

基于半径参数周期性缓慢变化的双种群遗传算法 *

刘 伟, 涂井先

(广东工业大学 应用数学学院, 广州 510520)

摘 要: 双种群遗传算法引进了主种群和辅助种群,通过控制半径参数的变化来控制辅助种群的变化从而实现种群多样性。但其存在半径参数变化过快导致辅助种群得不到控制的缺陷。针对该缺点,用余弦函数来刻画半径参数的变化,提出了基于半径参数周期性缓慢变化的双种群遗传算法。半径参数的这种变化规律既有利于维持种群多样性,又能增强局部搜索能力。为了估算半径参数的最大取值,给出了把个体与主种群中心的最大距离作为半径参数最大取值的方法。仿真实验表明,新算法优于当前一些较好的遗传算法。

关键词: 遗传算法; 双种群; 主种群; 辅助种群; 交叉繁殖; 半径参数

中图分类号: TP183; TP301.6 **文献标志码:** A **文章编号:** 1001-3695(2012)01-0043-04

doi:10.3969/j.issn.1001-3695.2012.01.011

Dual-population genetic algorithm based on
periodic slow change in radius parameter

LIU Wei, TU Jing-xian

(Faculty of Applied Mathematics, Guangdong University of Technology, Guangzhou 510520, China)

Abstract: Dual-population genetic algorithm introduced main population and reserve pop-ulation, it dominated the reserve population through controlling the change of radius parameter which maintained the diversity of group. However, the radius parameter changed too fast to control the reserve population. For the sake of overcoming this shortcoming, described the change of radius parameter by means of cosin function and proposed a dual-population genetic algorithm based on periodic slow change in radius parameter. The regularity for change of radius parameter was not only advantageous to keep the diversity of group, but also enhanced the local search. In order to estimate the maximum value of radius parameter, took the distance between individual and the centre of main population view as the maximum value of radius parameter. The results obtained show that the improved genetic algorithm is more effective than some current optimization algorithms.

Key words: genetic algorithm; dual-population; main population; reserve population; crossbreeding; radius parameter

0 引言

因为种群多样性过早地丧失,所以传统的遗传算法会出现早熟现象。为了克服遗传算法容易出现早熟现象的缺点,不少学者相继提出了基于适应值共享的策略^[1~3]和基于双种群的策略^[4~9],这些策略在一定程度上维持了种群多样性。基于适应值共享策略的缺点是要准确地设定小生境半径是很困难的。基于双种群的策略是近几年提出来的,成为维持种群多样性策略的研究热点之一。虽然有些双种群遗传算法通过两个种群中优秀个体互相迁移来达到维持种群多样性的目的,但是这种方法容易出现“统治现象”^[4~6]。有些双种群算法通过两个种群交叉繁殖来维持种群多样性^[7~9],有一定的优势。

Taejin 等人^[7]提出了基于自适应多样性控制的双种群遗传算法,该算法引进了主种群和辅助种群的概念,通过两个种群交叉繁殖来维持种群多样性。在辅助种群中,个体适应值是由个体与主种群中心的距离和半径参数决定的,该算法自适应地控制半径参数的变化以达到控制辅助种群变化的目的。半径参数呈现周期性的变化,在一个周期内,半径参数由 1 逐渐靠近上一代成功繁殖的两父代之间的距离,直至半径参数连续

若干代都不变小^[7]。半径参数在两个相邻周期的交接处会由一个远小于 1 的值增加到 1,而辅助种群不能适应半径参数跳跃性的变化。生物的进化是缓慢的,当生存环境变化很大时,生物不能迅速地适应新环境,相应地半径参数在进化过程中得缓慢地变化,这样辅助种群的变化才能跟得上半径参数的变化。因为个体与主种群中心距离的最大取值不超过 1,所以该算法将半径参数的最大取值设为 1 且固定不变。事实上,个体与主种群中心距离的最大取值可能远小于 1。例如,当主种群中心位于搜索空间的中间时,个体与主种群中心距离的最大取值可能不超过 0.5,此时把半径参数的最大值设为 0.5 是比较合理的。因为主种群中心是变化的,所以个体与主种群中心距离的最大取值也会发生变化,半径参数的最大取值也要随着变化。

针对半径参数变化过快和半径参数最大取值不合理的缺点,本文对文献[7]的算法进行了改进,提出了基于半径参数周期性缓慢变化的双种群遗传算法,其中辅助种群具有维持种群多样性和局部搜索的双重作用。为了防止半径参数出现跳跃性变化的现象,本文给出了用余弦函数刻画半径参数的变化规律。半径参数呈现周期性的变化,在一个周期内,半径参数

