

基于多准则决策的塔状粒子群优化算法

王益平, 蓝 艇*, 邬杨波

(宁波大学 信息科学与工程学院, 浙江 宁波 315211)

摘要: 针对经典粒子群优化算法存在早熟、收敛精度低和收敛速度慢的问题, 提出了一种新的改进算法. 该算法采用了塔状优化互联机制, 底层粒子群负责寻找局部最优解, 顶层粒子负责收集、反馈全局最优解, 为底层种群提供全局最优信息, 建立共享学习机制. 顶层粒子一旦发现停滞现象, 将通知底层粒子群采用细菌觅食优化、随机初始化等停滞优化策略, 以改善粒子群的收敛速度. 实验结果表明, 与同类算法相比, 改进算法具有更好的寻优能力, 改善了粒子群的收敛精度和收敛速度.

关键词: 粒子群优化; 塔状优化; 停滞优化; 多准则

中图分类号: TP301.6

文献标志码: A

文章编号: 1001-5132 (2015) 04-0048-05

粒子群优化(Particle Swarm Optimization, PSO)算法是一种缘于鸟类捕食的仿生算法^[1]. 该算法内部存在较少的调整参数, 适用于求解非线性问题, 因此得到了广泛的应用, 如目标定位、模式识别、图像处理以及车间调度等^[2-4].

通常 PSO 算法存在早熟、收敛精度低、收敛速度慢等问题^[5-6]. 局部极值限制了粒子群的搜索范围, 导致早熟现象; 早熟问题和迭代次数共同影响着算法的收敛精度; 算法的复杂度直接决定了粒子群的收敛速度. 在目标定位的实际工程应用中, 早熟问题容易引起系统在定位过程中陷入局部最优解的困境, 由此导致定位结果偏离定位目标的实际位置, 影响定位精度. 针对以上问题, 出现了一系列的改进算法, 如参数改进法、混合策略法、扰动法等^[7-8]. 近几年, 为了改善粒子群的全局寻优能力, 有研究提出了粒子群间共享信息的优化方案, 较好地解决了早熟问题, 如 LU 等^[9]提出的多种群分层粒子群优化(Hierarchical Structure Poly-Particle Swarm Optimization, HSPPSO)算法, 该算法增强了粒子群的全局寻优能力, 但运行复杂度增加, 收敛速度有待改善. 而有些方法具有快速收敛的特点, 如史小露等^[10]着重改善 PSO 算法的收敛

速度, 采用混合算法的思想, 以提高种群丰富度为目的, 提出了具有快速收敛和自适应逃逸功能的粒子群优化(Particle Swarm Optimization Algorithm with Fast Convergence and Adaptive Escape, FAPSO)算法, 有效地改善了 PSO 算法的收敛速度.

本文受到 HSPPSO 算法和混合策略法的启发, 综合考虑早熟和收敛速度的问题, 结合 2 种方法的优势, 提出一种基于多准则决策的塔状 PSO (Multi Criteria-Pyramid Particle Swarm Optimization, MCP PSO)算法. 该算法以塔状优化互联机制为基础, 引入基于多准则决策的停滞优化策略, 互取优势, 加强了粒子群的全局搜索能力, 同时也提高了收敛精度和收敛速度, 改善了 PSO 算法的整体性能.

1 PSO 算法

PSO 算法的优化过程包含个体学习和群体学习 2 个机制, 分别对应个体极值与全局极值. 为改善原始 PSO 算法的收敛精度和收敛速度, Shi 等^[11]提出惯性权重概念, 即标准粒子群优化(Standard Particle Swarm Optimization, SPSO)算法的概念. 设在 D 维空间中存在一个粒子群, 粒子总数为 N , 共进行 K 次迭代. 假使 $i=1, 2, \dots, N$, 第 i 个粒子在第

收稿日期: 2014-09-17.

宁波大学学报(理工版)网址: <http://journal.g.nbu.edu.cn/>

基金项目: 宁波市自然科学基金(2013A610120); 浙江省信息与通信工程重中之重学科开放基金(xkx11526).

