2009 年 4 月　　重庆师范大学学报(自然科学版)　　Apr. 2009
第 26 卷 第 2 期　　Journal of Chongqing Normal University(Natural Science)　　Vol. 26 No. 2

基于交叉变异操作的连续域蚁群算法研究*

刘正龙[1]，杨艳梅[2]
(1. 川北医学院 计算机与数学教研室；2. 西华师范大学 数学与信息学院，四川 南充 637000)

摘要：研究一种基于交叉变异操作的连续域蚁群算法，该算法对解的每一分量的可能取值组成一个动态的候选组，并记录候选组中的每一个可能取值的信息量。在蚁群算法的每一次迭代中，首先根据信息量选择解分量的初值，然后使用交叉、变异操作来确定全局最优解的值，通过相应算法设计，对于来自相对适应度较大的解的分量值，其变异的区域较小，成为局部搜索，反之，变异的区域较大，则构成全局搜索。同时，随着迭代次数的增多，分量值的变异幅度逐渐变小，这样可使收敛过程在迭代次数较多时得到适当的控制，以加速收敛。最后通过仿真实验，把交叉变异操作的连续域蚁群算法与遗传算法性能进行比较，证明了交叉变异操作的连续域蚁群算法具有较高的搜索较优解的能力，大大节约了计算时间。

关键词：交叉变异；连续域；蚁群算法
中图分类号：TP18　　文献标识码：A　　文章编号：1672-6693(2009)02-0087-03

在离散域组合优化问题中，蚁群算法的信息量留存、增减和最优解的选取都是通过离散的点状分布求解方式来进行的；而在连续域优化问题的求解中，其解空间是一种区域性的表示方法，而不是以离散的点集来表示的[1]。因此，连续域寻优蚁群算法是将所求问题解每一分量可能取值组成一个动态的候选组，并记录候选组中的每一个可能值的信息量，在蚁群算法的每一次迭代中，首先根据信息量选择解分量的初值，然后使用交叉、变异操作来确定全局最优解的值[2]。

1 算法设计

考虑如下有约束的非线性规划问题

$$\min G(x_1, x_2, \cdots, x_n) \tag{1}$$

使得

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \geqslant b_i, i = 1, 2, \cdots, r \tag{2}$$

这里目标函数 G 为任一非线性函数，约束条件构成一个凸区域。可采用不等式变换的方法求得包含这个凸区域的最小的 n 维立方体[1]。设该立方体为

$$l_i \leqslant x_i < u_i, i = 1, 2, \cdots, n \tag{3}$$

设蚂蚁数目为 m，在选取 m 个初始解以后，其第 i 个分量的 m 个取值构成了该分量的候选组。这里将解的 n 个分量看成 n 个顶点，在第 i 个顶点到第 $i+1$ 个顶点之间有 m 条连线，代表第 i 个分量候选组中的 m 个不同的候选值，并记其中第 j 条连线为 (i, j)，在 t 时刻 (i, j) 上的信息量记为 $\tau_{ij}(t)$。每只蚂蚁都从第一个顶点出发，按照一定的规则依次选择 $n-1$ 条连线。到达第 n 个顶点后，再从 m 条连线中选取某一条连线到达终点。每只蚂蚁所走过的路径代表一个解，其 n 条连线表示它的 n 个解。为了使解的分布具有多样性，在各个分量选取 m 个值后，对其实行交叉、变异操作，并将所得到的值作为新一代解的相应分量。在得到 m 个解后，要根据其适应度值更新各条边上的信息量[3]。每一次迭代后，要对各个分量的候选组中的值动态更新，并在新产生的值和候选组的值中选取信息量较高的 m 个值作为新的候选组。重复上述迭代过程。直至满足算法停止条件为止。

综上所述，基于交叉变异操作的连续域蚁群算法具体步骤可描述如下。

1.1 初始化

随机产生 m 个初始解，计算这 m 个初始解的适应度，由这 m 个初始解得到各个分量值的候选组，并根据候选组中的值按其所在解适应度计算其信息

* 收稿日期：2008-10-17
资助项目：川北医学院苗圃基金(08 基-17)
作者简介：刘正龙，男，讲师，硕士研究生，研究方向为分布式数据库、医学数据挖掘、生物数学模型。

量[4]。

1.2 迭代过程

While not 结束条件 do

① For $i = 1$ to n do // 对 n 个分量循环

For $k = 1$ to m do // 对 m 只蚂蚁循环

第 k 只蚂蚁根据状态转移概率在第 i 个分量的候选组内选择该分量的初值;