收稿日期: 2011-06-11; 修回日期: 2011-07-26 基金项目: 国家自然科学基金资助项目(60974077)
作者简介: 刘伟(1970-),男,副教授,硕士,主要研究方向为最优化理论与算法、进化计算;涂井先(1987-),男,广东河源人,硕士研究生,主要研究方向为智能计算(568205127@qq.com)。

先由大逐渐变为 0, 停滞若干代, 然后由 0 逐渐变大。半径参数的这种变化规律能很好地控制辅助种群的变化, 既有利于维持种群的多样性, 又增强了算法的局部搜索能力。为了准确计算半径参数的最大取值, 本文提出了个体与主种群中心的最大距离的概念和计算方法, 将它作为半径参数的最大取值, 半径参数的最大取值不超过 1 且是变化的。

1 相关工作

文献[7]的算法引进了主种群和辅助种群, 主种群中个体适应值是由目标函数决定的。两个种群同时进化, 每个种群的后代源于该种群内部繁殖的后代和它与另一个种群交叉繁殖的后代, 从每个种群与它的所有后代中进行自然选择更新种群。

1.1 辅助种群的适应值

辅助种群中个体适应值与个体和主种群中心之间的距离有关。辅助种群的适应值函数定义为

$$f_r(x) = 1 - |\delta - d(c, x)| \quad (1)$$

其中: δ 是半径参数, $0 \leq \delta \leq 1$; $x = (x_1, x_2, \dots, x_n)$ 为辅助种群中的个体; $c = (c_1, c_2, \dots, c_n)$ 是主种群的中心; $d(c, x)$ 是 c 与 x 之间的距离。距离采用标准化的欧式距离, 计算式如下:

$$d(c, x) = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{c_i - x_i}{b_i - a_i} \right)^2} \quad (2)$$

其中: a_i 是 x_i 的下界, b_i 是 x_i 的上界。主种群中心可以是所有个体的平均值, 也可以是当代最好的个体。文献[7]的算法将主种群所有个体的平均值作为主种群的中心, 显然 $c_i \in [a_i, b_i]$, $0 \leq d(c, x) \leq 1$ 。当 $d(c, x)$ 越靠近 δ 时, $f_r(x)$ 越大, 反之 $f_r(x)$ 越小; 当 $d(c, x) = \delta$ 时, $f_r(x)$ 取得最大值 1。

1.2 进化过程

两个种群间的交叉繁殖促使两个种群能互相传递信息。每个种群对应一种适应值评价方法, 两种适应值评价完全不同。虽然一个个体在某一种适应值评价中表现得很好, 但是该个体在另一种适应值评价中表现得不一定好甚至不好。因为繁殖的后代继承了两个种群的特点, 所以它们在两种适应值评价下都表现较好。文献[7]的算法不仅对每个种群内部进行繁殖, 而且对两个种群间进行交叉繁殖。

算法 1 文献[7]的算法的主要流程

a) 参数的初始化。设定主种群规模为 M , 辅助种群规模为 N , 进化代数为 t , 最大进化的代数为 T 。

b) 初始化主种群 $\bar{M}^0$ 和辅助种群 $\bar{N}^0$ 。

c) 对主种群和辅助种群进行繁殖操作。主种群进行繁殖操作产生的后代的集合记为 I_M , 辅助种群进行繁殖操作产生的后代的集合记为 I_N , 两种种群进行交叉繁殖操作产生的后代的集合记为 C 。

d) 对父代种群及其后代种群的个体进行适应值评价。

$$\bar{M}' = \bar{M}^t \cup I_M \cup C, \bar{N}' = \bar{N}^t \cup I_N \cup C$$

用 f_m 评价 $\bar{M}'$, 用 f_r 评价 $\bar{N}'$ 。其中 f_m 为目标函数。

e) 按优胜劣汰的原则从 $\bar{M}'$ 中选择 M 个体到主种群 $\bar{M}^{t+1}$ 中, 从 $\bar{N}'$ 中选择 N 个体到主种群 $\bar{N}^{t+1}$ 中。

f) 如果 $t \leq T$, $t = t + 1$, 返回步骤 c); 否则运算结束。

种群内部繁殖操作中的两个父代来自同一个种群, 交叉繁殖操作中的两个父代分别来自主种群和辅助种群。主种群繁殖操作产生的后代集合为 I_M , 辅助种群繁殖操作产生的后代

集合为 I_N , 交叉繁殖操作产生的后代集合为 C 。在繁殖操作后, 主种群的个体除了原来的个体外还包括 I_M 和 C , 辅助种群的个体除了原来的个体外还包括 I_N 和 C 。用目标函数评价主种群的个体, 用式(1)评价辅助种群的个体。文献[7]的算法通过控制 δ 的变化来控制辅助种群的变化, 并给出了 δ 的变化。按照优胜劣汰的原则, 从主种群中选择 M 个体到下一代主种群, 从辅助种群中选择 N 个体到下一代辅助种群。