第一作者: 王益平(1990-), 男, 浙江宁波人, 在读硕士研究生, 主要研究方向: 数据传输与监控. E-mail: wangyiping_nbu@163.com

*通信作者: 蓝艇(1976-), 男, 浙江宁波人, 讲师, 主要研究方向: 智能机器人控制. E-mail: lanting@nbu.edu.cn

k 次迭代时处于位置 $X_i^k=(x_{i1}^k, x_{i2}^k, \dots, x_{iD}^k)$, 速度为 $V_i^k=(v_{i1}^k, v_{i2}^k, \dots, v_{iD}^k)$, 其更新表达式为:

$$v_{id}^{k+1} = wv_{id}^k + c_1r_1(p_{id}^k - x_{id}^k) + c_2r_2(p_{gd}^k - x_{id}^k), \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1}, \quad (2)$$

$$w = w_{\max} - (w_{\max} - w_{\min}) \times k / K, \quad (3)$$

其中: w 为惯性权重, 采用线性递减策略, 如(3)式; c_1 、 c_2 为常数; r_1 、 r_2 为均匀分布在 $[0,1]$ 区间内的随机数; x_i 为第 i 个粒子的当前位置, p_i 为个体极值位置, p_g 为全局极值位置。

为了更好地改进 PSO 算法的性能, 有研究针对早熟、收敛速度慢等问题展开了相关的改进工作, 如基于高斯扰动的粒子群优化算法^[12]、HSPPSO 算法和 FAPSO 算法。其中 HSPPSO 算法在保持粒子数目的前提下, 通过调整粒子群的组织方式, 实现了粒子群间的信息共享, 相比传统 PSO 算法有效地提高了全局寻优能力, 但收敛速度有待改善。而 FAPSO 算法包含了混合算法的思想, 在寻优过程中, 每一进化环节包含全局搜索和局部搜索的操作。为了避免粒子群陷入局部最优, 在局部搜索环节采用混合算法对陷入局部极值的种群做优化处理, 提高寻优收敛速度。如何更好地提高算法的全局寻优能力, 加快收敛速度, 权衡整体性能是进一步研究 PSO 算法的改进方向。

2 MCPPSO 算法

为了综合改进 PSO 算法的全局寻优能力和收敛速度, 提出基于多准则决策的塔状粒子群优化算法, 主要包括 2 个方面。

(1) 在 PSO 算法的基础上, MCPPSO 算法引入塔状优化互联机制, 即采用粒子群分层的形式, 处于底层的粒子群被均分成若干更小单位的粒子群, 各自独立地收集局部极值信息, 并向顶层粒子传递局部极值信息; 顶层的粒子负责收集、反馈信息。顶层的粒子经过比较、筛选来自底层粒子群的局部极值, 获得当前全局极值(等同全局最优解), 同时比对前次全局极值, 判断系统是否停滞, 并向底层粒子群反馈全局极值和停滞信息。

(2) 实验发现, 种群的丰富度与求解全局极值之间存在着紧密的联系。因此, 寻找一种提高种群丰富度的方法成了改进算法的一个研究方向。细菌觅食优化(Bacteria Foraging Optimization Algorithm,

BFO)算法包括趋化、复制和迁移 3 个步骤。趋化操作强化了粒子群的局部挖掘能力, 迁移操作增加了粒子群的丰富度, 但算法的复杂度较大。而随机初始化策略与 BFO 算法的迁移操作存在相似性。由于省略了趋化、复制操作, 执行复杂度相比 BFO 算法大为降低。但因随机性较大, 收敛性不佳。综上所述, BFO 算法和随机初始化策略都能满足提高种群丰富度的要求。本研究在粒子群陷入局部极值时采用混合策略的思想, 结合两者的优点, 提出基于多准则决策的停滞优化策略, 从而改善早熟现象, 提高收敛速度。

2.1 塔状优化互联机制

在 MCPPSO 算法中, 粒子群被分成 2 个层次, 底层的种群 $S_l(l=1, 2, \dots, n)$ 负责寻找优解, 顶层的最优种群 S_{best} 负责收集、反馈信息。MCPPSO 算法的粒子群组织模型如图 1 所示。

