

一种均衡各速度项系数的多目标粒子群优化算法

耿焕同 赵亚光 陈 哲 李辉健

(南京信息工程大学计算机与软件学院 南京 210044)

摘 要 粒子群优化算法已成为求解多目标优化问题的有效方法之一,而速度更新公式中的惯性、局部和全局 3 个速度项的系数的动态合理设置是算法优化效率的关键问题。为解决现有算法仅单独设置各速度项系数导致优化效率不高的问题,提出了一种均衡各速度项系数的多目标粒子群优化算法。该方法旨在通过粒子的局部最优和全局最优的信息来引导种群的进化方向,动态调整每一个粒子速度项系数来均衡惯性、局部和全局 3 个速度项在搜索中的作用,从而更为准确地刻画算法的搜索能力和搜索精度,更好地平衡算法的探究和探索能力,进一步提高粒子群优化算法解决复杂多目标优化问题的效率。在 7 个标准测试函数上进行实验,并与 5 种经典的进化算法进行对比,结果表明新算法在综合指标 IGD 以及多样性评估指标 Δ 评分上具有更好的收敛速度和分布性,验证了新算法的有效性。

关键词 粒子群优化算法,均衡,速度项系数,自适应,多目标优化

中图分类号 TP183 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.12.045

Multi-objective Particle Swarm Optimization Algorithm with Balancing Each Speed Coefficient

GENG Huan-tong ZHAO Ya-guang CHEN Zhe LI Hui-jian

(College of Computer & Software, Nanjing University of Information Science & Technology, Nanjing 210044, China)

Abstract PSO has become one of the effective methods for solving multi-objective optimization problems, and the key of PSO is the proper settings of the inertial, local and global velocity coefficients. To solve the problem, separating settings for each speed coefficient in existing algorithm with ignoring potential relevancies, an improved multi-objective particle optimization for balancing each formula element was proposed. For the purpose of guiding the evolutionary particle swarm in a potential global optimum, our algorithm can dynamically adjust the speed of each particle coefficients to balance inertia, local and global effects of three speed items during the searching process. Thus the searching capability and accuracy of the new algorithm is more accurate. Meanwhile, our algorithm can not only balance the capacity of exploitation and exploration, but also improve the efficiency in solving complex multi-objective optimization problem. The experimental results indicate that the new algorithm outperforms other 5 classical evolutionary algorithms in terms of convergence speed and distribution on 7 multi-objective benchmark functions.

Keywords Particle swarm optimization, Balance, Speed coefficient, Adaptive, Multi-objective optimization

1 引言

多目标优化问题(Multi-Objective Optimization Problem, MOP)通常存在多个彼此冲突的目标,相比传统单目标优化问题,其优化结果不再是单一最优值而是 Pareto 最优解集,因其普遍存在于工程和科学领域,所以该类问题的求解方法一直以来都是学术界和工程界关注的热点^[1]。与传统数学方法相比,进化算法凭借其带方向随机搜索和一次运行即可获得逼近 Pareto 前沿解集的能力,被广泛应用于 MOP 问题的求解^[2]。

粒子群优化算法(Particle Swarm Optimization, PSO)^[3]是一种模拟鸟群迁徙行为的进化算法。2002 年 Coello 等人基于 PSO 算法首次提出用于求解多目标优化问题的多目标粒子群优化算法(Multi-Objective Particle Swarm Optimiza-

tion, MOPSO)^[4],其因易理解、易实现、快收敛、设置参数少等优点,已经被广泛应用在工程优化、数据挖掘、优化控制等领域^[5-7]。同时,多目标优化问题的最优解集特征和粒子群算法的快速收敛特征,使得多目标粒子群优化算法面临平衡进化过程中的探索(Exploration)和探究(Exploitation)等新的挑战,而解决办法最终都会体现在粒子速度更新方程上,主要包括 3 个部分^[8],分别为惯性项、全局和局部最优项。因此必须对 3 项进行合理设置,以便于更好地提高粒子群优化算法求解多目标问题的性能。

从国内外的文献调研来看,对惯性权重 w 和局部最优项加速系数 c_1 以及全局最优项加速系数 c_2 独立优化改进的研究虽有很多^[9-13],但是对于 3 个参数均衡协同自适应优化的研究成果甚少。Shi 和 Eberhart 于 1998 年首次提出了惯性权重的概念,提出了固定的惯性权重和 c_1, c_2 ,但是控制参数一般

到稿日期:2015-10-18 返修日期:2016-01-18 本文受国家自然科学基金(61403206),江苏省自然科学基金(BK20151458)资助。

耿焕同(1973-),男,教授,博士生导师,CCF 高级会员,主要研究方向为计算智能、多目标优化;赵亚光(1990-),女,硕士生,主要研究方向为多目标优化;陈 哲(1991-),男,硕士生,主要研究方向为多目标优化;李辉健(1989-),男,硕士生,主要研究方向为多目标优化。

具有问题依赖性,固定权值不能适应所有的问题;Wang H等根据种群所处进化状态和 Pareto 差熵,设计粒子运动的 3 个参数调节策略;Bansal J C 等提出了一种基于迭代次数变化选取固定惯性权重的策略,极大地改善了 PSO 算法的性能,但是线性权重算法也存在缺陷,迭代初期,即使粒子已接近全局最优点,由于其局部搜索能力较弱也可能被错过,迭代后期由于粒子全局搜索能力变弱,容易陷入局部最优,最大迭代次数较难预测,从而影响算法的性能;韩洪江等采用正弦调整和自适应相结合调整惯性权重的策略,综合考虑收敛速度和个体适应值,对惯性权重进行二次优化;Cooren Y 等从粒子运动参数选择角度提出平衡策略,如线性下降惯性权重参数和自适应参数等。它们虽然在性能上有所改进,但早熟收敛问题依然没有从根本上得到解决,尤其是后期粒子多样性变异导致收敛速度变慢,极易陷入局部最优。并且这些算法主要应用于单目标函数优化问题,在多目标优化方面的应用相对较少。

