

## 基于多层次信息交互的多目标粒子群优化算法

杨宁<sup>a</sup>, 霍炬<sup>b</sup>, 杨明<sup>a</sup>

(哈尔滨工业大学 a. 控制与仿真中心, b. 电气工程系, 哈尔滨 150001)

**摘要:** 为提高多目标优化算法的收敛性和多样性, 提出一种基于多层次信息交互的多目标粒子群优化算法. 在该算法中, 整个优化过程可分为标准粒子群优化层、粒子进化与学习层和档案信息交换层 3 个层次. 粒子进化与学习层保证了每次迭代都能得到更好的粒子位置; 档案信息交换层可以提供更好的全局最优. 优化算法各个层次之间通过信息交互, 共同提高算法的收敛性和多样性. 与 NSGA-II 和 MOPSO 算法的对比分析表明, 所提出算法具有良好的性能, 能够有效解决多目标优化问题.

**关键词:** 多目标优化; 多层次信息交互; 粒子群优化; 收敛性; 多样性

**中图分类号:** TP18

**文献标志码:** A

## Multi-objective particle swarm optimization algorithm based on the interaction of multi-level information

YANG Ning<sup>a</sup>, HUO Ju<sup>b</sup>, YANG Ming<sup>a</sup>

(a. Control and Simulation Center, b. Department of Electrical Engineering, Harbin Institute of Technology, Harbin 150001, China. Correspondent: YANG Ming, E-mail: myang@hit.edu.cn)

**Abstract:** In order to improve the convergence and diversity, a multi-objective particle swarm optimization algorithm based on the interaction of multi-level information is proposed. In this algorithm, the optimization is divided into the standard particle optimization layer, the particle evolution and learning layer and the archive information exchange layer. The particle evolution and learning layer ensures that a better particle position can be acquired in each iteration, while the layer of archive information exchange can provide a better global optimization. With the information interaction between different layers in this algorithm, the convergence and diversity are improved. Comparing this algorithm to the NSGA-II algorithm and the MOPSO algorithm, the results show that the proposed algorithm has better performance and can effectively solve the multi-objective optimization problem.

**Keywords:** multi-objective optimization; multi-level information interaction; particle swarm optimization; convergence; diversity

## 0 引言

在现实应用中, 许多问题都是多目标优化问题, 需要同时对多个目标进行优化. 然而, 需要优化的多个目标之间往往是相互冲突的, 难以优化. 传统的多目标优化方法通过加权将多目标优化问题转换为单目标优化问题, 但这需要待优化问题相应的先验知识, 因此该方法对一些多目标优化问题难以有效处理<sup>[1-2]</sup>. 基于种群的进化算法可以并行地搜索解空间中的多个解, 适合对多目标优化问题进行求解<sup>[3]</sup>. 在过去的数十年中, 许多学者提出了不同的进化算法来解决多目标优化问题, 如 Deb 等<sup>[4-5]</sup>提出了非劣分类遗传算

法(NSGA)和改进的非劣分类遗传算法(NSGA-II), Zitzler 等<sup>[6-7]</sup>提出了强度 Pareto 进化算法(SPEA)和改进的强度 Pareto 进化算法(SPEA2), Knowles 等提出了 Pareto 存档进化策略(PAES)<sup>[8]</sup>等.

粒子群优化算法(PSO)是由 Kennedy 等<sup>[9-10]</sup>于 1995 年提出的一种模拟群体行为的智能优化方法, 其核心思想是粒子之间通过信息共享机制, 相互借鉴经验, 促进整个种群的发展. 粒子群优化算法因其收敛速度快, 易于实现, 已在许多单目标优化问题中得到了应用. 近年来, 许多学者开始利用粒子群优化算法解决多目标优化问题, 其中较有代表性的是 Coello

收稿日期: 2015-01-25; 修回日期: 2015-09-14.

基金项目: 国家自然科学基金项目(61473100).

