

多策略改进的多目标粒子群优化算法

杨景明^{1,2}, 穆晓伟¹, 车海军^{1,2†}, 呼子宇¹, 侯宇浩¹

(1. 燕山大学 工业计算机控制工程河北省重点实验室, 河北 秦皇岛 066004;

2. 燕山大学 国家冷轧板带装备及工艺工程技术研究中心, 河北 秦皇岛 066004)

摘要: 为了进一步提高多目标粒子群优化算法的收敛性和多样性, 提出一种多策略改进的多目标粒子群优化算法. 建立具有精英粒子领导的异构更新模式并设置个体学习增强因子项, 促使种群能够快速寻找真实 Pareto 最优解. 引入外部档案冗余机制, 利用其变异及对种群的干扰策略增强解的多样性, 避免算法早熟现象的发生. 仿真实验结果表明, 与其他几种优化算法相比, 所提出的算法表现出较好的收敛性和多样性.

关键词: 多目标优化; 粒子群算法; 多策略; 增强因子; 冗余机制

中图分类号: TP273

文献标志码: A

Improved multi-objective particle swarm optimization algorithm based on multiple strategies

YANG Jing-ming^{1,2}, MU Xiao-wei¹, CHE Hai-jun^{1,2†}, HU Zi-yu¹, HOU Yu-hao¹

(1. Key Lab of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China; 2. National Engineering Research Center for Equipment and Technology of Cold Strip Rolling, Yanshan University, Qinhuangdao 066004, China)

Abstract: An improved multi-objective particle swarm optimization algorithm based on multiple strategies(MIMOPSO) is proposed for the optimization mechanism of particle swarm optimization, which can further improve the convergence and distribution. First, a heterogeneous learning mode and a leadership of the elite particles to individual learning pattern are established to prompt the population to find true Pareto optimal solutions quickly. Then, the variation and disturbance to the population strategy of the external file redundancy mechanism is used to enhance the diversity of the solution and avoid the premature phenomenon of the algorithm. Experimental results show that, comparing with other several kinds of optimization algorithm, the proposed algorithm has better convergence and diversity.

Keywords: multi-objective optimization; particle swarm optimization algorithm; multiple strategies; intensifying factor; redundancy mechanism

0 引言

面对现代工程中出现的复杂情况, 决策者往往需要同时处理多个目标, 以期达到最优效果. 而这些目标彼此相互制约, 计算复杂. 如何使多个目标在决策空间中同时尽可能最佳优化, 这类问题通常被称为多目标优化问题(MOP).

基于仿生学的粒子群算法^[1] 因其在解决单目标优化问题中的高效性而引起许多学者关注并被应用到多目标的优化问题中. 文献[2]提出的多目标粒子群优化算法(MOPSO)具有里程碑意义, 该方法利用自适应网格机制对外部档案进行维护, 并对粒子和粒

子的飞行区域进行变异, 保持了粒子群算法较快的收敛速度, 但在处理多峰问题时易陷入局部最优, 多样性较差; 文献[3]探究了粒子群不同飞行模式下的优化性能, 提出了一种实参自适应的异构粒子群优化算法, 可有效提高粒子动态寻优能力, 然而需要考虑不同时期异构模式的特性; 文献[4]提出了基于粒子群与模糊蚁群的混合多目标优化(H-MOPSO-FACO)算法, 将模糊蚁群算法的最优粒子作为粒子群算法的全局最优粒子进行寻优, 尝试将不同算法的优势进行融合; 文献[5]采用偏好选择方式对外部精英粒子进行了筛选而获得接近预期目标的 Pareto 解; 文献[6]提

收稿日期: 2016-01-21; 修回日期: 2016-05-13.

基金项目: 河北省高等学校创新团队领军人才培养计划项目(LJRC013); 国家冷轧板带及装备工程研究中心开放课题项目(2012006); 河北省自然科学基金面上项目(F2016203249).

作者简介: 杨景明(1957-), 男, 教授, 博士生导师, 从事冶金机械综合自动化、先进控制及工程应用等研究; 穆晓伟(1989-), 男, 硕士生, 从事冶金机械综合自动化、多目标决策的研究.

†通讯作者. E-mail: hjche@ysu.edu.cn

出了一种基于文化框架的约束多种群粒子群优化算法,通过构建文化传播信仰空间促进各子群的信息交流,从而解决有约束的多目标优化问题;文献[7]针对多目标优化的多子群优化方案,提出了一种协同进化多子群粒子群多目标优化算法,不同的种群分别优化不同的目标,从而避免了多目标优化分配中各目标彼此冲突的问题;文献[8]分析并简化了多目标综合学习粒子群算法,使得改进的多目标综合学习粒子群算法收敛速度进一步加快,获得的Pareto解集分布度更加均匀;文献[9]采用一种基于分解的离散粒子群多目标优化算法求解复杂邻域网络聚类问题,进一步拓展了粒子群算法解决多目标优化问题的能力;文献[10]提出了一种带有种群极值变异的改进多目标粒子群优化(IMOPEO-PLM)算法,与多目标极值优化(MOPEO)算法相比,其性能有显著的提升,但也导致了该算法时间复杂度的增加;文献[11]将量子理论引入粒子群优化算法,提出了一种基于量子粒子群优化(QPSO)和拥挤距离排序的多目标量子粒子群优化算法,为多目标优化理论和算法的应用提供了一种新的思路;文献[12]将目标空间的Pareto前端变换到平行格坐标形式的二维网格中,通过计算种群熵作为反馈信息设计具有自适应调节开发与开采过程的进化策略,使得算法的收敛性和多样性都得到了提升。

