

基于动态多粒子群的多目标优化算法

刘彬*, 张仁津

(贵州师范大学 数学与计算机科学学院, 贵阳 550001)

(* 通信作者电子邮箱 liubinn9@163.com)

摘要:为了让多目标粒子群优化算法在运行过程中保持粒子的多样性,提出了一种初始化方法和动态多粒子群协作的多目标优化算法。根据粒子群在决策空间中的分布情况动态增加或者减少粒子群数量;为避免粒子收敛速度过快,改进了决定粒子飞行速度的因素,速度值依赖于粒子当前速度惯性、粒子最优值、群最优值和所有群最优值。用五个测试函数对算法进行了测试并与多目标粒子群优化进行了比较,测试结果表明提出的算法优于多目标粒子群优化算法。

关键词:多目标优化;粒子群优化;局部搜索;全局最优解;局部最优解

中图分类号: TP18 **文献标志码:** A

Multi-objective optimization algorithm based on dynamic multiple particle swarms

LIU Bin*, ZHANG Renjin

(School of Mathematics and Computer Science, Guizhou Normal University, Guiyang Guizhou 550001, China)

Abstract: To keep the diversity of particles when multi-objective particle swarm optimization is running, a multi-objective optimization algorithm was proposed based on particle swarms initialization and dynamic multiple particle swarms cooperation. The quantity of swarms was increased or decreased dynamically according to the distribution of particle swarms in the decision space. To avoid converging too quickly, the factors, which affected the flying speed of a particle, were improved to depend on the current velocity inertia of the particle, the best value of the particle, the best value of the swarm which the particle belonged to, and the optimal value of all swarms. This algorithm was tested by five benchmark functions and compared with the multi-objective particle swarm optimization. The experimental results indicate that the proposed algorithm is superior to the multi-objective particle swarm optimization.

Key words: Multi-Objective Optimization (MOO); Particle Swarm Optimization (PSO); local search; global optimal solution; local optimal solution

0 引言

多目标优化 (Multi-Objective Optimization, MOO) 在科学和工程领域都有重要的研究价值^[1]。许多实际应用有多个目标需要同时优化,而这些需要优化的目标可能相互冲突,根本不存在一组解可以使所有目标函数都达到最优^[2]。早期解决多目标优化问题是把多个目标函数通过不同的权值线性相加,这样就可以把多目标优化转换成为单目标优化^[3],每次运行都可以得到一个解,通过多次运行就可以获得一组解,但这种方法得到的解受权值的影响比较大,很多时候求得的解与实际解相差很远。而理想的多目标优化希望通过一次运算就能获得足够多的解提供给决策者,并且这些解应该比较均匀地分布在 Pareto 前沿上。

粒子群优化算法 (Particle Swarm Optimization, PSO) 是受鸟群觅食行为的启发于 1995 年提出^[4],和遗传算法、蚁群算法一样都是模拟生物的行为,由于粒子群优化算法容易实现、收敛速度快,因此发展非常迅速。近些年,多目标粒子群优化在电力分配^[5]、投资项目选择^[6]、疾病诊断^[7]、股票交易^[8]、无线传感器路由^[9]等领域获得广泛应用。从多目标优化来看,主要关注的问题包括:1) 如何从粒子群中选择引导粒子搜索的粒子,从而使粒子能获得满意的非支配解;2) 如何保留在搜索过程获得的非支配解,并且要求保留下来的解能均

匀分布在 Pareto 前沿;3) 如何维持粒子群的多样性而不至于收敛到单个解。文献[10]提出一种把粒子群、遗传算法以及局部搜索相结合的多目标优化,在该算法中,粒子初始值由随机数赋值,粒子的速度计算采用的是常用的方法,依赖于粒子当前速度、粒子最优位置和粒子群最优位置,为了保持粒子的多样性,在粒子群每次移动到新位置后采用遗传算法,但遗传算法是一种基于概率的算法,并不能确保粒子的多样性,另外遗传算法会使收敛速度受到极大的影响。文献[11]为了增强粒子的多样性和均匀性,提出了一种与支配度、领域拥挤度和粒子差异度有关的储备集更新策略,通过一种拟小世界动态邻结构平衡粒子的全局搜索和局部搜索能力,这种方法对粒子包围区域的搜索能力比较强,但对于不在粒子包围的区域(如邻近边界的区域),这种方法并不能产生太大的影响。文献[12]为了保留在多目标粒子群优化中以 ϵ -支配时正常丢失的一些解,在接收一个非支配解到外部档案中时增加一个额外的目标函数,用一个计算量小且效率高的手段使 Pareto 前沿有更好的质量。这些方法主要关注的是优化过程采用的手段和方法,而对算法开始时粒子的多样性并未重视,普遍认为以随机数初始化的值分布就会好,但实际上基于概率的方法在这一点上并没有必然保障,由于粒子群算法收敛速度比较快,如果初始分布不好,对后期的影响会很大。文献[13]采用 Sobol 序列映射决策变量初始值,虽然能使粒子初