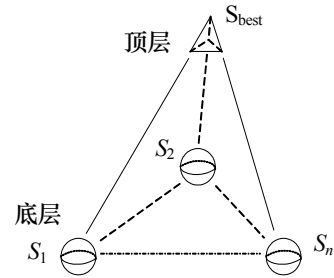


图1 MCPPSO 算法的粒子群组织模型

设 D 维空间中有一个粒子群, 总数为 N , 均分成 n 个种群, 共进行 K 次迭代。假使第 l 个种群中的第 i 个粒子在第 k 次迭代位于 $X_{i-l}^k=(x_{i1-l}^k, x_{i2-l}^k, \dots, x_{iD-l}^k)$, 速度为 $V_{i-l}^k=(v_{i1-l}^k, v_{i2-l}^k, \dots, v_{iD-l}^k)$ 。对应如下更新表达式:

$$v_{id-l}^{k+1} = wv_{id-l}^k + c_1r_1(p_{id-l}^k - x_{id-l}^k) + c_2r_2(p_{gd-l}^k - x_{id-l}^k) + c_3r_3(p_{gd}^k - x_{id-l}^k), \quad (4)$$

$$x_{id-l}^{k+1} = x_{id-l}^k + v_{id-l}^{k+1}, \quad (5)$$

$$w = w_{\max} - (w_{\max} - w_{\min}) \times k / K, \quad (6)$$

其中, 惯性权重 w 采用线性递减策略, $w_{\max}=0.9$, $w_{\min}=0.4$; c_1 、 c_2 、 c_3 为常数, 按经验取 $c_1=c_2=2.0$, $c_3=0.5$; r_1 、 r_2 、 r_3 为均匀分布在 $[0,1]$ 区间内的随机数; (5)式中 x_{i-l} 为第 l 个种群中第 i 个粒子的当前位置, p_{i-l} 为第 l 个种群中第 i 个粒子的个体极值位置, p_{g-l} 为第 l 个种群的局部极值位置, p_g 为 n 个种群的全局极值位置。

优化过程中,处于底层的种群将局部极值信息 p_{g-l} 向顶层汇总,由顶层比较得到 n 个种群的全局极值信息 p_g , 如(7)式:

$$p_g = B_{\text{best}}(p_{g-l}), l=1,2,\dots,n. \quad (7)$$

进入下一次迭代时,从顶层反馈回来的全局极值信息 p_g 将参与更新下一次迭代的种群信息. 种群间借助塔状优化互联机制,共享寻优信息,良好拓广了种群的多样性.

2.2 基于多准则决策的停滞优化策略

在塔状优化结构中,顶层粒子时刻监视全局极值信息,一旦发现停滞现象,便采取基于多准则决策的停滞优化策略(迭代 T 次后,全局极值持续未更新,则视为停滞).

为研究方便,假设存在 3 个底层种群. 一旦全局极值出现停滞,底层种群采取基于多准则决策的停滞优化策略: 种群 S_1 中的粒子按概率采取 BFO 寻优策略,寻找周围潜在的优解; 种群 S_2 的粒子根据随机初始化策略更新位置信息,仅当粒子的状况得到改善才保留已更新的信息,否则粒子将回到原有的位置; 种群 S_3 不进行任何操作.

停滞优化操作通过 BFO、随机初始化的策略,有效地避免了粒子群的早熟现象,有助于提高收敛速度; 同时,种群 S_2 及种群 S_3 的策略避免了盲目优化,增强了算法的稳定性和可靠性. 下面对 BFO 寻优策略和随机初始化策略进行具体分析.

2.2.1 BFO 寻优策略

BFO 算法是一种仿照大肠杆菌觅食过程的仿生算法, Passino 于 2002 年提出了这个概念^[13]. 该算法包括趋化、复制和迁移 3 个步骤,趋化操作的重要性最为突出(包括翻转、游动 2 个操作). 一旦种群出现群聚现象,个体将通过翻转、游动的方式去探索周围潜在的食物源,重新觅食.

假设陷入群聚的个体尝试向 D 个维度游动,同一方向上的最大步进次数为 L ,每次游动步长为 c . 一般用粒子群的健康度 J 表示粒子群的生存状况,记录当前健康度 J_{save} . 一轮翻转后,如果粒子群的健康度优于前次健康度,则粒子群执行游动操作,直至达到最大步进次数 L ; 如果粒子群的健康度恶化,则粒子群结束游动操作.

