

文章编号: 1001-0920(2012)06-0813-06

基于局部搜索与混合多样性策略的多目标粒子群算法

贾树晋¹, 杜斌^{1,2}, 岳恒³

(1. 上海交通大学 a. 自动化系, b. 系统控制与信息处理教育部重点实验室, 上海 200240;
2. 宝钢研究院 自动化研究所, 上海 201900; 3. 东北大学 自动化研究中心, 沈阳 110819)

摘要: 为了提高算法的收敛性与非支配解集的多样性, 提出一种基于局部搜索与混合多样性策略的多目标粒子群算法(LH-MOPSO). 该算法使用增广Lagrange乘法对非支配解进行局部搜索以快速接近Pareto最优解; 利用基于改进的Maximin适应值函数与拥挤距离的混合多样性策略对非支配解集进行维护以保留解的多样性, 同时引入高斯变异算子以避免算法早熟收敛; 最后针对多目标约束优化问题, 给出一种有效的约束处理方法. 实验研究表明该算法具有良好的优化性能.

关键词: 多目标优化; 粒子群算法; 增广Lagrange乘法; Maximin适应值函数; 拥挤距离
中图分类号: TP18 **文献标识码:** A

Local search and hybrid diversity strategy based multi-objective particle swarm optimization algorithm

JIA Shu-jin¹, DU Bin^{1,2}, YUE Heng³

(1a. Department of Automation, 1b. Key Laboratory of System Control and Information Processing of Ministry of Education, Shanghai Jiaotong University, Shanghai 200240, China; 2. Research Institute of Automation, Academy of BaoSteel, Shanghai 201900, China; 3. Research Center of Automation, Northeastern University, Shenyang 110819, China. Correspondent: JIA Shu-jin, E-mail: jiashujin_1@163.com)

Abstract: In order to improve the convergence and diversity performance, a local search and hybrid diversity strategy based multi-objective particle swarm optimization algorithm(LH-MOPSO) is proposed. LH-MOPSO makes full use of the augmented Lagrange multiplier method to approach the Pareto optimal solutions quickly, and the hybrid diversity strategy based on modified Maximin fitness function and crowding distance is used for maintaining the diversity of nondominated solutions. Meanwhile, Gaussian mutation operator is introduced to avoid LH-MOPSO premature convergence. Finally, an efficient constraint handling method is proposed. Simulation results show that LH-MOPSO has good performance.

Key words: multi-objective optimization; particle swarm optimization; augmented Lagrange multiplier method; Maximin fitness function; crowding distance

1 引言

实际工程与决策中的大量问题可归结为多目标优化问题. 多目标优化中不存在惟一的最优解, 而是存在多个相互之间无法比较优劣的非支配解^[1]. 多目标粒子群算法是解决多目标优化问题的有效方法, 已有不少的研究成果问世^[2], 其共同目标是: 1) 如何保持非支配解集的多样性, 即多样性策略问题; 2) 如何快速收敛到Pareto最优解, 即收敛性问题.

关于多样性策略的研究成果较多, 如Maximin策略^[3-4], 拥挤距离策略^[5-6], 自适应网格策略^[7], 小生境

策略^[8]等. 其中: 自适应网格策略计算复杂度低, 但划分的网格数对多样性保留效果有影响; 小生境策略的参数难以确定; Maximin策略能够自动“奖励”分散的解, “惩罚”聚集的解, 不需额外的多样性评价技术, 且根据适应值的正负性便可确定非支配解; 拥挤距离策略不需要额外参数, 能较全面地反映解的密度信息和拥挤程度. 研究表明, Maximin策略和拥挤距离策略是两种较好的多样性策略, 得到了广泛的研究.

关于收敛性问题, 目前的算法大都借助于粒子群本身的进化特性并辅之以变异算子^[6-7]. 粒子群算法

收稿日期: 2010-12-03; 修回日期: 2011-03-02.

基金项目: 国家重点基础研究发展计划项目(2009CB320604); 教育部科学技术研究重大项目(308007).

作者简介: 贾树晋(1982-), 男, 博士生, 从事智能优化算法、生产计划与调度的研究; 杜斌(1957-), 男, 教授级高工, 博士生导师, 从事过程控制、生产计划优化等研究.

易早熟收敛; 变异算子虽利于算法跳出收敛态, 增加解的多样性, 但其随机性使得搜索精度有限, 甚至会引起退化现象. 可见, 传统方法不足以有效解决收敛性问题.