收稿日期:2013-07-19。 基金项目:贵州省科学技术基金资助项目(黔科合 J 字 LKS[2013]29 号)。

作者简介:刘彬(1973-),男,湖南华容人,副教授,硕士,CCF 会员,主要研究方向:智能计算、可信计算; 张仁津(1963-),男,四川成都人,教授,主要研究方向:软件方法学、智能计算。

始值在决策空间范围均匀分布,但这种方法需要在算法运算前确定序列元素的个数,如果设置的值过大会导致粒子数过多,如果值过小又难保证粒子在决策空间的充分探索,而且在多目标优化中由于解空间的值比较多,采用单粒子群时,由于粒子群收敛很快,与多粒子群相比,更容易失去多样性,很容易失去 Sobol 序列带来的均匀分布的优势。因此,越来越多的科研人员研究用多粒子群解决多目标问题^[14]。针对多目标粒子群优化中存在的问题,本文提出一种基于动态多粒子群的多目标优化算法,主要工作包括:

- 1) 提出一种动态多粒子群的生成方法,可以使生成的粒子群在决策变量空间分布得更均匀;
- 2) 提出优化过程中粒子群动态变化机制,一方面防止产生过多的粒子群,另一方面确保粒子的多样性;
- 3) 提出一种多粒子群粒子飞行方法,粒子速度依赖于当前速度惯性、粒子最优值、群最优值和所有群最优值,避免粒子收敛过快。

1 MOO 的相关概念

一个多目标优化问题可以定义为一个三元组: (X, C, F) 。其中: X 是 n 维决策空间,故有 $x = (x_1, x_2, \dots, x_n) \in X$; C 是一组决策变量必须同时满足的约束集合; F 是含有 $m \geq 2$ 个目标函数的矢量,且 $F = (f_1(x), f_2(x), \dots, f_m(x))$ 。多目标优化就是让这些目标函数最小化。可用式(1)描述:

$$\begin{cases} \min F = [f_1(x), f_2(x), \dots, f_m(x)] \\ c_i(x) \leq 0 \end{cases}; i = 1, 2, \dots, I \quad (1)$$

其中 $c_i(x)$ 为第 i 个约束条件。由于多目标优化与单目标优化的概念不同,先介绍几个概念。

定义 1 可行解。如果 $x \in X$ 且满足 C 中的所有约束条件,称 x 为可行解。

定义 2 可行解集合。所有可行解 x 的集合称为可行解集合 X_f , 显然 $X_f \subseteq X$ 。

定义 3 Pareto 支配。假设 $x_A, x_B \in X_f$, 则 x_A 对 x_B 是 Pareto 支配的充要条件是: $\forall i = 1, 2, \dots, m, f_i(x_A) \leq f_i(x_B)$ 且 $\exists j = 1, 2, \dots, m, f_j(x_A) < f_j(x_B)$, 记为: $x_A > x_B$, 也称为 x_A 支配 x_B 。

定义 4 Pareto 最优解。如果一个可行解 x^* 满足条件: $\neg \exists x \in X_f$ 使 $x > x^*$, 则 x^* 被称为 Pareto 最优解(或非支配解)。

定义 5 Pareto 最优解集。所有 Pareto 最优解的集合称为 Pareto 最优解集。

多目标优化是为了找出非支配解的 Pareto 最优解集。由于 Pareto 最优解集可能具有过多的元素甚至是无限集合,会导致计算效率低甚至无法完成,所以通常用一个分布均衡的 Pareto 最优解集的子集代替。

