

基于动态参数的杂交粒子群优化算法

黄伟 罗世彬 王振国

(国防科技大学航天与材料工程学院 长沙 410073)

摘要 粒子群优化算法的局部搜索能力较差,搜索精度不够高,容易陷入局部极小解,且搜索性能对参数具有一定的依赖性。本文针对这些缺点,在借鉴遗传算法中杂交概念的基础上,进一步通过在速度进化方程中引进动态参数来提高算法的收敛速度和收敛率。经 LevyNo. 5 函数对改进算法的测试表明,相对杂交粒子群优化算法,该方法的收敛速度和平均收敛率均得到了不同程度的提高。

关键词 粒子群优化算法,优化,杂交,动态参数

中图分类号 TP18 文献标识码 A

Crossbreeding Particle Swarm Optimization Algorithm Based on Dynamic Parameter

HUANG Wei LUO Shi-bin WANG Zhen-guo

(Inst. of Aerospace and Materials Engineering, National University of Defense Technology, Changsha 410073, China)

Abstract The particle swarm optimization (PSO) algorithm is easy to trapped into local extremum, and its convergence speed is lower and the precision is worse in the late evolution. Furthermore, the parameter selection can affect the algorithm. Aimed at these disadvantages of PSO, based on using the crossbreeding concept in the genetic algorithm for reference, the new algorithm by introducing dynamical parameters in the evolution of the speed equation is proposed. The convergence speed and the convergence rate were improved. The new method are tested by function Levy No. 5 shows that the convergence speed and the average convergence rate was increased.

Keywords Particle swarm optimization, Optimization, Crossbreeding, Dynamic parameter

1 引言

粒子群优化算法(PSO)是 1995 年由美国心理学家 Kennedy 和电气工程师 Eberhart 共同提出的一种基于群体组织行为模型的随机寻优算法,它通过种群中粒子之间合作与竞争关系产生的群体智能来指导优化搜索过程^[1]。由于其通用性强,具有记忆功能,且原理简单,易于实现,在化工系统^[2]、电力系统^[3]、机械设计^[4]、通信^[5]、经济^[6]、图像处理^[7]、生物信息^[8]、航空航天^[9]等诸多领域得到了广泛的应用。

由于粒子群优化算法搜索能力较差,搜索精度低,容易陷入局部极小解以及对参数具有一定的依赖性,研究者们提出了许多改进型的粒子群优化算法^[10]。彭宇等^[11]采用单因子方差分析方法,探讨了不同参数设置水平与算法性能之间的基本联系。张更新等^[12]通过同时动态调整惯性权重和加速度权重对标准 PSO 算法进行了改进。在此基础上,冯春丽等^[13]分别通过在位置进化方程中引进动态参数,以及在速度、位置进化方程中同时引进动态参数来改进标准 PSO 算法。

本文针对标准粒子群优化算法的收敛速度、平均收敛率以及参数依赖性,借鉴遗传算法中的杂交概念,通过引入动态调整惯性权重和学习因子的策略来加强算法在性能上的平衡

性,以期达到更好的优化搜索效果。

2 标准粒子群优化算法

假设在 d 维搜索域中有 n 个粒子,它们组成一个种群。其中 $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$, $i=1, 2, \dots, n$ 为第 i 个粒子的位置向量, $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$ 为第 i 个粒子的速度向量,它们都是 d 维的。 $P_i = (p_{i1}, p_{i2}, \dots, p_{id})$ 是第 i 个粒子在优化过程中所找到的最优解,即个体极值 $pbest$; P_g 是整个粒子群搜索到的最优解,即全局最优解 $gbest$ 。

在找到这两个最优值后,粒子就可以跟踪它们来更新自己。那么第 t 代的第 i 个粒子进化到第 $t+1$ 代的第 j 维的速度和位置可用下面的进化方程计算得到^[1]:

$$v_{ij}(t+1) = \omega(t)v_{ij}(t) + c_1r_1(p_{ij}(t) - x_{ij}(t)) + c_2r_2(p_{gi}(t) - x_{ij}(t)) \quad (1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (2)$$