2 基于半径参数周期性缓慢变化的双种群遗传算法

在文献[7]的算法中, δ 变化太快会导致辅助种群的变化不能跟上 δ 的变化, 不利于维持种群多样性。本文提出通过缓慢地变化 δ 来控制辅助种群的变化。为了统一表示, 本文把基于半径参数周期性缓慢变化的双种群遗传算法简称为新算法。算法 1 也可以描述新算法的主要流程, 新算法与文献[7]的算法的区别在于调整 δ 的方法不同和半径参数的最大取值不同。

2.1 新算法中调整半径参数的方法

标准的遗传算法只用一个种群来探测和开发, 需要通过选择和繁殖的方法达到恰到好处地平衡探测和开发。在一些双种群遗传算法中, 两个种群在探测和开发中扮演着不同的角色。主种群倾向于探测, 辅助种群通过进化产生离主种群距离 δ 的个体为主种群提供开发新区域的机会。通过控制 δ 的变化来控制开发的程度。

本文把辅助种群中所有个体与 c 之间距离的平均值定义为辅助种群与 c 的距离, 将这个距离记为 d_{rm} :

$$d_{rm} = \frac{1}{N} \sum_{x \in N} d(c, x) \quad (3)$$

其中: N 为辅助种群。因为不同的优化问题有不同的特点以及初始化种群是随机的, 所以确定 δ 为何值时才更有利于进化是很难的, 任意取值的 δ 可能有利于进化。当 δ 较小时, 因为离 c 较近的个体容易存活, 所以 d_{rm} 也较小, 交叉繁殖产生的后代离 c 较近, 该算法有希望搜索到 c 附近的最优解; 当 δ 较大时, 因为离 c 较远的个体容易存活, 所以 d_{rm} 也较大, 交叉繁殖产生的后代离 c 较远, 该算法有希望搜索到离 c 较远的最优解; 当 δ 为 0 或者接近 0 时, 因为离 c 距离为 0 或接近 0 的个体容易存活, 所以 d_{rm} 接近 0, 交叉繁殖产生的后代离 c 很近, 该算法有希望得到高精度的最优解, 交叉繁殖增强了该算法的局部搜索能力。

基于上面的分析, 本文提出 δ 周期性缓慢变化的方法, 用余弦函数刻画 δ 的变化。 δ 呈现由大逐渐变为 0, 停滞若干代, 再由 0 逐渐变大的周期性变化。 δ 的这种变化规律既有利于维持种群多样性, 又增强算法的局部搜索能力。在一个周期 T 内, δ 的计算方法如式(4)所示:

$$\delta = \begin{cases} \delta_u \left| \cos\left(\frac{\pi}{T_1} \bar{T}_c\right) \right| & \text{if } \bar{T}_c \leq \frac{\bar{T}_1}{2} \\ 0 & \text{else if } \bar{T}_c \leq \frac{\bar{T}_1}{2} + \bar{T}_2 \\ \delta_u \left| \cos\left(\frac{\pi}{T_1} (\bar{T}_c - \bar{T}_2)\right) \right| & \text{otherwise} \end{cases} \quad (4)$$

其中: $\bar{T} = \bar{T}_1 + \bar{T}_2$; $\bar{T}_1$ 为一个周期内, δ 发生变化的代数; $\bar{T}_2$ 为一个周期内 δ 停滞为 0 的代数; $\bar{T}_c$ 为一个周期的当前代数; δ_u 为 δ 的最大取值, $0 \leq \delta_u \leq 1$ 。 $\bar{T}_c$ 初始值为 0, 此时 $\delta = \delta_u$ 。 δ 的变化周期包括 δ 发生变化的代数和 δ 停滞为 0 的代数两个阶段。

为了画图方便,假设 δ_u 是固定不变的。图 1 给出了一个周期内, δ 在 $T_1 = 150, T_2 = 15, \delta_u = 0.8$ 情况下的变化规律。当 T_c 从 0 增加到 $T_1/2$ 时, δ 从 δ_u 逐渐减小到 0, 此时 d_{rm} 也会逐渐减小; 当 T_c 从 $T_1/2$ 增加到 $T_1/2 + T_2$ 时, δ 恒为 0, 此时 d_{rm} 也会逐渐减小到一个与 0 接近的数; 当 T_c 从 $T_1/2 + T_2$ 增加到 T 时, δ 从 0 逐渐增加到 δ_u , 此时 d_{rm} 也会从与 0 很接近的数逐渐增加到与 δ_u 接近的数。

- 算法 2** 调整半径参数的算法流程
- a) $t = 0$ 。
 - b) $T_c = \text{mod}(t, T)$ 。
 - c) 利用式(4)计算 δ 。
 - d) 如果 $t \leq T, t = t + 1$, 返回步骤 b)。