2 动态多粒子群多目标优化

粒子群算法是基于生物进化算法中的一种,和其他进化算法类似,它也无法保证得到的结果一定是最优解。与其他进化算法相比,粒子群算法有一个重要的特点就是它的收敛速度快,因此用粒子群算法很容易得到局部最优解。为了减少这种可能,需要确保粒子在全局范围内搜索,同时在搜索的过程中让粒子保持多样性。为了让粒子能在全局范围内搜索并保持粒子的多样性,本文改进了粒子的初始化方法并以动态粒子群保持优化过程中粒子的多样性。

2.1 粒子群的结构

设在 d 维空间中有 M 个粒子群,每个粒子群含 N 个粒子,其结构如图 1 所示,粒子群之间通过所有群体最优值协作,即所有群体最优值引导粒子向最优值飞行。

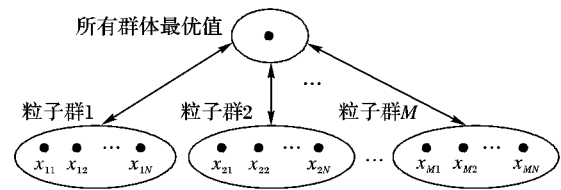


图1 粒子群结构

2.2 粒子的运动

第 j 个粒子群中第 i 个粒子在 t 时刻的位置表示为 $x_j^i(t) = (x_{j1}^i, x_{j2}^i, \dots, x_{jd}^i)$, 速度表示为 $v_j^i(t) = (v_{j1}^i, v_{j2}^i, \dots, v_{jd}^i)$, 到 t 时刻为止搜索到的最优位置(简称粒子最优值)表示为 $p_j^i(t) = (p_{j1}^i, p_{j2}^i, \dots, p_{jd}^i)$, 第 j 个粒子群中所有粒子到 t 时刻为止搜索到的最优位置(简称粒子群最优值)表示为 $l_j(t) = (l_{j1}^t, l_{j2}^t, \dots, l_{jd}^t)$, 全部粒子群中所有粒子到 t 时刻为止搜索到的全局最优位置(简称全局最优值)表示为 $g(t) = (g^1, g^2, \dots, g^d)$, 则第 j 个粒子群中粒子 i 在 $t+1$ 时刻的速度由式(2)计算。

$$v_j^i(t+1) = \omega v_j^i(t) + b_1 r_1 (p_j^i(t) - x_j^i(t)) + b_2 r_2 (l_j(t) - x_j^i(t)) + b_3 r_3 (g(t) - x_j^i(t)) \quad (2)$$

其中: b_1, b_2, b_3 为正常数,用于调整粒子自身、群体和全局所有粒子对粒子 i 的影响程度; r_1, r_2, r_3 为 $[0, 1]$ 内的随机数; 参数 ω 为惯性因子,为了在搜索的早期提高搜索范围会用一个大一些的值,在后期为了提高局部搜索能力会取一个比较小的值。设 ω 依据循环次数从 ω_s 线性递减至 ω_e 。设最大循环次数为 I_{\max} , 循环的当前次数为 I_c , 则 ω 的值可由式(3)求出:

$$\omega = \omega_s - (\omega_s - \omega_e) \frac{I_c}{I_{\max}} \quad (3)$$

其中: ω_s 为优化最初时的惯性因子, ω_e 为优化最末的惯性因子。

粒子群 j 中粒子 i 在 $t+1$ 时刻的位置由式(4)求出:

$$x_j^i(t+1) = x_j^i(t) + v_j^i(t+1) \quad (4)$$

由式(2)和式(4)可知,与 t 时刻位置相比,粒子 i 在 $t+1$ 时刻的位置变化受四个部分的影响,即 t 时刻的速度惯性、粒子最优值、群体最优值和全局最优值,在二维空间可表示为如图 2 所示,粗虚线表示各影响成分的实际值,细虚线表示各影响成分对速度的实际影响值,粗实线为各实际影响值的矢量和,圆圈表示粒子的位置。

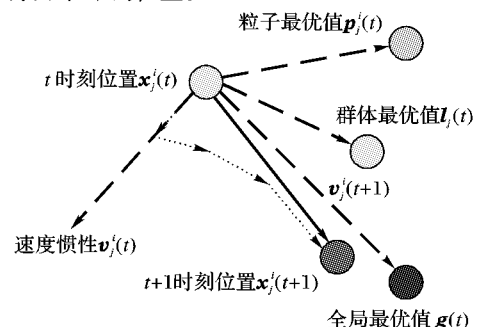


图2 二维空间中四因素对粒子位置的影响

2.3 粒子群的生成

2.3.1 普通粒子群的生成

首先生成 MN 个粒子,用定义域内的随机数初始化粒子,

以 k 均值聚类方法把这些粒子均分为 M 个群,每个群内用一个粒子作群的中心粒子,粒子群的中心粒子是与群内其他粒子距离最短的粒子,用 x_j^c 标识,按式(5) 求出。

$$x_j^c = \arg \left(\min \sum_{k=1}^N \|x_j^i - x_j^k\| \right); i \in [1, N] \quad (5)$$