为了提高算法收敛性与非支配解集的多样性, 本文在引入变异算子的同时, 提出一种新颖的局部搜索策略, 并结合 Maximin 策略与拥挤距离策略的优点, 提出了一种以 Maximin 为主、拥挤距离为辅的混合多样性策略. 最后给出一种有效的约束处理方法. 仿真实验表明了该算法的有效性.

2 粒子群算法

Kennedy 等人^[9]于 1995 年提出了粒子群算法; 随后, 文献 [10] 提出了经典的线性递减惯性权重粒子群算法, 以平衡算法的全局探测与局部开发能力. 在每次迭代中, 第 i 个粒子的速度和位置按下式更新^[10]:

$$\begin{aligned} v_i(k+1) &= \omega v_i(k) + c_1 r_1 [p_i(k) - x_i(k)] + \\ &\quad c_2 r_2 [p_g(k) - x_i(k)], \\ x_i(k+1) &= x_i(k) + v_i(k+1). \end{aligned} \quad (1)$$

其中: k 为迭代次数, ω 为惯性权重, c_1 和 c_2 为加速因子, r_1 和 r_2 为 $[0, 1]$ 之间服从均匀分布的随机数, p_i 为第 i 个粒子的最优位置, p_g 为全局最优粒子位置.

3 基于局部搜索与混合多样性策略的多目标粒子群算法

3.1 局部搜索策略

局部搜索多见于单目标粒子群算法^[11-12], 而对多目标粒子群算法引入局部搜索机制尚不多见. 本文提出一种新颖的局部搜索策略, 可以指导算法快速收敛到 Pareto 最优解.

3.1.1 局部搜索策略原理

在算法早期, 由于初始解远离 Pareto 最优解, 考虑到算法效率, 不宜引入局部搜索, 而应充分利用粒子群算法的全局搜索能力; 在算法中后期, 粒子群算法无法进一步有效提高解的质量, 这时应引入局部搜索以使算法收敛. 基于上述思想, 本文使用概率函数 P_{LS} 来判断是否进行局部搜索. P_{LS} 的表达式为

$$P_{LS} = \begin{cases} 0, & 0 < k < 0.5K; \\ 2k/K - 1, & 0.5K \leq k \leq K. \end{cases} \quad (2)$$

其中: K 为基于局部搜索与混合多样性的多目标粒子群算法 (LH-MOPSO) 总迭代次数, k 为当前迭代次数.

图 1 为局部搜索策略原理示意图 (以双目标为例). 每次迭代时, 选择非支配解集中每个目标的最优解 (点 A, B 对应的解) 并按概率 P_{LS} 进行局部搜索, 以逐渐逼近该目标的最优解 (f_{1opt} , f_{2opt} 对应的解). 当连续 3 次的搜索结果均无明显变化时, 停止边界解的

搜索, 这样便得到了分布广泛的 Pareto 边界. 与此同时, 每次迭代中随机选择一个非边界解 (C 点对应的解) 并按概率 P_{LS} 进行局部搜索, 通过求解以下优化问题得到一个 Pareto 最优解 (C' 点对应的解):

$$\begin{aligned} \min \quad & \sum_{l=1, \dots, m} \omega_l f_l(x_i). \\ \text{s.t.} \quad & \sum_{l=1, \dots, m} \omega_l = 1; \\ & h_i(x) = 0, \quad i = 1, \dots, p; \\ & g_j(x) \geq 0, \quad j = 1, \dots, q. \end{aligned} \quad (3)$$

ω_l 可由下式确定:

$$\begin{aligned} \omega_l &= \lambda_l / \sum_{r=1, \dots, m} \lambda_r, \\ \lambda_l &= (f_{l, \max} - f_l(x_i)) / (f_{l, \max} - f_{l, \min}). \end{aligned}$$

其中 $f_{l, \max}$ 和 $f_{l, \min}$ 分别对应当前非支配解集中第 l 个目标的最大、最小目标值.