为解决现有算法仅单独设置各速度项系数导致优化效率不高的问题,提出一种均衡各速度项系数的多目标粒子群优化算法(Multi-objective Particle Swarm algorithm with Balancing each Speed Coefficient, bcMOPSO)。其主要思想是借助自适应网格法^[15]来保持外部档案的多样性,并通过分析每个进化粒子与其局部和全局引导粒子在空间上的关系,动态赋予每个粒子独立的惯性权重,同时,也对每个粒子的局部和全局项进行动态自适应的调整。

2 问题定义

2.1 多目标优化问题

不失一般性,以最小化为例,多目标优化问题定义如下:

$$\begin{cases} \min F(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_M(\vec{x})\} \\ \text{s. t. } g_j(\vec{x}) \leq 0, 0 \leq j \leq p \\ h_k(\vec{x}) = 0, 0 \leq k \leq q \end{cases} \quad (1)$$

其中, $\vec{x}$ 为 n 维实数空间 R^n 的向量。 $f_1(\vec{x}), f_2(\vec{x}), \dots, f_M(\vec{x})$ 为相互冲突的 M 个优化目标,当 $M > 1$ 时,称为多目标优化。目标函数将决策空间映射到目标空间, $R^n \rightarrow \Omega \in R^M$, Ω 为 M 维目标空间。 $g_j(\vec{x})$ 为 p 个不等式约束, $h_k(\vec{x})$ 为 q 个等式约束。多目标优化算法的目标就是获得 Pareto 最优解集,同时使其 Pareto 最优前沿尽可能分布均匀并且收敛性好。

2.2 多目标粒子群优化算法及速度更新公式权重设置

在多目标粒子群优化算法中,一个无质量的粒子 i 可以由位置向量 x_i 和速度向量 v_i 表示,其中 $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,D}]$, $v_i = [v_{i,1}, v_{i,2}, \dots, v_{i,D}]$, $i = 1, 2, \dots, N$, N 是种群中的粒子个数, D 是决策变量个数,种群中的每一个粒子在进化过程中根据式(2)更新速度和位置。

$$\begin{cases} v_i(t+1) = wv_i(t) + c_1r_1(pbest_i - x_i(t)) + c_2r_2 \\ \quad (gbest - x_i(t)) \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases} \quad (2)$$

其中, t 表示迭代次数, $w \geq 0$ 表示惯性权重系数, $c_1, c_2 \geq 0$ 表示加速系数, r_1 和 r_2 是在 $[0, 1]$ 上均匀分布的随机数, $pbest_i$ 表示第 i 个粒子的局部最优解, $gbest$ 表示整个群体的全局最优解。将粒子群优化算法运用于多目标优化问题,关键问题之一是如何确定每个粒子的局部最优解 $pbest$ 和群体全局最

优解 $gbest$, 而非支配解之间不可比,迫使算法需要重新选择 $pbest$ 和 $gbest$ 。考虑到本文侧重于 3 个速度项系数的研究,因此 $pbest$ 和 $gbest$ 的选取见文献[15]中的策略。

在速度更新公式中,第一部分为粒子当前速度乘以权重进行加速,表示粒子对当前自身运动状态的信任,依据自身的速度进行惯性运动;第二部分为局部最优,表示粒子本身的思考;第三部分为全局最优,表示粒子间的信息共享与相互合作。图 1 是粒子的飞行过程示意图。

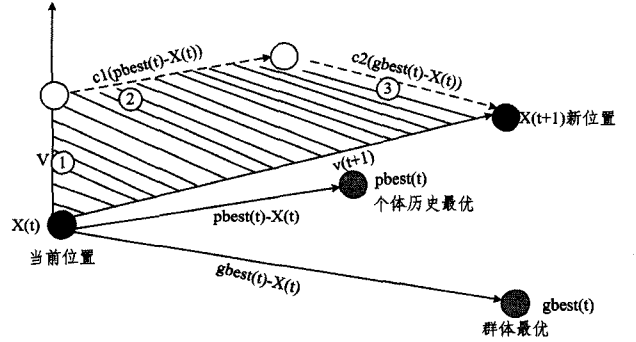


图 1 粒子飞行过程示意图

如图 1 所示,粒子从当前位置 $x(t)$ 出发,在 $v, c_1(pbest - x)$ 和 $c_2(gbest - x)$ 的作用下,分别沿速度矢量①—③方向运动,粒子速度 $v(t+1)$ 由速度矢量①、②和③合成。由于参数 r_1, r_2 的随机性,粒子会落在图中阴影区域的任意位置。

若没有①部分,即没有惯性作用,速度只取决于当前粒子个体极值($pbest$)和全局极值($gbest$)。若粒子位于全局最好的位置,则其保持静止,否则就飞向 $pbest$ 和 $gbest$ 的加权中心,在 $pbest$ 和 $gbest$ 附近局部搜索。如果 $c_1 = 0$,即粒子没有自我认知能力,由于所有粒子都快速向 $gbest$ 靠近,其收敛速度快,将导致算法陷入局部最优。如果 $c_2 = 0$,即粒子没有社会认知能力,粒子之间缺乏信息共享,算法缺乏全局搜索能力。如果 $c_1 = c_2 = 0$,粒子将一直沿着当前速度方向直线飞行,直到到达边界,此时算法失去寻优能力。由此分析不难看出,惯性权重 w 、加速系数 c_1 和 c_2 对粒子的飞行有着极其重要的影响。

在算法进化初期,为了加强算法探索能力,更好地定位潜在解空间,需赋予粒子较大的惯性权重,以此加速粒子的运动;同时需要增强全局最优引导粒子 $gbest$ 对粒子的影响力 c_1 ,确保寻优方向的正确性,此时粒子速度 v 以及 $gbest$ 的速度在搜索过程中起主导作用。进化中期,需要动态调整运动粒子、 $pbest$ 、 $gbest$ 三者间的权重系数 w, c_1, c_2 ,维持算法全局搜索和局部搜索能力的动态平衡。进化后期,通过逐渐减小惯性,减弱全局搜索能力,增强局部探究能力,有利于更快地定位到最优解的附近,提高局部搜索能力和解精度。