目前大多数粒子群多目标优化算法的研究主要集中在快速性、分布性和逼近性这3方面。为此,本文对粒子群算法进行改进,提出一种多策略改进的多目标粒子群优化(MIMOPSO)算法。该算法充分利用精英解集的信息,在标准粒子群算法更新公式中引入精英粒子学习的异构飞行模式;提出粒子个体最优的精英解更新策略,同时增强自身学习因子,以提高粒子群的局部探索能力;引入精英冗余集的概念,对外部档案中的前沿粒子进行变异操作,促使种群粒子进一步向前沿探索,对种群中部分粒子随机扰动,以增大种群多样性。通过与其他3种算法和自身各策略算法的对比分析可知,所提出算法不仅可保持快速的收敛性,而且具有良好的逼近性和分布性,也进一步表明,与其他算法相比,该算法的整体性能有一定提高。

1 基本概念

1.1 多目标优化

多目标优化问题的数学模型描述如下:

$$\begin{aligned} \min F(x) &= (f_1(x), f_2(x), \dots, f_m(x))^T, \\ \text{s.t. } x &\in \Omega^n. \end{aligned} \quad (1)$$

其中: $x = (x_1, x_2, \dots, x_n)$ 为 n 维决策变量, Ω^n 为决策变量可行解空间, $F: \Omega^n \rightarrow R^m$ 是由决策空间到

目标空间的映射, m 为目标空间的维数。有关求解多目标问题的Pareto最优解定义^[13]如下。

定义1 (Pareto支配) 设 x, y 分别为式(1)中的两个决策变量,若 x Pareto y 支配,则当且仅当

$$\begin{aligned} \forall i \in \{1, 2, \dots, n\}, \exists j \in \{1, 2, \dots, n\}: \\ f_i(x) < f_i(y) \wedge f_j(x) < f_j(y). \end{aligned} \quad (2)$$

定义2 (Pareto最优解) 当且仅当

$$\nexists y \in \Omega^n: y \prec x^* \quad (3)$$

时, $x^* \in \Omega^n$ 被称为 Ω^n 上的Pareto最优解,即 Ω^n 中不存在支配 x^* 的向量,此时 x^* 被称为Pareto最优解。

定义3 (Pareto最优解集) Ω^n 中所有Pareto最优解构成的集合称为Pareto最优解集(PS),即

$$\text{PS} = \{x \mid \nexists y \in \Omega^n: y \prec x\}. \quad (4)$$

定义4 (Pareto前沿) 在Pareto最优解集PS中,所有Pareto最优解对应的目标函数值所构成的曲线(面)称为Pareto前沿,即

$$\text{PF} = \{F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T \mid x \in \text{PS}\}. \quad (5)$$

1.2 标准粒子群算法

最初的粒子群算法的速度项并没有系数,后经Shi等^[14]的改进,引入惯性权重,从而形成了目前通用的粒子群算法更新公式。设种群中的一个粒子 i ($i \in N$, N 为种群规模)在 D 维搜索空间中, t 时刻到 $t+1$ 时刻的速度和位置更新公式如下:

$$\begin{aligned} v_i(t+1, d) = \\ wv_i(t, d) + c_1 \text{rand}(\text{gbest}(t, d) - x_i(t, d)) + \\ c_2 \text{rand}(\text{pbest}_i(t, d) - x_i(t, d)), \end{aligned} \quad (6)$$

$$x_i(t+1, d) = x_i(t, d) + v_i(t+1, d), \quad d = 1, 2, \dots, D. \quad (7)$$

其中: w 为惯性系数,表示前一时刻的速度对下一时刻速度的影响程度; c_1 和 c_2 为学习因子,分别表示社会学习能力和自身学习能力, c_1 越大表示粒子社会协作能力越强,越趋向于社会最优位置, c_2 越大表示粒子自身探索能力越强; gbest 为所有粒子经历的最优位置; pbest_i 为粒子 i 曾经经历的最优位置; rand 为 $(0, 1)$ 上产生的随机数; d 为粒子 i 的第 d 维变量。

2 改进多目标粒子群优化算法

2.1 粒子更新的异构模式

种群粒子的更新方式是粒子群算法寻优的动力,标准粒子群通过对粒子速度的大小和方向进行调节实现种群粒子向最优位置移动,这个过程需要一定的调节时间,同时也受到种群粒子不同学习行为方式的影响。Nepomuceno等^[3]就这一问题进行了研究,在不同时期,通过变换种群粒子不同信息获得途径的方式

有助于提高种群的寻优能力. 因此, 本文引入遗传算法中的交叉操作增加粒子的学习模式, 通过使种群中的粒子与精英粒子的信息交换提高种群粒子的寻优性能. 模拟 Deb 等^[15]提出的二项式交叉算子作为粒子第2种学习方式, 提出粒子更新的异构模式如下:

$$x_i(t+1, d) = \begin{cases} \chi(x_i(t, d)), & \text{rand} < \text{MR}; \\ 0.5(1-\theta)g_{fk}(t, d) + (1+\theta)x_i(t, d), & \text{else.} \end{cases} \quad (8)$$

$$\chi(x_i(t, d)) = x_i(t, d) + wv_i(t, d) + c_1\text{rand}(\text{gbest}(t, d) - x_i(t, d)) + \beta_i c_2\text{rand}(\text{pbest}_i(t, d) - x_i(t, d)). \quad (9)$$

$$\theta = \begin{cases} (2r_1)^{\frac{1}{\mu_1+1}}, & r_1 \leq 0.5; \\ \frac{1}{\gamma}, & \text{else.} \end{cases} \quad (10)$$

$$\gamma = [2(1-r_1)]^{\frac{1}{\mu_1+1}}. \quad (11)$$

其中: β_i 为第 i 个个体学习强化因子, 其更新策略将在 2.2.2 节说明; rand 为 $(0, 1)$ 上产生的随机数; MR 为模式转换因子; $g_{fk}(t, d)$ 为算法在 t 代时从外部档案 Pareto 最优解集中随机选择的第 k 个粒子; r_1 为 $(0, 1)$ 上的随机数; μ_1 为交叉分布指数; θ 为交叉因子.

