

Multi-Agent 协同进化算法研究

周铁军, 李 阳

(中南林业科技大学现代教育技术中心, 长沙 410004)

摘要: 与传统优化方法相比, 进化计算具有内在的并行性和自组织、自适应、自学习等智能特征, 它在许多领域显示出巨大优势并取得一定成功。研究 Multi-Agent 协同进化算法, 集成现有算法中的几种优势策略, 利用混合策略的思想结合具体问题设计算法, 并以实例说明该算法的有效性。

关键词: 多智能体; 进化算法; 蚁群算法

Research on Multi-Agent Co-evolutionary Algorithm

ZHOU Tie-jun, LI Yang

(Modern Education Technology Center, Central South University of Forestry and Technology, Changsha 410004)

【Abstract】 In comparison with traditional optimization methods, evolutionary computation due to its intrinsic parallelism and some intelligent properties, such as self-organizing, self-adaptation, and self-learning. Evolutionary computation has been successfully applied to many fields. This paper proposes a Multi-Agent co-evolutionary algorithm, designs a new hybrid algorithm by combining evolutionary algorithm with some field-special strategies, and proves their efficiency by several experiments.

【Key words】 Multi-Agent; evolutionary algorithm; Ant Colony Algorithm(ACA)

进化算法通常有 2 个基本步骤: (1) 每个个体通过变异生成一个后代; (2) 来自于父代和后代种群中较好的方案(个体)被选择为下一代种群的父代个体。协同进化算法基于传统进化算法, 将单一种群推广到 2 个或多个种群, 建立个体或种群之间的竞争、合作关系, 使其适应复杂系统的动态进化环境, 以达到种群优化的目的。文献[1]提出通用合作协同进化算法框架, 该算法的主要思想是以一般进化算法框架为基础, 将种群分为若干子种群, 再进一步考虑子种群之间基于合作关系的协同进化。本文基于此提出一种 Multi-Agent 协同进化算法(Multi-Agent Co-evolutionary Algorithm, MACA)。

1 MACA算法的详细设计

Multi-Agent 协同进化算法的优化框架如图 1 所示。

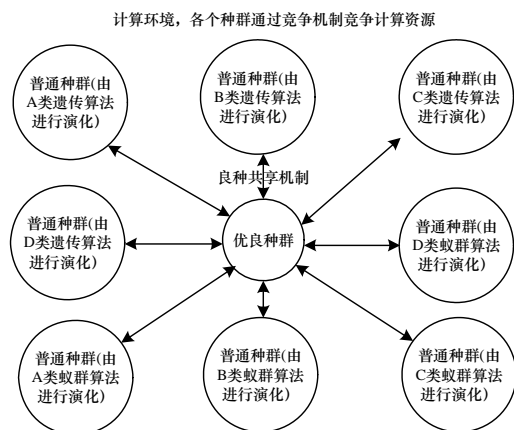


图 1 Multi-Agent 协同进化算法的优化框架

其中, 每个种群就对应于一个智能体, 每类进化算法对应采用不同参数设置的进化算法。A 类、B 类、C 类和 D 类

遗传算法分别代表 4 种不同参数组合的遗传算法。A 类、B 类、C 类和 D 类蚁群算法也分别代表 4 种不同参数组合的蚁群算法。在 Multi-Agent 协同进化算法中, 主要通过各种不同的遗传算法和蚁群算法来完成各个普通种群的进化。Multi-Agent 协同进化算法的具体执行流程见图 2。

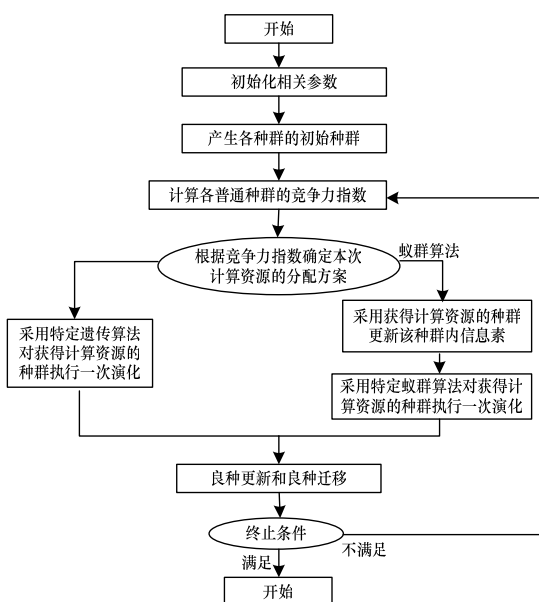


图 2 Multi-Agent 协同进化算法的执行流程

作者简介: 周铁军(1956—), 男, 教授, 主研方向: 数据挖掘, 网络安全; 李 阳, 硕士研究生

收稿日期: 2008-11-02 **E-mail:** lxyzhou@163.com

2 MACA算法的具体实现

遗传算法是求解组合优化问题的一种有效工具^[2-3],应用遗传算法求解组合优化问题,通常只能得到一个局部优化方案。文献[4]提出一种多群蚁群方法,通过不断地增加蚁群,直到搜索到该优化问题的最优解。基于智能体的计算通常被认为是求解复杂分布式问题的最有前途的方法,鉴于此,本文拟采用 Multi-Agent 协同进化算法来求解旅行商问题。