因此挖掘运动粒子、 $pbest$ 和 $gbest$ 间的潜在关系,从而实现惯性权重 w 、系数 c_1 和 c_2 的均衡调整,对获得算法全局搜索能力和局部搜索能力的动态平衡有着重要意义。

3 均衡各速度项系数的多目标粒子群优化算法(bcMOPSO)

3.1 惯性权重均衡策略

惯性权重 w 是 MOPSO 算法中极其重要的参数,使粒子

具有保持过去惯性的特点,即控制着速度上一代变化量对当前变化量的影响。惯性权重不仅影响粒子的局部和全局搜索能力,还影响算法的收敛性能。较大的 w 能够搜索到之前未能到达的区域,算法的全局搜索能力加强,有利于跳出局部极小点;较小的 w 能够更加细致地搜索附近的区域,算法的局部搜索能力增强,有利于算法收敛。

在早期 MOPSO 算法中, w 的值是静态的,这就导致其无法适应不同的问题。而在经典的多目标粒子群算法中,对惯性权重的设置如式(3)所示。其中 $w_{\max}$ 和 $w_{\min}$ 分别代表初始和最终的惯性权重的值, $iter_{\max}$ 代表最大迭代次数,通常情况下 $w_{\max}=0.9$, $w_{\min}=0.4$ 。 w 值随着迭代次数的增加而减小,算法的搜索策略也由全局搜索转变为局部搜索,但是其值只和最大迭代次数以及当前迭代次数有关,忽略了粒子的适应度值以及粒子的全局最优值和局部最优值等重要信息。

$$w(t+1)=w_{\max}-(w_{\max}-w_{\min})\times\frac{t}{iter_{\max}} \quad (3)$$

基于以上分析,针对传统惯性权重 w 中发现的问题,文中运用粒子现状信息与历史信息^[14],并且综合对粒子的个体极值以及全局极值的空间关系分析,动态、自引导设置粒子的 w , c_1 和 c_2 。仔细观察可以发现,每一代中的粒子大概分为3种情况:在最优值附近的粒子、介于两者之间的粒子、离最优值较远的粒子。如果粒子当前的目标值与 g_{best} 差距比较大,即认为该粒子与最优值差距较大,则使该粒子的飞行速度加快,使得搜索的范围更广;当粒子就在最优值附近,则减慢该粒子的飞行速度,在最优值附近进行搜索;介于两者之间的粒子用于平衡全局搜索与局部搜索。粒子的历史信息的变化对粒子的惯性权重也有着重要的影响。本文用进化速度来表示粒子历史信息。如果粒子的进化速度比较慢,则加快它的飞行速度;反之,则减慢飞行速度,经过调研,采用如下设置过程:每个粒子的惯性权重设置如式(4)所示,粒子进化速度的计算公式如式(5)所示。

$$w_i(t+1)=w_{\max}*[1-\frac{1}{\sqrt{(f_{p_{best}_i}(t)-f_i(t))^2+hv_i(t)^2+1}}] \quad (4)$$

$$hv_i(t)=\frac{1}{\sqrt{(f_{p_{best}_i}(t-1)-f_{p_{best}_i}(t-2))^2+1}} \quad (5)$$

其中, w_i 表示第 i 个粒子的惯性权重, $f_{p_{best}_i}$ 表示第 i 个粒子的个体最优函数值,它反映了粒子 i 的历史信息, $h v_i$ 表示第 i 个粒子的进化速度, f_i 表示第 i 个粒子的函数值,用来反映粒子 i 的当前现状, $f_{p_{best}}$ 表示群体最优函数值。

由式(4)可知,由于每个粒子下一代的惯性权重由当前代的目标值和全局最优值的差值及历史进化速度动态确定,因此每个粒子的惯性权重随着代数的不同而不同,并且在同一进化代中,每个粒子的惯性权重也不相同。不同问题的同一个粒子由于评估函数不同,惯性权重也不相同。经过实验验证,得出式(6),即取自适应惯性权重和传统的惯性权重中相对较大的一个值来平衡整个算法,避免部分粒子在运动的过程中陷入局部最优。

$$w_{\text{new}}=\max(w_i(t+1), w(t+1)) \quad (6)$$

3.2 局部最优项均衡策略

在粒子群算法的速度更新方程中,第二部分为自我认知部分,表示粒子对自身历史的思考,其中局部最优项加速系数 c_1 的设定对个体极值的寻优能力有着至关重要的影响。在自适应惯性权重策略中,分析了每一代中大致有3种粒子,如果当前粒子的目标值与局部最优值的差距较大,则应该加速粒子的飞行,每个粒子的 c_1 是由当前代的局部最优值和目标值所决定的,所以每个粒子的 c_1 随着代数的不同而不同,同一代进化的粒子中 c_1 也不同,因此对 c_1 设置为:

$$c_{1i}=\frac{1}{6}\pi-\arcsin(\frac{1}{\sqrt{(f_{p_{best}_i}(t)-f_i(t))^2+1}}) \quad (7)$$

其中, $f_{p_{best}_i}$ 为第 i 个粒子的局部最优值, f_i 为第 i 个粒子的目标值,由式(7)可以看出,如果当前粒子的局部最优值和目标值差距比较大的时候, c_1 应该加大,有利于整个算法更好地寻优,从而提高算法的收敛精度和收敛速度。

3.3 全局最优项均衡策略

在粒子群算法的速度更新方程中,第三部分为社会认知部分,表示粒子在群体中的信息共享和相互合作。 c_2 的设置影响着整个粒子的进化方向,在对粒子的不同状态进行分析之后,若当前粒子的目标值与全局最优值的差距比较大,则该粒子的飞行速度应该加快,即 c_2 也应该随之变大,经分析, c_2 是由当前代的全局最优值和目标值所决定的,所以,每个粒子的 c_2 随着代数的不同而不同,同一代进化的粒子中 c_2 的值也不同,具体公式如式(8)所示:

$$c_{2i}=\frac{2}{3}\pi-\arcsin(\frac{1}{\sqrt{(f_{p_{best}}(t)-f_i(t))^2+1}}) \quad (8)$$