其中: x_j^i, x_j^k 表示粒子群 j 中粒子 i 和 k ; $\|x_j^i - x_j^k\|$ 表示粒子 i, j 之间的距离,按式(6) 求出; $\arg(\cdot)$ 表示取主值,即若 x_j^i 与群内其他粒子之间的距离最小,则 $x_j^c = x_j^i$ 。

$$\|x_j^i - x_j^k\| = \sqrt{\left(\sum_{l=1}^d (x_j^{il} - x_j^{kl})^2 \right) / d} \quad (6)$$

其中 d 为决策变量的维数。

2.3.2 边界粒子群的生成

按随机值生成的粒子在决策空间内可能分布得不均匀,因此按以下方法进一步按需生成边界粒子群:

生成边界粒子时第 k 维分量下边界值 x_L^k 、上边界值 x_U^k 分别由式(7)、(8) 求出。

$$x_L^k = x_{\min}^k + cD_{\min} \quad (7)$$

$$x_U^k = x_{\max}^k - cD_{\min} \quad (8)$$

其中: c 为 $[0, 1]$ 内的随机数, D_{\min} 为设定的两粒子群距离的最小值的 $1/d$ 。然后针对决策变量的每一个边界测试与粒子群的距离,如果最小距离大于 D_{\min} ,则根据式(7)、(8) 生成一个此边界相邻的粒子群。

2.3.3 动态粒子群的生成

按式(6) 求出粒子群 a 与其他粒子群之间的距离,假设与粒子群 a 最近的粒子群为 b ,如果它们之间的距离大于 D_{\max} ,则需要生成一个粒子群 x_{M+1} ,群内第 i 个粒子第 k 维分量 x_{M+1}^{ik} 可按式(9) 计算。

$$x_{M+1}^{ik} = (x_a^{ik} + x_b^{ik}) / 2 + c_1(-1)^{\text{round}(0.5+c_2)} |x_a^{ik} - x_b^{ik}| / 2 \quad (9)$$

其中: c_1, c_2 为 $[0, 1]$ 内的随机数; $\text{round}(\cdot)$ 为取整函数,故 $\text{round}(0.5 + c_2)$ 为 0 或 1。

2.4 粒子越界处理

如果粒子群 j 中粒子 i 按式(4) 更新位置后超越了定义域界限,需要把粒子重新放回定义域内,将粒子的速度乘以一个 $(0, 1)$ 之间的调整系数 λ , λ 的初始值比较大。如果调整后仍然越界,继续减小系数,直到粒子的新位置落到决策空间内。新位置可按式(10) 计算:

$$x_j^i(t+1) = x_j^i(t) + \lambda v_j^i(t+1) \quad (10)$$

其中 λ 按式(11) 计算:

$$\lambda = 2 / (\gamma^2 + 2) \quad (11)$$

其中 γ 为调整次数,为避免停留在边界,当 γ 大于 3 后,粒子速度变为反向。在二维空间中粒子越界调整前后的情况如图 3 所示,粗虚线表示调整前的情况,细虚线表示调整后的情况,圆圈表示粒子的位置。

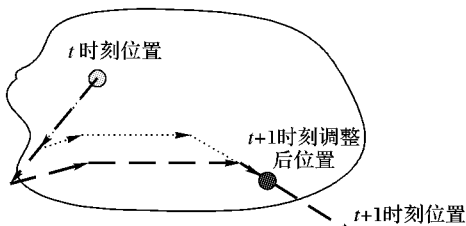


图3 粒子位置调整示意图

2.5 算法描述

1) 生成 MN 个粒子,用定义域内的随机数初始化粒子位置,以 k 均值聚类方法把这些粒子均分为 M 个群,按需生成边

界粒子群,所有粒子初始速度用定义域内随机数赋值,循环次数赋值为 0;

2) 如果某个粒子群与其他粒子群的最小距离大于 D_{\max} ,在它们之间插入新粒子群;

