

文章编号:1000-582X(2008)07-0781-05

多子群协同链式智能体遗传算法分析

曾孝平, 张晓娟, 李勇明

(重庆大学 通信工程学院, 重庆 400030)

摘 要:针对遗传算法(genetic algorithm, GA)易出现搜索效率不高和早收敛现象, 提出了一种多子群协同链式智能体遗传算法(multi-population agent genetic algorithm, MPAGA)。该算法采用多子群并行搜索模式、链式智能体结构, 引入动态邻域竞争和正交交叉等策略, 有效提高了算法性能。采用 3 个复杂多峰测试函数对算法进行优化性能测试结果表明, MPAGA 比普通智能体遗传算法有较快的收敛速度, 能有效防止早收敛现象。

关键词:遗传算法; 多子群; 智能体; 链式网络结构

中图分类号:TP181

文献标志码:A

Analysis of multi-population co-genetic algorithm for global numerical optimization

ZENG Xiao-ping, ZHANG Xiao-juan, LI Yong-ming

(College of Communication Engineering, Chongqing University, Chongqing 400030, P. R. China)

Abstract: In order to improve the low optimization efficiency and the premature convergence of genetic algorithms (GA), a multi-population agent co-genetic algorithm with a chain-like agent structure (MPAGA) was developed. This algorithm adopted a multi-population parallel searching mode, a chain-like agent structure, dynamic neighborhood competition, and an orthogonal crossover strategy. Three functions were used to test this algorithm. The experimental results show that MPAGA obtains higher optimization precision and converges to the domain close to global optima with higher speeds than other improved GAs.

Key words: genetic algorithm; multi-population; agent; chain-like agent structure

遗传算法是通过模拟生物在自然界的进化过程而形成的一种优化算法, 在众多领域中有良好的应用前景。但应用模型难度和复杂性较高时, 遗传算法易出现早收敛和搜索效率不高的现象。近年来人们提出了许多改进算法^[1-3]。其中文献[2]提出的基于智能体结构的多智能体遗传算法(multi-agent genetic algorithm, MAGA)具有较好的防止早收敛的性能, 但由于其采用 lattice 智能体结构, 邻域竞争不够动态且规模偏大, 使得种群的搜索效率不高。文献[3]提出的合作型协同进化遗传算法具有较快

的收敛速度, 但是破坏了相互不独立变量之间的相关性, 导致搜索到的满意解质量下降。

笔者结合智能体结构和多子群并行搜索的特点, 提出了一种多子群协同链式智能体遗传算法(multi-population agent genetic algorithm, MPAGA), 能有效防止早收敛现象, 并有较快的优化速度。

1 MPAGA 算法描述

该算法主要思路为: 首先整个种群被划分为多个子群; 其次, 每个子群独立遗传进化, 子群间通过

收稿日期: 2008-02-27

基金项目: 国家自然科学基金资助项目(30570473)