其中, $f_{p_{best}}$ 为粒子的全局最优值, f_i 为第 i 个粒子的目标值,由式(8)可以看出,若当前粒子的全局最优值和目标值差距比较大, c_2 加大,有利于整个粒子群更好地搜索到全局最优解,有利于粒子进行信息共享和交流。

这样粒子群优化算法可以通过调节运动方程的参数 w , c_1 和 c_2 来灵活控制整个算法的探究和探索过程。最后得出了算法的全新速度和位置更新方程,如式(9)所示:

$$\begin{cases} v_i(t+1)=w_{\text{new}}v_i(t)+c_{1i}r_1(p_{best_i}-x_i(t))+c_{2i}r_2 \\ \quad (g_{best}-x_i(t)) \\ x_i(t+1)=x_i(t)+v_i(t+1) \end{cases} \quad (9)$$

3.4 bcMOPSO 算法流程

参照多目标粒子群优化算法框架,结合惯性权重 w 和加速系数 c_1 , c_2 的均衡设置策略,给出 bcMOPSO 算法的主要流程,具体如下:

Step 1 参数设置与初始化。初始化粒子种群 $P(t=0)$, 种群规模为 N ; 初始化惯性权重 w 、加速系数 c_1 和 c_2 , 外部档案群体 G , 规模为 K ; 进化代数 t , 最大迭代次数为 $iter_{\max}$ 。

Step 2 适应度计算。评估种群 P 中每个粒子 $x(t)$ 对应的各个目标值。

Step 3 更新个体局部最优 p_{best} 和外部档案群体 G 。依据 Pareto 支配关系,首先对前后代的粒子进行选优确定新的粒子 $x(t)$;其次对每个粒子 $x(t)$ 的局部最优 p_{best} 进行更新;然后对外部档案群体 G 进行更新,当档案规模超过阈值 K 时,采用自适应网格法进行筛选和剔除。

Step 4 全局最优 g_{best} 选择。根据自适应网格法进行概率选择全局最优 g_{best} 。

Step 5 粒子更新操作。根据式(3)一式(6)依次计算每粒子 $x(t)$ 的惯性权重 w ; 根据式(7)和式(8)分别计算每粒子 $x(t)$ 的加速系数 c_1 和 c_2 ; 然后根据式(9)计算 $t+1$ 时刻的粒子速度和位置, 并进行粒子越界修正。

Step 6 算法终止判断。 $t=t+1$, 判断 t 是否小于或等于 $iter_{max}$, 若是则转 Step 2; 否则输出外部档案群体 G 后算法结束。

4 实验及结果分析

4.1 实验设计

4.1.1 多目标测试问题和对比较算法

文中采用 ZDT 系列^[16]和 FON、KUR^[17]共 7 个测试函数构成多目标测试问题集, 分别是 ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, FON, KUR。表 1 列出了 7 个多目标优化测试问题的函数表达式以及约束条件。为了验证 bcMOPSO 算法的优化性能, 选取了 5 种经典算法 agMOPSO^[15], GDE3^[18], SMP-SO^[19], MOPSO-PD^[20], SDMOPSO^[21] 作为对比算法。agMOPSO 算法采用了自适应网格技术来更新外部档案和选择全局最优解的策略; GDE3 算法采用差分进化策略; SMPSO 通过控制粒子速度, 对种群进行更新; MOPSO-PD 采用种群分解策略进行优化; SDMOPSO 定义了一种智能的方法来更新种群, 同时使用分解策略增加种群的多样性。

表 1 测试函数(以最小值为例)

问题	定义	约束条件
ZDT1	$f_1(x) = x_1$	
	$f_2(x) = g(x)(1 - \sqrt{x_1/g(x)})$	$n=30, 0 \leq x_i \leq 1$
	$g(x) = 1 + 9(\sum_{i=2}^n x_i)/(n-1)$	$i=1, 2, \dots, n$
ZDT2	$f_1(x) = x_1$	
	$f_2(x) = g(x)(1 - (x_1/g(x))^2)$	$n=30, 0 \leq x_i \leq 1$
	$g(x) = 1 + 9(\sum_{i=2}^n x_i)/(n-1)$	$i=1, 2, \dots, n$
ZDT3	$f_1(x) = x_1$	
	$f_2(x) = g(x)(1 - \sqrt{x_1/g(x)} - x_1 \sin(10\pi x_1)/g(x))$	$n=30, 0 \leq x_i \leq 1$
	$g(x) = 1 + 9(\sum_{i=2}^n x_i)/(n-1)$	$i=1, 2, \dots, n$
ZDT4	$f_1(x) = x_1$	
	$f_2(x) = g(x)(1 - \sqrt{x_1/g(x)})$	$n=10, 0 \leq x_i \leq 1$
	$g(x) = 1 + 10(n-1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i))$	$-5 \leq x_i \leq 5$ $i=2, 3, \dots, n$
ZDT6	$f_1(x) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$	
	$f_2(x) = g(x)(1 - (x_1/g(x))^2)$	$n=10, 0 \leq x_i \leq 1$
	$g(x) = 1 + 10(n-1) + \sum_{i=2}^n (x_i^2 - 10 \cos(4\pi x_i))$	$i=1, 2, \dots, n$
FON	$f_1(x) = 1 - \exp(-\sum_{i=1}^3 (x_i - 1/\sqrt{3})^2)$	$i=1, 2, 3$
	$f_2(x) = 1 - \exp(-\sum_{i=1}^3 (x_i + 1/\sqrt{3})^2)$	$x_i \in [-4, 4]$
KUR	$f_1(x) = \sum_{i=1}^{m-1} (-10 \exp(-0.2) \sqrt{x_i^2 + x_{i+1}^2})$	
	$f_2(x) = \sum_{i=1}^m (x_i ^{0.8} + 5 \sin x_i^3)$	$x \in [-5, 5]$

