

融合粒子群优化算法与蚁群算法的随机搜索算法*

A Stochastic Searching Algorithm in Combination with Particle Swarm Optimization Algorithm and Ant Colony Algorithm

支成秀,梁正友

ZHI Cheng-xiu, LIANG Zheng-you

(广西大学计算机与电子信息学院,广西南宁 530004)

(School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, 530004, China)

摘要:针对 PSO 算法与蚁群算法的优缺点,提出一种融合 PSO 算法与蚁群算法的混合随机搜索算法。该算法充分利用 PSO 算法的快速、全局收敛性和蚁群算法的信息素正反馈机制,达到优势互补,将这种优化方法拓展到求解连续空间问题,并通过实例来验证该算法对于单峰、多峰函数都能取得较好的优化效果。

关键词:搜索算法 粒子群算法 蚁群算法 连续函数优化

中图分类号:TP18 文献标识码:A 文章编号:1002-7378(2006)04-0231-03

Abstract: The particle swarm optimization algorithm (PSO) and the ant colony algorithm are employed to develop a hybrid stochastic searching algorithm. The fast convergence of PSO and the positive feedback mechanism of ant colony algorithm are used. The proposed algorithm is extended to solution of continuous function, and is used to deal with single peak and multi peaks functions in a sample, and show a good performance.

Key words: searching algorithm, PSO algorithm, Ant Colony algorithm, continuous function optimization

启发式智能方法近年来吸引了国际上众多专家、学者的关注和兴趣,启发式智能方法有诸如遗传算法、蚂蚁算法、模拟退火、禁忌搜索、粒子群优化算法等。它们在解决组合爆炸及 NP 类问题卓有成效。在针对各种具体问题本身所特有的特殊性和复杂性时,各种算法都表现出其自身的优劣性,都会面临时间性能和优化性能的双重挑战^[1]。

粒子群优化 (Particle Swarm Optimization, PSO) 算法是由 Eberhart 等^[2]于 1995 年提出的一类基于群智能的随机优化算法,是源于对鸟群的捕食等群体行为的模拟,主要用于连续空间函数的优化

问题。与遗传算法类似。一般地,其它进化算法能够应用较好的领域,PSO 算法亦能成功应用,如生产调度、多目标优化、组合优化等 NP 完全问题,并取得了较好的效果。PSO 还应用到很多的工业领域,如反应能源手电压控制,以及成分混合优化等。为了改善基本 PSO 算法的收敛性能,Y. Shi 等^[3]在 1998 年首次在速度进化方程中引入惯性权重。PSO 算法的优点:(1)具有大范围全局搜索能力;(2)搜索从群体出发,具有隐并行性;(3)搜索使用评价函数值启发;(4)收敛速度快,参数调整简单;(5)具有扩展性,容易与其他算法结合。存在的缺点是:在算法后期的局部搜索能力差,反馈信息利用不充分。

蚁群算法是 Dorigo^[4]于 1991 年提出的,源于自然界蚂蚁寻找从巢穴到食物源的最短路径的启发。蚁群自提出以来在旅行商问题 (TSP)^[5]、二次分配问题^[6]等著名的具有 NP 难的组合优化难题上获得

收稿日期:2006-07-17

作者简介:支成秀(1978-),女,广西全州人,硕士研究生,主要从事网络与并行分布式计算研究。

* 广西大学博士启动基金(编号:DD060008)。

成功应用。也引起了其它问题和学科领域的学者们的兴趣和关注。蚁群算法的优点:(1)具有信息正反馈机制;(2)分布式计算特征;(3)通用型随机优化方法;(4)具有启发式搜索特征。蚁群算法存在的缺点是:初期信息素匮乏,求解速度慢^[7]。

本文针对 PSO 算法与蚁群算法的优缺点,提出一种融合 PSO 算法与蚁群算法的新型的随机优化算法,充分利用 PSO 的快速、全局收敛性和蚁群算法的信息素正反馈机制,达到优势互补,将这种优化方法拓展到求解连续空间问题,并通过仿真实验来验证该算法对于单峰、多峰函数都能取得较好的优化效果。