式中, $i=1, 2, \dots, n$, $j=1, 2, \dots, d$, ω 为惯性权重因子, c_1, c_2 为正的学习因子, r_1, r_2 为 $[0, 1]$ 内均匀分布的随机数。

式(1)由三部分组成,第一部分是粒子当前速度与惯性权重的乘积,说明了粒子目前的状态对下一时刻状态的影响;第二部分是“认知”部分,代表粒子个体的信息积累;第三部分为“社会”部分,代表整个粒子种群的群体智能。三个部分共同

到稿日期:2010-01-20 返修日期:2010-03-29 本文受国防科技大学优秀研究生创新项目(B070101),湖南省研究生科研创新项目(3206)资助。

黄伟(1982-),男,博士生,主要研究方向为飞行器总体设计, E-mail: gladrain2001@yahoo.com.cn; 罗世彬(1976-),男,博士,副教授,主要研究方向为飞行器总体设计; 王振国(1960-),男,博士,教授,主要研究方向为高超声速推进技术。

决定了粒子的空间搜索能力;第一部分起到了平衡全局和局部搜索的能力;第二部分使粒子有了足够强的局部搜索能力,避免算法陷入局部极小值;第三部分体现了粒子间的信息共享。在这3部分的共同作用下粒子才能有效地到达最佳位置。

3 改进型杂交粒子群优化算法(MCPSO)

3.1 动态参数调整策略

粒子群优化算法的性能在很大程度上取决于算法的控制参数。研究发现,较大的 ω 有利于算法跳出局部极小值,提高算法的全局搜索能力。而较小的 ω 则有利于算法对当前搜索区域进行精确局部搜索,便于算法收敛。常见的权重变化公式有线性递减权重、自适应权重以及随机权重。同时,如果学习因子 c_1 相对于 c_2 较大,粒子更加倾向于自身找到的最优值,这样就没有充分利用种群中共享的信息,极易导致粒子在设计空间中过度地徘徊,降低收敛速度;反之,则粒子更加倾向于种群搜索到的最优值,这样就不能充分利用粒子本身的“经验”,极易导致粒子过早地涌向局部最优点,发生“早熟”现象^[9]。

为了平衡粒子群优化算法的全局搜索和局部改良能力,本文采用如下表达式来实现惯性权重 ω 的动态调整^[1]:

$$\omega = \begin{cases} \omega_{\min} - \frac{(\omega_{\max} - \omega_{\min}) \times (f - f_{\min})}{f_{\text{avg}} - f_{\min}}, & f \leq f_{\text{avg}} \\ \omega_{\max}, & f > f_{\text{avg}} \end{cases} \quad (3)$$

式中, ω_{\max} , ω_{\min} 分别表示惯性权重 ω 的最大和最小值, f 表示粒子当前的目标函数值, f_{avg} 和 f_{\min} 则分别表示当前所有粒子的平均和最小目标值。

当各粒子的目标值趋于一致或者趋于局部最优时,这种惯性权重动态调整策略将会促使惯性权重增加。而当各粒子的目标值相对比较分散时,这种策略将会使惯性权重减小。同时,对于目标函数值优于平均目标值的粒子,其对应的惯性权重因子较小,从而对该粒子实施了保护。反之,对于目标函数值差于平均目标值的粒子,其对应的惯性权重因子较大,将会使得该粒子向较好的搜索区域靠拢,从而提高了算法的收敛速度。