4.1.2 实验参数

在对比实验中, 所有算法的种群大小均设置为 $N=100$; 全局外部归档均设置为 $K=100$; 算法的最大迭代次数为 $iter_{max}=300$; 测试函数的评估次数均设置为 $FE=30000$ 。

4.1.3 性能指标

反转世代距离 (Inverted Generational Distance, IGD)^[22] 被用来度量真实 Pareto 前沿到近似 Pareto 前沿间的距离指标, 能综合评估算法获得近似的 Pareto 前沿的收敛性和多样性, IGD 的值越低, 表明算法获得的近似 Pareto 前沿的收敛性和多样性越好, 算法也就越接近真实的 Pareto 前沿。令 P 为真实的 Pareto 最优解集^[23], A 是由进化算法获得的近似 Pareto 最优解集, IGD 指标值的计算过程如式(10)所示:

$$IGD = \frac{1}{|P|} \sum_{i=1}^{|P|} Dist_i \quad (10)$$

$$Dist_i = \min_{j=1}^{|A|} \sqrt{\sum_{m=1}^M \left(\frac{f_m(p_i) - f_m(a_j)}{f_m^{\max} - f_m^{\min}} \right)^2} \quad (11)$$

其中, 式(11)是最小归一化的欧氏距离; $f_m^{\max}$ 和 $f_m^{\min}$ 分别是第 m 个目标上的最大值和最小值, $m=1, 2, \dots, M$, M 为目标个数; $p_i \in P, i=1, 2, \dots, |P|$; $a_j \in A, j=1, 2, \dots, |A|$ 。在本实验中, 7 个测试函数的真实 Pareto 最优解集的采样样本数均设置为 $|P|=1000$ 。

另外, 采用 Deb 提出的经典多样性评价指标 Δ ^[24] 进行评估, 指标 Δ 的定义如下:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}} \quad (12)$$

其中, d_i 是在 Pareto 最优解集 P 中相邻解的距离; $\bar{d}$ 是所有 d_i 的平均值; d_f 和 d_l 分别为算法获得的边界解与相应极端解的欧氏距离。 Δ 的值越小, 表明算法获得 Pareto 最优解的多样性越好。

4.2 实验结果及比较分析

表 2 列出了 bcMOPSO 和 5 种对等比较算法在 7 个多目标测试问题上的 IGD 性能指标值, 其中 IGD 的平均值(mean)和方差值(std)是同一算法在同一测试问题上独立运行 30 次的统计结果。算法的收敛性和多样性采用综合的 IGD 性能评价指标来进行评价。 t -test 值是本文算法与其他对等算法在同一测试问题上进行 t 检验^[10]时的值; “+”、“=”和“-”表示新算法获得的 IGD 值在显著性水平为 5% 的双尾 t 检验中分别优于、等于和劣于对应列的对等算法在对应的测试问题上的显著性区分结果; “Score”表示 bcMOPSO 算法显著优于对等算法在 7 个测试问题中的净胜得分, 即得“+”与得“-”的测试问题个数之差。除了在复杂问题 ZDT4 上, bcMOPSO 算法略低于 SMPSO 算法的优化结果外, 在其它 6 个测试函数上, bcMOPSO 算法获得的 IGD 值均好于 5 种对比算法, 这表明 bcMOPSO 算法具有更好的收敛性和多样性。同时, 在表 2 的 t 检验结果中, bcMOPSO 算法对其他 5 种对比算法的净胜得分均为正数, 在对 agMOPSO 和 GDE3 时的净胜得分均为 7 分, 对 SMPSO 时的净胜得分为 2 分, 对 MOPSO-PD 和 SDMOPSO 的净胜得分均为 5 分, 这表明 bcMOPSO 算法总体优于另外 5 种对比算法, 究其原因 bcMOPSO 算法通过均衡各速度项系数来动态地平衡整个算法的探索与探究过程。

表 2 5 种对比算法在 7 个测试函数上的 IGD 性能对比结果(最优情况用黑体表示)

测试函数		bcMOPSO	agMOPSO	GDE3	SMPSO	MOPSO-PD	SDMOPSO
ZDT1	Mean	5.06E-03	2.04E-02	1.80E-02	6.17E-03	5.30E-03	7.630E-03
	Std	2.62E-03	1.22E-02	1.22E-02	7.37E-02	3.90E-03	4.00E-03
	t-test		-6.98E+00	-5.43E+00	-1.76E+00	-2.33E+00	-9.43E+00
			+	+	+	+	+
ZDT2	Mean	3.51E-03	1.28E-02	1.90E-01	7.57E-02	2.78E-01	3.02E-02
	Std	4.48E-03	7.58E-03	9.59E-02	1.44E-02	3.80E-02	5.01E-03
	t-test		-9.86E+00	-6.43E+00	-2.50E+00	-2.06E+00	-6.02E+00
			+	+	+	+	+
ZDT3	Mean	1.20E-03	1.20E-02	3.07E-03	3.57E-02	3.06E-03	4.30E-02
	Std	8.27E-04	3.29E-03	5.89E-02	9.86E-03	4.13E-04	3.90E-03
	t-test		-4.89E+00	-1.89E+01	-8.76E-01	-1.29E-01	-4.48E+00
			+	+	=	=	+
ZDT4	Mean	3.88E-02	4.82E+00	8.51E-02	7.64E-04	4.23E-02	2.55E+00
	Std	1.84E-02	2.48E+00	1.63E-02	5.32E-04	3.21E-02	1.43E+00
	t-test		-2.38E+00	-9.78E+00	4.83E-03	-6.52E+00	-2.50E+00
			+	+	-	+	+
ZDT6	Mean	9.81E-04	3.24E-02	1.65E-02	1.56E-03	4.00E-03	1.30E-02
	Std	3.09E-03	7.92E-03	1.52E-01	3.42E-02	3.90E-03	3.58E-03
	t-test		-7.86E+00	-3.69E+00	-1.54E-01	7.92E+00	-7.33E+00
			+	+	=	+	+
FON	Mean	8.56E-03	4.78E-01	3.78E-01	9.56E-01	2.56E-01	1.96E-01
	Std	2.18E-03	3.47E-03	4.89E-02	3.89E-03	4.31E-02	3.62E-02
	t-test		-6.81E+00	-5.19E+00	-1.63E+00	5.26E+00	-1.94E-01
			+	+	+	+	=
KUR	Mean	9.71E-04	4.11E-01	8.56E-01	3.49E-02	3.98E-01	2.96E-01
	Std	1.49E-03	4.46E-02	9.24E-02	1.26E-02	4.56E-02	3.98E-02
	t-test		-4.81E+00	-3.19E+00	-1.93E-01	-4.16E+00	-1.54E-01
			+	+	=	+	=
Better(+)			7	7	3	6	5
Same(=)			0	0	3	1	2
Worse(-)			0	0	1	0	0
Score			7	7	2	5	5