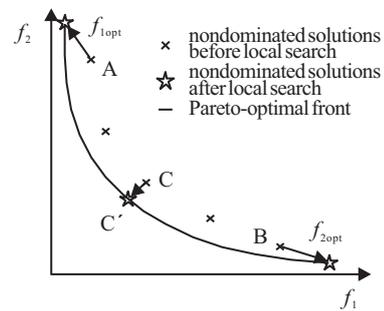


图 1 局部搜索策略原理图

可见, 在进行局部搜索时, 需要考虑原问题的约束条件, 即使原问题是无约束优化问题, 一般也会限定变量的取值范围. 因此, 上述局部搜索策略可归结为求解约束优化问题, 而增广 Lagrange 乘法^[13]是解决约束优化问题的一种有效方法.

3.1.2 增广 Lagrange 乘法

增广 Lagrange 乘法是罚函数法和 Lagrange 乘子法的有机结合, 其基本思想是通过构造增广 Lagrange 函数将约束优化问题转化为无约束优化问题, 然后使用序列无约束极小化方法 (SUMT) 逐渐逼近原约束优化问题的最优解.

对于一般单目标约束优化问题

$$\begin{aligned} \min \quad & f(x). \\ \text{s.t.} \quad & h_i(x) = 0, \quad i = 1, \dots, p; \\ & g_j(x) \geq 0, \quad j = 1, \dots, q. \end{aligned} \quad (4)$$

增广 Lagrange 函数可定义为

$$\begin{aligned} \phi(x, v, w, \sigma) &= \\ & f(x) - \sum_{i=1}^p v_i h_i(x) + \frac{\sigma}{2} \sum_{i=1}^p h_i^2(x) + \end{aligned}$$

$$\frac{1}{2\sigma} \sum_{j=1}^q \{[\max(0, w_j - \sigma g_j(x))]^2 - w_j^2\}. \quad (5)$$

其中: σ 为罚因子; v_i 和 w_j 为 Lagrange 乘子, 其迭代公式为

$$\begin{cases} v_i^{(t+1)} = v_i^{(t)} - \sigma^{(t)} h_i(x^{(t)}), \\ w_j^{(t+1)} = \max(0, w_j^{(t)} - \sigma^{(t)} g_j(x^{(t)})). \end{cases} \quad (6)$$

算法终止准则为

$$\delta(x^{(t)}) = \left\{ \sum_{j=1}^q [\min(g_j(x^{(t)}), w_j^{(t)}/\sigma^{(t)})]^2 + \sum_{i=1}^p [h_i(x^{(t)})]^2 \right\}^{\frac{1}{2}} < \varepsilon, \quad (7)$$

或 $t > \max DT$. 其中 $\max DT$ 为增广 Lagrange 乘法最大迭代次数, 以防止过多的函数评价.

增广 Lagrange 乘法法的算法步骤如下:

Step 1: 给定初始点 $x^{(0)}$, Lagrange 乘子向量初始估计 $v^{(1)}$ 和 $w^{(1)}$, 初始罚因子 σ , 放大因子 $\alpha > 1$, 参数 $\beta \in (0, 1)$, 允许误差 $\varepsilon > 0$, 置 $t = 1$.

Step 2: 以 $x^{(t-1)}$ 为初点, 解无约束优化问题

$$\min \phi(x^{(t-1)}, v^{(t)}, w^{(t)}, \sigma^{(t)}),$$

得解 $x^{(t)}$.

Step 3: 若 $\delta(x^{(t)}) < \varepsilon$ 或 $t > \max DT$, 则停止计算, 得到近似最优解 $x^{(t)}$; 否则, 转 Step 4.

Step 4: 若 $\delta(x^{(t)})/\delta(x^{(t-1)}) \geq \beta$, 则置罚因子 $\sigma := \alpha\sigma$, 转 Step 5; 否则, 直接转 Step 5.

Step 5: 利用式 (6) 更新 Lagrange 乘子, 置 $t := t + 1$, 转 Step 2.

对于 Step 2 的无约束优化问题, 应视其可导性灵活选择基于导数的优化算法或直接方法^[13]进行求解.

3.2 混合多样性策略

非支配解集的多样性是衡量多目标优化算法性能的重要指标. 本文在比较 Maximin 策略与拥挤距离策略差异的基础上, 结合两者的优点, 提出了一种新颖的混合多样性策略.

