

# 递进多目标粒子群算法的设计及应用

徐斌<sup>1</sup> 俞静<sup>2,3</sup>

(中央财经大学会计学院 北京 100081)<sup>1</sup> (中国科学院研究生院 北京 100190)<sup>2</sup>  
(中国科学院虚拟经济与数据科学研究中心 北京 100190)<sup>3</sup>

**摘要** 在现有递进进化技术研究的基础上,提出了一种递进多目标 PSO 算法,该方法每进化一定代数后以一定策略对群体进行重构,以提高算法对解空间的遍历性,从而较大程度上避免算法的早熟。该算法采用非劣解等级优先的选择方式复制后代,通过递进层次间对部分非劣解个体执行局部搜索,加快全局非劣解集的进化。采用递进 PSO 算法与非递进 PSO 算法对一些典型优化问题进行对比分析,验证了算法求解多目标函数优化问题的有效性。研究表明,通过研究惯性因子确定的随机数方法,比目前的固定数确定方法具有一定的先进性。

**关键词** 递进进化,多目标算法,粒子群算法,随机惯性因子

**中图分类号** O241, O244 **文献标识码** A

## Multi-objective PSO Algorithm Based on Escalating Strategy

XU Bin<sup>1</sup> YU Jing<sup>2,3</sup>

(School of Accountancy, Central University of Finance & Economics, Beijing 100081, China)<sup>1</sup>

(School of Graduate, China Academy of Sciences, Beijing 100190, China)<sup>2</sup>

(Fictitious Economy & Data Science, China Academy of Sciences, Beijing 100190, China)<sup>3</sup>

**Abstract** A multi-objective PSO algorithm based on escalating strategy was proposed. The main idea of this escalating strategy is to re-generate the whole evolutionary population with some technology, which results in a new population significantly indifferent from the old one while inheriting the evolutionary information from the history. By this way, the performance on global convergence can be enhanced, and premature can be avoided simultaneously. A neighborhood search procedure was imposed on some selected Pareto solutions to accelerate the evolution process for reaching a global Pareto set with well distribution. Some typical multi-objective optimization test problems were analyzed with escalation PSO and non-escalation PSO respectively to verify the effectiveness of the new algorithm. The details about how to select appropriate escalating parameters and their effect on the performance of EMPSO were also investigated to show that the EMPSO with random inertia weight factor has some advantage over than that of fixed inertia weight.

**Keywords** Escalation evolution, Multi-objective optimization, PSO algorithm, Random inertia weight

## 1 引言

粒子群优化算法(Particle Swarm Optimization, PSO)是 Kennedy 和 Eberhart 于 1995 年提出的一种优化算法<sup>[1]</sup>,算法的设计是模仿鸟类飞行的原理而进行的,通过设计个体全局最优解与整体全局最优解的方法来修正前进路线,从而达到全局最优的目标。PSO 算法由于容易理解、易于实现,一经提出就受到了学术界和实务界的热烈追捧。针对 PSO 算法的改进主要围绕如下两个方面展开:其一是把 PSO 算法的解空间的适应范围从连续空间拓展到离散空间<sup>[2]</sup>,从求解单目标问题逐步运用到求解多目标问题中去,并且从起初没有约束条件的优化问题逐步过度到包括约束条件的优化问题。其二针对 PSO 的操作算子进行改进,不断地通过加入诸如遗传

算子、改进操作算子以及融合类如基于小生境技术的启发式算法等等方法,从而加快了算法运行速度,提高了算法求解效果。

PSO 算法初始是针对求解单目标优化而提出的,不久就被引入到求解多目标问题领域中,称为 MOPSO 算法。国外学者对此进行了探索,取得了不少原创性的研究成果,虽然迄今有文献<sup>[3-6]</sup>研究了如何运用 PSO 算法求解多目标优化问题(Multi-Objective Optimization Problem, MOP),但是其基本求解思路并没有跳出传统的进化思路,这样就形成了无论 PSO 算法如何改进,基本上都是在随机初始群体的基础上对进化算子进行操作,使得算法容易早熟而不能够搜索到更优化的全局解空间。例如 Coello 和 Lechuga<sup>[7]</sup>在非劣最优概念的基础上应用了一个“容器”来记录已找到的非支配向量,并