表 3 列出了 bcMOPSO 算法和 5 种对等比较算法在 7 个多目标测试问题上的分布性 Δ 指标值,其中 Δ 的平均值(mean)和方差值(std)是同一算法在同一测试问题上独立运行 30 次的统计结果。实验结果表明,除了在 ZDT4 测试问题上,bcMOPSO 算法的多样性略差于 SMPSO 算法外,在其它

6 个测试问题上,bcMOPSO 算法的分布性 Δ 指标值均低于对比算法,这说明了本文提出的算法比 5 种对比算法具有更好的多样性,其主要原因是 bcMOPSO 算法使用了各速度项系数自适应调整策略,有效地平衡了整个算法的探索与探究过程。

表 3 5 种对比算法在 7 个测试函数上的 Δ 性能对比结果(最优情况用黑体表示)

测试函数		bcMOPSO	agMOPSO	GDE3	SMPSO	MOPSO-PD	SDMOPSO
ZDT1	Mean	4.26E-02	3.90E-01	5.31E-02	4.69E-02	3.98E-02	6.79E-02
	Std	1.25E-03	2.56E-03	3.01E-03	1.36E-03	1.98E-03	3.12E-03
ZDT2	Mean	4.65E-02	9.89E+00	5.05E-01	4.72E-01	4.98E-01	4.85E-01
	Std	2.01E-03	3.61E+00	8.45E-03	5.42E-02	6.45E-03	6.23E-03
ZDT3	Mean	5.94E-01	4.23E+00	6.25E-01	6.02E-01	9.56E-01	6.42E-01
	Std	1.85E-03	6.45E+00	5.21E-03	3.75E-03	2.89E-03	3.69E-03
ZDT4	Mean	3.29E-02	4.05E+00	5.29E+00	2.78E-02	4.98E-02	4.17E-02
	Std	2.02E-03	1.62E+00	1.12E+00	1.22E-03	2.32E-03	4.62E-03
ZDT6	Mean	6.45E-01	9.89E-01	7.78E-01	7.98E-01	8.98E-01	8.42E-01
	Std	2.03E-03	1.26E-02	1.22E-02	2.22E-02	3.12E-02	4.18E-02
FON	Mean	1.63E-01	7.56E-01	9.56E-01	2.56E-01	3.56E-01	3.26E-01
	Std	6.56E-03	2.15E-02	1.15E-03	7.15E-03	8.15E-03	2.15E-03
KUR	Mean	5.23E-02	6.23E-01	4.21E-01	4.99E-02	6.18E-01	2.13E-01
	Std	1.69E-03	3.12E-02	2.36E-02	1.09E-03	2.59E-02	3.61E-02

为了更直观地反映不同算法获得 Pareto 最优解集的收敛性和多样性情况,选取了 bcMOPSO, agMOPSO, MOPSO-PD 和 SDMOPSO 算法在 ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 测试函数上的某一次运行情况。从图 2 可看出, bcMOPSO 算法在 ZDT1 函数上具有更好的收敛性和多样性, agMOPSO 效果不理想, MOPSO-PD 以及 SDMOPSO 均没有收敛到真实 Pareto 最优前沿。在图 3 中, bcMOPSO 算法在 ZDT2 函数上获得了更好的收敛性和多样性, agMOSPO 搜索不全面,

MOPSO-PD 和 SDMOPSO 虽然分布性不错,但是收敛性不够。在图 4 中, bcMOPSO 算法在 ZDT3 函数上也获得了较好的收敛性和多样性,但 agMOPSO 不仅搜索不全面而且分布性不好。在图 5 中, bcMOPSO 算法在 ZDT4 函数上虽然有部分点没有完全收敛到真实的 Pareto 最优前沿,但具有很好的分布性,其主要原因是进化代数不够,而 agMOPSO 的收敛效果最不理想。在图 6 中, bcMOPSO 在 ZDT6 函数上不仅搜索到的 Pareto 最优解集分布更广,且收敛性也较好, agMOPSO

算法效果次之, MOPSO-PD 和 SDMOPSO 虽收敛性还可以,

但是分布性不好。

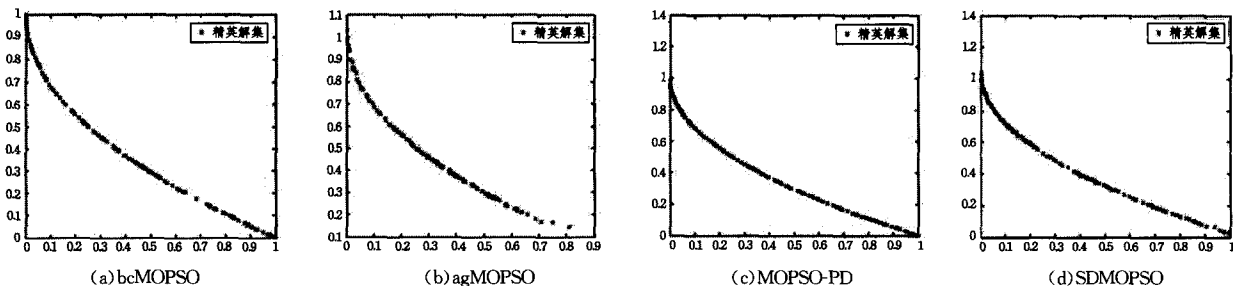


图2 不同算法在 ZDT1 上获得的 Pareto 最优解集(精英解集)