3.2.1 混合多样性策略原理

Maximin 策略起源于博弈理论, 由 Balling^[14]首次应用于多目标优化问题. 根据定义, 一个粒子 x_i 的 Maximin 适应度函数为

$$f_{\text{mm}}(x_i) = \max_{j=1, \dots, N} \left\{ \min_{l=1, \dots, m} \{f_l(x_i) - f_l(x_j)\} \right\}. \quad (8)$$

另外, 一种常用的多样性策略为拥挤距离策略. 图 2 给出了这两种策略的比较, 从中可以看出两种策略的差异. 其中: 点 A~E 代表 Pareto 最优前端上的 5 个点; A(1,5) (-0.5) 表示 A 点坐标为 (1,5), Maximin 适应值为 -0.5; 两个矩形周长分别表示 B 与 D 的拥挤

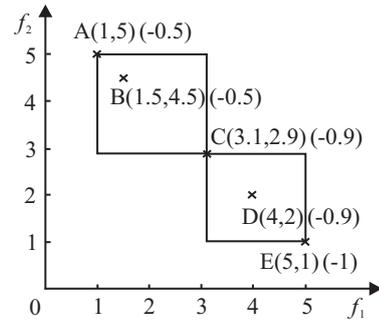


图 2 Maximin 策略与拥挤距离策略比较

距离. 假如非支配解集的上限为 4, 则需从中去除 1 个点. 直观上看, 显然应该去除 B; 根据 Maximin 策略, 应去除 A 或 B; 根据拥挤距离策略, 需去除 D. 由此可见, Maximin 策略与拥挤距离策略的区别是: 拥挤距离仅考虑该点前后两点的距离信息, 而不考虑该点的“偏置”信息, 即该点距哪个点更近; 与拥挤距离相反, Maximin 策略仅考虑“偏置”信息. 这两类策略单独使用都有失偏颇.

鉴于 Maximin 策略更符合直观结果, 计算复杂度更低, 本文提出一种以 Maximin 策略为主、拥挤距离策略为辅的混合多样性策略. 首先, 根据 Maximin 适应值从小到大排序, 若几个解的 Maximin 适应值相同 (如图 2 中 A 和 B 点, C 和 D 点), 则对这些解使用拥挤距离排序. 按照这种方法, 图 2 最终的排序结果为 E-C-D-A-B, 需要去除的点为 B, 与直观结果相同.

混合多样性策略与 Maximin 策略相比, 并未明显提高算法复杂度, 这是由于算法仅需对 Maximin 适应值相等的解进行拥挤距离排序, 而一般非支配解集中 Maximin 适应值相等的概率较小.

3.2.2 Maximin 适应值函数的改进

传统的 Maximin 适应值函数存在以下 3 个问题: 1) 若各目标值不在同一数量级, 则按式 (8) 计算得到的 Maximin 适应值有偏向性, 应先对目标值进行标准化处理^[4]; 2) 被支配解的存在会影响非支配解的 Maximin 适应值^[14], 应先剔除被支配解, 再计算 Maximin 适应值; 3) 为了获得广泛的 Pareto 最优前端, 应保留边界解, 令其 Maximin 适应值最小即可保留边界解. 改进后的 Maximin 适应值函数定义为

$$\begin{cases} f_l(x_i) = \frac{f_l(x_i) - \min_l(f_l(x_i))}{\max_l(f_l(x_i)) - \min_l(f_l(x_i))}, \\ f_{\text{mm}}(x_i) = \max_{\substack{x_i, x_j \in \text{PS} \\ i \neq j}} \left\{ \min_l \{f_l(x_i) - f_l(x_j)\} \right\}, \\ f_{\text{mm}}(\bar{x}) = f_{\text{mm}}(\bar{x}) = \min_{x_i \in \text{PS}} f_{\text{mm}}(x_i). \end{cases} \quad (9)$$

其中: $l = 1, \dots, m$; \bar{x}, \bar{x} 分别表示非支配解集 PS 中两个边界解 (以双目标为例).

3.3 变异算子

粒子群算法具有快速收敛的特点,但在多目标算法中,过快的收敛速度可能导致算法早熟.目前解决这个问题的主要方法是引入变异算子.因均匀变异算子容易引起退化现象,增加算法运行时间,而高斯变异算子不仅能有效避免算法早熟收敛,且具有一定的局部搜索能力,有利于找到更好的非支配解,故本文采用高斯变异算子.在变异操作具体实现中采用一种递减变异策略^[7],在算法初期作用于所有粒子以扩大搜索空间,随后变异的粒子逐渐减少以使算法收敛.