(1) 趋化操作: 1) 如果 $J_{\text{health}}^i(i,d,l+1)$ 优于 $J_{\text{health-save}}^i(i,d,l)$, 且未达最大游动次数,则继续游

动操作; 2) 如果 $J_{\text{health}}^i(i,d,l+1)$ 次于 $J_{\text{health-save}}^i(i,d,l)$, 则第 i 个粒子结束当前维度下的游动,进入下一维度的翻转操作; 3) N 个粒子在 D 个维度下遍历执行 1) 和 2) 步骤. 其中, $J_{\text{health}}^i(i,d,l+1)$ 代表第 i 个粒子在第 d 个维度上的第 $(l+1)$ 次迭代的健康度,所谓“优于”视实际适应度函数而定,而非简单表示“大于”或“小于”的含义.

(2) 复制操作: 1) 趋化操作结束后,计算各粒子的健康度,按序排列; 2) 在当前粒子群中,淘汰一半健康度较差的个体,复制另一半较优的粒子,维持等量的粒子数目.

(3) 迁移操作: 1) 如果 $P_i > P_{\text{constant}}$, 第 i 个粒子执行迁移操作,随机初始化该粒子的位置; 2) 如果 $P_i < P_{\text{constant}}$, 第 i 个粒子在原地停留,其中 P_{constant} 表示一个常数阈值, P_i 表示均匀分布在 $[0,1]$ 区间内的随机数.

2.2.2 随机初始化策略

随机初始化策略的基本步骤为:

(1) 如果 $P_i > P_{\text{constant}}$, 随机初始化第 i 个粒子的位置. 计算第 i 个粒子的健康度,若粒子的状况得到改善,保留已更新的位置、个体极值信息,否则恢复第 i 个粒子的位置、个体极值信息.

(2) 如果 $P_i < P_{\text{constant}}$, 第 i 个粒子在原地停留. 其中 P_{constant} 表示一个常数阈值,取值区别于 BFO 寻优策略中的 P_{constant} 取值, P_i 表示均匀分布在 $[0,1]$ 区间内的随机数.

2.3 MCPPSO 算法的流程步骤

MCPPSO 算法步骤(图 2)为:

Step1 随机初始化种群的速度和位置信息,设定最大迭代次数 K 、最大停滞次数 T 和收敛精度 δ , 计算局部极值和全局极值.

Step2 按照(6)式计算当前惯性权值 w , 按照(4)和(5)式更新底层各种群粒子的速度和位置信息,以及种群局部极值和粒子的个体极值.

Step3 更新 n 个种群的全局极值,判断是否出现停滞现象. 若全局极值出现停滞现象,执行 Step4; 否则跳转到 Step5.

Step4 执行基于多准则决策的停滞优化操作,之后执行 Step5.

Step5 判断是否达到最大迭代次数或是收敛精度. 若满足条件,则输出结果,否则返回 Step2.

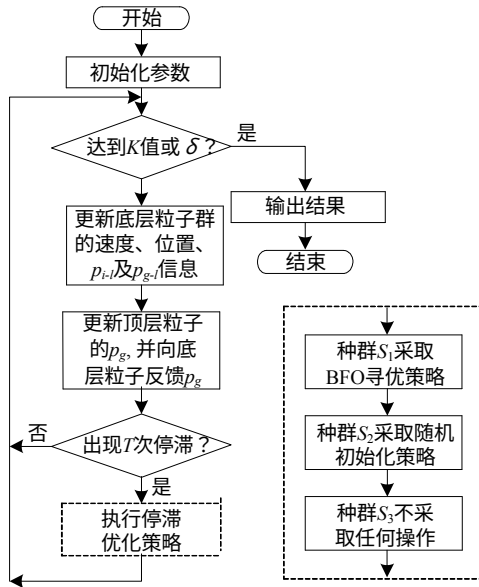


图2 MCPPSO 算法流程

3 仿真实验

3.1 测试函数

选取4个经典的多峰测试函数检验 MCPPSO 算法的性能. 表1给出了测试函数的名称、搜索空间以及全局最优. 表达式^[14]为:

表1 测试函数

函数	名称	搜索空间	全局最优
f_1	Rosenbrock	$x_i \in [-30, 30]$	$f(x)=0, x_i=0$
f_2	Rastrigin	$x_i \in [-5.12, 5.12]$	$f(x)=0, x_i=0$
f_3	Griewank	$x_i \in [-600, 600]$	$f(x)=0, x_i=0$
f_4	Ackley	$x_i \in [-50, 50]$	$f(x)=0, x_i=0$

Rosenbrock 函数:

$$f_1 = \sum_{i=1}^{N-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2), -30 \leq x_i \leq 30.$$

Rastrigin 函数:

$$f_2 = \sum_{i=1}^N (x_i^2 - 10 \cos(2\pi x_i) + 10), -5.12 \leq x_i \leq 5.12.$$

Griewank 函数:

$$f_3 = \frac{1}{4000} \sum_{i=1}^N x_i^2 - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, -600 \leq x_i \leq 600.$$

Ackley 函数:

$$f_4 = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^N x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^N \cos 2\pi x_i\right) + 20 + e, -50 \leq x_i \leq 50.$$

3.2 仿真结果

仿真实验中参数设为: 维数 $D=30$, 最大迭代次数 $K=2000$, 粒子总数 $N=60$, 每个函数运行 50 次后取平均. 以 SPSO 和 HSPPSO 算法为比较对象, 在 4 种测试函数下仿真各算法的性能, 结果见表 2.

表2 3种算法的测试结果比较

函数	算法	最小值	最大值	平均值	方差
f_1	SPSO	2.325e+1	8.071e+1	2.990e+1	1.005e+2
	HSPPSO	1.014e+1	8.498e+1	2.780e+1	7.443e+1
	MCPPSO	7.673e+0	1.766e+1	1.188e+1	5.041e+0
f_2	SPSO	1.031e+0	2.092e+1	1.038e+1	2.047e+1
	HSPPSO	4.505e+0	1.849e+1	9.892e+0	9.524e+0
	MCPPSO	2.028e+0	1.585e+1	9.036e+0	1.165e+1
f_3	SPSO	2.913e-3	4.981e-2	1.985e-2	1.383e-4
	HSPPSO	3.338e-4	3.749e-2	9.003e-3	7.327e-5
	MCPPSO	0	2.955e-2	6.503e-3	6.612e-5
f_4	SPSO	9.321e-1	3.096e+0	2.035e+0	2.017e-1
	HSPPSO	3.143e-3	2.120e+0	7.507e-1	5.515e-1
	MCPPSO	2.931e-14	1.927e-13	5.759e-14	8.305e-28

从表2可知, MCPPSO 算法的平均收敛精度和收敛方差均优于 SPSO 和 HSPPSO 算法. 平均收敛精度越高, 算法的全局寻优能力越好; 方差越小, 算法的稳定性越高. 此外, 在 Griewank 和 Ackley 测试函数中, MCPPSO 算法表现出了极佳的性能, 即 MCPPSO 算法能够胜任多峰函数的寻优任务.

图3和图4是3种算法在 Ackley 和 Griewank 测试函数下的收敛性能比较. 从图中可看出: (1)相比 SPSO 和 HSPPSO 算法, MCPPSO 算法持续表现出良好的递减趋势, 收敛速度最快; (2) MCPPSO 算法的收敛精度最高, 与表1中的全局最有结果一致. (3)在 Griewank 测试函数中, HSPPSO 算法在迭代后期逐渐赶超 SPSO 算法的收敛精度, 但优势不大.