2.1 蚁群算法的实现

本文蚁群算法的优化流程如图 3 所示。

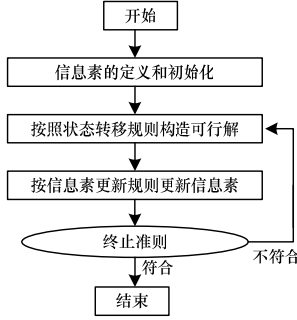


图 3 蚁群算法的执行过程

本文分别通过局部更新规则、全局更新规则和信息素挥发规则来逐步完成对信息素的更新。

(1)信息素的定义和初始化。在应用蚁群算法时,一项非常重要的事务就是如何定义方案(解)空间上的信息素非常重要。笔者将给定节点安排在其他节点后的概率定义为方案空间上的信息素。将信息素定义为一个矩阵 K , 它的维度为 $n \times n$, n 代表所有节点的数目。对于任意的元素 $K[i][j]$, 它的含义是将节点 i 安排到节点 j 后的概率。在本阶段中,信息素矩阵 K 中的元素都被初始化为 λ_0 (一个相对较小的数值)。

(2)状态转移规则。状态转移规则如式(1)所示。位于当前节点 i 的蚂蚁采用状态转移规则来选择下一步要达到的节点 j 。假设将第 k 只蚂蚁还没有周游到的城市集定义为 $allowed_k$, t 时刻节点 i 和节点 j 之间的信息素为 $\tau_{ij}(t)$, 节点 i 和节点 j 之间距离的倒数作为该线路的启发值, 即 $\eta_{ij} = 1/d_{ij}$ 。这样, 第 k 只蚂蚁在当前节点 i 按式(1)的概率来选择下一步要达到的节点 j :

$$P_{ij}^k(t) = \begin{cases} 1 & q \leq q_0 \text{ \& } j = \arg(\min_{p \in allowed_k} d_{ip}) \\ \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{p \in allowed_k} [\tau_{ip}(t)]^\alpha [\eta_{ip}]^\beta} & q > q_0 \text{ \& } j = allowed_k \end{cases} \quad (1)$$

其中, q 是一个均匀分布在 $[0, 1]$ 上的随机数; $q_0 (0 \leq q_0 \leq 1)$ 为实现给定的参数; α 和 β 分别表示信息素和启发值的权重。

(3)局部更新规则。当构建成 TSP 的一个完整解后, 蚂蚁遍历了这个解包含的各条边, 并应用局部更新规则改变了这些边上的信息素水平:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + L_{gb} / (L_k \times S) \text{ if } edge(i, j) \in Path_k \quad (2)$$

其中, L_{gb} 表示当前得到的最优解的长度; $Path_k$ 和 L_k 分别表示第 k 只蚂蚁在当前迭代中得到的路径和路径长度; S 表示蚂蚁的数量。

(4)全局更新规则。当所有蚂蚁都构建好自己的完整解后, 蚁群优化主体将应用全局优化规则来更新当前最优解包含的各条边上的信息素水平。如果给定迭代中得到的最优解并不由于当前最优解, 则当前最优解包含的各条边上的信息

素水平不被更新。全局更新规则如式(3)所示。

$$\tau_{ij}(t+1) = \tau_{ij}(t) + L^* / L_{gb} \text{ if } edge(i, j) \in Path_{gb} \quad (3)$$

其中, L^* 表示已有文献中得到的最优解的长度; $Path_{gb}$ 和 L_{gb} 分别表示当前最优路径及其长度。

(5)信息素衰减规则。当每次迭代完成后, 蚁群优化主体都要执行信息素衰减规则。每条边上的信息素更新都要按如下信息素衰减规则进行:

$$\tau_{ij}(t+1) = \begin{cases} \min\{\tau_{max}, (1-\rho)\tau_{ij}(t+1)\} & \text{if } \tau_{ij}(t+1) > \tau_{max} \\ \max\{\tau_{min}, (1-\rho)\tau_{ij}(t+1)\} & \text{other} \end{cases} \quad (4)$$