3.4 约束处理

不同于 NSGA-II 基于锦标赛方法进行约束处理^[15],多目标粒子群算法处理约束的关键是排序问题.本文设计的排序规则为:1)对于可行解,排序依次考虑 Maximin 适应值、拥挤距离;2)对于不可行解,按违反约束的总量从小到大排序;3)可行解优先于不可行解.

3.5 算法流程

Step 1: 随机产生 N 个粒子并存储于 P_0 中;将粒子的个体最优位置 p_i 设为当前位置 x_i ;置初始非支配解集 $A_0 = \emptyset$,并给定非支配解集的规模 \bar{A} ;令迭代次数 $k = 0$.

Step 2: 将 P_0 中的非支配解更新到 A_0 中,并根据混合多样性策略排序,形成 A_1 ,令 $P_1 = P_0$.

Step 3: 令 $k := k + 1$,开始循环迭代.

Step 4: P_k 中的粒子 i 按式(1)更新位置并判断是否变异,其中粒子 i 的全局最优解 p_g 在 A_k 前 20% 的非支配解中随机选取.将产生的子代种群 Q_k 与 P_k 合并并存储于 R_k 中,这时 R_k 共包含 $2N$ 个粒子.

Step 5: 根据混合多样性策略对 R_k 进行排序,选择前 N 个粒子组成下一代种群 P_{k+1} ,更新 P_{k+1} 中各粒子的 p_i ,并将 R_k 中的非支配解更新到 A_k 中.

Step 6: 对 A_k 进行局部搜索,删除其中的被支配解,根据混合多样性策略对 A_k 进行排序.若 $|A_k| > \bar{A}$,则取前 \bar{A} 个粒子构成 A_{k+1} ;否则,令 $A_{k+1} = A_k$.

Step 7: 若不满足算法终止条件,则返回 Step 3 继续迭代;否则,结束程序,输出 A_{k+1} 中的非支配解.

4 实验研究

4.1 测试环境设置

本文选用 6 个标准测试函数(ZDT1~ZDT4^[16], Kita^[7], TNK^[15])来比较 LH-MOPSO 与其他 3 种典型的多目标粒子群算法(maximinPSO^[3], NSPSO^[5], MOPSO^[7])的性能.算法采用 C++ 编程实现,对每个测试函数分别进行 20 次独立实验.

多目标优化算法一般通过算法收敛性、非支配

解的多样性及算法复杂度来衡量其性能.本文选取 \mathcal{I} , Δ 指标^[15]和 CPU 时间 Time 分别衡量算法的收敛性、非支配解的多样性及算法复杂度,并通过近似 Pareto 最优前端来直观地比较算法性能.

4.2 算法参数设置

为了公平起见,除特别说明外,4 种算法的参数设置相同.具体设置如下:对于 ZDT1~ZDT4,群体规模 $N = 100$;对于 Kita 和 TNK,群体规模 $N = 30$.非支配解集规模 \bar{A} 均设为 100.由于 ZDT4 具有大量的局部极值点,优化难度较高,将 maximinPSO, NSPSO 和 MOPSO 的迭代次数设为 250;对应其余测试函数的迭代次数设为 100.为了说明 LH-MOPSO 收敛的快速性,将 LH-MOPSO 的迭代次数设为其余 3 种算法的一半.对于 ZDT1~ZDT4,设惯性权重 ω 从 1 线性递减到 0.4,加权因子 $c_1 = c_2 = 2.0$;对于 Kita 和 TNK,设 $\omega = 0.4$, $c_1 = c_2 = 1.0$.变异概率设为 0.5.局部搜索参数设置为: $\sigma = 2$, $v^{(1)} = w^{(1)} = 1$, $\alpha = 2$, $\beta = 0.5$, $\varepsilon = 10^{-6}$,max DT= 50.

下面讨论上述参数设置的合理性并藉此分析各参数对算法性能的影响.对于惯性权重 ω ,加速因子 c_1 和 c_2 ,通过实验说明其对 LH-MOPSO 的影响.设计两组实验:第 1 组未加入局部搜索,第 2 组加入局部搜索.选取常用的 3 组参数进行比较:参数组 1^[3,5]: ω 从 1 线性递减到 0.4, $c_1 = c_2 = 2.0$;参数组 2: $\omega = 0.729$, $c_1 = c_2 = 1.4945$;参数组 3^[7]: $\omega = 0.4$, $c_1 = c_2 = 1.0$,其余参数同前.每种参数设置下运行 20 次,采用 \mathcal{I} 指标评价算法性能(不同参数对 Δ 指标和 Time 的影响较小).