到稿日期:2009-10-03 返修日期:2009-12-03 本文受中央财经大学 211 工程第 3 期,国家自然科学基金委创新研究群体科学基金项目数据挖掘与智能知识管理理论与应用研究(70921061),人民币汇率制度改革相关问题研究(70950002),主权财富基金(70840010)资助。

徐斌(1966-),男,博士,主要研究方向为财务决策支持系统、数值仿真, E-mail: zhuohongxubin@163.com;俞静(1969-),女,博士后,主要研究方向为数据分析。

用这些解来指导其他粒子的飞行。Parsopoulos 和 Vrahatis<sup>[8]</sup>应用了权重聚合的方法, Hu 和 Eberhart<sup>[9]</sup>应用了动态邻近的 PSO 算法来求解多目标优化问题。国内研究也不甘落后,对 PSO 算法的改进与 MOPSO 算法的研究迄今也取得了令人乐观的成果。张利彪、周春光等(2004)<sup>[10]</sup>通过选取全局目标和个体目标平均数的方法优化粒子优化方向。李宁和邹彤等(2005)<sup>[11]</sup>采用小生境技术,并加入部分变异和边界变异的机制大大提高了多目标优化解集合的分散性和均匀性。张宝菊、单国全等(2006)<sup>[12]</sup>提出用粒子进化速度的变步长的改进方法来改进进化结果。徐杰和黄德先(2007)<sup>[13]</sup>、王金华和尹泽勇(2007)<sup>[14]</sup>则通过在 PSO 操作算子中加入遗传算法操作算子改进了 PSO 算法。可见,国内外学术界学者虽然对 PSO 算法进行了很多的改进,但是几乎无一例外地都是在 PSO 原有操作因子的基础上进行的。令人可喜的是,师瑞峰和周弘等(2005)<sup>[15]</sup>针对遗传算法的改进提出了一种递进化遗传算法,该方法是在每进化一定代数后以一定策略对群体进行重构,以提高算法对解空间的遍历性,从而较大程度上避免算法的早熟。该算法采用非劣解等级优先的选择方式复制后代,降低算法的时间复杂性,通过递进层次间对部分非劣解个体执行局部搜索,加快全局非劣解集的进化。本文准备在借鉴 Coello 等人<sup>[16]</sup>的小群体多目标进化中进化至一定代数后对小群体采取重新初始化的思路,以及师瑞峰等人<sup>[15,17]</sup>提出的递进遗传算法的基础上提出一种基于 PSO 算法的递进多目标 PSO 算法,从而为 PSO 算法的改进提出新的思路。

但是,令人遗憾的是,对于 PSO 这一充满前景的算法的改进还是局限于上述所叙述的两个方面,递进化思想没有得到应用。因此本文准备在借鉴已有递进化思想的基础上,提出一种递进粒子群算法来求解多目标问题。

## 2 PSO 算法

粒子群优化算法(PSO)源于对鸟类寻食行为的研究,研究者发现鸟群在飞行过程中经常突然改变方向、散开、聚集,从而造成表面上其飞行行为具有不可预测性,但是其总体行为保持一致性,个体与个体之间也保持着最适当的距离。发现者通过对类似生物群体行为的研究,发现生物群体中存在着一种社会共享机制,它为群体的进化提供了一种社会信息共享机制,也为群体的进化提供了一种优势,这便是 PSO 算法形成的基础。

PSO 算法中的每个粒子就是解空间中的一个解,它根据自己的飞行经验和同伴的飞行经验来调整自己的飞行。每个粒子在飞行过程中所经历过的最好位置,就是整个群体目前找到的最优解。前者叫做个体极值(pBest),后者叫做全局极值(gBest)。实际操作中通过由优化问题所决定的适应值(Fitness Value)来评价粒子,这些粒子通过上述两个极值不断更新自己,从而产生新一代群体。