算法 2 给出了调整 δ 的算法流程。图 2 给出了整个进化过程中 δ 的变化规律。其中 $T = 500$, 计算 δ 所用的参数与图 1 的一样。 T_c 等于 t 对 T 求余数, T_c 随着 t 的增加呈现出周期性的变化, T_c 在一个周期内从 0 增加到 $T - 1$ 。在计算出 T_c 后, 本文利用式(4)计算 δ , δ 呈现出周期性的变化。

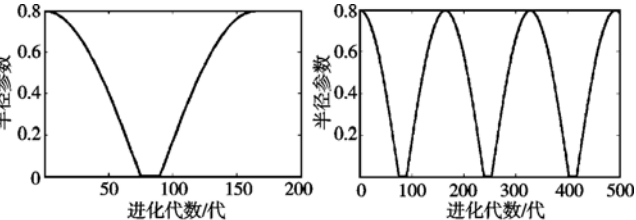


图1 一个周期内 δ 的变化规律 图2 整个进化过程中 δ 的变化规律

2.2 新算法中计算半径参数的最大取值的方法

辅助种群中离 c 距离 δ 处的个体适应值最好, 一些双种群遗传算法通过控制 δ 的变化来控制辅助种群的变化。如何设定 δ_u 是重要的, 如果 δ_u 太小, 辅助种群的所有个体都分布在以 c 为中心的很小的邻域内。没有个体位于这个邻域外的区域, 这样不利于辅助种群遍历整个搜索空间。如果 δ_u 太大, 当 δ 足够大时, 辅助种群中位于搜索空间边界上的个体容易存活, δ 的变化对辅助种群的变化无影响。虽然辅助种群也能遍历整个搜索空间, 但是辅助种群倾向于分布在搜索空间的边界上, 影响了种群多样性的维持。因为任何个体与 c 的距离不超过 1, 所以文献[7]的算法在任何情况下将 δ_u 值设为 1。事实上, 在大多数情况下将 δ_u 设为 1 是不合理的。本文提出了一种计算 δ_u 行之有效的方法, 用个体与 c 的最大距离估计 δ_u 。

本文将 $\bar{d} = \max\{d(c, x) | x \in S\}$ 定义为个体与 c 的最大距离, 其中 $S = \prod_{i=1}^n [a_i, b_i]$ 。容易证明:

$$\bar{d} = \sqrt{\frac{1}{n} \sum_{j=1}^n \left(\frac{\max(c_j - a_j, b_j - c_j)}{b_j - a_j} \right)^2}$$
 (5)

因为对 $\forall x \in S$, 有 $d(c, x) \leq \bar{d}, \exists x \in S$, 满足 $d(c, x) = \bar{d}$ 。由式(5)可以看出, $0 \leq \bar{d} \leq 1, \bar{d}$ 随着 c 的变化而变化。 δ_u 为

$$\delta_u = \bar{d}$$
 (6)

当 δ 从 0 到 δ_u 变化时, 辅助种群能遍历整个搜索空间; 当 δ 小于 δ_u 时, 辅助种群分布在搜索空间的内部; 当 δ 取值为 δ_u 时, 辅助种群分布在搜索空间的边界上, 所以 δ_u 是合理的。

3 仿真实验

本文选择了五个经典、复杂的多峰函数优化问题^[10], 测试

了三种算法的求解精度和收敛速度。
 f_1 是多峰函数, 有四个点达到全局最大值, 最大值约为 2.1188, 有大量的点达到局部最大值。

$$\max f_1(x, y) = 1 + x \sin(4\pi x) - y \sin(4\pi y) + \frac{\sin(6\sqrt{x^2 + y^2})}{6\sqrt{x^2 + y^2} + 10^{-15}}$$
$$x, y \in [-1, 1]$$

f_2 是 Needle-in-a-haystack 问题, 当 $a = 0.05, b = 3$, 其全局最大值为 3 600, 有五个点达到局部最大值, 其中只有一个点达到全局最大值。

$$\max f_2(x, y) = \left(\frac{b}{a + x^2 + y^2} \right)^2 + (x^2 + y^2)^2 \quad x, y \in [-5.12, 5.12]$$

f_3 的全局最小值为 0, 有无穷多个点达到局部最小值。

$$\min f_3(x, y) = (x^2 + y^2)^{0.25} (\sin^2 50(x^2 + y^2)^{0.1} + 1.0)$$
$$x, y \in [-5.12, 5.12]$$

f_4 是 Schaffer 函数, 全局最小值为 0。

$$\min f_4(x, y) = 0.5 + \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{[1 + 0.001(x^2 + y^2)^2]^2} \quad x, y \in [-10, 10]$$

f_5 是 Rana 函数, 全局最优解约为 -512。

$$\min f_5(x, y) = x \sin \sqrt{|y + 1 - x|} \cos \sqrt{|y + 1 + x|} + (y + 1) \cos \sqrt{|y + 1 - x|} \sin \sqrt{|y + 1 + x|}$$
$$x, y \in [-512, 512]$$

