一种求解函数优化的混合蚁群算法*

熊伟清, 陈 烽, 魏 平
(宁波大学 信息科学与工程学院, 浙江 宁波 315211)

摘 要: 将遗传算法与蚁群算法中的协同模型进行有机结合, 在蚁群算法中引入交叉、变异、选择算子来改进基本蚁群算法, 克服了蚁群算法不太适合求解连续空间优化问题的缺陷。通过测试函数表明该方法具有较好的收敛速度和稳定性, 求解结果好于遗传算法。

关键词: 模拟进化; 蚁群算法; 遗传算法; 函数优化

中图法分类号: TP301.6 **文献标识码**: A **文章编号**: 1001-3695(2005)07-0051-03

A Mixed Ant Colony Algorithm for Function Optimization

XIONG Wei-qing, CHEN Feng, WEI Ping
(School of Information Science & Engineering, Ningbo University, Ningbo Zhejiang 315211, China)

Abstract: The genetic algorithm is combined with the cooperated model of ant colony algorithm. The crossover, mutation and selection operator are proposed to improve the basic ant colony algorithm. The limitation that the algorithm doesn't fit to solve continuous space optimization is overcome. The tested function shows that the method has the better convergence speed and the stability, the solution is better than genetic algorithm's.

Key words: Simulating Evolution; Ant Colony Algorithm; Genetic Algorithm; Function Optimization

20 世纪 90 年代, 意大利学者 M. Dorigo 等人从生物进化的机理中受到启发, 通过模拟自然界蚂蚁寻找从蚁穴到食物源的最短路径的行为, 提出了一种全新的模拟进化算法——蚁群算法。从此, 吸引了许多研究人员对该算法进行研究, 并成功地运用于解决多种组合优化问题, 如旅行商问题(TSP) 、二次分配问题(QAP) 、车间调度问题(JSP) , 取得了一系列较好的实验结果。蚁群算法是一种启发式随机搜索算法, 众多的研究结果已经证明, 蚁群算法具有很强的发现较好解的能力。这是因为该算法不仅利用了正反馈原理, 在一定程度上可以加快进化过程, 而且是一种本质上并行的算法, 不同个体之间不断进行信息的交流和传递, 从而能够相互协作, 有利于发现较好的解[1]。

1 基本蚁群算法系统模型

Ant System 最先用于求解 TSP 问题, 下面就以 TSP 问题为例来说明 Ant System[2]。设 m 为蚁群数量; d_{ij} 为城市 i, j 之间的距离; $\tau_{ij}(t)$ 为 t 时刻连接城市 i 和 j 的路径 (i, j) 上的残留信息量, 初始时刻各路径上信息量相等, 设 $\tau_{ij}(0) = C$(C 为常数) ; η_{ij} 表示城市 i 转移到城市 j 的期望程度, 可根据某种启发式算法具体确定, 在 TSP 问题中一般取 $\eta_{ij} = 1/d_{ij}$。

蚂蚁 $k(k = 1, 2, \cdots, m)$ 根据各条路径上的信息量决定转移方向, t 时刻蚂蚁 k 从城市 i 向城市 j 转移的概率 $P_{ij}^k(t)$ 计算如下:

$$P_{ij}^k(t) = \begin{cases} \dfrac{\tau^{\alpha}_{ij}(t) \cdot \eta^{\beta}_{ij}(t)}{\sum\limits_{s \in \mathrm{allowed}_k} \tau^{\alpha}_{is}(t) \cdot \eta^{\beta}_{is}(t)} & j \in \mathrm{allowed}_k \\ 0 & \mathrm{otherwise} \end{cases} \quad (1)$$

其中, $\alpha(\alpha \geq 0)$ 为轨迹的相对重要性, $\beta(\beta \geq 0)$ 为期望的相对重要性, $\mathrm{allowed}_k = \{0, 1, \cdots, n-1\} - \mathrm{tabu}_k$, 表示蚂蚁 k 下一步允许选择的城市。人工蚁群和自然蚁群系统不同之处在于人工蚁群系统具有一定记忆力, $\mathrm{tabu}_k(k = 1, 2, \cdots, m)$ 用于记录蚂蚁 k 所走过的城市, 集合 tabu_k 随着进化过程进行动态调整。随着时间的推移, 以前留下的信息逐渐消逝, 用参数 $\rho(0 \leq \rho < 1)$ 表示轨迹的持久性, $1-\rho$ 表示信息素衰减度。在每一只蚂蚁完成对所有 n 个城市的访问后(也即一个循环结束后) , 各路径上的信息量要根据下式作调整:

$$\tau_{ij}(t+n) = \rho \cdot \tau_{ij}(t) + (1-\rho) \cdot \triangle\tau_{ij} \quad (2)$$

$$\triangle\tau_{ij} = \sum_{k=1}^{m} \triangle\tau_{ij}^k \quad (3)$$

其中, $\triangle\tau_{ij}^k$ 表示第 k 只蚂蚁在本次循环中留在路径 (i, j) 上的信息量, $\triangle\tau_{ij}$ 表示本次循环中路径 (i, j) 上的信息量的增量。

$$\triangle\tau_{ij}^k = \begin{cases} \dfrac{Q}{L_k} & \text{若第 } k \text{ 只蚂蚁在时刻 } t+1 \text{ 之间经过路径} (i, j) \\ 0 & \text{否则} \end{cases} \quad (4)$$

其中, Q 是体现蚂蚁所留轨迹数量的一个常数, L_k 表示第 k 只蚂蚁在本次循环中所走总路径的长度。

2 用于函数优化的混合蚁群算法设计

2.1 多维函数优化问题的数学模型

设一般多维优化问题的数学模型如下:

$$\begin{aligned} & \min f(x_1, x_2, \cdots, x_n) \\ & \mathrm{st.}\ g_i(x_1, x_2, \cdots, x_n) \geq 0, i = 1, 2, \cdots, m \\ & \quad h_i(x_1, x_2, \cdots, x_n) = 0, \ i = 1, 2, \cdots, l \end{aligned} \quad (5)$$

其中, $f(x)$ 为目标函数, $g_i(x_1, x_2, \cdots, x_n)$ 和 $h_i(x_1, x_2, \cdots, x_n)$ 称为约束函数; 向量 $X = (x_1, x_2, \cdots, x_n)$ 称为决策变量; $g_i(x_1, x_2,$

收稿日期: 2004-07-10; 修返日期: 2004-09-18
基金项目: 国家自然科学基金资助项目(60272034)

$\cdots, x_n) \geq 0$ 称为不等式约束; $h_i(x_1, x_2, \cdots, x_n) = 0$ 称为等式约束。

引入惩罚函数,即在适当条件下,可以把含约束函数的优化问题转化为如下域约束优化问题:

$$\min f(x_1, x_2, \cdots, x_n) \tag{6}$$

$$\text{st.}\ a_i < x_i < b_i, (i = 1, 2, \cdots, n)$$

因此下面使用域约束优化问题来探讨。

2.2 算法设计

设系统中的蚂蚁数为 m,在选取 m 个初始解以后,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。记其中第 j 条连线为 (i, j),在 t 时刻它上面的信息量记为 $\tau_{ij}(t)$。每只蚂蚁都从第一个顶点出发,按照一定的策略依次选择 n 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 个分量。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,所得到的值加入相应分量的候选组中,所得到的值还作为新一代解的相应分量。在得到 m 个新解后要根据它们的适应度值更新各分量候选组值的信息量,在每一次迭代后选取最好的 m 个解,这就产生了此次进化迭代的新个体。重复这样的进化迭代过程,直至满足停止条件。

(1) 编码方式。本文所用的算法采用实数编码。优化函数的维数决定了蚂蚁每一循环中需经过的路径数,蚂蚁走过的第一条路径对应函数的第一个变量,走过的第二条路径对应函数的第二个变量,依此类推。

(2) 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它们所在解的适应度计算它们的信息量。

(3) 蚂蚁状态转移规则。蚂蚁 k 的第 i 个分量选择第 i 个分量候选组中第 j 个值的概率为

$$P_{ij}^k = \tau_{ij}(t) / \sum_{r=1}^{m} \tau_{ir}(t) \tag{7}$$

其中, $\tau_{ij}(t)$ 表示第 i 个分量候选组中第 j 个值的信息量。本文采用轮盘赌方式来实现蚂蚁的状态转移。

(4) 信息量全局更新规则[3]。在得到新一代解以后,按照式(2)、式(3)更新各个分量的候选组中的值的信息量 $\triangle\tau_{ij}$,其中 $\triangle\tau_{ij}^k$ 的更新规则如下:

$$\triangle\tau_{ij}^k = \begin{cases} w^* f_k & \text{若第 } k \text{ 只蚂蚁的解的第 } i \text{ 个分量选中第 } j \text{ 个候选值} \\ 0 & \text{否则} \end{cases} \tag{8}$$