实验结果列于表 1.显然,对于 ZDT1~ZDT4,参数组 1 最佳;对于 Kita 和 TNK,参数组 3 最佳.其余 3 种算法的最佳 ω , c_1 , c_2 设置与 LH-MOPSO 相同.其原因是:ZDT1~ZDT4 是高维测试函数,解空间范围广,粒子群算法应尽可能扩大搜索空间以避免早熟收敛,故应选择参数组 1;Kita 和 TNK 是低维函数,解空间有限,应将重点放在局部搜索方面,故应选择参数组 3.另外,从表 1 中可以看出, ω , c_1 , c_2 对未加入局部搜索的算法性能有重要影响;而加入局部搜索后,性能差距已不甚明显,说明局部搜索的引入使得 LH-MOPSO 对 ω , c_1 , c_2 参数变化不敏感,增加了算法的鲁棒性.

对于群体规模与迭代次数的设置,经实验发现,LH-MOPSO 在其较小时也具备较强的寻优能力,而其余 3 种算法则需要较大的参数设置才能达到较好的性能.本文选取的群体规模和迭代次数较好地兼顾了 maximinPSO, NSPSO 和 MOPSO 的算法性能及算法之间性能差异的可辨识度,同时考虑了测试函数的

表 1 参数组 ω, c_1, c_2 对 LH-MOPSO 算法性能的影响 (γ 指标)

参数组	无局部搜索						有局部搜索					
	ZDT1	ZDT2	ZDT3	ZDT4	Kita	TNK	ZDT1	ZDT2	ZDT3	ZDT4	Kita	TNK
1	0.0024	0.0032	0.0067	2.9643	0.0882	0.0171	0.0021	0.0027	0.0054	0.4811	0.0059	0.0054
2	0.0134	0.0152	0.0078	6.0742	0.4144	0.0149	0.0025	0.0033	0.0059	0.5723	0.0042	0.0048
3	0.1481	0.2859	0.2409	15.033	0.0083	0.0042	0.0034	0.0058	0.0075	1.2935	0.0020	0.0016

复杂性.

对于变异概率的设置, 需要平衡解的质量和收敛速度两方面. 如果变异概率过大, 则导致搜索空间过大, 虽能提高解的质量, 但延长了收敛速度; 反之, 虽能提高收敛速度, 但易陷入局部最优. 在借鉴文献[7]实验结果的基础上, 考虑到设置的变异概率能使算法前半段引入变异^[7], 在算法后半段引入局部搜索, 平衡了算法的“探索”与“开发”能力. 对于局部搜索参数, 则使用增广 Lagrange 乘子法的常用参数设置^[13].

4.3 实验结果与分析

图 3 和图 4 为 4 种算法的近似 Pareto 最优前端与理论 Pareto 最优前端的对比. 表 2 为 4 种算法对应的 γ 指标, Δ 指标及 CPU 时间 Time (均值), 其中最优结果用黑体标出.

从实验结果可得以下结论: 从算法收敛性来看,

得益于局部搜索策略的引入, 尽管迭代次数较少, 但 LH-MOPSO 仍具有最好的收敛性能 (γ 值较小); 且由图 3 和图 4 可知, LH-MOPSO 均收敛到测试函数的 Pareto 最优前端, 说明该算法具有快速收敛性. 从解的多样性来看, LH-MOPSO 除在 Kita 上稍逊于 maximinPSO 外, 在其余 5 个测试函数上均得到了最好结果 (Δ 值较小), 表明混合多样性策略较单纯使用 Maximin 策略或拥挤距离策略更优越. 从算法运行时间来看, LH-MOPSO 的算法复杂度介于 maximinPSO 和 NSPSO 之间, 处于第 3 位 (LH-MOPSO 的迭代次数是其余算法的一半), 表明 LH-MOPSO 在提高算法性能的同时, 并未引起算法复杂度的大幅提升. 另外应看到, MOPSO 的算法复杂度最低, 因随机变异算子的作用, 在迭代次数较少时不易收敛, 故在优化高维测试函数 ZDT1~ZDT4 时性能较差. 如果运行相同的时间, MOPSO 的性能并不弱于 maximinPSO 和 NSPSO,