数值实验是在 MATLAB7.0 中完成。本文将基于半径参数周期性缓慢变化的双种群遗传算法 (记为 PRDPGA) 与另外两个遗传算法进行比较 (文献[7]中的基于自适应多样性控制的双种群的遗传算法 (记为 ADCDPGA) 和文献[3]中的基于适应值共享的遗传算法 (记为 FSGA))。在三种算法中, $p_c = 0.9, p_m = 0.1, T = 500$ 。在新算法和文献[7]的算法中, $M = 50, N = 100$ 。在新算法中, $T_1 = 150, T_2 = 15$ 。在文献[3]的算法中, $M = 30$ 。三种算法对每一个测试函数都独立重复运行 20 次。为了比较求解精度, 本文比较了 20 次实验解的平均值和标准差, 如表 1 所示。为了比较收敛速度, 本文给出了一次实验的进化曲线如图 3~7 所示。

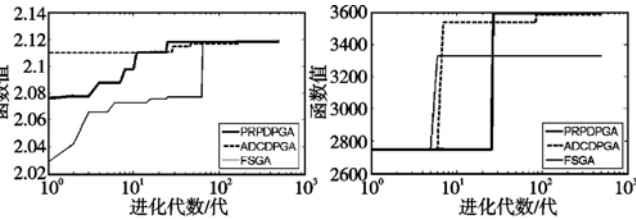


图3 f_1 的进化曲线 图4 f_2 的进化曲线

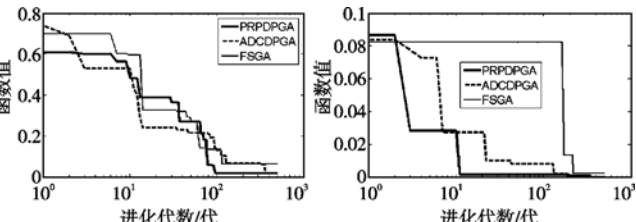


图5 f_3 的进化曲线 图6 f_4 的进化曲线

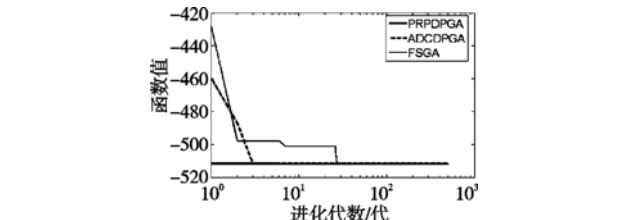


图7 f_5 的进化曲线

从表 1 可以看出, 新算法的求解精度要优于文献[3]和文

献[7]的算法。从图 3~7 可以看出,新算法的收敛速度要优于文献[3]和文献[7]的算法。

表 1 五个测试函数收敛精度的比较

函数	算法	平均值	标准差
f_1	PRPDPGA	2.1185786	2.3072322 E-04
	ADCDPGA	2.1183343	8.4444353 E-04
	FSGA	2.1071478	1.7814954 E-02
f_2	PRPDPGA	3.5956088 E+03	4.3298206
	ADCDPGA	3.4509127 E+03	2.7073117 E+02
	FSGA	3.0298345 E+03	3.8365504 E+02
f_3	PRPDPGA	2.2326195 E-02	8.2034001 E-03
	ADCDPGA	2.3493362 E-02	1.0740854 E-02
	FSGA	3.3417862 E-02	1.2192675 E-02
f_4	PRPDPGA	3.0065451 E-04	4.0365425 E-04
	ADCDPGA	3.7897113 E-04	4.8866365 E-04
	FSGA	2.3892511 E-03	2.0118706 E-03
f_5	PRPDPGA	-5.1175292 E+02	1.1039606 E-01
	ADCDPGA	-5.1173261 E+02	7.9328448 E-02
	FSGA	-5.1170777 E+02	1.5707474 E-09

4 辅助种群的分布规律

辅助种群中个体的适应值由 c 和 δ 决定,个体离 c 的距离越接近 δ ,它的适应值越高,该个体就越容易存活。双种群遗传算法通过控制 δ 的变化来控制辅助种群的变化,需要准确设定 δ 的变化规律来控制辅助种群。倘若辅助种群没有很好地受到 δ 的变化规律的控制,那么 δ 的设定没有多大意义。