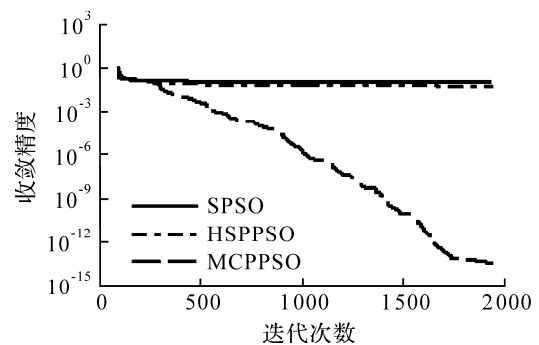


图3 Ackley 函数收敛性能比较

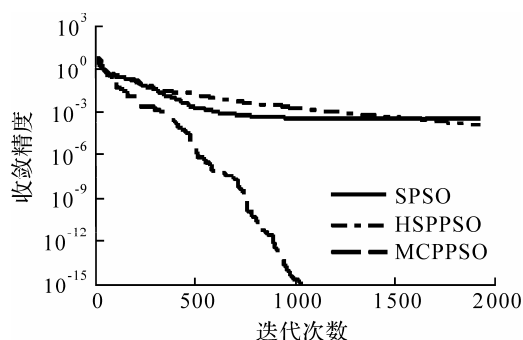


图4 Griewank 函数收敛性能比较

4 结语

针对粒子群优化算法存在早熟、收敛精度低和收敛速度慢的问题,提出了改进方案:(1)提出塔状优化互联机制,种群间共享信息;(2)在停滞优化中引入多准则决策,动态地适应寻优过程。相比SPSO和HSPPSO算法,本文提出的MCPPO算法具有更好的全局寻优能力,改善了粒子群的收敛精度和收敛速度。

参考文献:

- [1] Kennedy J, Eberhart R C. Particle swarm optimization [C]//Proc of IEEE Int Conf Neural Networks, 1995: 1942-1948.
- [2] 崔静静, 孙延明, 车兰秀. 改进细菌觅食算法求解车间作业调度问题[J]. 计算机应用研究, 2011, 28(9):3324-3326.
- [3] 张军, 詹志辉, 陈伟能, 等. 计算机智能[M]. 北京: 清华大学出版社, 2009:121-123.
- [4] LI Yong-ming, LAI Han, LU Liu-yi, et al. Dynamic brain magnetic resonance image registration based on inheritance idea and PSO[C]//Proc of IEEE International Conference on Biomedical Engineering and Informatics, 2011:263-267.
- [5] 顾宏杰, 许力. 利用带感知能力的粒子群算法求解约束优化问题[J]. 计算机应用, 2011, 31(1):85-88.
- [6] 代军, 李国, 徐晨, 等. 一种新的粒子群优化算[J]. 计算机工程, 2010, 36(9):192-194.
- [7] 李丽, 牛奔. 粒子群优化算法[M]. 北京: 冶金工业出版社, 2009:70-74.
- [8] 任小波, 杨忠秀. 粒子群优化算法的改进[J]. 计算机工程, 2010, 36(7):205-207.
- [9] LU Lin, LUO Qi, LIU Jun-yong, et al. An Improved Particle Swarm Optimization Algorithm[C]//Proc of IEEE International Conference on Granular Computing, 2008: 486-490.
- [10] 史小露, 孙辉, 李俊, 等. 具有快速收敛和自适应逃逸功能的粒子群优化算法[J]. 计算机应用, 2013, 33(5): 1308-1312.
- [11] Shi Y, Eberhart R C. A modified particle swarm optimizer [C]//Proc of IEEE World Congress on Computational Intelligence, 1998:69-73.
- [12] 朱德刚, 孙辉, 赵嘉, 等. 基于高斯扰动的粒子群优化算法[J]. 计算机应用, 2014, 34(3):754-759.
- [13] Kevin M P. Biomimicry of bacterial foraging for distributed optimization and control[J]. IEEE Control Systems Magazine, 2002, 22(3):52-67.
- [14] 刘波. 粒子群优化算法及其工程应用[M]. 北京: 电子工业出版社, 2010:117-119.

Pyramid Particle Swarm Optimization Based on Multi Criteria

WANG Yi-ping, LAN Ting*, WU Yang-bo

(Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China)

Abstract: Aiming at the problems of Particle Swarm Optimization (PSO), such as premature, low convergence precision and slow convergence rate, a newly improved algorithm is proposed in which the whole particles are organized in a pyramid structure. In the optimizing process, the swarms in the bottom layer share information among groups under the coordination of the global optimal particle in the top layer. Once the swarms stop moving, the groups in the bottom layer will take multi criterions to optimize the swarms' convergence rate. Experimental results suggest that the proposed algorithm bears better efficient and stronger optimizing ability, and improves optimizing precision and convergence rate more than some other existing optimization algorithms.

Key words: particle swarm optimization; pyramid optimization; stagnation optimization; multi criterions

(责任编辑 史小丽)