当 rand 小于 MR 时, 粒子 i 保持带个体学习强化因子的标准飞行方式; 否则, 以交叉的方式进行位置更新. 粒子以不同方式飞行有利于粒子动态地调节学习能力, 从而避免算法陷入停滞状态.

2.2 基于外部档案领导粒子的更新策略

2.2.1 全局最优粒子的选择

全局最优粒子对于种群的引导至关重要, 其所在位置是种群重点探索的区域. 本文选取外部档案中的粒子作为全局最优领导粒子, 该领导粒子选择策略分为两种情况: 1) 除去极端点外, 以 75% 的概率选择档案中拥挤距离最大的粒子作为全局最优领导粒子; 2) 当外部档案中只存在一个或两个极端粒子时, 随机选择其中一个粒子作为全局最优领导粒子. 偏重选择拥挤距离最大的粒子作为全局最优粒子, 有利于种群对最优解附近稀疏区间进行探索.

2.2.2 个体最优粒子的选择

合理地选择个体最优粒子有助于提高种群对局部空间的开发能力, 便于粒子多区域寻找最优解. 与全局最优粒子从外部档案获得信息来源类似, 个体最优粒子同样根据外部档案解的信息进行选择. 本文采用严格 Pareto 支配(即粒子 i 在每一个目标维度都支配粒子 j) 关系, 设计一种带有粒子学习强化项的个体最优更新策略, 从而加快粒子向前沿飞行. 以二维目标为例, 针对原始个体最优粒子和精英粒子对种群

粒子的引导作简要描述: 一般情况下, 种群粒子依靠个体历史所经过的最优点向前沿飞行, 如图 1 中的虚线箭头所示的折线路径, 需要多次迭代才能搜索到前沿. 当种群粒子的支配区存在前沿解时, 本文选择种群粒子支配区中的前沿粒子作为个体引导粒子, 构建种群粒子直接针对前沿粒子的飞行方式, 从而提高算法的收敛速度.

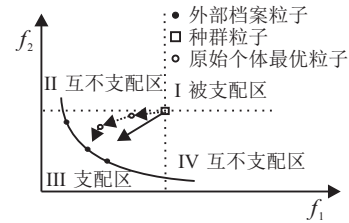


图 1 粒子搜索前沿

个体最优粒子更新原则如下: 1) 当更新个体最优粒子时, 优先选择粒子支配区中的一个外部存档粒子作为其对应的个体最优粒子. 2) 当支配区中无精英粒子时, 比较粒子 i 与个体最优粒子的支配关系. 若受个体最优粒子支配, 则个体最优粒子保持不变; 若其支配个体最优粒子, 则替代个体最优粒子; 若互不支配, 则随机选择其一更新个体最优粒子.

在个体最优粒子更新时, 重点强调 β 的变化. 当种群粒子 i 作为个体最优粒子时, 其个体学习强化因子 β_i 也将更新增大, 这将会加强下一代粒子的自身学习能力, 相对减弱其社会协作性, 有利于种群粒子对自身局部空间的探索, 提高算法的开发能力; 当粒子 i 经过一代位置更新后, β_i 恢复初始值, 以防止粒子因学习的过度强化而脱离其社会协作性.

2.3 精英档案构建机制

2.3.1 外部档案的维护

随着粒子群算法迭代次数的增加, 求得的非支配解也将增多, 甚至会超过外部档案的预设值. 因此, 需要对外部档案进行实时维护, 以引导种群向 PF 搜索, 保障 Pareto 前沿上解的分布.

外部档案粒子的选取可描述为: 1) 将种群粒子和外部档案粒子进行非支配排序, 选择等级最高的粒子进入外部档案; 2) 当粒子数量超过档案规模时, 根据拥挤距离删除拥挤距离最小的一个粒子, 再次进行拥挤距离排序, 重复前一步操作, 直至粒子数量满足档案规模; 3) 当粒子数量未达到外部档案规模时, 将等级最高的粒子全部存储到外部档案中.

2.3.2 带有冗余集的外部档案机制

基于以上外部档案的维护, 在算法初期由于种群对可行域空间的搜索范围有限, 得到的最优解数量不多, 外部档案能够有效保留精英解信息. 但随着迭代次数的递增, 得到的非支配解数量急剧增加, 虽然

拥挤距离删除机制保障了最优解集的规模,但也丢失了部分最优解信息.此外,以全局最优粒子为中心的种群寻优方式易导致粒子群算法出现早熟现象,诱使整个种群陷入局部最优.为了弥补这一缺陷并充分利用 Pareto 前沿粒子的位置信息,本文提出一种外部档案冗余集机制.冗余集中的粒子由外部档案排序后,随机选择 R_{size} 个精英粒子经过变异构成, R_{size} 由 α 控制, $R_{\text{size}} = \alpha A_{\text{size}}$, 其中 A_{size} 为外部档案规模.

可采用 Deb 等^[15] 提出的变异公式,即

$$\text{pop}'_{\text{new}}(d) = \text{pop}'_{\text{new}}(d) + \eta(\max(d) - \min(d)). \quad (12)$$

$$\eta = \begin{cases} (2r_2)^{\frac{1}{\mu_2+1}} - 1, & r_2 < 0.5; \\ 1 - [2(1 - r_2)]^{\frac{1}{\mu_2+1}}, & \text{else.} \end{cases} \quad (13)$$

$$V = e^{-\left(\frac{D}{d=1} \left(\frac{1}{D} \sum_{i=1}^N v_i\right)\right)} + 0.5 \quad (14)$$

其中: r_2 为 $(0, 1)$ 上产生的随机数, μ_2 为变异分布指数, η 为变异因子, pop'_{new} 为在外部档案排序后随机选择的粒子, $\max(d)$ 、 $\min(d)$ 分别为粒子第 d 维的上界和下界.

冗余集有两个作用: 1) 对种群形成干扰, 避免种群陷入局部最优; 2) 通过变异产生的粒子重新进行档案维护, 增加精英解集的多样性.

上述冗余机制流程可表述如下.