作者简介: 曾孝平(1956-), 男, 重庆大学教授, 博士生导师, 主要从事通信信号处理方向研究, (E-mail)zxp@ccee.cqu.edu.cn。

共享智能体进行遗传信息的分享与传递,以实现多个子群协同寻找满意解的目的。以下具体介绍 MPAGA 的主要算子。

1.1 多子群协同链式智能体结构

子群规模 L_{size} 为 6, 共享智能体个数 S 为 2 的多子群协同链式智能体结构如图 1 所示。

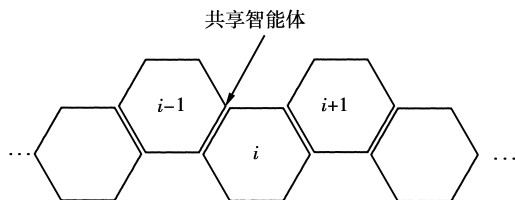


图 1 多子群协同链式智能体结构

图中一个六边形表示一个规模为 L_{size} 的子群, 整个种群由 M 个子群组成。图 1 中六边形的各个顶点表示智能体, 相邻六边形的共有顶点表示共享智能体, 子种群 i 内为链式智能体结构, 如图 2 所示, 其中圆圈代表智能体, 圆中数字代表该智能体在网格中的位置, 处于 i 子种群中第 j 个格点上的智能体表示为 $\mathbf{L}_{i,j} = (l_{i,j,1}, l_{i,j,2}, \dots, l_{i,j,n})$, $i=1, 2, \dots, M, j=1, 2, \dots, L_{\text{size}}, n$ 为维数。其邻域定义为: $\mathbf{L}_{\text{Neighbors}_{i,j}} = \{\mathbf{L}'_{i,j}, \mathbf{L}''_{i,j}\}$, 如式(1)。智能体与其左右邻域中的智能体进行竞争与合作, 共享智能体将上一子群的遗传信息传送到当前子种群, 有利于遗传信息在种群中的扩散。

$$\mathbf{L}'_{i,j} = \begin{cases} \mathbf{L}_{i,L_{\text{size}}}, & j=1, \\ \mathbf{L}_{i,j-1}, & j \neq 1; \end{cases} \quad \mathbf{L}''_{i,j} = \begin{cases} \mathbf{L}_{i,L_{\text{size}}}, & j=L_{\text{size}}, \\ \mathbf{L}_{i,j+1}, & j \neq L_{\text{size}}. \end{cases} \quad (1)$$

$$n_{i,j,k} = \begin{cases} b_{i,j,lk}, m_{i,j,k} + U(-1,1) \times (m_{i,j,k} - l_{i,j,k}) < b_{i,j,lk}; \\ b_{i,j,uk}, m_{i,j,k} + U(-1,1) \times (m_{i,j,k} - l_{i,j,k}) > b_{i,j,uk}; \\ m_{i,j,k} + U(-1,1) \times (m_{i,j,k} - l_{i,j,k}), \text{ else.} \end{cases} \quad (2)$$

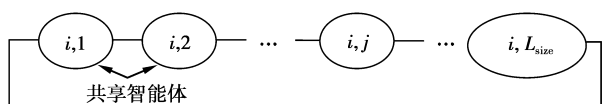


图 2 子种群内部链式智能体结构

1.2 动态邻域竞争算子

智能体生存在图 1 所示的智能体网络结构中。 $\mathbf{L}_{i,j} = (l_{i,j,1}, l_{i,j,2}, \dots, l_{i,j,n})$ 表示位于 (i,j) 格点的当前智能体, $\mathbf{M}_{i,j} = (m_{i,j,1}, m_{i,j,2}, \dots, m_{i,j,n})$ 为其邻域中能量最大的智能体, 即: $\mathbf{M}_{i,j} \in \mathbf{L}_{\text{Neighbors}_{i,j}} \quad \forall a \in$

$\mathbf{L}_{\text{Neighbors}_{i,j}}, \text{Eng}(\mathbf{a}) \leq \text{Eng}(\mathbf{M}_{i,j})$ 都成立, 其中 $\text{Eng}(\cdot) = -f(\cdot)$ 表示智能体能量, $f(\cdot)$ 为目标函数。如果 $\text{Eng}(\mathbf{L}_{i,j}) > \text{Eng}(\mathbf{M}_{i,j})$, $\mathbf{L}_{i,j}$ 就可继续存活, 否则其格点将被 $\mathbf{N}_{i,j} = (n_{i,j,1}, n_{i,j,2}, \dots, n_{i,j,n})$ 占据。 $\mathbf{N}_{i,j}$ 有 2 种生成策略, 这 2 种策略通过竞争概率 P_{co} 进行选择工作。

策略 1 中, $\mathbf{N}_{i,j}$ 按式(2)产生。其中, l 为变量的下界; u 为变量的上界; $m_{i,j,k}$ 为 $\mathbf{M}_{i,j}$ 的第 k 维变量; $l_{i,j,k}$ 为 $\mathbf{L}_{i,j}$ 的第 k 维变量; $b_{i,j,lk}$ 和 $b_{i,j,uk}$ 分别为 $l_{i,j,k}$ 的上下边界。

$$n_{i,j,k} = \left(p_{\text{match}} + \sum_{t=2}^N \left(p_{\text{match}} \times (1 - p_{\text{match}})^{(t-1)} \right) + p_{\text{com}} \right) \times m_{i,j,k} \quad (3)$$