End for k

对所选择的第 i 个分量的 m 个初值施行交叉、变异操作生成第 i 个分量的新一代 m 个不同的候选值,并加入候选组;

End for i

根据所选择的一个分量构成 m 个新一代解,并计算新一代解的适应度值;

② 修改各分量的候选组中各候选值的信息量;

③ 取各分量的候选组中的信息量较高的 m 个值作为新的候选组

End while

在上述算法中,蚂蚁在得到新一代的解以后,算法则相应地更新各分量候选组中相应值的信息量 $\tau_{ij}(t)$,并按下式更新的 $\Delta\tau_{ij}^k$

$$\Delta\tau_{ij}^k = \begin{cases} W \cdot f_k, \text{若蚂蚁 } k \text{ 的解的第 } i \text{ 个分量选中第 } j \text{ 个候选值} \\ 0, \quad \text{否则} \end{cases} \tag{4}$$

在(4)式中,W 为一常数;f_k 表示第 k 只蚂蚁所对应解的适应度。

1.3 自适应的交叉和变异操作

上述算法的迭代过程,对第 i 个分量所选择的 m 个初值进行了交叉、变异等操作。为了加速收敛并防止局部优化,可采用具有自适应性的交叉、变异操作,根据解的自适应度值 f 可按照下式定义其相对适应度 F [5]

$$F = \begin{cases} \dfrac{f - f_{\min}}{f_{\max} - f_{\min}}, \text{若 } f_{\max} \neq f_{\min} \\ 1, \quad \text{否则} \end{cases} \tag{5}$$

在(5)式中,$f_{\max}$ 和 $f_{\min}$ 分别表示本代群体中最大和最小的适应度值。在交叉操作中,设 $x_i(1)$ 和 $x_i(2)$ 为进行交叉操作的第 i 个分量的两个初值,其所在解的适应度分别为 f_1、f_2,记 $f_{\mathrm{ave}} = (f_1 + f_2)/2$。这里以 f_{ave} 来衡量 $x_i(1)$ 和 $x_i(2)$ 的整体适应度,也作为交叉操作所产生的结果 $x_i'(1)$ 和 $x_i'(2)$ 的适应度值。可根据 f_{ave} 响应的相对适应 F_{ave} 度来决定实际交叉概率 p_c,取 $p_c = p_{\mathrm{cros}}(1 - F_{\mathrm{ave}})$,其中 p_{cros} 为系统预设的交叉概率[6]。这样,对于来自整体适应度较大的一对的分量值,其实行交叉的概率较小;反之,实行交叉的概率就较大。随机产生 $p \in [0,1]$,若 $p > p_c$,则进行交叉操作。取 $b = 2 \times (1 - F_{\mathrm{ave}})$,产生两个随机数 $c_1, c_2 \in [-b, b]$,使 $c_1 + c_2 = 1$,将 $x_i(1)$ 和 $x_i(2)$ 作仿射组合产生交叉结果值 $x_i'(1)$ 和 $x_i'(2)$ 来替换 $x_i(1)$ 和 $x_i(2)$,其表达式为

$$\begin{aligned} x_i'(1) &= c_1 x_i(1) + c_2 x_i(2), \\ x_i'(2) &= c_2 x_i(1) + c_1 x_i(2) \end{aligned} \tag{6}$$

由此,对于来自整体适应度较大的一对解的分量值,他们实行交叉的幅度较小;反之亦然。

在对某个分量值进行的交叉变异操作中,可根据 x 所在解的相对适应度值 F 来决定实际变异概率 p_m,取 $p_m = p_{\mathrm{mutate}}(1 - F)$,其中 p_{mutate} 表示算法中预设的变异概率。这样,对于来自整体适应度较大的解分量值实行变异的概率较小,可使较优的解分量尽可能予以保留。随机产生 $p \in [0,1]$,$p > p_m$,则进行变异操作[7]。设对 x_i 进行变异操作得到 x_i',记 x_i 值的上、下界分别为 u_i, l_i,为保证交叉操作的结果 x_i' 仍然在子区间 $[l_i, u_k]$ 中,设

$$d_i = \max\{u_i - x_i, x_i - l_k\} \tag{7}$$

取

$$x_i' = \begin{cases} x_i + \delta(F,t) d_i, \text{若 } l_i - x_i \leqslant \delta(F,t) d_i \leqslant u_i - x_i \\ x_i - \delta(F,t) d_i, \quad \text{否则} \end{cases} \tag{8}$$