图 8 给出了文献[7]的算法在求解 f_1 时 δ 和 d_m 的变化规律。从图 8 可以看出,在文献[7]的算法中, δ 在一个周期内由 1 逐渐变小,然后一会儿变大一会儿又变小,最后跳跃性变为 1。相应地, d_m 始终在 0.3~0.5 之间变化,因为 δ 变化太快,所以辅助种群的变化跟不上 δ 的变化。该算法把 δ_u 设为 1,因为 δ_u 偏大,所以个体倾向于分布在搜索空间边界上,这就导致 d_m 也大, δ 的这种变化规律影响了算法的求解效果。图 9 给出了新算法在求解 f_1 时 δ 和 d_m 的变化规律。从图 9 可以看出,在新算法中, δ 在一个周期内先由 0.6 逐渐减少到 0, δ 停滞若干代,然后逐渐变大到 0.6。相应地, d_m 由 0.5 逐渐接近到 0,在 0 附近停滞若干代,然后逐渐靠近 0.5, d_m 的变化能跟上 δ 的变化。根据式(5)(6), δ_u 围绕在 0.6 周围变化,远小于 1,进一步说明了新算法的优越性。因为 c 是变化的,所以 δ_u 也是变化的,新算法在辅助种群的分布规律上优于文献[7]的算法。

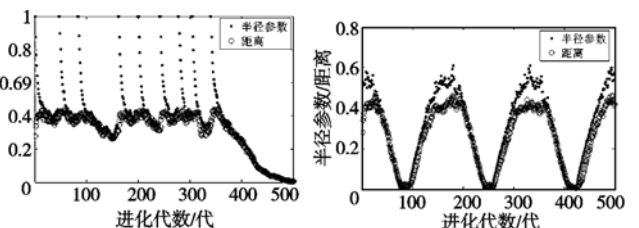


图 8 文献[7]的算法中, δ 和 d_m 的变化规律

5 参数的变化对新算法的影响

表 2 给出了 T_1 取不同的值时新算法在求解 f_2 时的求解精度。由表 2 可以看出,当 T_1 较小时,新算法求解精度不高;当 T_1 较大时,新算法求解精度较高。图 10 给出了 T_1 取不同值时新算法在求解 f_2 时的进化曲线。由图 10 可以看出,当 T_1 较小时,

新算法收敛速度较慢;当 T_1 较大时,新算法收敛速度较快。图 11 给出了 $T_1 = 10$ 和 $T_2 = 15$ 时 δ 和 d_m 的变化规律。由图 11 可以看出,当 T_1 很小时, δ 的变化很快,辅助种群的变化远跟不上 δ 的变化,这是导致新算法求解效果低的主要原因。 T_1 应该取足够大才能保证辅助种群跟得上 δ 的变化,本文建议 T_1 取值为 $[100, 300]$ 。

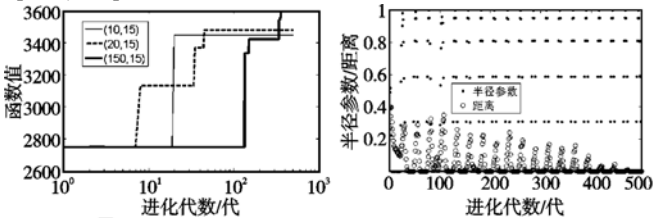


图 10 T_1 取不同值时, f_2 的进化曲线(a、b 表示 $T_1=a$ 和 $T_2=b$)

图 11 当 $T_1=10$ 和 $T_2=15$ 时, δ 和 d_m 的变化规律

表 2 T_1 取值不同时的仿真实验

函数	(T_1, T_2)	平均值	标准差
f_2	(10, 15)	3.3661115 E+03	3.3101898 E+02
	(20, 15)	3.4955742 E+03	2.5721545 E+02
	(100, 15)	3.5969227 E+03	4.6572637
	(150, 15)	3.5956088 E+03	4.3298206
	(300, 15)	3.5977953 E+03	4.7513114

表 3 给出了 T_2 取不同的值时新算法在求解 f_4 时的求解精度。由表 3 可以看出,当 T_2 较小时,新算法求解精度不高;当 T_2 较大时,新算法求解精度较高。图 12 给出了 T_2 取不同值时新算法在求解 f_4 时的进化曲线。由图 12 可以看出,当 T_2 较小时,新算法收敛速度较慢;当 T_2 较大时,新算法收敛速度较快。图 13 给出了 $T_1 = 150$ 和 $T_2 = 5$ 时 δ 和 d_m 的变化规律。由图 13 可以看出,当 T_2 很小时,半径参数的停滞为 0 的代数较小,导致 d_m 不能达到很小的值,主种群的局部搜索能力不强。 T_2 应该取足够大才能增强新算法的局部搜索能力,本文建议 T_2 取值为 $[15, 25]$ 。