Step 1: 如果外部档案排序后粒子规模 A_n 小于 R_{size} , 则将 A_n 个粒子全部复制到冗余集中; 否则, 从 A_n 中随机选取 R_{size} 个粒子复制到冗余集中.

Step 2: 根据式 (12)、(13) 对冗余集中的粒子进行变异操作.

Step 3: 根据式 (14) 判断种群扰动条件, 若种群速度 V 小于速度限制 V_{limit} , 则随机选择不大于 R_n (R_n 为迁入种群数量阈值) 个粒子替换种群适应度最差的 R_n 个粒子, 否则转 Step 4.

Step 4: 根据第 2.3.1 节将冗余集粒子纳入到外部存档中.

Step 5: 清空冗余集, 结束.

2.4 MIMOPSO 算法流程

MIMOPSO 算法具体流程如下.

Step 1: 设定 MIMOPSO 相关参数, 初始化种群及速度, 更新外部存档, 更新个体最优和全局最优.

Step 2: 根据式 (8) 对种群进行更新, 对粒子位置和速度进行边界限定, 同时计算种群目标向量, 然后复原上代的种群个体学习强化因子 β .

Step 3: 更新外部档案, 并依据策略 2.3.2 构建冗余集机制, 对种群进行扰动和外部档案的重新维护.

Step 4: 依据 2.2.1 节和 2.2.2 节的领导粒子更新策略更新全局最优粒子和个体最优粒子, 并更新下代的种群个体学习强化因子 β .

Step 5: 判断是否达到最大迭代次数, 若是, 则输出外部存档粒子, 否则转 Step 2 继续循环.

MZMOPSO 算法流程如图 2 所示.

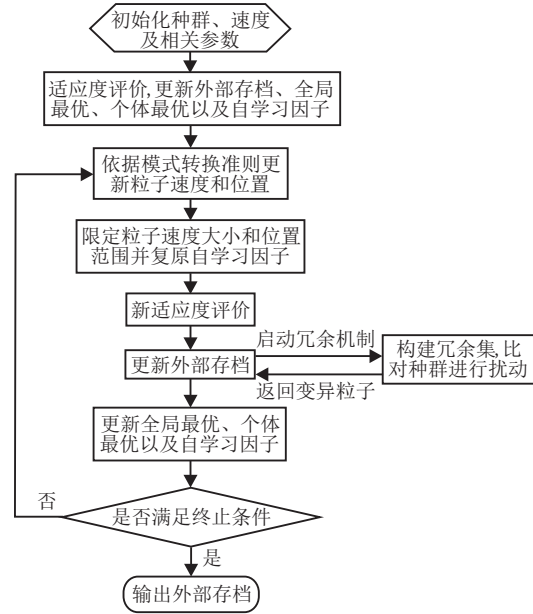


图 2 MIMOPSO 算法

3 性能测试与实验结果分析

3.1 测试函数及性能指标

本文选择经典的多目标优化测试函数集 ZDT^[16] 和 DTLZ^[17] 进行计算比较, 并采用综合指标 Inverted Generational Distance (IGD)^[18] 对所提出的算法进行评价. IGD 是度量真实 Pareto 前端与算法获得的近似 Pareto 前端之间的距离指标. IGD 值越低, 表明算法获得的近似 Pareto 前端的收敛性和多样性越好, 越接近真实 Pareto 前端, 其计算公式如下:

$$\text{IGD}(P, P^*) = \frac{\sum_{i=1}^{|P|} d(P_i, P^*)}{|P|}. \quad (15)$$

其中: P 为理想 PF 上的一组均匀采样, P^* 为通过多目标算法求得 Pareto 解集, $|P|$ 为种群 P^* 的规模, $d(P_i, P^*)$ 为 P_i 与种群 P^* 之间的最小欧氏距离.

3.2 仿真实验结果与分析

为验证本文所提出算法的有效性, 设置 3 种应用于多目标优化问题的对比算法, 其中包括 MOPSO 算法^[2]、自适应进化多目标粒子群优化 (AEP SO) 算法^[19] 和非支配排序算法 II (NSGA II)^[15]. MIMOPSO 算法各参数设置为: 惯性系数 $w = 0.117$, 社会学习因子 $c_1 = 1.193$, 个体学习因子 $c_2 = 1.193$, 初始各个体学习增强因子 $\beta_i = 1$, 增强后的 $\beta'_i = 2$, 模式转换因子

MR = 0.95,冗余集控制系数 $\alpha = 0.5$,交叉、变异分布指数分别为 $\mu_1 = 2$ 、 $\mu_2 = 5$,粒子迁入规模 $R_n = 5$,通过实验观察可设定 $V_{limit} = 0.387$.对于两目标优化问题(ZDT系列函数),所有算法种群规模均设为 $N = 100$,外部档案规模 $A_{size} = 100$,最高评价次数FEAS = 30 000次;对于三目标优化问题(DTLZ系列函数),所有算法的种群规模均设为 $N = 150$,外部档案规模 $A_{size} = 150$,最高评价次数FEAS = 100 000次.4种算法分别独立运行30次,IGD性能指标的平均值(Mean)和标准差(Std)如表1所示,其中黑体加粗字体为各算法在同一测试问题上获得的最优值.