通常情况下,学习因子 c_1, c_2 固定为常数,并且取值为2。但是在实际的应用过程中,可以根据不同的问题适当地进行策略的调整。如果两个学习因子在优化过程中随时间进行不同的变化,就会使得在优化过程的初始阶段,粒子具有较大的自我学习能力和较小的社会学习能力,加强了算法的全局搜索能力。而在优化过程的后期,粒子具有较大的社会学习能力和较小的自我学习能力,有利于算法收敛到全局最优解。本文采取的学习因子变化公式为^[14]。

$$c_1 = c_{1,\text{ini}} + \frac{c_{1,\text{fin}} - c_{1,\text{ini}}}{t_{\text{max}}} t \quad (4)$$

$$c_2 = c_{2,\text{ini}} + \frac{c_{2,\text{fin}} - c_{2,\text{ini}}}{t_{\text{max}}} t \quad (5)$$

式中, $c_{1,\text{ini}}, c_{2,\text{ini}}$ 分别代表学习因子 c_1 和 c_2 的初始值, $c_{1,\text{fin}}, c_{2,\text{fin}}$ 代表学习因子 c_1 和 c_2 的迭代终值。对于大多数情况,采用如下的参数设置效果较好:

$$c_{1,\text{ini}} = 2.5, c_{2,\text{fin}} = 0.5, c_{2,\text{ini}} = 0.5, c_{1,\text{fin}} = 2.5 \quad (6)$$

3.2 杂交粒子群优化算法

借鉴遗传算法中的杂交概念,在每次迭代过程中,根据杂

交概率选取指定数量的粒子放入杂交池内。池中的粒子随机两两杂交,产生同样数目的子代粒子(child),并用子代粒子替代亲代粒子(parent)。子代位置由父代位置进行算术交叉得到^[14]:

$$\text{child}(x) = p \cdot \text{parent}_1(x) + (1-p) \cdot \text{parent}_2(x) \quad (7)$$

式中, p 是 $[0,1]$ 之间的随机数。

子代的速度可以由下式计算得到:

$$\text{child}(v) = \frac{\text{parent}_1(v) + \text{parent}_2(v)}{|\text{parent}_1(v) + \text{parent}_2(v)|} |\text{parent}_1(v)| \quad (8)$$

3.3 MCPSO 算法步骤及基本流程

MCPSO 算法步骤如下:

步骤1 随机初始化种群中各粒子的位置和速度;

步骤2 评价每个粒子的适应度,将当前各粒子的位置和适应值存储在各粒子的 pbest 中,将所有 pbest 中适应值最优的个体的位置和适应值存储于 gbest 中;

步骤3 分别用式(1)和式(2)更新粒子的速度和位移;

步骤4 采用式(3)更新惯性权重因子 ω ,分别采用式(4)和式(5)更新学习因子 c_1, c_2 ;

步骤5 对每个粒子,将其适应值与其经历过的最好位置作比较,如果较好,则将其作为当前的最好位置;

步骤6 比较当前所有 pbest 和 gbest 的值,更新 gbest;

步骤7 根据杂交概率选取指定数量的粒子放入杂交池内,池中的粒子随机两两杂交,产生同样数目的子代粒子,子代的位置和速度分别按式(7)和式(8)计算,在此过程中,保持 pbest 和 gbest 不变;

步骤8 若满足停止条件(通常为预设的运算精度或迭代次数),搜索停止,输出结果,否则返回步骤3继续搜索。

MCPSO 算法的流程如图1所示。

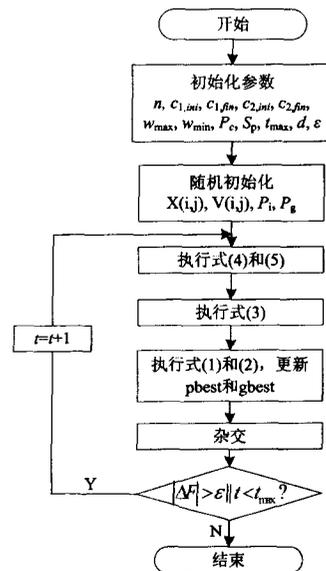


图1 MCPSO 算法流程图

4 算法验证

为检验本文提出的改进型杂交 PSO 算法的性能,选择以下典型函数进行测试,并与文献[12]提出的速度动态调整 PSO,文献[13]提出的速度、位置同时动态调整 PSO 以及杂交 PSO^[14]进行了对比试验。