作者简介: 杨宁(1986—), 男, 博士生, 从事图像处理、视觉测量的研究; 杨明(1963—), 男, 教授, 博士生导师, 从事飞行器制导与控制、复杂系统仿真理论与方法等研究.

等<sup>[1]</sup>提出的多目标粒子群优化算法MOPSO. 然而, 在目前已有的多目标粒子群优化算法中, 并没有充分利用计算过程中得到的相关信息, 在每一步迭代中仅利用了个体最优和全局最优; 同时, 在优化过程中, 仅是粒子之间相互交换信息, 而外部档案中存储的非支配解之间的信息交换并没有得到重视, 这些都影响了优化算法的性能.

为了提高多目标粒子群优化算法的收敛性和多样性, 本文提出一种基于多层次信息交互的多目标粒子群优化算法(MLII-MOPSO). 该算法在粒子群优化的基础上, 将进化操作引入相应的粒子操作中, 充分利用每一步迭代得到的粒子信息; 同时, 在外部档案存储的非支配解间进行信息交换操作, 通过信息交换提供更好的全局最优. 最后通过与NSGA-II和MOPSO算法的对比分析表明, 所提出算法具有良好的性能, 能够有效解决多目标优化问题.

## 1 多目标优化问题

为了更好地理解多目标优化问题, 给出如下定义<sup>[12-13]</sup>.

**定义 1**(多目标优化问题) 在空间  $\Omega$  中寻找向量  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ , 使得目标函数  $F(\mathbf{x})$  中的多个目标  $f_i(\mathbf{x})(i = 1, 2, \dots, m)$  最小, 即

$$\min_{\mathbf{x} \in \Omega} F(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})\}. \quad (1)$$

**定义 2**(Pareto 支配) 向量  $\mathbf{u} = (u_1, u_2, \dots, u_m)$  支配向量  $\mathbf{v} = (v_1, v_2, \dots, v_m)(\mathbf{u} \prec \mathbf{v})$  当且仅当  $\mathbf{u}$  不大于  $\mathbf{v}$ , 且  $\mathbf{u}$  中存在小于  $\mathbf{v}$  的分量, 即

$$\forall i \in \{1, 2, \dots, m\}, u_i \leq v_i \wedge \exists i \in \{1, 2, \dots, m\}, u_i < v_i. \quad (2)$$

**定义 3**(Pareto 最优解) 向量  $\mathbf{x} \in \Omega$  为 Pareto 最优解当且仅当空间  $\Omega$  中不存在  $\mathbf{x}' \in \Omega$  使得向量  $\mathbf{v} = (f_1(\mathbf{x}'), f_2(\mathbf{x}'), \dots, f_m(\mathbf{x}'))$  支配向量  $\mathbf{u} = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))$ .

**定义 4**(Pareto 最优解集) 给定一多目标优化问题, 其优化的目标函数为  $F(\mathbf{x})$ , 则其 Pareto 最优解集  $P$  定义为

$$P = \{\mathbf{x} \in \Omega | \neg \exists \mathbf{x}' \in \Omega, F(\mathbf{x}') \prec F(\mathbf{x})\}. \quad (3)$$

**定义 5**(Pareto 前沿) 给定一多目标优化问题, 其优化的目标函数为  $F(\mathbf{x})$ , Pareto 最优解集为  $P$ , 则其 Pareto 前沿 PF 定义为

$$\text{PF} = \{\mathbf{u} = F(\mathbf{x}) | \mathbf{x} \in P\}. \quad (4)$$

多目标优化问题就是获取一组在目标空间尽量靠近 Pareto 前沿(收敛性好)且均匀分布(多样性好)的解.