表1 IGD指标统计

性能函数	算法				
	MIMOPSO	AEPSO	MOPSO	NSGA II	
ZDT1	Mean	3.925e-03	4.761e-03	6.918e-03	4.887e-03
	Std	(6.700e-05)	(2.964e-04)	(3.837e-04)	(1.875e-04)
	p-value	-	1.604e-15	8.815e-28	3.271e-23
ZDT2	Mean	4.006e-03	1.669e-01	6.904e-03	4.977e-03
	Std	(2.696e-05)	(2.650e-01)	(5.034e-04)	(3.643e-04)
	p-value	-	2.160e-03	6.222e-24	1.163e-14
ZDT3	Mean	4.430e-03	1.017e-02	7.263e-03	8.182e-03
	Std	(1.371e-04)	(4.895e-03)	(5.987e-04)	(9.202e-03)
	p-value	-	4.733e-07	1.930e-21	3.325e-02
ZDT4	Mean	4.194e-03	3.927e+00	2.715e+00	9.596e-03
	Std	(1.353e-04)	(2.270e+00)	(1.777e+00)	(2.438e-03)
	p-value	-	2.262e-10	3.281e-09	6.218e-13
ZDT6	Mean	2.448e-03	3.399e-03	4.069e-03	2.773e-03
	Std	(1.140e-04)	(1.801e-03)	(7.640e-04)	(2.776e-04)
	p-value	-	7.016e-03	5.502e-12	1.215e-06
DTLZ1	Mean	5.083e-02	1.965e+00	9.008e+00	2.253e-02
	Std	(3.165e-02)	(7.466e-01)	(3.597e+00)	(1.164e-03)
	p-value	-	1.901e-14	3.799e-14	3.132e-05
DTLZ2	Mean	5.497e-02	5.322e-02	7.316e-02	6.320e-02
	Std	(1.870e-03)	(1.366e-03)	(2.900e-03)	(2.560e-03)
	p-value	-	3.244e-04	6.711e-24	6.248e-15
DTLZ4	Mean	5.307e-02	1.791e-01	1.480e-01	9.202e-02
	Std	(1.091e-03)	(3.643e-02)	(2.154e-02)	(1.585e-01)
	p-value	-	7.949e-18	4.591e-21	1.888e-01
DTLZ7	Mean	6.403e-02	7.659e-02	8.446e-02	1.502e-01
	Std	(3.253e-03)	(3.852e-03)	(4.998e-03)	(1.525e-01)
	p-value	-	2.069e-13	2.954e-17	4.451e-03
	+ / = / -	-	8/0/1	9/0/0	7/1/1

表1中还记录了本文算法与其他对比算法在同一测试问题上的显著水平为5% t -检验的 p -value值;“+”、“=”、“-”表示本文算法在IGD值 t -检验中分别优于、类似于、劣于各对比算法在同一测试问题中的显著性区分结果.

3.3 对两目标优化问题求解性能比较

从表1中可以看出,MIMOPSO算法在ZDT系列函数的最终求解结果中获得了较优的结果,并且相较于对比算法,其在各函数的优化中均具有显著性优

势.图3绘制了各算法对部分ZDT测试函数在3 000次、12 000次评价后得到的PF情况,分别对应于图3中的ZDT2-1、ZDT3-1和ZDT2-2、ZDT3-2.

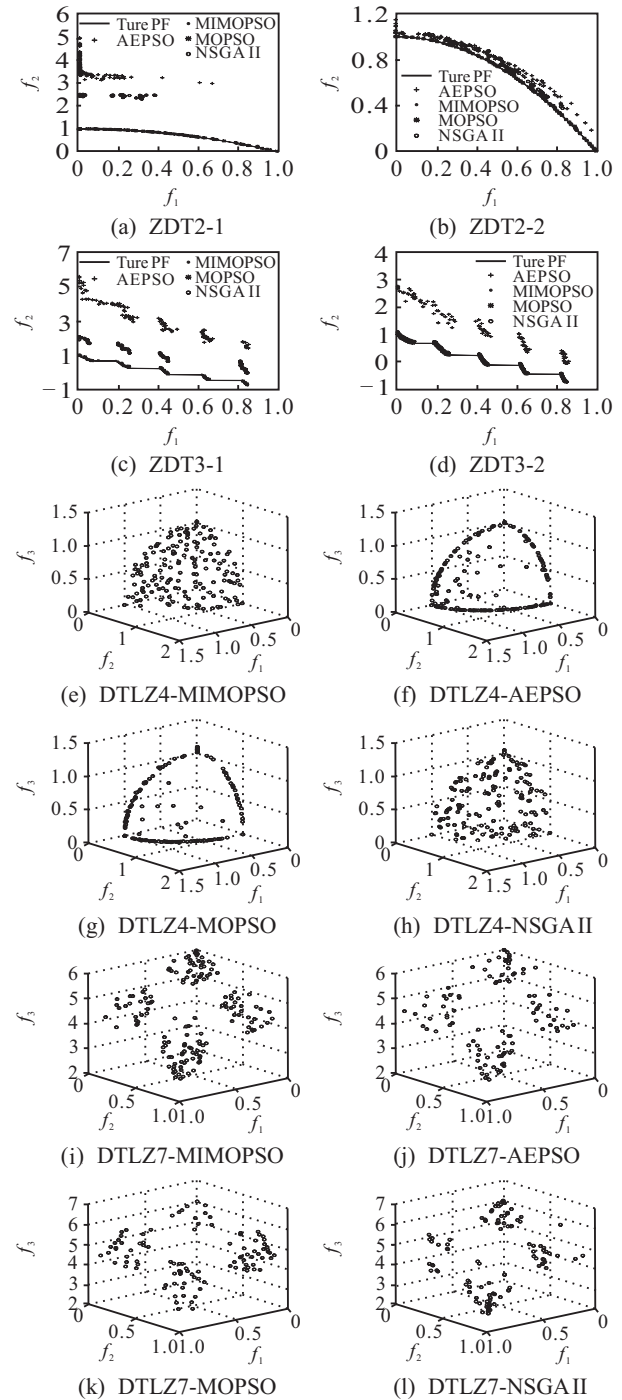


图3 各算法部分测试函数评价后的PF

从图3可知:MIMOPSO算法和MOPSO算法能在ZDT2、ZDT3函数的3 000次评价中较快收敛到理想PF,而AEPSO算法和NSGA II算法尚未收敛,说明MIMOPSO算法的精英粒子领导策略能快速引导粒子向Pareto前沿搜索,MOPSO算法通过对粒子飞行区间的变异方式获得了同样效果.虽然MIMOPSO算法在ZDT2函数的3 000次评价时只寻优到了部分Pareto最优解,多样性较MOPSO算法差,但在12 000

次评价后已能覆盖整个理想PF,说明MIMOPSO算法能够通过冗余机制很好地跳出局部最优,保证解的多样性.尤其在多峰函数ZDT4的优化中,MIMOPSO算法相较于AEPSo算法、MOPSO算法和NSGA II算法获得了最优的IGD值,反映出其很好的收敛性及在理想PF上良好的分布性,说明MIMOPSO算法有较快的收敛速度和较强的探索开发能力.

