

粒子群算法种群多样性控制方法研究

陈基漓^{1,2}

(1 桂林理工大学 信息科学与工程学院, 广西 桂林 541004; 2 广西空间信息与测绘重点实验室, 广西 桂林 541004)

摘 要: 粒子群算法是一直典型的群体智能算法, 种群多样性是影响其优化性能的一个重要因素, 对几种影响种群多样性的方法如微粒的初始位置、迭代过程微粒个体执行变异操作、异步更新策略、带线性递减惯性权重的异步更新策略等进行研究. 并通过对典型测试函数进行实验, 给出了几种能提高微粒群算法优化性能的多样性控制方法.

关键词: 粒子群算法; 多样性; 变异; 异步更新; 线性递减惯性权重

中图分类号: TP301.6

文献标识码: A

文章编号: 1000-7180(2013)06-0006-04

The Research on the Control Methods of Population Diversity in Particle Swarm Algorithm

CHEN Ji-li^{1,2}

(1 College of Information Science and Engineering, Guilin University of Technology, Guilin 541004, China;

2 Guangxi Key Laboratory of Spatial Information and Geomatics, Guilin 541004, China)

Abstract: Particle swarm optimization algorithm is a typical swarm intelligence algorithm, the diversity of the population is an important factor to affect the optimize performance of particle swarm optimization algorithm. Several methods that control the population diversity are analyzed, such as the initial position of the particle, the mutation of particle that perform during the iterative process, asynchronous update strategy, and asynchronous update strategy with a linearly decreasing inertia weight. Some test functions are used for simulation experiments, and several control methods of the diversity that can improve the performance of particle swarm optimization are given.

Key words: Particle Swarm Optimization (PSO); diversity; mutation; Asynchronous update pattern; linearly decreasing inertia weight

1 引言

粒子群(PSO)算法通过模拟鸟群觅食的飞行行为, 实现最优解的搜索. PSO 算法具有收敛速度较快, 易于实现等优点^[1]. PSO 算法存在的问题是容易陷入局部最优解中, 对于复杂问题后期收敛速度较慢等缺陷^[2].

粒子群算法出现早期收敛, 陷入局部最优解的一个主要原因是种群多样性随着收敛次数的递增而衰减^[3]. 种群多样性是影响算法搜索范围的一个重要指标. 当种群具有较高的多样性时, 微粒个体分布

在搜索空间中呈分散状态, 种群的搜索空间较大, 具有很强的全局搜索能力; 反之当种群具有较低的多样性时, 微粒个体分布则相对集中, 种群的搜索空间较小, 具有较强的局部搜索能力^[4]. 种群多样性程度的高低, 对算法的收敛速度、求解精度都有很大的影响.

种群多样性是指群体中各个体之间的差异程度, 微粒群模型中, 个体的特征包括位置、速度、历史最优值、认知能力等^[5], 本文重点研究微粒个体位置多样性、最优值多样性及其控制方法.

收稿日期: 2012-09-01; 修回日期: 2012-11-05

基金项目: 广西空间信息与测绘重点实验室开放基金项目(桂科能 1103108-16)

2 粒子群算法

2.1 算法基本原理

微粒群算法中,一个微粒对应于一个独立的解,微粒的维数与问题解的维数一致.每个微粒的特征包括微粒的当前位置(即对应于搜索空间中的一个解),飞行速度,微粒自身所经过的历史最好位置,整个微粒群的参数包括微粒个数,整个群体所经过的全局历史最好位置.群体中的所有微粒通过跟踪自身历史最好位置以及追随群体历史最好位置在搜索空间中寻找最优解.

算法采用适应度函数衡量微粒当前位置与实际最优解之间的差异.算法首先随机初始化所有微粒的位置及速度,并计算每个微粒的适应度函数值,每次迭代都记录每个微粒的历史最好位置以及整个微粒群的全局历史最好位置,每个微粒根据自身历史最好位置和群体的全局最好位置更新速度和位置,通过不断迭代,逼近最优解对应的位置.