策略 2 中, $\mathbf{N}_{i,j}$ 按照式(3)产生。式(3)中, 拆分概率 p_{match} 为 $[0,1]$ 随机数, p_{com} 定义为

$$p_{\text{com}} = 1 - \left[p_{\text{match}} + \sum_{t=2}^N \left(p_{\text{match}} (1 - p_{\text{match}})^{(t-1)} \right) \right].$$

动态竞争策略为在竞争过程中, 当第 j 个智能体经过上述竞争后将被立即更新。设第 j 个智能体在竞争前表示为 $\mathbf{L}_{i,j}^{\text{pre}}$, 竞争后表示为 $\mathbf{L}_{i,j}^{\text{post}}$, 其对应的 $\mathbf{M}_{i,j}$ 可表示为式(4)。

$$\mathbf{M}_{i,j} = \begin{cases} \max(\mathbf{L}_{i,L_{\text{size}}}^{\text{pre}}, \mathbf{L}_{i,j+1}^{\text{pre}}), & j=1; \\ \max(\mathbf{L}_{i,L_{\text{size}}}^{\text{post}} - 1, \mathbf{L}_{i,1}^{\text{post}}), & j=L_{\text{size}}; \\ \max(\mathbf{L}_{i,j-1}^{\text{post}}, \mathbf{L}_{i,j+1}^{\text{pre}}), & \text{else.} \end{cases} \quad (4)$$

1.3 正交交叉算子

与一般的随机交叉算子^[4-9]相比, 正交交叉算子可以确定一些数量较少的均匀分布在搜索空间内的有代表性个体。笔者只对该算子作简要介绍, 若需要其他细节, 请参看文献[10]。

正交交叉算子是将 $\mathbf{L}_{i,j}$ 和 $\mathbf{M}_{i,j}$ 两父代个体寻求两子代个体的过程转换为 F 因素 Q_2 水平的正交试验, 从正交表 $\mathbf{A}_4(2^3)$ 中获得能量最高的 2 种组合作为 2 个子代。本算法中 F 因素是随机生成的, Q_2 水平按式(6)产生。式(5)表示由 $\mathbf{L}_{i,j}$ 和 $\mathbf{M}_{i,j}$ 确定的搜索空间 $[x_{i,j,l}, x_{i,j,u}]$ 。第 b 维的定义域被量化为 Q_2 水平 $\beta_{b,1}, \beta_{b,2}, \dots, \beta_{b,Q_2}$, 其中 $\beta_{b,k}$ 表示如式(6)。在 MPAGA 中 Q_2, F 和 M_2 分别取值为 2、3 和 4, 正交矩阵采用 $\mathbf{A}_4(2^3)$, 表示如下:

$$\mathbf{A}_4(2^3) = \begin{bmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 & 1 & 2 & 2 & 2 & 1 \\ x_{i,j,l} = (\min(l_{i,j,1}, m_{i,j,1}), \dots, \min(l_{i,j,n}, m_{i,j,n})); \\ x_{i,j,u} = (\max(l_{i,j,1}, m_{i,j,1}), \dots, \max(l_{i,j,n}, m_{i,j,n})). \end{bmatrix} \quad (5)$$

$$\beta_{b,k} = \begin{cases} \min(l_{i,j,b}, m_{i,j,b}), & k=1; \\ \min(l_{i,j,b}, m_{i,j,b}) + (k-1) \times \left(\frac{|l_{i,j,b} - m_{i,j,b}|}{Q_2 - 1} \right), & 2 \leq k \leq Q_2 - 1; \\ \max(l_{i,j,b}, m_{i,j,b}), & k=Q_2. \end{cases} \quad (6)$$

1.4 自适应变异算子

本算子按式(7)产生新智能体 $N_{i,j}$ 。

$$n_{i,j,k} = \begin{cases} l_{i,j,k}, & U(0,1) < p_m; \\ l_{i,j,k} + G\left(0, \frac{1}{t}\right), & \text{else.} \end{cases} \quad (7)$$

式中: p_m 表示变异概率; $G(0, 1/t)$ 产生高斯分布的随机数, t 为当前进化代数。按照文献[11], 最恰当的 p_m 应设置为 $1/n$ 。