3.4 三目标优化问题求解性能比较

对于三目标问题,图3给出了DTLZ4和DTLZ7函数在100000次评价后的最优PF.结合表1的计算数值和图3可知,MIMOPSO算法在DTLZ4和DTLZ7函数的求解收敛精度和均匀性上明显优于AEPSo算法和MOPSO算法.除了DTLZ4函数的IGD值与NSGA II算法无差异外,MIMOPSO算法在DTLZ7函

数的优化上显著优于NSGA II算法;对于DTLZ1函数的优化,MIMOPSO算法略逊于NSGA II算法,但优于AEPSo算法和MOPSO算法,显著性差异明显;对于DTLZ2函数,同样可以从表1看出,MIMOPSO算法与对比算法显著性差异明显,AEPSo算法取得了最好的优化效果,MIMOPSO算法次之,MOPSO算法和NSGA II算法优化效果较差.由此可知,MIMOPSO算法对于三目标的DTLZ函数也具有良好的优化性能.

3.5 算法的收敛性比较

图4列出了4种算法的IGD指标下降趋势图.图4中记录的数据为30次独立实验得到的平均值,其中ZDT系列函数为每600次评价记录一次IGD指标,DTLZ系列函数为每2000次评价记录一次IGD指标.

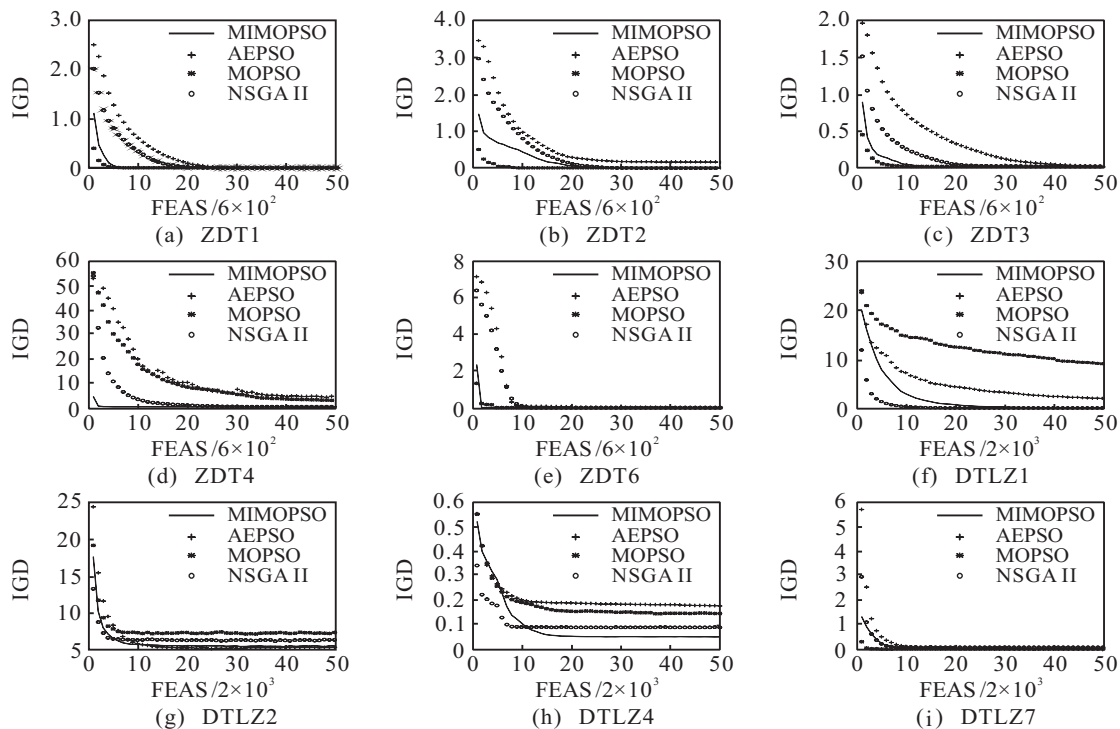


图4 各算法IGD指标趋势

从图4中能看到MIMOPSO算法的IGD曲线在ZDT4和ZDT6函数中下降最快,而在其他函数中处在大多数算法之下,位于次最优位置.但就表1的最终收敛精度而言,MIMOPSO算法在9个测试问题中获得7个函数的最优值,在显著性上,MIMOPSO算法胜出AEPSo算法8个测试函数,劣于AEPSo算法1个测试函数,全部胜于MOPSO算法,胜于NSGA II算法7个测试函数,相平于该算法1个测试函数,劣于该算法1个测试函数.由此说明MIMOPSO算法在平衡收敛速度和求解多样性的同时获得了良好的性能.

3.6 各策略性能对比实验分析

为了验证所提策略对算法的提升作用,本节设计4种自身对比实验.MIMOPSO-I为采用标准粒子更

新方式的算法;MIMOPSO-II算法为采用一般领导粒子更新策略,即个体最优粒子由种群粒子和个体最优粒子的支配原则更替,全局最优粒子随机从外部档案中选取获得;MIMOPSO-III算法为采用NSGA II算法中的非支配排序外部存档更新策略;MIMOSPO-IV为删除冗余机制的算法.上述算法除了与MIMOSPO算法的某项策略机制不同外,其余各参数及实验设置与3.2节均相同,表2给出了实验统计结果.