2.2 标准粒子群算法流程

标准粒子群算法流程如下^[1]:

(1)初始化.算法首先初始化群体规模,个体的速度、位置等初始值,并计算每个微粒的适应度函数值;

(2)对每个微粒,比较其当前适应度函数值与自身的历史最好位置,如果前者更好,则更新个体的历史最好位置;对每个微粒,比较其适应度函数值与微粒群全局最好位置,若前者更好,则更新微粒群全局最好位置;

(3)根据式(1)调整每个微粒的速度和位置,其中 p_i 为单个微粒的历史最好位置, p_g 为整个微粒群的全局历史最好位置, c_1 、 c_2 为学习因子, r_1 、 r_2 为 0 至 1 之间的随机数, w 为惯性权重;

(4)如果满足终止条件则算法停止,否则转(2)继续下一代的迭代运算.算法终止条件一般可设为指定的迭代次数或者是所求最优解的精确度.

$$v_i(t+1) = \omega v_i(t) + c_1 r_1() (p_i - x_i(t)) + c_2 r_2() (p_g - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

由式 1) 可知,影响微粒个体速度值有三个因素. $\omega v_i(t)$ 为惯性部分,微粒根据自身的速度进行惯性运动, w 值越大则说明微粒速度受自身惯性影响程度越大,有利于跳出局部最优值,而较小的 w 值则有利于算法收敛. $c_1 r_1() (p_i - x_i(t))$ 侧重考虑微粒当前位置与该微粒的历史最好位置之间的差异,

重点考虑微粒自身的飞行经验,体现的是微粒个体的认知能力. $c_2 r_2() (p_g - x_i(t))$ 则考虑微粒当前位置与整个微粒群的全局历史最好位置之间的差异,实现微粒群中个体之间的信息共享,体现的是微粒群整体的认知能力.

3 位置多样性

位置是微粒个体的一个最基本的属性,位置多样性则是影响算法搜索范围的一个重要指标.影响位置多样性的因素包括微粒群中个体的初始位置,迭代过程中微粒个体位置的变化等.标准微粒群算法中微粒个体初始位置及速度采用随机方式生成,速度更新公式中的学习因子采用固定值,惯性权重采用简单的随机方式生成,迭代过程中微粒个体位置更新只考虑微粒前次迭代所处位置及速度更新,无变异情况出现.针对这些问题,一些学者提出了改进算法,如曾建潮提出随机惯性权重^[6], Y Shi 提出线性递减惯性权重等^[7],这些改进算法都在一定程度上提高了微粒群算法的收敛性,能有效地避免过早陷于局部有化解中.本文从微粒初始位置分布、迭代过程增加变异策略两种途径入手,提出控制微粒位置多样性的方法.

3.1 群体初始值均匀分布

标准微粒群算法中微粒初始位置由随机数决定,当微粒数量偏少时,随机性对初始位置的分布影响更为突出.群体中的微粒的初始位置在搜索空间中的分布越广泛,则微粒初始位置的多样性程度越高.群体初始值均匀分布方式将微粒个体均匀散落在搜索空间中,将搜索空间按照微粒群中微粒个数分为若干子区间,每个子区间中分布一定数量的微粒个体,在每个子区间中用随机数方式产生微粒个体的初始位置.

设群体中微粒个数为 N ,搜索空间某一维取值范围为 $[0, X]$,初始值均匀分布方式将搜索空间分成 10 个子区间,每个子区间中放 $N/10$ 个个体,子区间中 $N/10$ 个个体的分布以随机数方式实现.标准微粒群算法中微粒初始分布实现方式为:

对于每一个微粒 i

微粒 i 中的每一维 j

$$\text{particles}[i].\text{position}[j] = \text{random}() * X;$$

其中 funDim 为微粒维数.采用均匀分布,则初始分布实现方式更改为:

对于每一个微粒 i

微粒 i 中的每一维 j

```

particles[i].position[j]=
random() * (X/10)+(i/(double)N/
(X/10.0)) * 10;