(下转第 170 页)

- [4] Pawlak Z. Rough sets[J]. International J of Computer and Information Sciences, 1982, 11(5): 341-356
- [5] Pawlak Z. Rough sets[M]. Theoretical Aspects of Reasoning About Data. Netherlands; Kluwer, 1991: 45-57
- [6] 杨春燕, 张拥军. 可拓策划[M]. 北京: 科学出版社, 2002: 51-58
- [7] 李立希, 杨春燕, 等. 可拓策略生成系统[M]. 北京: 科学出版社, 2006: 122-129

- [8] 胡寿松, 何亚群. 粗糙决策理论与应用[M]. 北京: 航空航天大学出版社, 2006: 5-20
- [9] 蔡文, 杨春燕. 可拓营销[M]. 北京: 科学技术文献出版社, 2000: 11-23
- [10] 张文修, 吴伟志, 等. 粗糙集理论与方法[M]. 北京: 科学出版社, 2001: 5-16
- [11] 陈文伟, 杨春燕, 黄金才. 可拓知识与可拓知识推理[J]. 哈尔滨工业大学学报, 2006, 38(7): 1094-1096

(上接第 166 页)

$$f_4(X) = \sum_{i=1}^5 [i \cos((i-1)x_1 + i)] \times \sum_{j=1}^5 [j \cos((j+1)x_2 + j)] + (x_1 + 1, 42513)^2 + (x_2 + 0, 80032)^2 \quad (9)$$

式中, $-10 \leq x_1; x_2 \leq 10$ 。该函数具有 760 个局部极值点, 全局最优解为 $x = (-1.3068, -1.4248)$, 最优值为 -176.1375 。

在测试过程中, 选取种群规模 $n=20$, $c_{1,ini}=2.5$, $c_{1,fin}=0.5$, $c_{2,ini}=0.5$, $c_{2,fin}=2.5$, 终止条件为最优适应值与最优解之差 $\epsilon < 0.0001$, 限定最大迭代次数 $t_{max}=130$, $\omega_{max}=0.9$, $\omega_{min}=0.6$, 杂交概率 $P_c=0.9$, 杂交池的大小比例 $S_p=0.2$ 。由于粒子群优化算法的随机性, 只有进行大量测试才能评估算法的性能, 因此每次优化计算 100 次。共做 5 次试验, 考察达到最优解的次数, 并且求出平均每次所用的进化次数。其中速度动态调整 PSO 算法中其他参数的设置见文献[12], 速度、位置同时动态调整 PSO 算法中其他参数的设置见文献[13]。杂交 PSO 算法中 $c_1=2$, $c_2=2$, $\omega=0.7^{[14]}$ 。详细结果如表 1 所列。

表 1 算法性能的比较

项目	速度动态调整 PSO ^[12]	速度位置动态调整 PSO ^[13]	杂交 PSO	MCP SO
1				
达到最优次数	87	97	91	94
平均进化代数	109.08	99.57	66.64	50.62
2				
达到最优次数	88	96	90	95
平均进化代数	109.20	96.85	67.91	54.27
3				
达到最优次数	89	98	90	93
平均进化代数	108.33	98.06	66.84	53.31
4				
达到最优次数	89	97	91	95
平均进化代数	108.26	97.15	68.57	46.19
5				
达到最优次数	86	96	93	95
平均进化代数	107.22	97.97	65.66	52.76

可以看出, 经改进后的杂交粒子群优化算法达到最优解的代数明显多于杂交粒子群优化算法, 而且平均进化代数低于杂交粒子群优化算法。可见, 在混合粒子群优化算法中, 适当引进参数的动态调整策略, 可以在一定程度上提高算法的收敛速度和收敛率。