1.5 MPAGA 算法流程

笔者采用的精英保留策略为用当前子群的当前代最优个体替换下一代子群中的最差个体, 使得种群能准确收敛^[12]。退出进化条件是当代的全局最优值与上一代全局最优值差的绝对值连续 TIMES_OUT 次小于 ε (即: $\text{times} < \text{TIMES_OUT}$), 则认为种群完成遗传, 退出进化。具体流程如下:

Begin

$t=0$;

初始化种群 $p(t)$

while 没有达到推出条件 do

将 $p(t)$ 排列在多子群协同链式智能体结构中, 得到 $\text{subpop}_1(t), \text{subpop}_2(t), \dots, \text{subpop}_M(t)$;

for $m=1:M$

subpop_m(t) 进行邻域动态竞争,

得到 subpop_m($t+1/4$);

subpop_m($t+1/4$) 进行正交交叉,

得到 subpop_m($t+2/4$);

subpop_m($t+2/4$) 进行自适应变异,

得到 subpop_m($t+3/4$);

subpop_m($t+3/4$) 进行精英保留,

得到 subpop_m($t+1$);

更新该子群的最优个体信息;

end for

按顺序将 $\text{subpop}_1(t), \text{subpop}_2(t), \dots,$

$\text{Subpop}_M(t)$ 组合成 $p(t+1)$;

从 M 个子群的最优个体中选出 t 代的可行解;

更新进化计数器 times;

$t=t+1$;

end

End

2 实验结果和分析

为了验证本文算法的有效性, 笔者参考了一些论文所采用的经典测试函数^[13-16]对 MPAGA 和 MAGA 的性能进行测试, 以下函数均为高欺骗性的多峰复杂函数, 其中 f_{\min} 为函数的最小值。

$$f_1(x) = \sum_{i=1}^n -x_i \sin(\sqrt{|x_i|}),$$

$$x_i \in [-500, 500]^n,$$

$$f_{\min} = -418.95n;$$

$$f_2(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10],$$

$$x_i \in [-5.12, 5.12]^n,$$

$$f_{\min} = 0;$$

$$f_3(x) = \frac{1}{4\,000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

$$x_i \in [-600, 600]^n,$$

$$f_{\min} = 0.$$

笔者选用: 平均最小值、平均退出遗传代数、平均运行时间 3 个指标来衡量算法的性能。

测试所采用的参数设置如下: 种群规模为 66, $L_{\text{size}} = 6$, $S = 2$, $p_c = 0.1$, $p_m = 0.1$, $P_{\text{co}} = 0.5$, $\text{TIMES_OUT} = 10$, 最大进化代数 $T = 1\,000$ 。平均优化结果为随机运行 50 次的平均值; V 为平均最小值, G 为平均退出代数, T 为平均运行时间。实验条件为: CPU 主频 2.80 GHz, 内存 0.99 GB。

表 1 为 MAGA 和 MPAGA 两算法优化 2~100 维函数效果对比。

由表 1 可看出: 随着维数 n 的增加, MPAGA 的运行时间增长幅度远小于 MAGA, 同时其平均最小值更接近标准最小值, 说明 MPAGA 能有效防止早收敛现象, 且搜索过程较稳定, 运行时间较短。

图 3 为函数 f_2 在多种维数情况下的收敛性能测试图。由图可知, 收敛曲线分为 2 部分: 全局搜索区和局部搜索区。在全局搜索区, MPAGA 的搜索经历代数比 MAGA 少, 曲线下降剧烈, 说明 MPAGA 具有较强的全局搜索能力。在局部搜索区 MPAGA 不仅搜索经历代数比 MAGA 少, 而且搜索到的满意值较接近全局最优值, 说明 MPAGA 比 MAGA 有较强的局部搜索能力, 能较好地防止早收敛现象。