```

若搜索空间取值范围为 $[-X, X]$, 则微粒初始位置分布增加一个随机数判断, 使初始值取值为正、负的概率相同即可. 以均匀初始值分布为例, 将实现方式修改为:

```

对于每一个微粒 i
微粒 i 中的每一维 j
    if (random() > 0.5)
        particles[i].position[j]=
        random() * (X/10)+(i/(double)N
/(X/10.0)) * 10;
    else
        particles[i].position[j]=
        - random() * (X/10)+
(i/(double)N/(X/10.0)) * 10;

```

3.2 个体变异

变异是进化算法中的常用算子, 通过对一定比例的个体产生较大的突变, 增强解空间的搜索能力. 变异算子思想来源于物种进化中的基因突变, 与基因突变类似, 变异算子能扩大算法的解搜索范围, 对算法寻优能力的提升起关键作用. 标准微粒群算法没有采用变异算子, 微粒个体位置的更新由自身及群体的认知能力决定, 当迭代进行到一定次数时, 微粒容易出现向局部最优解聚集的现象, 使算法过早陷入局部优化解中. 当搜索出现聚集时, 微粒群体的多样性程度就随之降低. 在算法中增加变异操作, 通过变异使微粒位置在搜索空间中产生较大幅度的变化, 保持一定的多样性程度.

随机变异是一种最简单的变异策略, 该变异策略最先在实数编码的遗传算法中应用. 随机变异在指定的取值范围内, 用随机值替代将要发生变异的基因值. 在标准微粒群算法中增加随机变异操作, 算法每完成一次迭代, 根据变异率随机选择一些微粒个体进行变异操作. 为方便操作, 进行变异操作的微粒只随机选择其中的一个维度进行变异. 具体实现流程为:

对于每一个微粒 i 根据变异率决定是否需要进行变异操作, 如果是, 则

```

{
d=(int)(Math.random() * funDim);
particles[i].position[d]=X * Math.random
();

```

4 最优值多样性

微粒群全局历史最好位置体现整个群体的认知能力, 在速度更新公式中也起着关键的作用. 标准微粒群算法采用的是同步更新模式, 在一次迭代过程中, 当所有的微粒都完成速度、位置、个体历史最好位置的更新之后, 再对全局历史最好位置进行更新. 实际上, 当某一个微粒完成更新之后, 如果该微粒的个体历史最好位置已经优于整个微粒群的全局历史最好位置, 同步更新模式无法满足全局历史最好位置的实时更新, 导致对群体整体认知能力对微粒个体的影响作用减弱.

4.1 异步更新模式

罗建宏提出异步更新模式, 异步更新模式主要目的是提高微粒群全局历史最好位置更新的实时性. 异步更新模式实现策略是当每个微粒发生更新之后, 若该微粒的历史最好位置已经优于群体全局历史最好位置, 则更新全局历史最好位置, 同时对其他微粒进行更新. 使全局历史最好位置能实时反映出整个微粒群体的认知能力^[8]. 在异步更新模式下, 共享信息能够得到实时更新. 异步更新模式能提高整个微粒群体全局历史最好位置以及单个微粒个体历史最好位置的多样性程度.

异步更新模式与同步更新模式相比, 对搜索空间的最优位置能实时给出直接的反馈, 而同步更新模式一次迭代只反馈一次.

4.2 线性递减惯性权重的异步更新模式

上述对微粒个体多样性的控制方式以提高多样性程度为主, 实际上, 多样性程度并非总是越高越好. 随着迭代次数的增加, 局部搜索能力的提高也对解的质量有影响. 在异步更新模式中引入线性递减惯性权重, 除了采用异步更新模式提高最优值多样性之外, 随着迭代次数的增强, 惯性权重以线性方式减小, 微粒个体的惯性程度降低, 微粒个体位置的多样性程度相应减弱, 以提升解的局部搜索能力.

线性递减惯性权重在迭代初期, 具有较大的惯性权重值, 微粒个体依赖自身的惯性程度较高, 算法具有较强的搜索能力, 能在搜索空间中较大范围进行优化解的寻找. 随着迭代次数的增加, 惯性权重值逐步减小, 算法的开发能力相应增强, 以满足搜索速度的需求. 惯性权重按迭代次数在 $[0.5, 0.1]$ 范围内递减.