显然,在 PSO 算法中每个粒子都可以看作是解空间中的一个点。如果粒子的群体规模为  $N$ ,则第  $i(i=1,2,\dots,N)$  个粒子的位置可表示为  $X_i$ ,它所经历的“最好”位置记为  $pBest[i]$ ,它的速度用  $V_i$  表示,群体中的“最好”粒子位置用  $gBest$  表示,所以粒子  $i$  根据下面的公式来更新自己的深度和位置:

$$V_i = wV_i + c_1 \times rand() \times (pBest[i] - X_i) + c_2 \times rand() \times (gBest - X_i)$$

$$X_i = X_i + V_i$$

其中, $c_1, c_2$  为常数,称为学习因子; $rand()$  是  $[0,1]$  上的随机数, $w$  为惯性因子(inertia weight)。上述公式由 3 部分组成,第 1 部分是粒子先前的速度,说明了粒子目前的状态;第 2 部分是认知部门(cognition modal),表示粒子本身的思考;第 3 部分为社会信息共享部分(social modal)。3 部分共同决定了粒子的空间搜索能力。第 1 部分起了平衡全局和局部搜索的能力,第 2 部分使得粒子具有足够强大的全局和局部搜索的能力,第 3 部分体现了粒子之间的信息共享能力,这样 3 部分的共同作用才能够有效地到达最好位置。当然,粒子在不断地根据速度调整自己的位置时,还要受到最大速度  $V_{max}$  的限制,当个体速度  $V_i$  超过最大速度  $V_{max}$  时,将被限制为  $V_{max}$ 。

## 3 递进多目标 PSO 算法

### 3.1 搜索策略

#### 3.1.1 非劣解分级筛选

MOP 的一般数学描述形式如下:

$$\min(f_1(x), f_2(x), \dots, f_m(x)) \quad x \in D \quad (1)$$

其中, $f_i(x)(i=1,2,\dots,m)$  表示第  $i$  个优化目标, $x=(x_1, x_2, \dots, x_n)^T$  为决策变量, $D$  表示决策变量的可行域。基于 Pareto 最优解的 PSO 算法在对个体求解目标值之后,需要对群体进行 Pareto 分级,以获得当前群体的不同非劣解等级的子集,本文采用文献[18]中的 Pareto 分级策略。

假设进化群体规模为  $N$ ,在计算出每一代群体中各个体的  $m$  个目标函数值之后,对群体进行 Pareto 分级。首先考察整个群体,通过个体的目标函数值与其他个体目标函数值之间的比较,选出非劣点(设有  $n_1$  个),令其所对应的决策变量为第 1 级非劣解。然后考察剩余  $N-n_1$  个个体,按照同样的法则选出其中的非劣解(设有  $n_2$  个),作为第 2 级非劣解。此过程持续至整个群体分级完毕。形如式(1)的 MOP,群体分级的详细实现过程参见文献[15]。

#### 3.1.2 稀疏原则

一般 PSO 算法在求解多目标问题时基本上都是设计类似于单目标的适应度函数来进行比较,这种方法简便直观,便于操作,但是却违背了多目标 Pareto 解集的本质。尽管目前有的论文也充分地利用了稀疏原则来确定各个解的排序,但是基本上也是通过设计一个能够反映稀疏情况的适应度函数来达到目的间接方法。这种方法虽然在一定程度上能够反映解的稀疏程度,但是如何处理来自于不同群体所产生的解之间的比较却留下不尽完美之处。本文准备在此研究的基础上提出一种直接反映解的稀疏程度的排序法则。具体方法叙述如下:对于新旧群体中的 Pareto 解集按照上述方法进行排序分级,对于分级相同的解则按照稀疏度进行排序,排序第一的默认适应度最好,依次类推。

#### 3.1.3 精英保留策略

算法开始运行时,设置一个外部的伴随非劣解集以保留迄今为止所发现的非劣解。每代进化完成后,将当前非劣解集加入伴随非劣解集后对外部群体重新分级,剔除其中的劣解后更新伴随非劣解集。这一精英保留策略可以使算法随整个进化逐步收敛至全局 Pareto 非劣解集。

#### 3.1.4 随机惯性因子和学习因子

伴随群体重新进行 Pareto 分级,剔除劣解后选取距离其

他个体距离最大的部分非劣解个体进行局部搜索;