3) 粒子群 j 中粒子 i 的粒子外部档案集 P_j^i 为空,粒子群 j 的群外部档案集 L_j 为空,所有粒子的全局最优解外部档案集 G 为空;

4) 每个粒子的当前位置按非支配解规则放入其粒子外部档案集中,用粒子所属群的群外部档案集中与粒子当前位置最近的值作为粒子最优位置 $p_j^i(t)$;

5) 每个粒子群中所有粒子的当前位置按非支配规则放入群外部档案集中,用群外部档案集中与粒子当前位置最近的位置作为粒子 i 的最好群位置 $l_j(t)$;

6) 所有粒子的当前位置按非支配规则放入全局粒子群外部档案集,按 Pareto 存档演化策略 (Pareto Archived Evolution Strategy, PAES)^[15] 的自适应栅格技术将目标空间划分超立方体,把 G 中粒子位置对应的目标向量确定粒子所在的立方体,为每个非空格子计算适应度值,其值为粒子个数的倒数,以轮盘赌方法为每个粒子从 G 中随机选择一个位置做粒子的全局最优位置 $g(t)$;

7) 计算所有粒子的速度,求出粒子新位置,如果新位置越界,按 2.4 节方法处理;

8) 求粒子群之间的距离,如果某个粒子群与其他粒子群的最小距离大于 D_{\max} ,在它们之间插入新粒子群,把粒子初始化,生成粒子外部档案和群外部档案;

9) 如果某个粒子群与其他粒子群的最小距离小于 D_{\min} ,则删除一个粒子群,把被删除群的群外部档案按非支配原则合并到保留的粒子群中;

10) 与边界最近的粒子群到边界的距离如果大于 D_{\max} ,插入边界粒子群,生成粒子外部档案和群外部档案;

11) 如果达到最大循环次数,停止搜索,返回 G ,否则转到 4)。

3 性能测试分析

3.1 初始化比较

为了测试粒子在决策空间的分布情况,选用 CEC09 讨论会采用的测试函数 $UF1 \sim UF5$ ^[16], $UF1 \sim UF5$ 详情如下:

$UF1$ 是将以下两个目标函数优化到最小值:

$$f_1 = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left[x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right) \right]^2$$

$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} \left[x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right) \right]^2$$

其中: $J_1 = \{j | j \text{ 是奇数且 } 2 \leq j \leq n\}$, $J_2 = \{j | j \text{ 是偶数且 } 2 \leq j \leq n\}$; 搜索空间为 $[0, 1] \times [-1, 1]^{n-1}$ 。 $UF1$ 的 Pareto 前沿面为:

$$f_2 = 1 - \sqrt{f_1}; 0 \leq f_1 \leq 1$$

$UF1$ 的 Pareto 最优解集是:

$$x_j = \sin\left(6\pi x_1 + \frac{j\pi}{n}\right); j = 2, 3, \dots, n, 0 \leq x_1 \leq 1$$

$UF2$ 是将以下两个目标函数优化到最小值:

$$f_1 = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} y_j^2$$

$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} y_j^2$$

其中: $J_1 = \{j | j \text{ 是奇数且 } 2 \leq j \leq n\}$, $J_2 = \{j | j \text{ 是偶数且 } 2 \leq$

$j \leq n\}$, 且满足

$$y_j = \begin{cases} x_j - \left[0.3x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1 \right] * \\ \cos\left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_1 \\ x_j - \left[0.3x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1 \right] * \\ \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_2 \end{cases}$$

搜索空间为 $[0,1] \times [-1,1]^{n-1}$ 。

UF2 的 Pareto 前沿面为:

$$f_2 = 1 - \sqrt{f_1}; 0 \leq f_1 \leq 1$$

UF2 的 Pareto 最优解集是:

$$x_j = \begin{cases} \left[0.3x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1 \right] * \\ \cos\left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_1 \\ \left[0.3x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1 \right] * \\ \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_2 \end{cases}; 0 \leq x_1 \leq 1$$

UF3 是将以下两个目标函数优化到最小值:

$$f_1 = x_1 + \frac{2}{|J_1|} \left(4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2 \right)$$

$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|} \left(4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2 \right)$$

其中: $J_1 = \{j|j \text{ 是奇数且 } 2 \leq j \leq n\}$, $J_2 = \{j|j \text{ 是偶数且 } 2 \leq$