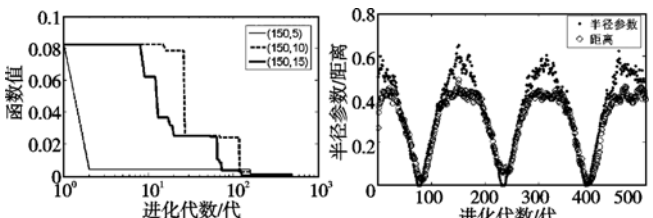


图 12 T_2 取不同值时, f_4 的进化曲线(a,b 表示 $T_1=a$ 和 $T_2=b$)

图 13 当 $T_1=150$ 和 $T_2=5$ 时, δ 和 d_m 的变化规律

表 3 T_2 取值不同时的仿真实验

函数	(T_1, T_2)	平均值	标准差
f_4	(150, 5)	7.6128439 E-04	9.5115736 E-04
	(150, 10)	3.5718563 E-04	2.8748957 E-04
	(150, 15)	3.0065451 E-04	4.0365425 E-04
	(150, 20)	4.6644618 E-04	4.8681571 E-04
	(150, 25)	3.6721029 E-04	2.9537101 E-04

6 结束语

本文提出了基于半径参数周期性缓慢变化的双种群遗传算法。半径参数呈现周期性缓慢的变化,在一个周期内,半径参数由大逐渐变为 0,停滞若干代,再由 0 逐渐变大。这种变化规律既有利于维持种群多样性,又增强了算法的局部搜索能力。本文也提出了计算半径参数最大取值的 (下转第 51 页)

测试结果与著名的 WMWALKSAT 算法的结果作比较。在 WMWALKSAT 的第 35 个版本中^[8],一共提供了七种不同翻转变量选择策略,分别为 Random、Productsum、Reciprocal、Additive、Best、Exponential、Tabu。经测试,其中 Best 策略是效果最好的一种。Best 基本思想为从不被满足的子句含有的变量中挑选权重和最小的变量进行翻转,不断接近最优解。实验环境使用 AMD Athlon TM 64 × 2 Dual Core Processor 4400 + 2.30 GHz,1.00 GB 内存。本文对每个问题用并行蚁群算法与串行蚁群算法分别求解 10 次,每次以最大搜索次数(对目标函数的评价次数)为停止准则,然后比较它们求得的解的质量。结果如表 1~3 所示。其中,表 1、2 中#1 表示本文算法,表 3 中加速比 = 串行时间/并行时间,并行效率 = 加速比/2(2 表示为双核 CPU)。对每个测试实例,若子句数超过 1 000 的则设最大搜索次数为 50 000,否则设为 20 000。

表 1 不同变量数的计算结果

序号	变量数	子句数	搜索次数	目标值	最优值		最优值		最优值		最优值	
					#1	文献[8]	#1	文献[8]	#1	文献[8]	#1	文献[8]
1	40	1 132	50 000	6 394	5 140	5 123	5 113	5 091	5 130	5 106	10.8	11.0
2	43	1 212	50 000	6 904	5 771	5 762	5 771	5 725	5 771	5 752	0.0	14.0
3	60	1 200	50 000	6 826	5 628	5 580	5 575	5 534	5 601	5 558	16.7	16.7
4	64	1 408	50 000	7 966	6 729	6 661	6 682	6 616	6 701	6 636	13.7	16.9
5	70	800	20 000	4 419	4 286	4 272	4 268	4 254	4 275	4 264	6.2	6.5

表 2 不同子句数的计算结果

序号	变量数	子句数	搜索次数	目标值	最优值		最优值		最优值		最优值	
					#1	文献[8]	#1	文献[8]	#1	文献[8]	#1	文献[8]
1	42	136	20 000	772	675	675	675	671	675	673	0.0	1.6
2	42	400	20 000	2 298	1 894	1 865	1 878	1 791	1 887	1 836	5.9	27.4
3	42	840	20 000	4 712	3 760	3 755	3 742	3 729	3 751	3 741	6.5	9.0
4	42	1 528	50 000	8 602	6 806	6 769	6 787	6 744	6 798	6 754	6.3	8.2
5	42	1 690	50 000	9 440	7 390	7 371	7 368	7 343	7 380	7 354	7.5	8.1

表 3 算法并行化后的计算结果

变量数	子句数	搜索次数	#1 运行加速比	#1 并行效率	最小运行时间/s		平均运行时间/s	
					串行时间	并行时间	串行时间	并行时间
40	1 132	5 000	1.106	0.553	1.014	0.78	1.03	0.931
41	1 640	5 000	1.598	0.799	1.326	0.826	1.339	0.838
42	1 528	5 000	1.675	0.838	1.326	0.795	1.348	0.805
43	1 212	5 000	1.529	0.765	1.045	0.686	1.061	0.694
60	1 200	5 000	1.605	0.803	1.185	0.733	1.202	0.749
64	3 648	5 000	1.825	0.913	2.855	1.606	2.95	1.616
70	700	5 000	1.535	0.767	1.005	0.645	1.01	0.658