1 PSO 算法与蚁群算法在函数优化中的优化机理

1.1 PSO 算法的优化机理^[2]

在 PSO 算法中,每个粒子具有速度和位置,且对应一个可行解,通过不断调整自己的位置来搜索新解,且都能记住自己搜索到的最好解,记作 P_{id} ,以及整个粒子群经历过的最好位置,即目前搜索到的最优解,记作 P_{gd} 。每个粒子都有一个速度 V 和位置 X ,且根据下面两个公式来更新自己的速度和位置。

$$\begin{cases} v_{id}(t+1) = \omega \cdot v_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 (p_{gd}(t) - x_{id}(t)), & (1) \\ x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), & (2) \end{cases}$$

其中 v_{id} 表示第 i 个粒子在第 d 维上的速度, x_{id} 表示第 i 个粒子在 d 维上的位置, ω 为惯性权重, r_1 和 r_2 为均匀分布于 $[0, 1]$ 之间的随机数, c_1 和 c_2 为加速度限定因子。

PSO 算法步骤描述如下。

第 1 步:初始化粒子群,包括群体规模,每个粒子的位置和速度,最大循环次数;

第 2 步:计算每个粒子的适应度值;

第 3 步:对每个粒子,用它的适应度值和个体极值 P_{id} 比较,如果较好,则替换 P_{id} ;

第 4 步:对每个粒子,用它的适应度值和全局极值 P_{gd} 比较,如果较好,则替换 P_{gd} ;

第 5 步:根据公式(1)、(2)更新粒子的速度和位置;

第 6 步:如果满足结束条件(误差足够好或达到最大循环次数)退出,否则回到第 2 步。

1.2 蚁群算法的优化机理

为了将蚁群算法应用到求解连续空间问题且以求解函数最大值。我们对蚂蚁寻优思想作了修改和

引申。每只蚂蚁处在一个可行解的位置,蚂蚁 i 的转移规则为:按转移概率移动各蚂蚁;蚂蚁在半径 r 的领域内进随机搜索。按转移概率 $P(k, j)$ 进行转移是蚂蚁个体根据信息素强度和启发式函数进行全局范围内的移动。

$$P(k, j) = \frac{T^\alpha(j) \cdot \eta^\beta(j)}{\sum_i T^\alpha(i) \cdot \eta^\beta(i)}, \quad (3)$$

式中, $\eta(j) = k \cdot a^{\Delta F_{ij}}$, $\Delta F_{ij} = F_j - F_i$, F_i 为蚂蚁 i 所在位置的函数值, $T(j)$ 为蚂蚁 j 所处位置的信息素数量, $a < 1$, α 和 β 为非负参数。

按概率转移也就是进行全局搜索,如果蚂蚁 k 向蚂蚁 j 进行转移,则蚂蚁 j 的信息素增量增加 Q (Q 为常数) 个信息素,即

$$\Delta T_j = \Delta T_j + Q, \quad (4)$$

领域搜索就是进行局部搜索,就是在半径 r 的领域内随机搜索,如新位置的目标函数值较原值大,则取新位置为蚂蚁的当前位置,否则不变化。半径 r 每结束一次循环就缩减,以便在后期求解越来越精。一次循环结束则更新信息素,信息素更新模型为:

$$T(k) = \rho \cdot T(k) + \Delta T(k), \quad (5)$$

其中, $T(k)$ 是信息素轨迹强度, ρ 表示轨迹的持久性, $0 \leq \rho < 1$, 将 $(1 - \rho)$ 理解为轨迹衰减度。

2 PSO 算法与蚁群算法融合

2.1 融合的设计思想

PSO 算法与蚁群算法融合的设计思想是采用 PSO 算法的快速性、全局收敛性作为前期搜索,将 PSO 算法搜索到的各粒子的历史最优值作为后期蚁群算法的初始信息素分布,充分利用蚁群算法的正反馈机制以及求解效率高等特征。融合算法的总体流程见图 1。

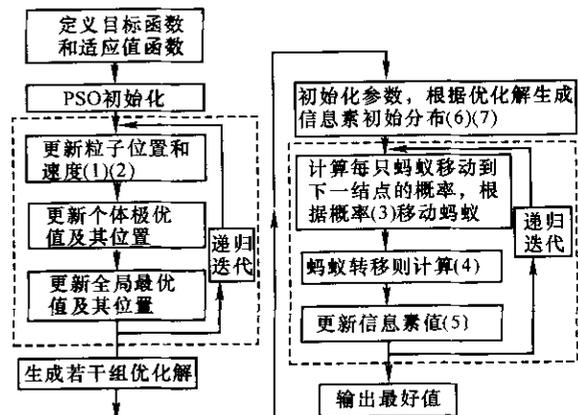


图 1 PSO 算法与蚁群算法融合的结构

2.2 融合算法中蚁群算法的衔接

结合所要求解的连续函数优化问题,对蚁群算

法的各蚂蚁位置及信息素初始分布重新设置。

蚂蚁 i 位置 $X(i)$ 对应 PSO 算法所求解的各粒子历史最优位置 $P(i)$, 即:

$$X(i) = P(i), \quad (6)$$

蚂蚁信息素初始分布利用各蚂蚁所在位置的评价函数值, 公式为:

$$T(i) = k \cdot a^{-f(x_i)}, \quad (7)$$

其中 k 为大于 0 的常数, $0 < a < 1$, $f(x_i)$ 为目标函数值, X_i 为蚂蚁 i 的位置, 根据实际问题定义 a, k 的大小, 这样目标函数值越大, 则 X_i 所在位置留下的信息素越多。

2.3 算法设计

融合 PSO 算法与蚁群算法的随机优化算法分为两部分, 首先以带惯性权重的 PSO 算法产生出各粒子最优位置值, 然后再应用蚁群算法对找出的最优位置值进一步优化调整。融合算法的步骤描述如下。

第 1 步: 进行 PSO 算法的寻优过程, 得出各粒子的历史最优值;

第 2 步: 初始化蚁群最大循环代数 N_c ;

第 3 步: 按(6)式初始化蚁群位置, 按(7)式初始化信息素, $\text{count} = 1$, 找出最好值及其相应位置;

第 4 步: 对每只蚂蚁按(3)式把 $X^{(k)}$ 变成 $X^{(j)}$, 且计算(4)式, 以半径 r 进行领域搜索, 如果局部搜索好解则替换, 否则不变化;

第 5 步: 更新最优值及其所对应的位置;

第 6 步: 对每只蚂蚁 k , 按(5)式更新信息素且;

第 7 步: $r = 0.9 * r$ (半径缩减);

第 8 步: $\text{count}++$, 若 $\text{count} < N_c$, 则转第 4 步;

第 9 步: 输出最好值及其位置。

3 实例验证

用 3 个典型函数算例验证融合 PSO 算法与蚁群算法的随机优化算法, 验证过程用 Matlab 工具编程实现, 其中, $\alpha = \beta = 1, r = 0.5, \rho = 0.7, Q = 1$ 。

例 1 Ackley 函数:

$$F = 20 \exp(-0.2 \sqrt{\frac{1}{2} \sum_{j=1}^2 x_j^2}) +$$

$$\exp(\frac{1}{2} \sum_{j=1}^2 \cos(2\pi x_j)) - 20 - \exp(1), -10 \leq x_j \leq 10, j = 1, 2$$

是一个经典测试函数, 它在一个几乎平坦的区域内由余弦波调制形成一系列峰或谷, 使曲面起伏不平。

用本文提出的算法计算该函数 20 次, 算法的粒子数与蚂蚁数 $m = 20$, PSO 算法循环 10 次, 蚁群算法循环 20 次。与文献[8]的蚁群算法采用 $m = 50, 10$ 次循环, 计算 20 次平均值的比较结果见表 1。

表 1 求解 Ackley 函数的比较结果

算法	最优值	最优位置	平均值
文献[8]算法	-0.001044	(-0.000236, -0.000283)	-0.001177
本文算法	-0.00001414	(-0.00000033, -0.00000498)	-0.00006003

例 2 Branin 函数:

$$\max F(X) = -(x_2 - bx_1^2 + cx_1 - d)^2 - e(1 - f)\cos x_1 - e,$$

其中, $b = 5.1/4\pi^2, c = 5/\pi, d = 6, e = 10, f = 1/8\pi, x_1 \in [-5, 10], x_2 \in [0, 15]$ 。本文算法与例 1 的参数一样, 与文献[9]算法用 20 只蚂蚁, 循环 1000 次的结果比较见表 2。

表 2 求解 Branin 函数的比较结果

	代数	最优值	最优值位置
文献[9]算法	71	-0.397981926517719	(9.42373, 2.4646665)
文献[9]算法	32	-0.397970397512475	(-3.14525, 12.279453)
本文算法	20	-0.39788735873956	(9.4247924, 2.4750104)

例 3 函数 Rosenbrock:

$$f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2, -2.048 \leq x_i \leq 2.048, i = 1, 2, \text{该函数是局部有些凹的双变量二次函数。}$$