表 1 MAGA 和 MPAGA 优化 2~100 维函数效果对比

函数	维数	MAGA			MPAGA		
		V	G	T/s	V	G	T/s
f_1	2	-830.798 6	23.02	0.357 9	-837.965 5	21.06	0.188 1
	10	-4 142.452 3	55.96	8.753 7	-4 189.828 8	26.26	5.115 2
	50	-18 242.887 8	328.74	49.701 8	-20 949.138 2	27.82	6.047 8
	100	-33 803.142 2	339.96	128.510 9	-40 608.497 5	28.10	10.624 9
f_2	2	0	19.06	0.268 2	0	16.40	0.133 2
	10	2.819 9	19.38	3.877 8	0	19.02	3.201 2
	50	2.643 6	102.46	16.061 8	0	19.22	4.211 8
	100	0.690 3	145.40	61.218 7	0	19.26	8.437 5
f_3	2	$6.353\ 1\times 10^{-6}$	33.14	0.361 7	0	22.82	0.188 2
	10	$9.079\ 0\times 10^{-5}$	88.64	14.681 5	$1.312\ 5\times 10^{-15}$	33.76	6.516 3
	50	0.665 8	270.70	42.203 7	0.334 2	35.66	6.868 3
	100	0.791 1	356.60	158.064 7	0.408 7	48.54	13.784 5