在(8)式中,$\delta(F,t)$ 是一个 $[-1,1]$ 间的随机数,它趋近于 0 的概率随 F 和 t 的增大而增加,其表达式为

$$\delta(F,t) = r(1 - F)^{1+\lambda t} \tag{9}$$

在(9)式中,r 是一个 $[-1,1]$ 间的随机数,λ 为决定非一致性程度的参数,它具有调节局部搜索区域的作用,其取值可在 $[0.000\,1, 0.000\,3]$ 之间,这样,对于来自相对适应度较大的解的分量值,其变异的区域较小,成为局部搜索;反之,变异的区域较大,则构成全局搜索。同时,随着迭代次数的增多,分量值的变异幅度逐渐变小,这样可使收敛过程在迭代次数较多时得到适当的控制,以加速收敛。

2 仿真算例

基于交叉变异操作的连续域蚁群算法和遗传算法分别通过带约束非线性优化软件测试[8] $G_1 \sim G_5$。设 $m = 50$,$p_{\mathrm{cros}} = 0.9$,$p_{\mathrm{mutate}} = 0.5$。对每一问题进行

10 次计算,他们达到最优解平均所需要的代数和计算时间如表 1 所示。

表 1　基于交叉变异操作的连续域蚁群算法与遗传算法性能比较

问题	达到最优解所需的代数		达到最优解所需的时间/s	
	遗传算法	连续域蚁群算法	遗传算法	连续域蚁群算法
G_1	923.99	675.33	73.69	66.85
G_2	4 845.71	3 479.54	365.70	245.72
G_3	5 672.68	4 946.42	478.06	371.53
G_4	5 284.01	3 127.10	433.24	269.34
G_5	4 697.24	3 766.49	336.38	252.49
平均	4 284.726	3 198.976	337.434	241.186

3 结论

对上述问题使用遗传算法平均需要 4 284.726 次迭代才能达到最优解,而基于交叉变异操作的连续域蚁群算法平均需要 3 198.976 次迭代才能达到最优解,证明了交叉变异操作的连续域蚁群算法具有较高的搜索较优解的能力,大大节约了计算时间。

参考文献:

[1] 段海滨. 蚁群算法原理及其应用[M]. 北京:北京科学出版社,2005:175-187.

[2] 汪镭,吴铁军. 蚁群算法在连续空间寻优问题求解中的应用[J]. 控制与决策,2003,18(1):45-48.

[3] 段海滨,王道波,于秀芬. 一种求解连续空间优化问题的改进蚁群算法[J]. 系统仿真学报,2007,19(5):39-42.

[4] 高尚,钟娟,莫述军. 连续优化问题的蚁群算法研究[J]. 微机发展,2003,13(1):21-22.

[5] 陈崚,沈洁,秦玲. 蚁群算法求解连续空间优化问题的一种方法[J]. 软件学报,2002,13(12):2317-2323.

[6] 陈崚,沈洁,秦玲. 蚁群算法进行连续参数优化新途径[J]. 系统工程理论与实践,2003,23(3):48-53.

[7] Preux P H, Talbi E G. Towards Hybrid Evolutionary Algorithms International[J]. Transactions in Optional Res-earch, 1999,6:557-570.

[8] Shen J, Chen L. A new approach to solving nonlinear programming[J]. Journal of Science and Systems Engin-eering, 2002,11(1):28-36.

Research into Continual Domain Ant Colony Algorithm Based on Overlapping Mutation Operation

LIU Zhen-long[1], *YANG Yan-mei*[2]

(1. Dept. of Mathematics and Computer, North-Sichuan Medical College;
2. Mathematics & Information College, West China Normal University, Nanchong Sichuan 637000, China)

Abstract: This paper studies a continual-domain ant colony algorithm based on the overlapping mutation operation, which forms a dynamic candidate group to the solution of each component possible value, and records each possibility value information content in the candidate group. In each iteration of ant colony algorithm, firstly, it should choose the starting value of solution component according to the information content, and then it should use overlapping and variation operation to determine the overall optimal solution value. Through the corresponding algorithm design, from the relative sufficiency big solution component value, its variation region is small and becomes the partial search. Otherwise, the variation region is big and then constitutes the overall situation search. At the same time, along with iterative number of times increase, the component value variation scope becomes gradually small, as this may enable the restraining process a lot when the iterative number of times is under the suitable control in accelerating convergence. Finally through the simulation experiment, compared the overlapping variation operation continual domain ant colony algorithm with the genetic algorithm performance. Conclusion has proven the overlapping variation operation continual domain ant colony algorithm has the high search superior solution ability and saves the computing time greatly.

Key words: overlapping mutation; continual domain; ant colony algorithm

(责任编辑　游中胜)