, 其求得的解优于 MMAS 而劣于 PGACO。MMAS 约 80 次迭代后收敛至其最优解, 相较于前两者, PGACO 在整个迭代过程中能不断地搜索并构建新的可行解, 直至 850 代附近才收敛, 且解质量明显更高, 具有更强的全局搜索能力。因此, 算法加入摄动后, 当趋于过早收敛时能有效地干扰收敛过程, 移动解至其邻域中, 加深了解空间的探索, 有利于生成更优的可行解。

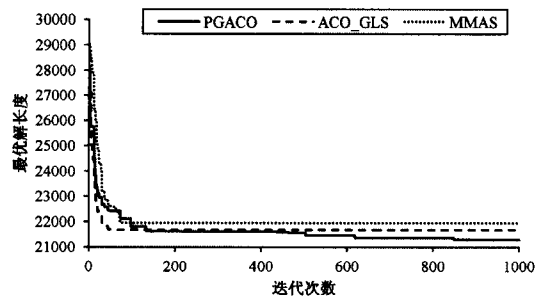


图 1 实例 kroA100 使用 3 种算法求解的全局最优解进化过程

由图 1 可见, PGACO 与 ACO_GLS 在 50 代之前的进化较为类似, 这是因为 PGACO 是在加入 GLS 的 ACO 基础上, 为了改善其容易过早收敛至非最优解的问题, 进一步提高生成解质量而提出的, 所以两种算法的基础一致, 而前期进化过程中并没有收敛的趋势, 因此没有使用摄动, 一旦符合过早收敛的判断条件, PGACO 将使用摄动方法使当前解向其邻域空间移动, 并利用惩罚因子和蚁群算法的正反馈特性指导蚂蚁构建解路径, 优化搜索过程, 生成更优的可行解。

结束语 加入导向性局部搜索的蚁群算法能通过动态修改目标函数来处理蚁群算法陷入局部最优解的不足, 但混合

后的算法容易过早收敛,造成算法在执行初期便无法再产生更优的可行解,以致于算法对解空间探索不充分,给搜索结果和质量带来很大的损失。针对该问题,本文提出一种带有摄动的导向性蚁群算法 PGACO,并设计了一种新的摄动方法。这种方法能够充分利用导向性局部搜索中惩罚因子在寻优过程积累的导向信息以及蚁群算法的正反馈特性指导搜索过程,当算法存在过早收敛的趋势时,通过在启发式信息上加入惩罚因子的影响来优化解构建的过程,使当前解移动到相邻的解空间中,以搜索更多的可能解。实验结果表明,与最大最小蚁群算法以及文献[3]中加入导向性局部搜索的蚁群算法相比,本文算法改善了过早收敛的问题,有效提升了算法性能,使其具有更强的全局搜索能力、更高的解质量,同时有更快的执行速度,其总体求解时间优于加入导向性局部搜索的蚁群算法,时效性较好。

参 考 文 献

[1] Dorigo M. Optimization, Learning and Nature Algorithms[D]. Milan: Dipartimento di Elettronica, Politecnico di Milano, 1992

[2] Dorigo M, Birattari M, Stützle T. Ant colony optimization[J]. Computational Intelligence Magazine, IEEE, 2006, 1(4): 28-39

[3] Hani Y, Amodeo L, Yalaoui F, et al. Hybrid optimization method for the facility layout problem[M]// Swarm Intelligence: Focus on Ant and Particle Swarm Optimization, 2007: 331-342

[4] Zhao N, Wu Z, Zhao Y, et al. Ant colony optimization algorithm with mutation mechanism and its applications[J]. Expert Systems with Applications, 2010, 37(7): 4805-4810

[5] Marinakis Y, Marinaki M, Doumpos M, et al. A hybrid ACO-GRASP algorithm for clustering analysis[J]. Annals of Opera-

tions Research, 2011, 188(1): 343-358

[6] Tairan N, Zhang Q. Population-based guided local search: Some preliminary experimental results[C]// 2010 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2010: 1-5

[7] Vansteenwegen P, Souffriau W, Berghe G V, et al. A guided local search metaheuristic for the team orienteering problem[J]. European Journal of Operational Research, 2009, 196(1): 118-127

[8] Hani Y, Amodeo L, Yalaoui F, et al. Ant colony optimization for solving an industrial layout problem[J]. European Journal of Operational Research. 2007, 183: 633-642