从表2中可知,MIMOPSO算法在9个测试问题中取得了6个函数的最优值.在显著性上,MIMOPSO算法胜于MIMOPSO-I算法两个测试函数,持平于该算法7个测试函数,劣于该算法0个测试函数,并且除了DTLZ2的标准差外都要小于其他测试函数的标

准差,说明种群更新的异构模式对于不同测试问题,其优化性能均稳定,提高了种群在搜索空间的寻优能力,优于种群标准的单一更新方式;与MIMOPSO-II算法对比,MIMOPSO算法胜出5个测试函数,持平于该算法两个测试函数,劣于该算法两个测试函数,说明精英领导粒子的更新策略能够有效引导种群快速寻找Pareto最优解;相较于MIMOPSO-III算

法,MIMOPSO算法胜出7个测试函数,持平于该算法两个测试函数,劣于该算法0个测试函数,说明改进的外部存档机制在维护精英解集和多样性方面起到了积极作用;对比于MIMOPSO-IV算法,MIMOPSO算法胜出7个测试函数,持平于该算法两个测试函数,劣于该算法0个测试函数,说明冗余机制能够迫使种群跳出局部最优,防止算法早熟现象的发生。

表2 各策略IGD指标统计

性能函数		算法				
		MIMOPSO	MIMOPSO-I	MIMOPSO-II	MIMOPSO-III	MIMOPSO-IV
ZDT1	Mean	3.925e-03	3.975e-03	4.928e-02	6.123e-03	2.424e-02
	Std	(6.700e-05)	(9.543e-05)	(2.307e-02)	(1.002e-03)	(5.278e-02)
	p-value	-	1.706e-02	1.197e-11	3.859e-13	4.381e-02
ZDT2	Mean	4.006e-03	4.011e-03	1.351e-01	6.214e-03	9.650e-01
	Std	(2.696e-05)	(5.292e-05)	(2.174e-01)	(1.095e-03)	(2.918e-01)
	p-value	-	6.343e-01	2.547e-03	6.198e-12	2.648e-17
ZDT3	Mean	4.430e-03	4.480e-03	5.892e-02	5.588e-03	1.846e-02
	Std	(1.371e-04)	(1.530e-04)	(2.528e-02)	(2.267e-04)	(3.663e-02)
	p-value	-	1.690e-01	1.316e-12	4.387e-22	4.473e-02
ZDT4	Mean	4.194e-03	7.470e-02	3.830e-03	5.981e-03	6.963e+00
	Std	(1.353e-04)	(2.623e-01)	(7.486e-05)	(7.212e-04)	(3.887e+00)
	p-value	-	1.517e-01	2.094e-13	1.893e-14	1.028e-10
ZDT6	Mean	2.448e-03	2.479e-03	2.525e-03	3.967e-03	2.456e-03
	Std	(1.140e-04)	(1.169e-04)	(1.576e-04)	(1.463e-03)	(1.185e-04)
	p-value	-	2.421e-01	3.648e-02	2.277e-06	8.033e-01
DTLZ1	Mean	5.083e-02	6.132e-02	9.125e-02	6.945e-02	1.417e+01
	Std	(3.165e-02)	(4.679e-02)	(1.510e-01)	(3.572e-02)	(2.946e+00)
	p-value	-	2.634e-01	1.716e-01	4.537e-02	9.024e-22
DTLZ2	Mean	5.497e-02	5.472e-02	5.346e-02	5.570e-02	6.246e-02
	Std	(1.870e-03)	(1.759e-03)	(3.328e-03)	(2.241e-03)	(2.887e-03)
	p-value	-	6.043e-01	1.362e-02	1.377e-01	1.489e-13
DTLZ4	Mean	5.307e-02	5.371e-02	8.011e-02	5.482e-02	7.379e-02
	Std	(1.091e-03)	(1.174e-03)	(1.034e-01)	(1.997e-03)	(9.164e-02)
	p-value	-	4.330e-02	1.637e-01	6.544e-05	2.257e-01
DTLZ7	Mean	6.403e-02	6.396e-02	6.619e-02	6.490e-02	1.272e+00
	Std	(3.253e-03)	(4.003e-03)	(4.190e-03)	(6.641e-03)	(5.049e-01)
	p-value	-	9.304e-01	4.907e-02	4.215e-01	1.032e-13
+ / = / -		-	2/7/0	5/2/2	7/2/0	7/2/0

上述结果表明,多策略改进粒子群算法显著性地提高了粒子群算法的寻优性能。

3.7 时间复杂度分析

本文的仿真实验是对相同的测试函数进行优化,因此为了计算简便,各算法以种群规模和外部存档大小作为时间消耗标准。其中,MIMOPSO算法、AEPSON算法和NSGA II算法都采用了非支配排序准则维护外部存档,基于NSGA II算法的时间复杂度为 $O(N^2)$,而AEPSON算法在此基础上引入了种群自适应变异策略,这部分的时间复杂度为 $O(N)$ 。此外,领导粒子更新时间复杂度为 $O(2N)$,考虑在最差情况下,AEPSON算法的时间复杂度为 $O(N^2) + O(3N)$,即AEPSON算法的时间复杂度也为 $O(N^2)$;与AEPSON算法相似,MIMOPSO算法虽然在多个策略上对粒子群算法进行了改进,但主要是粒子选择策略的转换,并没有影响算法时间的消耗,即使自学习因子 β 的引入也与个体最优粒子更新在同一循环体