$j \leq n\}$, 且满足

$$y_j = x_j - x_1^{0.5(1.0 + \frac{3(j-2)}{n-2})}; j = 2, 3, \dots, n$$

搜索空间为 $[0,1]^n$ 。

UF3 的 Pareto 前沿面为:

$$f_2 = 1 - \sqrt{f_1}; 0 \leq f_1 \leq 1$$

UF3 的 Pareto 最优解集是:

$$x_j = x_1^{0.5(1.0 + \frac{3(j-2)}{n-2})}; j = 2, 3, \dots, n, 0 \leq x_1 \leq 1$$

UF4 是将以下两个目标函数优化到最小值:

$$f_1 = x_1 + \frac{2}{|J_1|} \left(\sum_{j \in J_1} h(y_j) \right)$$

$$f_2 = 1 - x_1^2 + \frac{2}{|J_2|} \left(\sum_{j \in J_2} h(y_j) \right)$$

其中: $J_1 = \{j|j \text{ 是奇数且 } 2 \leq j \leq n\}$, $J_2 = \{j|j \text{ 是偶数且 } 2 \leq$

$j \leq n\}$, 且满足

$$y_i = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right); j = 2, 3, \dots, n$$

和 $h(t) = \frac{|t|}{1 + \exp(2|t|)}$; 搜索空间为 $[0,1] \times [-2,2]^{n-1}$ 。

UF4 的 Pareto 前沿面为:

$$f_2 = 1 - f_1^2; 0 \leq f_1 \leq 1$$

UF4 的 Pareto 最优解集是:

$$x_j = \sin\left(6\pi x_1 + \frac{j\pi}{n}\right); j = 2, 3, \dots, n, 0 \leq x_1 \leq 1$$

UF5 是将以下两个目标函数优化到最小值:

$$f_1 = x_1 + \left(\frac{1}{2N} + \varepsilon\right) |\sin(2N\pi x_1)| + \frac{2}{|J_1|} \left(\sum_{j \in J_1} h(y_j) \right)$$

$$f_2 = 1 - x_1 + \left(\frac{1}{2N} + \varepsilon\right) |\sin(2N\pi x_1)| + \frac{2}{|J_2|} \left(\sum_{j \in J_2} h(y_j) \right)$$

其中: $J_1 = \{j|j \text{ 是奇数且 } 2 \leq j \leq n\}$, $J_2 = \{j|j \text{ 是偶数且 } 2 \leq j \leq n\}$, N 是整数, $\varepsilon > 0$, 且满足

$$y_i = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right); j = 2, 3, \dots, n$$

和 $h(t) = 2t^2 - \cos(4\pi t) + 1$; 搜索空间为 $[0,1] \times [-1,1]^{n-1}$ 。

UF5 具有 $2N + 1$ 个 Pareto 最优解:

$$\left(\frac{i}{2N}, 1 - \frac{i}{2N}\right); i = 0, 1, \dots, 2N$$

MOPSO 采用随机数生成 100 个粒子; 本文算法是生成固定粒子群数为 10, 每个粒子群有 3 个粒子, 其他粒子群为生成的边界粒子群和 D_{\min} 生成的粒子群(设 $D_{\min} = 0.1$ 和 0.2)。按式(6) 求出粒子间的距离(本文算法是求粒子群之间的距离), 两种算法各测 30 次, 每次求出粒子(群) 之间的最大值。30 次测试出的最大值、最小值如表 1 所示。从表 1 中可知, 本文方法在 $D_{\min} = 0.1$ 和 0.2 分别测试 30 次获得的最大值与最小值相差比较接近, 而 MOPSO 的最大值和最小值有比较大的差距, 说明完全用随机数生成的初始粒子分布不稳定, 而用本文相对比较稳定, 而且比较容易控制粒子密度。

表 1 30 次测试粒子(群) 之间最大值的比较

距离	本文算法		MOPSO
	$D_{\min} = 0.1$	$D_{\min} = 0.2$	
最大值	0.0976	0.1985	0.3558
最小值	0.0763	0.1841	0.2687
平均值	0.0829	0.1962	0.3294