现有的 PSO 算法基本上都是赋予学习因子  $c_1, c_2$  和惯性因子  $w$  为常数,但是其实这些因子本身也对算法的进化速度与进化效果有影响。但是,学习因子  $c_1, c_2$  乘以  $rand()$  的效果实际已经达到了随机取数的目的,因此只需要赋予惯性因子  $w$  随机数即可。

### 3.2 递进多目标 PSO 算法

本文对现有多目标 PSO 算法的最大改进之处是在现有算法中加入了递进结构:算法每进化规定代数后,保留已找到的非劣解集信息,并对整个群体进行大规模而有序的更新和重构,以使算法从搜索空间中的一个区域跳跃至另一区域,从而避免算法陷入早熟;而重构后的群体又能以一定的形式继承先前的搜索信息,以保持整体搜索过程的一致性。每一次群体重构称为搜索过程向前递进一层。本文采用的递进策略具体实现方式为:利用外部非劣解集伴随群体保存迄今为止找到的非劣解集信息;在多目标 PSO 算法进化至规定代数后,从伴随群体中引入部分非劣解个体进入下一递进进化层,并通过初始化方式生成下一进化层进化群体不足的个体。本文提出的进化算法与现有多目标 PSO 算法的另一不同之处是,在每代进化完成后对其中非劣解个体执行局部搜索策略,通过这一改进可使算法获得更好非劣解集。本文提出的递进多目标 PSO 算法如图 1 所示,其实现过程如下。

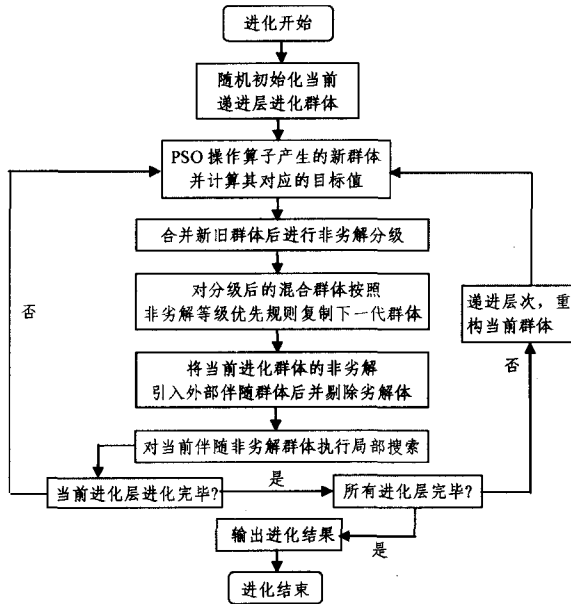


图 1 递进多目标 PSO 算法结构

(1)生成外部非劣解集伴随群体并置为空集,置当前递进层次为 1;

(2)初始化进化群体,计算群体中个体的目标值,根据目标值对群体进行 Pareto 分级;

(3)将当前群体的非劣解个体加入外部伴随群体,对伴随群体重新进行 Pareto 分级,剔除劣解后更新伴随群体;

(4)对当前群体执行 PSO 操作,生成新群体;

(5)将新、旧群体合并为复制候选集,剔除重复个体后,按照非劣解等级由低到高逐级复制个体进入下一代群体,若某一等级个体复制后整个群体超出进化群体规模,则最后进入的等级内部按照由疏到密的顺序逐一复制个体进入下一代,直至下一代进化群体达到限定规模;

(6)计算新一代进化群体中个体的目标值,根据目标值对群体进行 Pareto 分级;

(7)将进化群体中的非劣解个体加入外部伴随群体后;

(8)若当前进化层次进化尚未完毕,转(4);若当前进化层次进化完毕,但总层数未达到规定值,则将进化层次向前递进一层,转(2);若达到所规定递进层次,则转下一步;

(9)输出伴随非劣解集为所求解多目标优化问题的非劣解集,递进优化完毕。

## 4 算例测试

实验使用了文献[19,20]中提出的测试函数 1 和测试函数 2。其分别表示如下:

测试函数 1:

$$\min f_1(x, y) = (x^2 + y^2)^{1/8}$$

$$\min f_2(x, y) = ((x-0.5)^2 + (y-0.5)^2)^{1/4}, x, y \in [-10, 10]$$

测试函数 2:

$$\min f_1 = x_1$$

$$\min f_2 = 1 - [f_1/g]^2$$

$$g = 1 + (9/(n-1)) \sum_{i=2}^n x_i \quad x_i \in [0, 1] \quad n=30$$

为了比较不同算法的优劣,除了可以把不同算法所获得的 Pareto 有效解前沿绘制在一张图形上进行直观比较之外,也可以运用 Zitzler<sup>[21]</sup>提出的评价算法对两两之间非劣解集优劣的  $C$  指标进行评价,其定义公式如下:

$$C(A, B) = |\{b \in B \mid \exists a \in A: a > b\}| / |B|$$

其中,  $A, B$  是两种算法优化获得的非劣解集,  $C$  指标是一种值域定义在  $(0, 1)$  上、用来刻画  $(A, B)$  之间偏序性能的指标;  $C(A, B) = 1$  说明对于  $B$  集中任一非劣解,  $A$  集中总存在“优于”它的解个体;  $C(A, B) = 0$ , 说明对于  $B$  集中任一解个体,  $A$  集中都不存在“优于”它的解个体。

### 4.1 递进与非递进多目标 PSO 算法比较

对于测试函数 1 和测试函数 2, 本文运用递进和非递进多目标 PSO 算法进行求解, 它们共同参数设置如下: 种群规模  $POPSIZE=300$ , 非劣解保留群体规模  $RESERVE=500$ , 进化代数  $MaxGen=100$ , 惯性因子  $weight=0.7$ , 学习因子  $c_1=c_2=1$ 。对于递进算法则设置递进层  $ESCALATION=10$ , 于是它们的 Pareto 解集如图 2 所示。

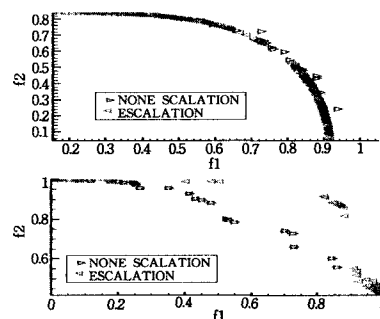


图 2 测试函数 1 和测试函数 2 的递进与非递进 PSO 算法 Pareto 解比较图

针对测试函数 1 来说, 递进算法与非递进算法获得的 Pareto 解的个数分别为 218 和 23, 图 1 也显示递进算法要优于非递进算法。经过简单计算, 可以得到算法比较的数据如下:

C(递进算法,非递进算法)=0.6087

C(非递进算法,递进算法)=0.0321

针对测试函数 2 来说,递进算法与非递进算法获得的 Pareto 解的个数分别为 217 和 36,图 1 也显示递进算法要优于非递进算法。经过简单计算,可以得到算法比较的数据如下:

C(递进算法,非递进算法)=0.0556

C(非递进算法,递进算法)=0.0092

#### 4.2 基于随机惯性因子和固定惯性因子的递进多目标 PSO 算法比较

一般来说,目前对 PSO 算法的改进基本上都是集中在整体算法结构上进行的,当然这是算法改进的主流研究方向,但是也不能够忽略一些参数的取值方法对最终求解的影响。本文对此进行了探索,通过设置惯性因子  $weight=0.7rand(0,1)$  来测试其对 Pareto 解的影响,其它参数如上,并且采用递进算法结构进行求解。测试结果如图 3 所示。



图 3 基于固定和随机惯性因子的测试函数 1 和测试函数 2 的递进 PSO 算法 Pareto 解比较图

针对测试函数 1 来说,基于随机数和固定数的递进算法所获得的 Pareto 解的个数分别为 228 和 218,图 1 也显示递进算法要优于非递进算法。经过简单计算,可以得到算法比较的数据如下:

C(随机数惯性因子,固定数惯性因子)=0.2064

C(固定数惯性因子,随机数惯性因子)=0.1930

针对测试函数 2 来说,基于随机数和固定数的递进算法所获得的 Pareto 解的个数分别为 216 和 217,图 1 也显示递进算法要优于非递进算法。经过简单计算,可以得到算法比较的数据如下:

C(随机数惯性因子,固定数惯性因子)=0.9992

C(固定数惯性因子,随机数惯性因子)=0.0691

**结束语** 本文提出了一种递进多目标 PSO 算法,在此基础上对 PSO 算法中的惯性因子固定数取值改为随机数取值进行了尝试,并且运用两个比较常见的测试函数对算法进行了测试比较,比较结果显示本文的改进有助于 Pareto 解的优化。但是,本文的研究仅限于两个目标函数,而对于超过两个目标函数的优化问题研究将是进一步研究的课题。

#### 参考文献

[1] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[C]// Proc. of the 6th Int'l Symposium on Micro Machine and Human Science. Piscataway, NJ: IEEE Service Center, 1995:39-43

[2] Kennedy J, Russell C. Eberhart: A Discrete binary version of the

particle swarm algorithm[J]. IEEE Transactions, 1997; 4104-4108

[3] Parsopoulos K E, Vrahatis M N. Particle swarm optimizer in noisy and continuously changing environments [M] // M H Hamza, ed. Artificial Intelligence and Soft Computing. Iasted, ACTAPress, 2001; 289-294

[4] Parsopoulos K E, Vrahatis M N. Particle swarm optimization method for constrained optimization problems[C]// Euro-Int'l Sympo Computational Intelligence 2002. Slovakia, 2002

[5] Eberhart R C, Hu X. Human tremor analysis using particle swarm optimization[C]// IEEE Congress on evolutionary computation (CEC1999). Washington D. C., USA, 1999

[6] Shiand Y, Eberhart R. A modified particle swarm optimizer [C] // IEEE Int'l Conf on Evolutionary Computation. Anchorage, Alaska, 1998

[7] Coello C A, Lechuga M S. MOPSO: A proposal for multiple objective particle swarm optimization[C]//IEEE Congress on Evolutionary Computation (CEC 2002). Honolulu, Hawaii, USA, 2002

[8] Parsopoulos K E, Vrahatis M N. Particle swarm optimization method in multiobjective problems [C] // Proc. of the ACM Symp. on Applied Computing 2002 (SAC 2002). New York: ACM Press, 2002; 603-607

[9] Hu X, Eberhart R C. Multiobjective optimization using dynamic-neighborhood particle swarm optimization[C]//IEEE Congress on Evolutionary Computation (CEC 2002). Honolulu, Hawaii, USA, 2002

[10] 张利彪,周春光,马铭,等. 基于粒子群算法求解多目标优化问题[J]. 计算机研究与发展, 2004, 41(7): 1286-1291

[11] 李宁,邹彤,孙德宝,等. 基于粒子群的多目标优化算法[J]. 计算机工程与应用, 2005, 23: 43-46

[12] 张宝菊,单国全,齐名军. 求解非线性约束优化问题改进的粒子群算法[J]. 天津师范大学学报, 自然版, 2006, 26(2): 73-76

[13] 徐杰,黄德先. 基于混合粒子群算法的多目标车辆路径研究[J]. 计算机集成制造系统, 2007, 13(3): 573-579

[14] 王金华,尹泽勇. 基于 NSGA-II 和 MOPSO 融合的一种多目标优化算法[J]. 计算机应用, 2007, 27(11): 2817-2823

[15] 师瑞峰,周泓,谭小卫. 递进多目标遗传算法[J]. 系统工程理论与实践, 2005, 12: 48-56

[16] Ceollo C A C, Pulido G T. A micro-genetic algorithm for multiobjective optimization[A]// Proceedings of the 1st International Conference on Evolutionary Multi-Criterion Optimization [C]. Springer-Verlag, 2001: 126-140

[17] 师瑞峰,周泓. 一种求解双目标 flow shop 排序问题的进化算法[J]. 管理科学学报, 2007, 10(5): 11-19

[18] Deb K. Multi-objective Optimization Using Evolutionary Algorithms[M]. Wiley, John & Sons, 2001

[19] Joanna L, Eiben A E. A multi-sexual genetic algorithm for multi-objective optimization[C]// Toshio Fukuda, Takeshi Furuhashi, eds. The 1996 Int'l Conf on Evolutionary Computation. Nagoya, Japan, 1996

[20] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms[J]. Empirical results, Evolutionary Computation, 2000, 8(2): 173-195

[21] Zitzler E. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications [D]. Switzerland: Swiss Federal Institute of Technology, 1999