同时, 不难看出, 相比文献[12]提出的速度动态调整 PSO 算法, 借鉴遗传算法中的杂交概念不仅能提高算法的收敛速度, 而且能提高算法的收敛率; 而相比文献[13]提出的速度、位置同时动态调整 PSO 算法, 改进杂交粒子群优化算法并不能提高算法的收敛率, 但可以在很大程度上提高算法的收敛速度。

在所考察的这 4 种改进 PSO 算法中, 本文提出的 MCP SO 算法在收敛速度上的效果明显强于其他 3 种改进 PSO 算法。从一个侧面可反映出, 把粒子群优化算法的优点和其他智能算法的优点结合起来, 可以形成更加行而有效的优化算法, 提高算法的收敛速度和收敛率。当然, 这可能要付出时间上的代价。但就不同的问题而言, 这还是值得的。

结束语 本文在借鉴遗传算法中的杂交概念形成杂交粒子群优化算法的基础上, 引进参数动态调整策略, 在保证进化过程中解的多样性的同时, 提高了算法的收敛速度、平均收敛率和全局寻优能力。测试函数的对比结果表明, 改进型杂交粒子群优化算法的效果较好。对于这种改进型杂交粒子群优化算法在航空航天领域(如飞行器动力系统各部件的性能寻优)的应用还需做进一步的深入研究。

参考文献

- [1] 王凌, 刘波. 微粒群优化与调度算法[M]. 北京: 清华大学出版社, 2008
- [2] Cockshott A R, Hartman B E. Improving the fermentation medium for Echinocandin B production part II: Particle swarm optimization [J]. Process Biochemistry, 2001, 36: 661-669
- [3] Abido M A. Optimal design of power system stabilizers using particle swarm optimization [J]. IEEE Transactions on Energy Conversion, 2002, 17: 406-413
- [4] Jeon J Y, Okuma M. Acoustic radiation optimization using the particle swarm optimization algorithm [J]. International Journal of Series C-Mechanical Systems Machine Elements and Manufacturing, 2004, 47: 560-567
- [5] Zhang X G, Yu L, Zheng Y, et al. Two-stage adaptive PMD compensation in a 10 Gbit/s optical communication system using particle swarm optimization algorithm [J]. Optics Communications, 2004, 231: 233-242
- [6] Pavlidis N G, Parsopoulos K E, Vrahatis M N. Computing Nash equilibria through computational intelligence methods [J]. Journal of Computational and Applied Mathematics, 2005, 175: 113-136
- [7] Caorsi S, Donelli M, Lommi A, et al. Location and imaging of two-dimensional scatterers by using a particle swarm algorithm [J]. Journal of Electromagnetic Waves and Applications, 2004, 18: 481-494
- [8] Rasmussen T K, Krink T. Improved hidden Markov Model training for multiple sequence alignment by a particle swarm optimization-evolutionary algorithm hybrid [J]. Biosystems, 2003, 72: 5-17
- [9] 王允良. 飞行器总体参数优化的进化算法及其应用研究[D]. 西安: 西北工业大学, 2006
- [10] 谢晓锋, 张文俊, 杨之廉. 微粒群算法综述[J]. 控制与决策, 2003, 18(2): 129-134
- [11] 彭宇, 彭喜元, 刘兆庆. 微粒群算法参数效能的统计分析[J]. 电子学报, 2004, 32(2): 209-213
- [12] 张更新, 赵辉, 王红君, 等. 基于动态参数的微粒群算法(PSO)的研究[J]. 天津理工大学学报, 2005, 21(4): 42-44
- [13] 冯春丽, 唐毅谦, 赵悦. 一种改进的微粒群优化算法[J]. 南京航空航天大学学报, 2006, 38(sup): 58-61
- [14] 龚纯, 王正林. 精通 MATLAB 最优化计算[M]. 北京: 电子工业出版社, 2009