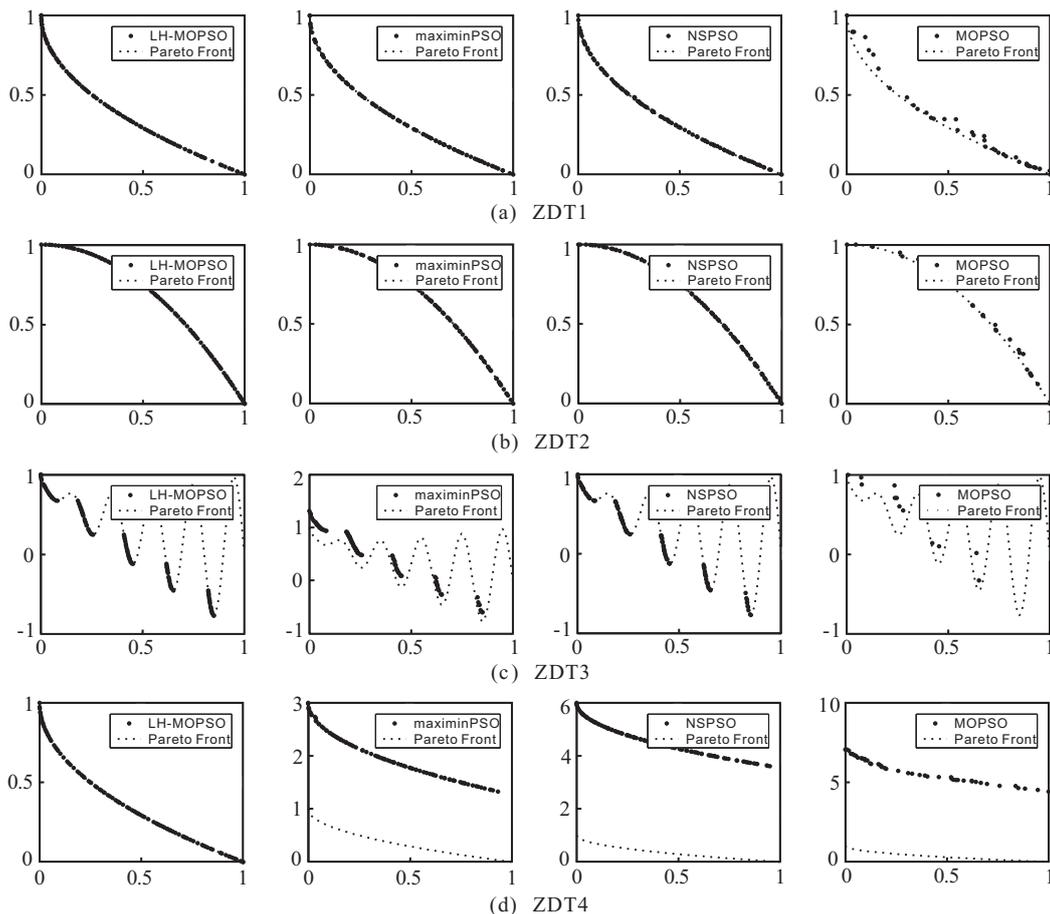


图 3 4 种算法对应 ZDT1, ZDT2, ZDT3, ZDT4 的近似 Pareto 最优前端

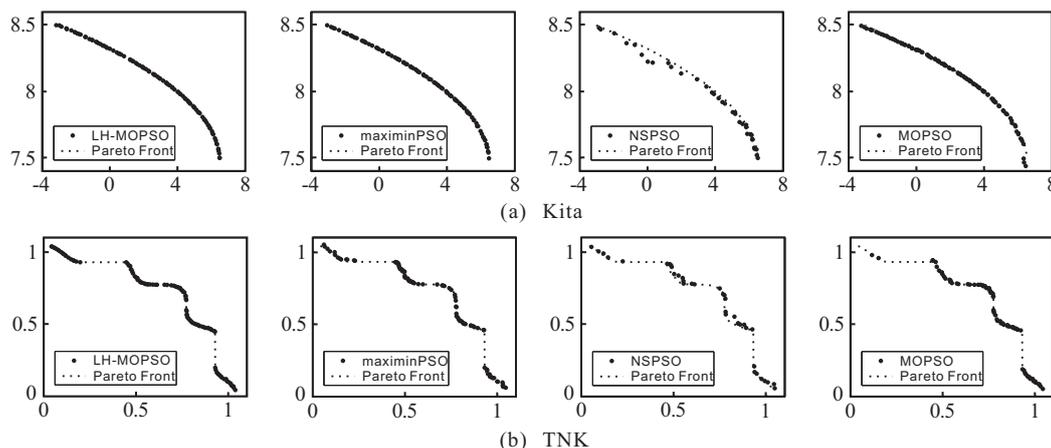


图4 4种算法对应 Kita, TNK 的近似 Pareto 最优前端

表2 LH-MOPSO, maximinPSO, NSPSO 和 MOPSO (分别用 1, 2, 3, 4 表示) 的 γ 指标, Δ 指标和 CPU 时间 Time 对比