[9] Chehade H, Yalaoui F, Amodeo L, et al. Ant colony optimization for assembly lines design problem[C]// Proceedings of the 8th International FLINS'08 Conference on Computational Intelligence in Decision and Control. Madrid, 2008: 1135-1140

[10] 邢立宁, 陈英武. 基于混合蚁群优化的卫星地面站系统任务调度方法[J]. 自动化学报, 2008, 34(4): 414-418

[11] Voudouris C, Tsang E P K, Alsheddy A. Guided local search [M]. Springer US, 2010

[12] Stützle T, López-Ibáñez M, Pellegrini P, et al. Parameter adaptation in ant colony optimization [M]. Autonomous Search. Springer Berlin Heidelberg, 2012: 191-215

[13] Tsutsui S, Fujimoto N. Parallel ant colony optimization algorithm on a multi-core processor [M]. Swarm Intelligence. Springer Berlin Heidelberg, 2010: 488-495

[14] Dan Z, Hongyan H, Yu H. The Optimal Selection of the Parameters for the Ant Colony Algorithm with Small-Perturbation[C]// 2010 International Conference on Computing, Control and Industrial Engineering (CCIE). IEEE, 2010. 2: 16-19

(上接第 150 页)

步的工作是考虑直接从原数据集中利用非参数方法(如核密度估计方法)估计连续值属性的概率密度函数,然后计算条件属性之间的互信息,以度量条件属性之间的相关性,并研究本文算法在离散情况下得到的结论是否依然成立。

参 考 文 献

[1] Pawlak Z. Rough Sets [J]. International Journal of Computer and Information Science, 1982, 11(5): 341-356

[2] 苗夺谦, 李道国. 粗糙集理论、算法与应用 [M]. 北京: 清华大学出版社, 2008

[3] Skowron A, Rauszer C. The discernibility matrices and functions in information systems [C]// Slowinski R, ed. Intelligent Decision Support, Handbook of Applications and Advances of the Rough Sets Theory. Kluwer, Dordrecht, 1992

[4] 苗夺谦, 王珏. 粗糙集理论中知识粗糙性与信息熵关系的讨论 [J]. 模式识别与人工智能, 1998, 11(1): 34-40

[5] Qian Y, Liang J, Pedrycz W, et al. Positive approximation: an accelerator for attribute reduction in rough set theory [J]. Artificial Intelligence, 2010, 174(9/10): 597-618

[6] Chen D, Zhao S, Zhang L, et al. Sample Pair Selection for Attribute Reduction with Rough Set [J]. IEEE Transactions on Knowledge and Data Engineering, 2012, 24(11): 2080-2093

[7] Hu Q, Xie Z, Yu D. Hybrid attribute reduction based on a novel fuzzy-rough model and information granulation [J]. Pattern Recognition, 2007, 40(12): 3509-3521

[8] Tsang E, Chen D, Yeung D. Attributes reduction using fuzzy rough sets [J]. IEEE Transactions on fuzzy system, 2008, 16

(5): 1130-1141

[9] Yao Y Y, Zhao Y. Discernibility matrix simplification for constructing attribute reducts [J]. Information Sciences, 2009, 179(5): 867-882

[10] Wang J, Wang J. Reduction algorithms based on discernibility matrix; the ordered attributes method [J]. Journal of Computer Science and Technology, 2001, 16(6): 489-504

[11] Qian Y, Liang J. Combination Entropy & Combination Granulation in Rough Set Theory [J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2008, 16(2): 179-193

[12] 王钰, 刘三阳, 刘振华. 相容 RS 理论中的属性约简算法 [J]. 计算机科学, 2003, 30(2): 161-162

[13] Parthala N, Shen Q. Exploring the boundary region of tolerance rough sets for feature selection [J]. Pattern Recognition, 2009, 42(5): 655-667

[14] Jensen R, Shen Q. New Approaches to Fuzzy-Rough Feature Selection [J]. IEEE Transactions on Fuzzy Systems, 2009, 17(4): 824-838

[15] Yao Y Y, Zhao Y. Attribute reduction in decision-theoretic rough set models [J]. Information Sciences, 2008, 178(17): 3356-3373

[16] Cover T M, Thomas J A. Elements of information theory (Second Edition) [M]. Hoboken: Wiley-Interscience, 2006

[17] Blake C L, Merz C J. UCI Repository of Machine Learning Databases[OL]. [2011]. <http://www.ics.uci.edu/~mllearn/ML-Repository.html>

[18] Mitchell T M. Machine Learning [M]. 北京: 机械工业出版社, 2003: 55-73