(5) 交叉过程。受遗传算法启发,引入交叉操作[4]用来进行全局搜索,这里采用算术交叉。算术交叉即按照下面的公式产生两个个体,其中 a 是事先给定的或随机产生的 $[0, 1]$ 之间的一个实数,设 x, y 进行杂交产生 x 和 y',则算术交叉的公式为

$$x' = a \cdot x + (1 - a) \cdot y \tag{9}$$

$$y' = (1 - a) \cdot x + a \cdot y \tag{10}$$

(6) 变异过程。算法通过交叉算子已接近最优解邻域时,利用变异算子提高局部随机搜索能力。设 x_i 进行变异操作得到 x_i',记 x_i 上下界分别为 u_i 和 l_i,为了保证变异操作的结果仍然在 $[l_i, u_i]$ 间,设 $d_i = \max(u_i - x_i, x_i - l_i)$。

$$\text{取 } x_i' = \begin{cases} x_i + \delta(F, t) d_i & \text{若 } x_i - l_i \leq \delta(F, t) d_i \leq u_i - x_i \\ x_i - \delta(F, t) d_i & \text{否则} \end{cases} \tag{11}$$

这里 $\delta(F, t)$ 的具体表达式为[5]

$$\delta(F, t) = r(1 - F)^{1+\lambda t} \tag{12}$$

其中, r 是一个 $[-1, 1]$ 间的随机数; λ 为决定非一致性程度的一个参数,它具有调节局部搜索区域的作用,其取值可在 0.000 1 到 0.000 3 之间。这样,对于来自相对适应度较大的解的分量值,它变异的区域较小,成为局部搜索;反之,变异的区域就较大,构成全局搜索。同时随着迭代次数的增多,分量值的变异的幅度逐渐变小,这样可使收敛过程在代数较多时得到适当的控制以加速收敛[6]。

(7) 选择最优解。采用式(13)进行选择操作。

$$f(x) = \begin{cases} C_{\max} - F(x) & \text{若 } C_{\max} \geq F(x) \\ 0 & \text{否则} \end{cases} \tag{13}$$

式(13)中, $C_{\max}$ 是一个常数, $C_{\max}$ 可以用多种方法来选取。一般通过估计 $F(x)$ 在 x 取值范围内的最大值来确定这个常数;或者可以将 $C_{\max}$ 取为当前种群中 $F(x)$ 的最大值。选取 m 个适应值最大的解,将这 m 个解作为初始解再次迭代。

3 实验测试

选用 De Jong 等提出的常用优化函数[4]来测试算法的效果。测试时 $P_c = 0.8$, $P_m = 0.2$, $\rho = 0.9$, $w = 0.1$。

测试函数(1): $\sum_{i=1}^{n} x_i^2$, $-5.12 \leq x_i \leq 5.12$。该函数在 $(x_1, x_2) = (0, 0)$ 有一个全局最小值 0。表 1 为蚂蚁数是 20,进化迭代 500 次,单独运行共计五次的运行结果。

表 1 测试函数(1)的运行结果

参数 x_1	1.62181E-05	7.00081E-06	-1.62247E-09	-2.78210E-08	7.73786E-06
参数 x_2	1.16690E-06	4.52206E-07	1.02411E-09	-9.11989E-10	2.17013E-04
函数值	2.64388E-10	4.92158E-11	3.68121E-18	7.74838E-16	4.71546E-08

测试函数(2): $100(x_1^2 - x^2)^2 + (1 - x_1)^2$, $-2.048 \leq x_i \leq 2.048$。该函数在 $(x_1, x_2) = (1, 1)$ 有一个全局最小值 0,虽然在求极小值时它是单峰值的函数,但它却是病态的,难以进行全局极小化。表 2 为蚂蚁数是 30,进化迭代 500 次,单独运行共计五次的求解结果。

表 2 测试函数(2)的运行结果

参数 x_1	0.999999999	0.999999995	1.000000001	1.000000026	1.000000049
参数 x_2	0.999999998	0.999999991	1.000000003	1.000000051	1.000000096
函数值	3.34521E-17	2.07532E-17	3.66818E-18	6.62109E-16	2.70152E-15