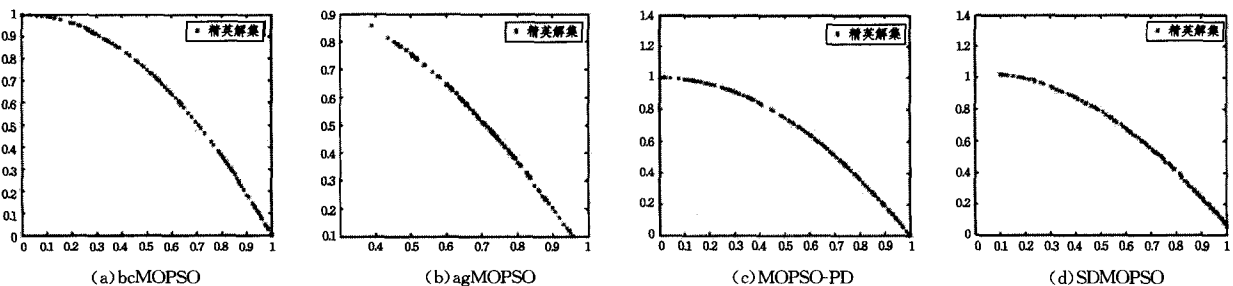


图3 不同算法在 ZDT2 上获得的 Pareto 最优解集(精英解集)

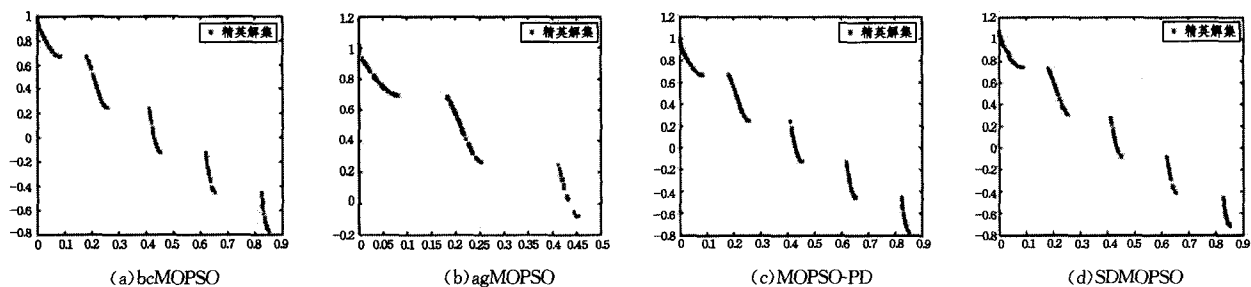


图4 不同算法在 ZDT3 上获得的 Pareto 最优解集(精英解集)

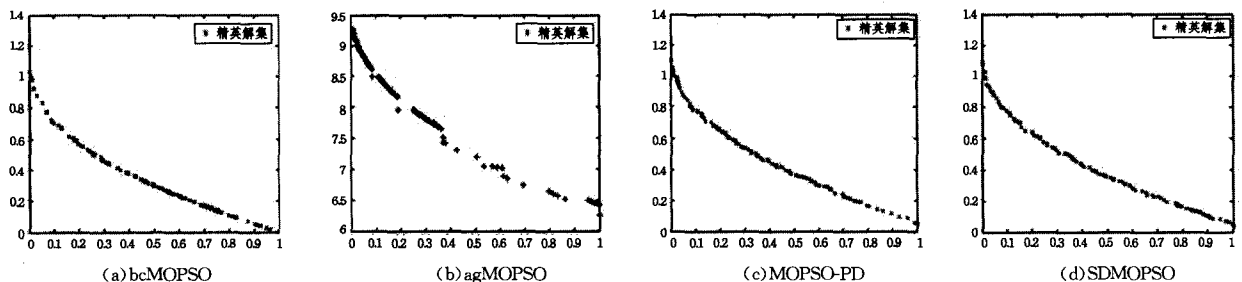


图5 不同算法在 ZDT4 上获得的 Pareto 最优解集(精英解集)

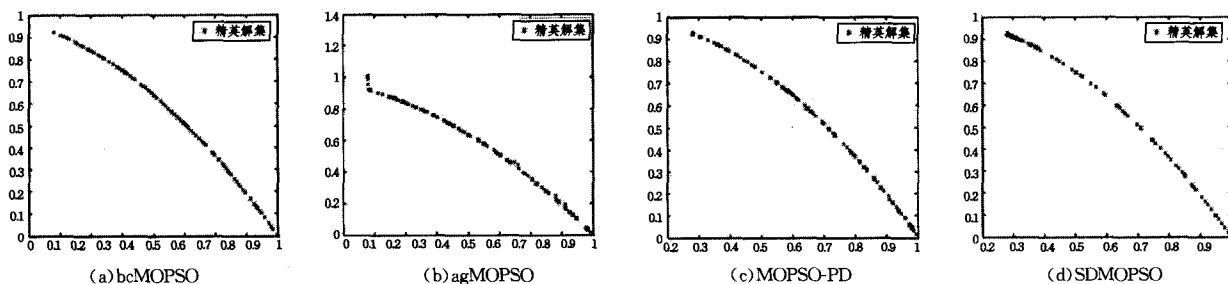


图6 不同算法在 ZDT6 上获得的 Pareto 最优解集(精英解集)

与对比算法的优化效果相比, bcMOPSO 算法能够保持比较好的结果主要有以下几个原因。首先, 算法能够利用当前和历史信息自适应调整粒子飞行速度和方向, 从而赋予每个粒子在全局搜索策略和局部搜索策略中进行动态切换的