中,并没有增加额外的时间花费,时间复杂度仍为 $O(N)$;而全局最优粒子可直接从外部档案中选择,反而减少了时间计算;最大的时间消耗来自于外部档案的维护,其最坏情况下的时间复杂度约小于 $O(N^2) + O((3/2)N^2 \times \log(2N))$,因为 N 为有限较小正整数,故 $\log(2N) \ll N^2$,即档案更新时间复杂度近似为 $O(N^2)$ 。由于冗余集策略的引入对外部档案更新产生影响,所需时间计算量约为 $O((1 + \alpha)^2 N^2) + O(((2 + \alpha)\alpha/2)N^2 \log((1 + \alpha)N)) + O(\alpha N)$,构建冗余机制的时间复杂度取决于 α 的大小,由于本文中 α 为(0.5, 1)之间的常数,其时间复杂度也可记为 $O(N^2)$ 。因此,MIMOPSO算法在严格的外部存档更新(2.4节 Step 1和 Step 3)下,算法初期外部档案的规模小于 N ,随着算法的迭代,档案中粒子数目逐渐被限定到 N ,由此可得改进算法的时间消耗小于最大耗时,即为 $O(3N^2) + O(2N) \approx O(N^2)$ 。MOPSON算法虽然未采用非支配方式对外部档案进行维护,但对自适

应变策略的应用提升了算法的时间消耗,从而其最大的时间计算量为 $O(N^2)$ 。

可以看出,虽然MIMOPSO算法的时间复杂度有所提升,但仍然与AEPPO算法和NSGA II算法处于同一数量级,且大于等于MOPSO算法的时间复杂度。

4 结论

本文提出了一种多策略改进的粒子群算法,将遗传算法的进化方式融入粒子群算法中对多目标空间进行探索,使其在保留了粒子群算法快速收敛性的同时,又具有进化算法对全局空间的探索能力;合理地选择领导粒子及适时增强个体学习能力能够有效引导粒子向Pareto前沿飞行;建立外部档案的冗余机制一方面有助于充分发掘精英解的信息,产生更有可能接近理想前沿的Pareto解,另一方面冗余集产生的种群扰动可提高解的多样性并促使种群跳出局部最优状态。以上改进策略可有效兼顾粒子群算法的探索和开采过程,不会因算法的过度探索而降低收敛速度,也不会因算法的快速收敛而易出现早熟现象。与其他算法的仿真实验对比分析验证了MIMOPSO算法在大多数测试问题的收敛速度、收敛精度以及PF分布性上都有明显提高。因此,MIMOPSO算法是一种整体性能更优、鲁棒性更强的多目标优化方法。

尽管MIMOPSO算法整体性能有所提升,但多个策略的改进引入了多个参数设定。通过实验或人为方式选取各参数的值并不能很好地适应算法针对不同问题的优化。因此,如何探究MIMOSPO算法各参数对不同问题的自适应调节,使各策略有效地协同配合,发挥算法的最大快速寻优性能,值得进一步研究。

参考文献(References)

- [1] Kennedy J. Particle swarm optimization[M]. Encyclopedia of Machine Learning. Berlin: Springer, 2010: 760-766.
- [2] Coello C A C, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 256-279.
- [3] Nepomuceno F V, Engelbrecht A P. A self-adaptive heterogeneous pso for real-parameter optimization[C]. 2013 IEEE Congress on Evolutionary Computation(CEC). Cancun: IEEE, 2013: 361-368.
- [4] Elloumia W, Abraham N B A, Alimia A M. The multi-objective hybridization of particle swarm optimization and fuzzy ant colony optimization[J]. J of Intelligent and Fuzzy Systems, 2014, 27(1): 515-525.
- [5] Wang Y, Yang Y. Particle swarm with equilibrium strategy of selection for multi-objective optimization[J]. European J of Operational Research, 2010, 200(1): 187-197.
- [6] Daneshyari M, Yen G G. Constrained multiple-swarm particle swarm optimization within a cultural framework[J]. IEEE Trans on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2012, 42(2): 475-490.
- [7] Zhan Z H, Li J, Cao J, et al. Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems[J]. IEEE Trans on Cybernetics, 2013, 43(2): 445-463.
- [8] Ali H, Khan F A. Attributed multi-objective comprehensive learning particle swarm optimization for optimal security of networks[J]. Applied Soft Computing, 2013, 13(9): 3903-3921.
- [9] Gong M, Cai Q, Chen X, et al. Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition[J]. IEEE Trans on Evolutionary Computation, 2014, 18(1): 82-97.
- [10] Zeng G Q, Chen J, Li L M, et al. An improved multi-objective population-based extremal optimization algorithm with polynomial mutation[J]. Information Sciences, 2016, 330: 49-73.
- [11] 施展, 陈庆伟. 基于QPSO和拥挤距离排序的多目标量子粒子群优化算法[J]. 控制与决策, 2011, 26(4): 540-547.
(Shi Z, Chen Q W. Multi-objective quantum-behaved particle swarm optimization algorithm based on QPSO and crowding distance sorting[J]. Control and Decision, 2011, 26(4): 540-547.)
- [12] 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.
(Hu W, Yen G G, Zhang X. Multi-objective particle swarm optimization based on Pareto entropy[J]. J of Software, 2014, 25(5): 1025-1050.)
- [13] Lin Q, Li J, Du Z, et al. A novel multi-objective particle swarm optimization with multiple search strategies[J]. European J of Operational Research, 2015, 247(3): 732-744.
- [14] Shi Y H, Eberhart R. A modified particle swarm optimizer[C]. IEEE World Congress on Computational Intelligence. Anchorage: IEEE, 1998: 69-73.
- [15] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Trans on Evolutionary Computation, 2002, 6(2): 182-197.
- [16] Zitzler E, Deb K, Thiele L. Comparison of multi-objective evolutionary algorithms: Empirical results[J]. Evolutionary Computation, 2000, 8(2): 173-195.
- [17] Deb K, Thiele L, Laumanns M, et al. Scalable multi-objective optimization test problems[C]. Proc of the Congress on Evolutionary Computation(CEC-2002). Honolulu: IEEE, 2002: 825-830.
- [18] Neri F, Cotta C. Memetic algorithms and memetic computing optimization: A literature review[J]. Swarm and Evolutionary Computation, 2012, 2: 1-14.
- [19] Chen M Y, Zhang C Y, Luo C Y. Adaptive evolutionary multi-objective particle swarm optimization algorithm[J]. Control and Decision, 2009, 24(12): 1851-1855.