测试函数(3): $10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i))$, $-5.12 \leq x_i \leq 5.12$。该函数在 $(x_1, x_2) = (0, 0)$ 有一个全局最小值 0。表 3 为蚂蚁数是 20,进化迭代 200 次,单独运行共计五次的运行结果。

表 3 测试函数(3)的运行结果

参数 x_1	1.65082E-05	-9.10451E-06	1.53861E-06	-5.67409E-07	6.82128E-10
参数 x_2	3.02549E-10	-1.99199E-09	4.62202E-07	8.97938E-04	-4.60828E-08
函数值	5.40661E-08	1.64452E-08	5.12042E-10	1.59962E-04	4.21402E-13

测试函数(4): $\sum_{i=1}^{n}(-x_i \sin \sqrt{|x_i|})$, $-500 \leq x_i \leq 500$。全局最小值是函数的极小值 $= -n \times 418.9829$,取 $x_i = 420.9678$, $i = 1, 2, \cdots, n$。表 4 为采用蚂蚁数是 30,进化迭代 1 000 次,单独运行共计五次的运行结果。

表 4　测试函数(4) 的运行结果

参数 x_1	420.969	420.969	420.969	420.943	420.97
参数 x_2	420.969	420.969	420.969	420.969	420.969
函数值	-837.966	-837.966	-837.966	-837.966	-837.966

测试函数(5): $a(x_2-bx_1^2+cx_1-d)^2+e(1-f)\cos(x_1)+e$, 其中 $-5\le x_1\le 10$, $0\le x_2\le 15$; 且 $a=1, b=5.1/(4\pi^2), c=5/\pi, d=6, e=10, f=1/(8\pi)$。函数在 $(x_1,x_2)=(-\pi,12.275)$, $(\pi,2.275)$, $(9.42478,2.475)$ 处有全局最小值 0.39788。表 5 为蚂蚁数是 20, 进化迭代 200 次, 单独运行共计五次的运行结果。

表 5　测试函数(5) 的运行结果

参数 x_1	3.14155	9.42473	3.14144	3.14198	3.14155
参数 x_2	2.27532	2.47495	2.27603	2.27470	2.27529
函数值	0.397887	0.397887	0.397888	0.397888	0.397887

测试函数(6): $-\cos(x_1)\cdot\cos(x_2)e^{-(x_1-\pi)^2-(x_2-\pi)^2}$。其中, 对 $i=1,2$, $-100\le x_i\le 100$, 函数在 $(x_1,x_2)=(\pi,\pi)$ 处有一个全局最小值 -1。表 6 为蚂蚁数是 30, 进化迭代 1 000 次, 单独运行共计五次的运行结果。

表 6　测试函数 6 运行结果

参数 x_1	3.14158	3.14174	3.14159	3.14159	3.14153
参数 x_2	3.14159	3.14159	3.14159	3.14159	3.14159
函数值	-1	-1	-1	-1	-1

4　有关算法参数的选择

(1) 信息素残留系数 ρ的选取

在蚁群算法中, 人工蚂蚁是具有人类记忆功能的, 随着时间的推移, 以前留下的信息将要逐渐消逝。在算法模型中用参数 1 - ρ表示信息素挥发度, ρ是信息素残留系数。信息素挥发度 1 - ρ的大小直接关系到蚁群算法的全局搜索能力及其收敛速度。由于信息素挥发度 1 - ρ的存在, 当要处理的问题规模比较大时, 会使那些从来未被搜索到的路径(可行解) 上的信息量减小到接近于 0, 因而降低了算法的全局搜索能力, 而且当 1 - ρ过大时, 以前搜索过的路径被再次选择的可能性过大, 也会影响到算法的随机性能和全局搜索能力; 反之, 通过减小信息素挥发度 1 - ρ 虽然可以提高算法的随机性能和全局搜索能力, 但又会使算法的收敛速度降低。

表 7 为不同 ρ求解测试函数(2) 的运行结果。蚂蚁数是 30, ρ∈{0.3, 0.5, 0.7, 0.9, 0.95, 0.99}, 进化迭代 2 000 次, 对每一 ρ单独运行共计五次优化, 从中取最好的结果。