5 仿真实验

5.1 测试函数

对采用不同多样性控制方法的微粒群优化算法, 使用如下4个测试函数进行函数优化实验, 这些函数是常用的优化测试函数, 且全局最小值都为0.

函数 f1: Ackleys

$$f(x) = 20 + e - 20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) \quad -30 \leq x_i \leq 30$$

函数 f2: Griewank

$$f(x) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) \quad -600 \leq x_i \leq 600$$

函数 f3: Rastrigin

$$f(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10] \quad -5.12 \leq x_i \leq 5.12$$

函数 f4: Sphere

$$f(x) = \sum_{i=1}^n x_i^2 \quad -100 \leq x_i \leq 100$$

5.2 参数设置

根据不同的多样性控制方法, 对以下几种微粒群模型进行测试, 固定惯性权重为0.4, 无变异操作, 微粒群初始位置随机分布的为模型1; 固定惯性权重为0.4, 无变异操作, 微粒群初始位置均匀分布的为模型2; 固定惯性权重为0.4, 有变异操作, 微粒群初始位置随机分布的为模型3; 固定惯性权重为0.4, 有变异操作, 微粒群初始位置均匀分布的为模型4; 固定惯性权重为0.4, 异步更新模式为模型5; 模型1即为同步更新模式; 线性递减惯性权重(0.5—0.1)异步更新模式为模型6. 对每种模型, 针对每一个测试函数, 测试函数维数分别设为2, 10, 30, 50, 微粒群中微粒数量设为100, 迭代次数设为500, 每个算法运行次数为100次, 记录最优值和平均值.

5.3 仿真实验结果

测试函数 f1 至 f4 的实验结果表明, 微粒个体初始位置采用均匀分布的微粒群模型优化性能优于

采用随机分布的模型, 采用变异策略的微粒群模型优化性能远强于比未采用变异策略的模型, 采用异步更新策略的模型性能也优于同步更新策略, 在异步更新策略中使用线性递减惯性权重的模型优于异步更新模型. 微粒群中微粒个数分别设为50, 30, 对以上测试函数进行实验, 也获得同样的结论.

6 结束语

微粒群算法中的多样性是影响算法优化能力的关键因素, 本文从微粒初始位置分布、变异策略、最优值多样性等方面入手对控制多样性的方法进行了分析, 对常用测试函数的仿真实验表明, 变异策略、初始值均匀分布、异步更新模式均对提升算法的优化性能有较好的作用. 进一步的研究是在微粒群算法中建立自适应的多样性控制策略, 并应用于实际的优化问题中.

参考文献:

- [1] Kennedy J, Eberhart R C. Particle swarm optimization [C] // Proc IEEE International Joint Conference on Neural Networks. Australia: Perth, 1995.
- [2] 吴启迪, 康琦, 汪镭. 自然计算导论[M]. 上海: 上海科学技术出版社, 2011.
- [3] 方伟, 孙俊, 须文波. 一种多样性控制的粒子群优化算法[J]. 控制与决策, 2008, 23(8): 863—868.
- [4] 焦巍, 刘光斌. 基于多样性反馈的粒子群优化算法[J]. 计算机工程, 2009, 35(22): 202—204.
- [5] 程适, 史玉回. 基于 L_1 范式的粒子群算法群体多样性研究[J]. 计算机科学, 2011, 38(7): 190—194.
- [6] 胡建秀, 曾建潮. 具有随机惯性权重的 PSO 算法[J]. 计算机仿真, 2006, 23(8): 164—167.
- [7] Shi Y, Eberhart R C. Empirical study of particle swarm optimization [C] // Proceedings of the 1999 Congress on Evolutionary Computation. Piscataway, NJ, IEEE Service Center, 1999: 1945—1950.
- [8] 罗建宏, 张忠能. 并行仿真的粒子群优化算法异步模式研究[J]. 计算机仿真, 2005, 22(6): 68—70.

作者简介:

陈基漓 女, 硕士, 副教授. 研究方向为智能计算、数据挖掘.