测试函数	γ				Δ				Time			
	1	2	3	4	1	2	3	4	1	2	3	4
ZDT1	0.0021	0.0033	0.0078	0.0166	0.4088	0.6038	0.5737	0.6544	2.9505	4.3747	6.7603	2.3283
ZDT2	0.0027	0.0047	0.0099	0.0104	0.3803	0.6433	0.5426	0.7097	2.7267	4.0731	6.3185	2.2187
ZDT3	0.0059	0.0812	0.0064	0.0632	0.5607	0.7155	0.7343	0.8361	2.9454	4.3490	7.1873	2.3283
ZDT4	0.4811	3.1802	5.8578	8.1735	0.4089	0.8284	0.9051	0.8843	9.8379	12.218	20.593	8.9521
Kita	0.0020	0.0081	0.0301	0.0183	0.6177	0.6029	0.7504	0.6227	1.2422	1.7133	2.3445	1.4117
TNK	0.0016	0.0079	0.0126	0.0063	0.6494	0.8295	0.8469	0.7210	1.1692	1.4587	2.0937	0.8957

但与 LH-MOPSO 相比存在一定差距. LH-MOPSO, maximinPSO 和 NSPSO 使用本文提出的约束处理方法均可有效优化 Kita 和 TNK, 说明了该约束处理方法的有效性.

综上所述, 局部搜索的引入能够指导非支配解快速接近 Pareto 最优解, 极大提高了算法的收敛性能和收敛速度, 同时使得 LH-MOPSO 对其他参数 (如 ω, c_1, c_2 等) 变化不敏感, 增加了算法的鲁棒性. 另外, 混合多样性策略在对 Maximin 适应值函数改进的基础上, 充分结合了 Maximin 策略和拥挤距离各自的优点, 能够有效保留分布性能良好的非支配解, 增加了解的多样性, 使其分布更加均匀.

5 结论

本文提出的 LH-MOPSO 算法的主要贡献在于将局部搜索的思想引入多目标粒子群算法, 从而指导算法快速收敛到 Pareto 最优解, 有效地解决了收敛性问题. 其次, 本文对两种常用的多样性策略进行了比较, 并对 Maximin 适应值函数进行了改进, 进而提出了一种以 Maximin 为主、拥挤距离为辅的混合多样性策略, 该策略结合了 Maximin 策略和拥挤距离策略各自的优点, 使多样性策略更趋合理. 另外, 高斯变异算子的引入有效避免了算法早熟收敛. 最后针对多目标约束优化问题, 提出了一种有效的约束处理方法. 实验研究表明, LH-MOPSO 具有良好的收敛性和分布

性能. 下一步的研究重点是引入新的局部搜索策略, 以及 LH-MOPSO 在实际工程与决策中的应用等.

参考文献(References)

- [1] 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.
(Gong M G, Jiao L C, Yang D D, et al. Research on evolutionary multi-objective optimization algorithms[J]. J of Software, 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]. Int J of Computational Intelligence Research, 2006, 2(3): 287-308.
- [3] Li X D. Better spread and convergence: Particle swarm multiobjective optimization using the maximin fitness function[C]. Proc of the Genetic and Evolutionary Computation Conf. Washington, 2004: 117-128.
- [4] 徐鸣, 沈希, 马龙华, 等. 一种多目标粒子群改进算法的研究[J]. 控制与决策, 2009, 24(11): 1713-1718.
(Xu M, Shen X, Ma L H. Research on modified multi-objective particle swarm optimization[J]. Control and Decision, 2009, 24(11): 1713-1718.)
- [5] Li X D. A non-dominated sorting particle swarm optimizer for multiobjective optimization[C]. Proc of the Genetic and Evolutionary Computation Conf. Chicago, 2003: 37-48.

(下转第826页)