在本文的蚁群优化主体中, 为防止算法陷入到局部极值点, 每条边上的信息素水平都控制在 $[\tau_{min}, \tau_{max}]$ 范围内。

2.2 遗传算法的实现

(1)选择操作。选择算子在当前解的临域内不断地进行探索, 以期得到更为优秀的解。即在选择算子的作用下, 当前解将被它的临域精英解所替代。本文的选择操作采用相邻交换策略进行临域解的搜索。

(2)交叉操作。交叉操作是遗传算法中最重要的一种操作, 对于每对须进行交叉的父代个体, 本文的交叉算子要执行多次交叉, 然后从父代个体和子代个体中选择 2 个最优个体作为交叉的结果, 交叉算子采用相同位优先拷贝规则(the same-site-copy-first principle)。

(3)变异操作。在优化过程中, 主要通过变异操作的执行来增强整体算法的全局搜索能力。由于本文的选择算子采用相邻交换策略, 因此此处变异算子采用 shift-change 策略。

3 MACA算法的性能分析

在本部分, 笔者采用一些常用的旅行商问题来测试本方法的有效性。为保证优化结果的稳定性, 笔者对每个旅行商问题都进行了 10 次优化, 以 10 次优化结果的平均值作为最终的求解结果。将本方法的优化结果和其他 3 种已经公开的优化方法进行比较, 如表 1 所示。

表 1 参与比较的 4 种不同方法

符号	优化算法名称	文献
M1	标准蚁群算法	—
M2	多种群蚁群算法	文献[4]
M3	蚁群系统算法	文献[5]
M4	Multi-Agent 协同进化算法	本文

所有的优化结果都是在主频为 2.4 GHz、内存为 256 MB 的 Pentium IV 个人电脑上运行得到。采用以上 4 种方法优化 6 个基准问题的实验结果如表 2 所示。

表 2 采用 4 种不同方法求解 6 个测试实例的仿真实验结果

实例名	最优结果	优化偏差/(%)				计算时间/s			
		M1	M2	M3	M4	M1	M2	M3	M4
LIN318	42 029	0.88	0.54	0.82	0.09	18.89	17.23	55.06	11.37
PCB442	50 778	1.07	0.40	1.07	0.14	35.00	31.91	100.22	21.95
ATT532	27 686	0.67	0.56	0.67	0.11	41.42	37.34	114.97	27.98
U574	36 905	1.52	0.60	1.61	0.16	71.20	64.11	202.63	36.85
RAT783	8 806	1.76	1.41	1.98	0.18	159.28	142.86	459.34	78.67
PR1002	259 045	1.82	1.55	1.45	0.18	159.42	142.61	470.08	81.55
平均值		1.29	0.84	1.26	0.14	80.87	72.68	233.72	38.23

无论是优化结果还是计算时间, 本方法都明显优于其他方法。数据实例表明, 本文提出的求解 TSP 的 Multi-Agent 协同进化算法无论在优化质量还是计算时间上都明显优于其他方法。

(1)本文方法在优化质量上明显优于其他方法。采用 4 种

不同方法求解 6 个测试实例的优化偏差的分析如图 4 所示。

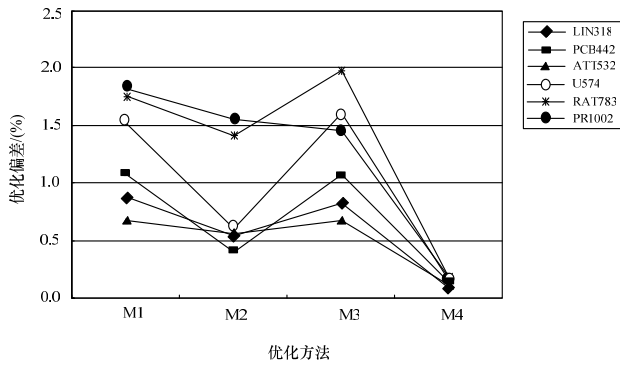


图 4 采用 4 种方法求解 6 个实例的优化偏差分析

(2) 本文方法在计算时间上明显优于其他方法。采用 4 种不同方法求解 6 个测试实例的计算时间的分析如图 5 所示。

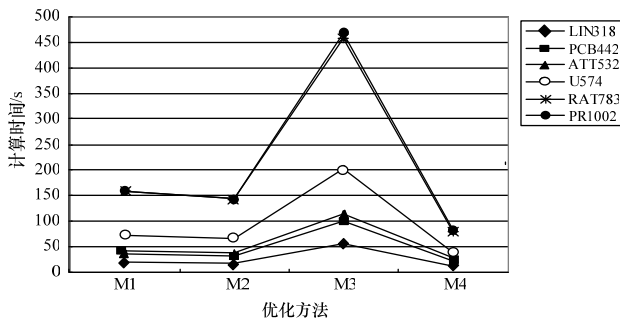


图 5 采用 4 种方法求解 6 个实例的计算时间分析

编辑 金胡考

(上接第 189 页)