能力, 以平衡算法的探索和探究能力。其次不同于传统算法凭经验设定粒子速度更新项系数的策略, 新算法对于每一项系数都考虑到了局部最优和全局最优的信息, 并且对于每个粒子都设置了独立的 w, c_1 和 c_2 , 增加了种群的多样性, 因此

新算法所得到的 Pareto 最优解集更优。

结束语 针对现有算法仅单独设置各速度项系数导致优化效率不足的问题,提出了一种均衡各速度项系数的多目标粒子群优化算法。新算法将局部最优和全局最优信息分别引入到速度更新公式的 3 项中。综合历史和现状信息对每个粒子的惯性权重 w 进行动态自适应设置,并对 c_1, c_2 也进行了动态自适应设置,采用 3 项均衡来引导种群的进化方向,有效地兼顾了探索和探究的过程,以此来更好地平衡整个算法的收敛性和多样性,极大地提高了搜索效率。通过 ZDT 系列、FON、KUR 测试函数测试,与 5 种经典的算法作对比,在 IGD 性能指标和多样性评估指标上都获得了一定的优势,表明新算法在总体上具有更好的优化性能。下一步工作可以将本文的均衡策略引进高维动态多目标粒子群优化算法中,拓宽粒子群优化算法的应用领域。

参考文献

- [1] Lalwani S, Singhal S, Kumar R, et al. A comprehensive survey: applications of multi-objective particle swarm optimization(mopso) algorithm[J]. Transactions on Combinatorics, 2013, 2(1): 39-101
- [2] Gong Mao-guo, Jiao Li-cheng, Yang Dong-dong, et al. Research on evolutionary multi-objective optimization algorithms [J]. Journal of Software, 2009, 20(2): 271-289(in Chinese)
公茂果, 焦李成, 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
- [3] Kennedy J, Eberhart R C. Particle swarm optimization[C]// IEEE International Conference on Neural Networks. 1995: 1942-1948
- [4] Coello Coello C A, Lechuga M S. Mopso: a proposal for multiple objective particle swarm optimization[C]// IEEE Congress on Evolutionary Computation, 2002: 1051-1056
- [5] Mukhopadhyay A, Maulik U, Bandyopadhyay S, et al. Survey of multiobjective evolutionary algorithms for data mining: part II [J]. IEEE Transactions on Evolutionary Computation, 2014, 18(1): 20-35
- [6] Onieva E, Hernandez-Jayo U, Osaba E, et al. A multi-objective evolutionary algorithm for the tuning of fuzzy rule bases for uncoordinated intersections in autonomous driving[J]. Information Sciences, 2015, 321(c): 14-30
- [7] Wei J, Jia L. A novel particle swarm optimization algorithm with local search for dynamic constrained multi-objective optimization problems[C]// IEEE Congress on Evolutionary Computation. 2013: 2436-2443
- [8] Yao Xin, Yang Sheng-xiang. Evolutionary computation for dynamic optimization problems[M]. Springer-Verlag Berlin Heidelberg, 2013
- [9] Shi Y, Eberhart R. A modified particle swarm optimizer[C]// IEEE World Congress on Computational Intelligence, 1998: 69-73
- [10] Wang H, Yen G G. Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system [J]. IEEE Transactions on Evolutionary Computation, 2015, 19(1): 1-18
- [11] Bansal J C, Singh P K, Saraswat M, et al. Inertia weight strategies in particle swarm optimization[C]// 2011 Third World Congress on Nature and Biologically Inspired Computing. 2011: 633-640
- [12] Han Hong-jiang, Li Zheng-rong, Wei Zhen-chun. An adaptive inertia weight particle swarm optimization and its simulation research[J]. Journal of System Simulation, 2006, 18(10): 2969-2971(in Chinese)
韩江洪, 李正荣, 魏振春. 一种自适应粒子群优化算法及其仿真研究[J]. 系统仿真学报, 2006, 18(10): 2969-2971
- [13] Cooren Y, Clerc M, Siarry P. MO-TRIBES. An adaptive multiobjective particle swarm optimization algorithm[J]. Computational Optimization and Applications, 2011, 49(2): 379-400
- [14] Smith E A, Boyd R, Richerson P J. Culture and the evolutionary process[J]. American Anthropologist, 1987, 89(1): 203-205
- [15] Hu Wang, Yen G G, Zhang Xin. Based on pareto entropy multi-objective particle swarm optimization [J]. Journal of Software, 2014, 25(5): 1025-1050(in Chinese)
胡旺, Yen G G, 张鑫. 基于 Pareto 熵的多目标粒子群优化算法 [J]. 软件学报, 2014, 25(5): 1025-1050
- [16] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results[J]. IEEE Transactions on Evolutionary Computation, 2000, 8(2): 173-195
- [17] Deb K, Thiele L, Laumanns M, et al. Scalable multi-objective optimization test problems[C]// IEEE Congress on Evolutionary Computation. 2002: 825-830
- [18] Kukkonen S, Lampinen J. Performance assessment of generalized differential evolution 3 with a given set of constrained multi-objective test problems[C]// IEEE Congress on Evolutionary Computation. 2009: 1943-1950
- [19] Nebro A J, Durillo J J, Garcia-Nieto J, et al. SMPSO: a new pso-based metaheuristic for multi-objective optimization[C]// IEEE symposium on Computational intelligence in multi-criteria decision-making. 2009: 66-73
- [20] Zhao Y, Liu H L. Multi-objective particle swarm optimization algorithm based on population decomposition[M]. Intelligent Data Engineering and Automated Learning-IDEAL 2013. Springer Berlin Heidelberg, 2013: 463-470
- [21] Moubayed N A, Petrovski A, Mccall J. A novel smart multi-objective particle swarm optimisation using decomposition [M]. Parallel Problem Solving from Nature, PPSN XI. Springer Berlin Heidelberg, 2010: 1-10
- [22] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731
- [23] ETH Zurich [EB/OL]. [2015-09-25]. <http://people.ee.ethz.ch/~sop/download/supplementary/testproblem>
- [24] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197