3.2 解的可靠性比较

为了便于比较, 在此用 IGD (Inverted Generational Distance) 度量, 详情参考文献[17]。 P^* 是在目标空间沿前沿面均匀分布的点的集合, A 是对前沿的一个近似点的集合, 从 P^* 到 A 的平均距离定义为:

$$IGD(P^*, A) = \frac{1}{|P^*|} \sum_{p \in P^*} d(p, A) \quad (12)$$

其中 $d(p, A)$ 是 p 到 A 中点的最小欧几里得距离。

选用 CEC09 讨论会采用的测试函数 UF1 ~ UF5 与 MOPSO^[18] 进行比较, 取 30 次测试值的平均值, 结果如表 2 所示。从表中可以看出, 本文算法获得的 Pareto 解普遍要比 MOPSO 小, 说明本文算法获得的 Pareto 解与实际值更接近, 因此本文算法的解更优。这也说明通过改进初始化方法以及用多粒子群能使多目标优化获得性能更优的 Pareto 解集。

表 2 IGD 的平均值、最大值和最小值比较

测试函数	本文算法			MOPSO		
	平均 IGD	最小 IGD	最大 IGD	平均 IGD	最小 IGD	最大 IGD
UF1	0.010846	0.004563	0.011234	0.018263	0.005812	0.024568
UF2	0.006732	0.003025	0.010457	0.008521	0.004569	0.013785
UF3	0.035271	0.018217	0.054826	0.043684	0.023179	0.608843
UF4	0.042358	0.030584	0.048659	0.043677	0.034829	0.056281
UF5	0.315598	0.206756	0.395420	0.351841	0.296647	0.461284

4 结语

本文主要研究了用动态多粒子群保持多目标粒子群优化算法如何保持粒子多样性的问题,首先从粒子群生成着手,根据粒子群的分布情况在决策变量空间动态生成粒子群,为避免粒子群收敛过快改变了粒子选择粒子最优解和群最优解的方法,并在粒子群搜索解的过程中动态增加或减少粒子群的个数。测试表明这个方法能有效改变粒子的分布情况,对粒子群多目标优化算法的改进有一定的意义。

参考文献:

- [1] 公茂果,焦成,杨咚咚,等. 进化多目标优化算法研究[J]. 软件学报, 2009, 20(2): 271 - 289.
- [2] REYES-SIERRA M, COELLO C A C. Multi-objective particle swarm optimizers: a survey of the state-of-the-art [J]. International Journal of Computational Intelligence Research, 2006, 2(3): 287 - 308.
- [3] MARLER R T, ARORA J S. Survey of multi-objective optimization methods for engineering [J]. Structural and Multidisciplinary Optimization, 2004, 26(6): 369 - 395.
- [4] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceedings of the 1995 IEEE International Conference on Neural Networks. Piscataway: IEEE, 1995: 1942 - 1948.
- [5] KUMAR R, SHARMA D, SADU A. A hybrid multi-agent based particle swarm optimization algorithm for economic power dispatch [J]. International Journal of Electrical Power and Energy Systems, 2011, 33(1): 115 - 123.
- [6] RABBANI M, BAJESTANI M A, KHOSHKOU G B. A multi-objective particle swarm optimization for project selection problem[J]. Expert Systems with Applications, 2010, 37(1): 315 - 321.
- [7] CHANG R I, LIN S Y, HUNG Y. Particle swarm optimization with query-based learning for multi-objective power contract problem[J]. Expert Systems with Applications, 2012, 39(3): 3116 - 3126.
- [8] BRIZA A C, NAVAL JR P C. Stock trading system based on the multi-objective particle swarm optimization of technical indicators on

end-of-day market data[J]. Applied Soft Computing, 2011, 11(1): 1191 - 1201.