由表 2、图 2 可以看出, SVM 能有效克服过度拟合问题, 表现出优越的泛化性能。相比其他几种方法在封闭测试与开放测试表现出的不均衡性, 基于结构风险最小的 SVM 方法能更好地适合垃圾邮件过滤这一具有高维特征、小样本等特性的任务。

本文还对比了 6 种不同核函数的性能, 如表 3 所示。可以看出, 线性核的性能最佳, 计算复杂度也最低。不同 C 参数对于垃圾邮件过滤性能的影响如图 3 所示。

表 3 6 种核方法、核参数的分类准确率对比

核函数	核参数	封闭测试	开放测试
线性核	$c=0$	0.978 8	0.979 5
多项式核	$d=1, s=1, c=0$	0.978 8	0.979 5
多项式核	$d=1, s=2, c=0$	0.978 8	0.979 5
多项式核	$d=2, s=1, c=0$	0.927 4	0.900 3
径向基核	$g=1$	0.834 5	0.790 3
径向基核	$g=2$	0.834 5	0.790 3

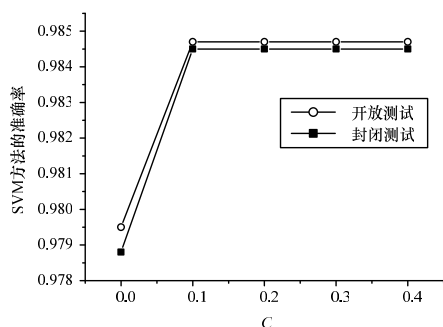


图 3 不同 C 参数对 SVM 性能的影响

编辑 张帆

4 结束语

本文的 Multi-Agent 协同进化算法取得了比较理想的优化结果, 从实践应用的角度验证了本文方法的可行性, 使所设计的算法更加有效或具有所期望的某种特别性态。下一步的工作是改进和完善算法。

参考文献

- [1] Potter M A. The Design and Analysis of a Computational Model of Cooperative Coevolutionary[D]. Fairfax County, Virginia, USA: George Mason University, 1997.
- [2] Choi I C, Kim S I, Kim H S. A Genetic Algorithm with a Mixed Region Search for the Asymmetric Traveling Salesman Problem[J]. Computers and Operations Research, 2003, 30(5): 773-786.
- [3] Tsai H K, Yang J M, Tsai Y F, et al. An Evolutionary Algorithm for Large Traveling Salesman Problems[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2004, 34(4): 1718-1729.
- [4] Jun Ouyang, Yan Guirong. A Multi-group Ant Colony System Algorithm for TSP[C]//Proceedings of the 3rd International Conference on Machine Learning and Cybernetics. New York, USA: [s. n.], 2004: 117-121.
- [5] Dorigo M, Gambardella L M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem[J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 1-26.

图 3 表明, 当 C 参数达到一定值后, 其取值对分类性能影响不大。在实际应用中可通过实验方法确定最佳 C 参数。

5 结束语

本文提出基于 SVM 的中英文混合垃圾邮件过滤方法。SVM 方法在高维特征空间以及小样本数据中表现出优越的泛化性能, 相比贝叶斯模型、最大熵模型, 更适合垃圾邮件过滤任务。大规模语料库分类实验表明, 其比采用 Good-Turing 算法平滑的朴素贝叶斯模型具有更好的泛化性能, 比最大熵模型具有更好的分类精度。

参考文献

- [1] Giménez J, Márquez L. SVMTool: A General POS Tagger Generator Based on Support Vector Machines[C]//Proceedings of the 4th International Conference on Language Resources and Evaluation. Lisbon, Portugal: [s. n.], 2004.
- [2] Pang Xiuli, Feng Yuqiang, Jiang Wei. A Chinese Anti-spam Filter Approach Based on Support Vector Machine[C]//Proceedings of International Conference on Management Science & Engineering. [S. l.]: IEEE Press, 2007.
- [3] Joachims T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features[C]//Proc. of the 10th European Conference on Machine Learning. Chemnitz, Germany: [s. n.], 1998.
- [4] 任 禾, 曾隽芳. 一种基于信息熵的中文高频词抽取算法[J]. 中文信息学报, 2006, 20(5): 40-43.
- [5] 姜 维, 王晓龙, 关 毅, 等. 基于多知识源的中文词法分析系统[J]. 计算机学报, 2007, 30(1): 137-145.