表 1、2 中,目标值表示该问题可能取得的最大权重和,其只有在所有子句均成立时方可获得。最优值、最差值、平均值 为利用两算法进行 10 次求解得到的统计值,三者的值越大、越

接近目标值,则表明对应解的质量越高;而标准差表示算法的稳定性,其值越小则算法越稳定。从表 1、2 的结果比较中可以看到,本文算法较 WMWALKSAT 算法在解的质量上有了明显改善,并且取得了较小的标准差,说明本文算法有更好的稳定性。从表 3 可知,并行化后的算法取得了较好的加速比效果及并行效率。

4 结束语

本文提出一种用于多核 CPU 中求解加权 MAX-SAT 的并行蚁群算法,针对加权 MAX-SAT 的特点改进了蚁群算法模型,并使其并行化。改进后的算法对蚁群进化控制更加直接,运行效率也得到大大的提高,同时算法搜索效率也更加高效,算法在正反馈性、内在并行性上都能有效地避免求解问题陷入局部最优的缺点;同时,算法还能更好地控制求解问题的时间和代价,在确保寻找到最优解的前提下极大地提高了算法的收敛速度。实验表明,该算法是有效并切实可行的。

参考文献:

[1] BORCHERS B,FURMAN J. A two-phase exact algorithm for MAX-SAT and weighted MAX-SAT problems[J]. *Journal of Combinatorial Optimization*,1999,2(4):299-306.

[2] De GIVRY S,LARROSA J,MESEGUER P,et al. Solving MAX-SAT as weighted CSP[C]//*Principles and Practice of Constraint Programming*. Berlin:Springer,2003:363-376.

[3] ALSINEL T,MANYA F,PLANES J. An efficient solver for weighted MAX-SAT[J]. *Journal of Combinatorial Optimization*,2008,41(1):61-73.

[4] LIN Han, SU Kai-le. Exploiting inference rules to compute lower bounds for MAX-SAT solving[C]//*Proc of the 20th International Joint Conference on Artificial Intelligence*. San Francisco: Morgan Kaufmann Publishers,2007: 2334-2339.

[5] ELLABIB I,CALAMAI P,BASIR O. Exchange strategies for multiple ant colony system[J]. *Information Sciences*,2007,177(5): 1248-1264

[6] MANFRIN M,BIRATTARI M,STUTZLE T, et al. Parallel ant colony optimization for the traveling salesman problem[C]//*Proc of the 5th International Workshop on Ant Colony Optimization and Swarm Intelligence*. Berlin:Springer,2006: 224-234.

[7] 林奋,周育人. 求解可满足问题的改进的蚁群算法[J]. *计算机工程与应用*,2009,45(3):42-44.

[8] Walksat Home Page. Walksat[EB/OL]. <http://www.cs.rochester.edu/~kautz/walksat/>,2009.

(上接第 46 页)方法。仿真实验显示,新算法既能求解到较高精度的解,也有较强的全局搜索能力。

参考文献:

[1] GOLDBERG D E,RICHARDSON J. Genetic algorithms with sharing for multimodal function optimization[M]. Hillsdale: Lawrence Erlbaum,1987.

[2] 于歆杰,王赞基. 对适应值共享遗传算法的分类及评价[J]. *模式识别与人工智能*,2001,14(1):42-47.

[3] ANTONIO D C,CLARDIO D S. On the role of population size and niche radius in fitness sharing[J]. *IEEE Trans on Evolutionary Computation*,2004,8(6):580-592.

[4] 方必和,于蕾蕾. 基于淘汰机制的双种群遗传算法[J]. *计算机技术与发展*,2009,19(9):101-104.

[5] 李军华,黎明,袁丽华. 一种改进的双种群遗传算法[J]. *小型微型*

计算机系统,2008,29(11):2099-2102.

[6] 雷振宇,蒋玉明. 一种基于自主计算的双种群遗传算法[J]. *计算机工程*,2010,36(24):189-191.

[7] TAEJIN P,KWANG R R. A dual-population genetic algorithm for adaptive diversity control[J]. *IEEE Trans on Evolutionary Computation*,2010,14(6):580-592.

[8] TAEJIN P,KWANG R R. A dual-population genetic algorithm with evolving diversity[C]//*Proc of Congress on Evolutionary Computation*. [S. l.]:IEEE Press,2007:3516-3522.

[9] TAEJIN P,CHOE R,KWANG R R. Adjusting population distance for the dual-population genetic algorithm[C]//*Lecture Notes in Computer Science*,vol 4830. Berlin:Springer-Verlag,2007:171-180.

[10] 焦李成,杜海峰,刘芳,等. 免疫优化计算、学习与识别[M]. 北京:科学出版社,2006.