表 7　测试函数(2) 不同 ρ的运行结果

残留系数	0.3	0.5	0.7	0.9	0.95	0.99
最优解	0.00200286	0.00193025	0.000216982	6.50993E-08	6.45960E-08	1.68076E-06

由实验结果不难看出, 在其他参数相同的情况下, 信息素挥发度 1 - ρ的大小对蚁群算法的收敛性能影响极大。特别是当 1 - ρ很小时, 由于路径上的残留信息占主导地位, 信息正反馈的作用相对较弱, 搜索的随机性增强, 更能找到全局最优解。而在 1 - ρ比较大时, 由于信息正反馈的作用占主导地位, 搜索的随机性减弱, 虽然算法收敛速度加快, 但易于陷入局部最优状态。因而, 关于蚁群算法中信息素挥发度 1 - ρ的选择, 必须综合考虑算法的全局搜索能力和收敛速度两项性能指标, 针对具体问题的应用条件和实际要求, 在全局搜索能力和收敛速度两方面作出合理或折中的选择。本文蚁群算法中 ρ的选择宜取为 0.9 ～0.99 之间。

(2) 蚁群数量的选择

蚁群数量多可以提高蚁群算法的全局搜索能力以及算法的稳定性。但蚂蚁数目增大后, 会使大量曾被搜索过的路径上的信息量的变化比较平均, 信息正反馈的作用不明显, 搜索的随机性虽然得到了加强, 但收敛速度减慢。反之, 蚁群数量少, 特别是当要处理的问题规模比较大时, 会使那些从来未被搜索到的解(路径) 上的信息量减小到接近于 0, 搜索的随机性减弱, 虽然收敛速度加快, 但会使算法的全局性能降低, 算法的稳定性差, 容易出现过早停滞现象。

表 8 为不同蚂蚁数目求解测试函数(2) 的结果。蚂蚁数 m∈{10, 20, 30, 40, 50}, ρ= 0.9, 进化迭代 2 000 次, 对每一 m 单独运行共计五次优化, 从中取最好的结果。

表 8　测试函数(2) 在不同蚂蚁数量的运行结果

蚂蚁数	10	20	30	40	50
最优解	2.61340E-04	3.07185E-06	7.68062E-06	1.38685E-09	1.22072E-05

可见, 蚂蚁数过小会使算法过早收敛, 跳不出局部最优; 而蚂蚁数过大会使算法计算繁重, 收敛减慢, 同时庞大的计算得不到回报, 如例子中 $m=50$ 就属于这一情况。所以蚂蚁数应取 20 ～40 之间, 这样既保证算法执行的速度, 同时亦能求得较好的最优值, 保证算法的效率。

5　总结

本文使用一种实数编码的改进蚁群算法, 此法可以有效地克服基本蚁群算法不能求解连续参数优化问题的缺陷。本文中的算法将遗传算法与蚁群算法中的协同模型进行有机结合, 改变了一般蚁群算法中每一阶段固定可选路径的做法, 使它们动态地在变化, 而每阶段对各路径所对应的分量进行的交叉、变异等遗传操作使可选路径趋于多样化、全局化, 使对应值可以在连续空间里选取, 从而解决了用蚁群算法求解连续参数优化的问题。实例的仿真实验结果充分说明, 在连续空间函数的优化问题中, 本文给出的改进蚁群算法模型是实用而有效的, 求解结果是令人满意的。

参考文献:

[1] olorni A, Dorigo M, Maniezzo V. Distributed Optimization by Ant Colonies[C]. Elsevier: Proc. the 1st European Conf. Artificial Life, Pans, 1991. 134-142.

[2] Dorigo M, Gambardella LM. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem[J]. IEEE Trans. on Evolutionary Computation, 1997, 1(1): 53-66.

[3] Sutzle T, Hoos H. The Max-Min Ant System and Local Search for the Traveling Salesman Problem[C]. Proc. of ICEC ' 97, IEEE the 4th Int. Conf. on Evolutionary Computation, 1997.

[4] Zbigniew Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs[M]. 北京: 科学出版社, 2000.

[5] 熊伟清, 俞舜浩, 赵杰煜. 具有分工的蚁群算法及应用[J]. 模式识别与人工智能, 2003, 16(3): 328-333.

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

作者简介:

熊伟清, 男, 副教授, 研究方向为进化计算、软件工程; 陈烽, 男, 研究方向为进化计算; 魏平, 女, 副教授, 研究方向为进化计算、数据库。