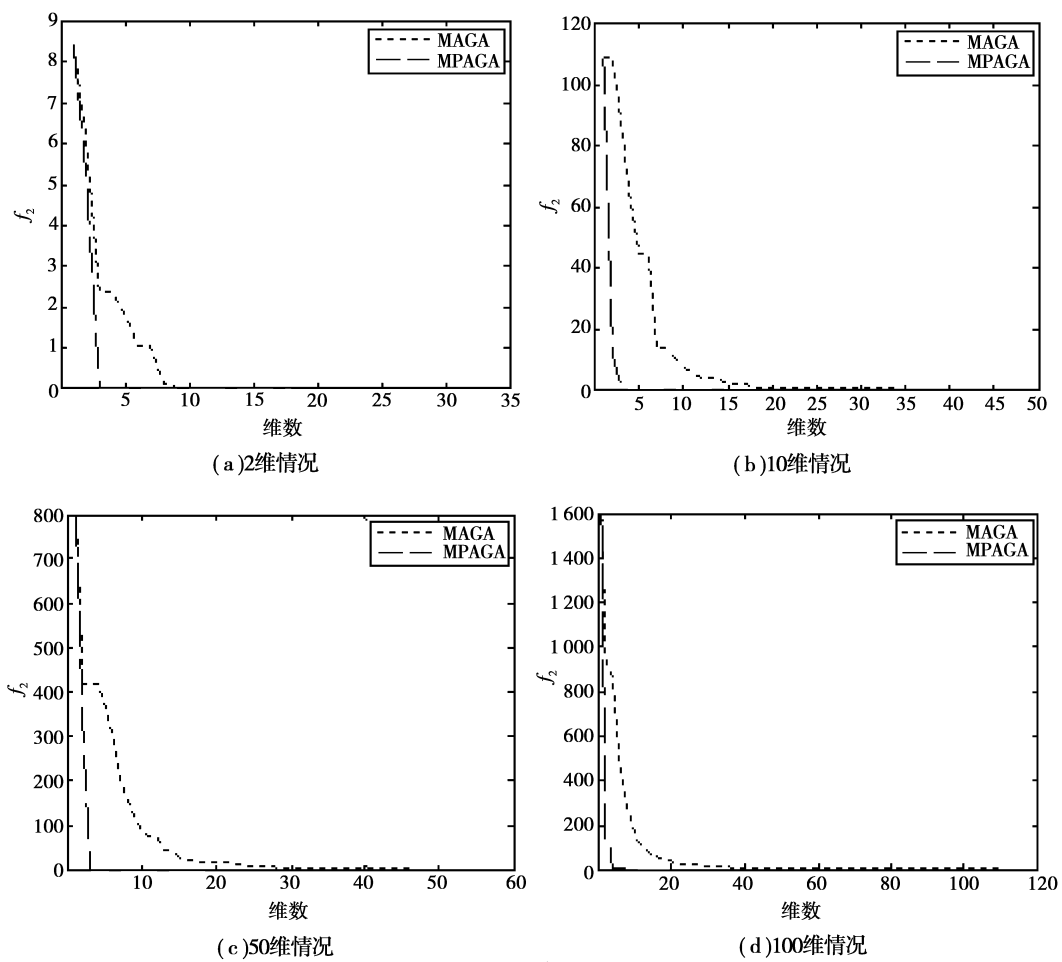


图 3 MPAGA 与 MAGA 对函数 f_2 多种维数下寻优的收敛性能对比

3 结 语

笔者针对遗传算法易出现搜索效率不高和早收敛现象,提出了一种多子群协同链式智能体遗传算法(MPAGA)。该算法通过链式智能体结构、多子群并行搜索模式、动态邻域竞争和正交叉等改进策略,提高了算法的优化性能。测试结果说明本算法比普通智能体遗传算法能更有效地防止早收敛现象,并有较快的收敛速度。

参考文献:

- [1] SRINIVAS M, PATNAIK L M. Adaptive probabilities of crossover and mutation in genetic algorithms [J]. IEEE Transactions on Systems, Man and Cybernetics, 1994, 24(4):656-667.
- [2] ZHONG W C, LIU J, JIAO L C. A multi-agent genetic algorithm for global numerical optimization [J]. IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, 2004, 34(2):1128-1141.
- [3] 孙晓燕, 巩敦卫. 变种群规模合作型协同进化遗传算法及其在优化中的应用 [J]. 控制与决策, 2004, 19(12): 1437-1440.
SUN XIAO-YAN, GONG DUN-WEI. Varying population size cooperative co-evolutionary genetic algorithm and its application in optimization [J]. Control and Decision, 2004, 19(12):1437-1440.
- [4] HONG J H, CHO S B. Efficient huge-scale feature selection with speciated genetic algorithm [J]. Pattern Recognition Letters, 2006, 27(2):143-150.
- [5] DOAN S, HORIGUCHI S. An efficient feature selection using multi-criteria in text categorization [J]. IEEE Computer Society, 2004(11):86-91.
- [6] CLUNE J, GOINGS S, PUNCH B, et al. Investigations in meta-GAs: panaceas or pipe dreams? [C] // Proc of Workshop on Genetic and Evolutionary Computation, June 25-29, 2005, Washington, D. C. USA. New York: ACM Press, 2005:235-241.
- [7] JUANG C F, LIOU Y C. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design [J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3):240-255.
- [8] ZHANG J, CHUNG H S H, HU B J. Adaptive probabilities of crossover and mutation in genetic algorithms based on clustering technique [C] // Proceeding of the 2004 Congress on Evolutionary Computation, June 19-23, 2004, Portland, OR, United States. New York: Institute of Electrical and Electronics Engineers Inc, 2004:2280-2287.
- [9] YUAN X H, CAO L, XIA L Z. Adaptive genetic algorithm with the criterion of premature convergence [J]. Journal of Southeast University: English Edition, 2003, 19(1): 40-43.
- [10] LEUNG Y W, WANG Y. An orthogonal genetic algorithm with quantization for global numerical optimization [J]. IEEE Transactions on Evolutionary Computation, 2001, 5(1):41-53.
- [11] ZBIGNIEW MICHALEWICZ, DAVID B FOGEL. 如何求解问题: 现代启发式方法 [M]. 曹宏庆, 李艳, 董红斌, 等译. 北京: 中国水利水电出版社, 2003.
- [12] 张文修, 梁怡. 遗传算法的数学基础 [M]. 西安: 西安交通大学出版社, 2000.
- [13] YAO X, LIU Y, LIU G M. Evolutionary programming made faster [J]. IEEE Transactions on Evolutionary Computation, 1999, 3(2):82-102.
- [14] PAN Z J, KANG L S. An adaptive evolutionary algorithm for numerical optimization in simulated evolution and learning [C] // Lecture Notes in Artificial Intelligence. Berlin, Germany: Springer-Verlag, 1997: 27-34.
- [15] MUHLENBEIN H, SCHLIERKAMP-VOOSEN D. Predictive models for the breeder genetic algorithm [J]. Evolution Computation, 1993, 1(1):25-49.
- [16] ZHU F M, GUAN S. Feature selection for modular GA-based classification [J]. Applied Soft Computing, 2004, 4(4):381-393.

(编辑 张 苹)