- [9] 岳林,易本顺,肖进胜. 能量平衡与 QoS 保障的无线传感器机会路由[J]. 湖南大学学报: 自然科学版, 2011, 38(11): 82 - 87.
- [10] MOUSA A A, EL-SHORBAGY M A, ABD-EL-WAHED W F. Local search based hybrid particle swarm optimization algorithm for multiobjective optimization [J]. Swarm and Evolutionary Computation, 2012, 3(1): 1 - 14.
- [11] 任子晖,王坚. 动态拓扑结构的多目标粒子群优化算法[J]. 同济大学学报: 自然科学版, 2011, 39(8): 1222 - 1226.
- [12] HERNÁNDEZ-DÍAZ A G, SANTANA-QUINTERO L V, COELLO COELLO A C, *et al.* Improving the efficiency of ϵ -dominance based grids [J]. Information Sciences, 2011, 181(15): 3101 - 3129.
- [13] 聂瑞,章卫国,李广文,等. 一种自适应混合多目标粒子群优化算法[J]. 西北工业大学学报, 2011, 29(5): 695 - 701.
- [14] ZHANG Y, GONG D W, DING Z H. Handling multi-objective optimization problems with a multi-swarm cooperative particle swarm optimizer [J]. Expert Systems with Applications, 2011, 38(11): 13933 - 13941.
- [15] KNOWLES J D, CORNE D W. Approximating the nondominated front using the Pareto archived evolution strategy [J]. Evolutionary Computation, 2000, 8(2): 149 - 172.
- [16] ZHANG Q F, ZHOU A M, ZHAO S Z, *et al.* Multiobjective optimization test instances for the CEC2009 special session and competition[EB/OL]. (2009 - 04 - 20) [2013 - 01 - 19]. <http://decs.essex.ac.uk/staff/zhang/MOEAcompetition/cec09testprobJem0904.pdf>.
- [17] COELLO COELLO C A, LECHUGA M S, LECHUGA M S. MOPSO: a proposal for multiple objective particle swarm optimization [C]// CEC'02: Proceedings of the 2002 Congress on Evolutionary Computation. Piscataway: IEEE, 2002: 1051 - 1056.
- [18] LI H, ZHANG Q F. Multiobjective optimization problems with complicated Pareto sets MOEA/D and NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 284 - 302.

(上接第 3371 页)

6 结语

不良网页的大量存在,不仅严重影响了搜索引擎的检索效率,而且降低了用户体验的满意度。因此,搜索服务非常有必要对不良网页进行识别和过滤。在设计和实验过程中,我们编写了爬虫软件,爬取了大量的网页,建立了实验数据集,并对实验数据集中的网页进行了清洗和过滤,分析了不良网页的特征,给出了行之有效的特征提取方法。鉴于 KNN 分类算法的计算量较大,需要将所有训练文本都存储起来,对空间及时间的要求都比较高,本文对 KNN 压缩算法进行了改进,并通过 Hadoop 平台上提供的 MapReduce 编程模型进行了分布式并行计算,不仅给出了实验结果,而且还对不同环境下的实验结果进行了分析和比较,结果证明了改进的 KNN 算法和基于 MapReduce 模型进行并行化计算的有效性。

参考文献:

- [1] HU W M, WO O, CHEN Z Y, *et al.* Maybank: recognition of pornographic Web pages by classifying texts and images [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(6): 1019 - 1034.
- [2] 施聪莺. 针对青春期少年的网络色情文本过滤技术研究[D]. 南京: 南京师范大学, 2010.
- [3] 吴慧玲,沈建京,贺广生. 基于不良文本信息过滤预处理方法的

研究[J]. 网络安全技术与应用, 2006(11): 61 - 63.

- [4] 苏贵洋,马颖华,李建华. 一种基于内容的信息过滤改进模型[J]. 上海交通大学学报, 2004, 38(12): 2030 - 2034.
- [5] 崔虹燕,蒋念平. 一种改进的多级信息安全过滤模型[J]. 情报理论与实践, 2006, 29(5): 615 - 617.
- [6] 杨晓懿,刘嘉勇. 基于内容的信息安全过滤技术[J]. 信息安全, 2004(4): 47 - 49.
- [7] LEE P Y, HUI S C, FONG A C M. An intelligent categorization engine for bilingual Web content filtering [J]. IEEE Transactions on Multimedia, 2005, 7(6): 1183 - 1190.
- [8] DU R, SAFAVI-NAINI R, SUSILO W. Web filtering using text classification [C]// ICON 2003: Proceedings of the 2003 11th IEEE International Conference on Networks. Piscataway: IEEE, 2003: 325 - 330.
- [9] WAI H H, PAUL A W. Statistical and structural approached to filtering Internet pornography [C]// Proceedings of the 2004 IEEE International Conference on System, Man and Cybernetics. Piscataway: IEEE, 2004, 5: 4792 - 4798.
- [10] 潘文锋. 基于内容的垃圾邮件过滤研究[D]. 北京: 中国科学院计算技术研究所, 2004.
- [11] 刘慧. 基于 KNN 的中文文本分类算法研究[D]. 成都: 西南交通大学, 2010.
- [12] 文思. 基于 Hadoop 的 K 近邻分类算法的研究[D]. 广州: 华南理工大学, 2011.