本文算法与例 1 的参数一样, 与文献[10]的蚁群算法中蚂蚁数为 50 只, 循环 30 次的结果比较见表 3。

表 3 求解 Rosenbrock 函数的比较结果

算法	最优值	平均最优值	获得最优的比率(%)
文献[10]算法	3905.926	3902.184	55.8
本文算法	3905.926	3902.666	60.2

例 1 和例 2 的函数是多峰函数, 例 3 的函数是单峰函数, 从表 1~3 结果来看, 本文算法对于单峰函数和多峰函数在迭代次数、计算结果等方面与其它现存的蚁群算法相比都具有相当的优势。

4 结束语

本文设计的融合算法充分利用 PSO 算法快速收敛性及蚁群算法的正反馈机制, 达到优势互补。本文算法对 3 个经典测试函数的求解收到良好的效果。这种群智能算法的融合, 可以克服经典方法易陷



图3 细化算法对比实验结果

a. 原始图像; b. 快速细化算法结果; c. 模板细化算法结果; d. 本文算法细化结果

表1 3种算法的运行时间

算法	运行时间(ms)
快速细化算法	5.782
文献[5]算法	9.148
本文算法	6.506

4 结束语

本文在快速细化算法的基础上提出了一种新的快速细化算法。从实验结果可见,本文的细化算法不仅速度快,同时还保证了对图像的完全细化,细化效果较快速细化算法和改进的图像模板算法有一定提高。细化后的图像能够很好的保持原图像的特征,并且没有出现双像素宽的情况,是一种有效的指纹快速细化方法,有较好的应用价值。

参考文献:

- [1] 张昊,徐刚.基于四邻域的二值图像细化算法[J].信息技术与信息化,2004,6(3):24-27.
- [2] 吕俊白.一种有效的二值图像细化算法[J].计算机工程,2003,29(18):147-148.
- [3] 穆国燕,田俊霞,陈树中.一种有效的指纹图像分割和细化方法[J].计算机工程,2002,28(10):82-83.
- [4] 冯星奎,李林艳,颜组泉.一种新的指纹图像细化算法[J].中国图像图形学报,1999,4A(10):835-838.
- [5] 王家隆,郭成安.一种改进的图像模板细化算法[J].中国图像图形学报,2004,9(3):297-301.

(责任编辑:韦廷宗)

(上接第233页)

于局部优点的常见弊病,在科学与工程优化问题中具有良好的应用前景。

作为新型的进化算法,PSO算法与蚁群算法的研究时间较短,理论基础有待进一步研究。尤其是对于求解连续空间的优化问题,算法的收敛性和参数选定都需要探讨,这些都值得进一步研究。

参考文献:

- [1] 王凌.智能优化算法及其应用[M].北京:清华大学出版社,2001:154-157.
- [2] KENNEDY J, EBERHART R C. Particle swarm optimization; proceeding of the IEEE International Conference on Neural networks, IV[C]. Piscataway: IEEE Service Center, 1995:1942-1948.
- [3] SHI Y, EBERHART R C. A modified particle swarm optimizer; proceedings of the IEEE Congress on Evolutionary Computation, Piscataway [C]. USA: IEEE Service Center, 1998:6973.
- [4] COLORMI A, DORIGO M, MANIEZZO V. Distributed

optimization by ant colonies; proc Europ Conf Artif Life[C]. Paris: Elsevier Publishing, 1991:134-142.

- [5] ROCCO P. Stability of P D control for industrial robot arms [J]. IEEE Trans Robot Automat, 1996, 12(4): 606-614.
- [6] MANIEZZO V, COLORMI A. Ant system applied to the quadratic assignment problem [J]. IEEE Trans Knowl Data Eng, 1999, 11(5): 769-778.
- [7] 吴庆洪,张纪会,徐心和.具有变异特征的蚁群算法[J].计算机研究与发展,1999,36(10):1240-1245.
- [8] 杨勇,宋晓峰,王建飞,等.蚁群算法求解连续空间优化问题[J].控制与决策,2003,18(5):62-65.
- [9] 陈焯.用于连续函数优化的蚁群算法[J].四川大学学报,2004,36(6):119-122.
- [10] 熊伟清,余舜浩,魏平.用于求解函数优化的一个蚁群算法设计[J].微电子学与计算机,2003(1):23-25, 30.

(责任编辑:邓大玉)