## 2 MLII-MOPSO 算法

在 MLII-MOPSO 算法中, 整个优化可分为标准粒子群优化层、粒子进化与学习层和档案信息交换层. 在优化初始阶段, 通过标准粒子群优化算法获得粒子的新位置, 并判断粒子新位置与旧位置的优劣情况. 如果粒子新位置没有优于旧位置, 则进入粒子进化与学习层, 通过粒子进化和粒子对全局最优的学习, 获得更好的粒子新位置. 同时, 在优化过程中获得的非支配解会存储到一个外部存储空间(即外部档案)中, 在每次迭代完成后, 随机选取外部档案中一定数目的非支配解进行信息交换, 从而提高档案中非支配解的质量, 为粒子群寻优以及粒子进化与学习提供更好的全局最优. MLII-MOPSO 算法中各个层次的信息交互关系如图 1 所示.

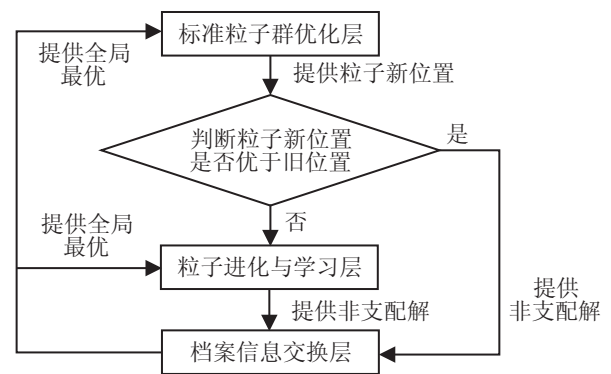


图 1 MLII-MOPSO 算法中各个层次信息交互关系

### 2.1 标准粒子群优化算法

粒子群优化算法通过一群没有体积和重量的粒子在解空间中的运动对优化问题求解. 每个粒子在解空间中的位置代表优化问题的一个候选解, 粒子的速度对应粒子飞行的方向和距离. 在每一次迭代中, 粒子跟随两个极值进行运动, 一个是粒子至今发现的个体最优解  $p_{\text{best}}$ , 另一个是全种群至今发现的全局最优解  $g_{\text{best}}$ .

假设粒子群中存在  $m$  个粒子, 第  $i$  个粒子在  $D$  维搜索空间中的位置为  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ , 飞行速度为  $\mathbf{v}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ . 第  $i$  个粒子迄今发现的个体最优解记为  $\mathbf{p}_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ , 全种群迄今发现的全局最优解记为  $\mathbf{p}_g = (p_{g1}, p_{g2}, \dots, p_{gD})$ . 在每一次飞行中, 粒子根据下式更新自己的速度和位置:

$$\mathbf{v}_{id} = w\mathbf{v}_{id} + c_1r_1(\mathbf{p}_{id} - \mathbf{x}_{id}) + c_2r_2(\mathbf{p}_{gd} - \mathbf{x}_{id}), \quad (5)$$

$$\mathbf{x}_{id} = \mathbf{x}_{id} + \mathbf{v}_{id}. \quad (6)$$

其中:  $w$  为惯性权重因子,  $c_1$ 、 $c_2$  为学习因子,  $r_1$ 、 $r_2$  为  $[0,1]$  之间的随机数.

粒子速度和位置更新公式主要由3部分组成: 1) 粒子先前的速度, 表示粒子的状态; 2) 认知部分, 表示粒子自身经验对其飞行的影响; 3) 社会部分, 反映粒子间的协同合作和知识共享<sup>[14-15]</sup>.

## 2.2 粒子进化和最佳学习

粒子群优化算法通过粒子的飞行在解空间中逐代搜索, 最后得到问题的最优解. 如果在一次迭代中, 粒子的新位置没有优于旧位置, 则这种情况显然不利于算法的收敛. 为了解决这一问题, 将进化操作与最佳学习策略引入粒子群优化算法.

在一次迭代完成之后, 判断粒子的新位置是否优于旧位置, 如果没有, 则进行粒子的进化与最佳学习, 具体步骤如下.

**Step 1:** 将粒子飞行后得到的粒子新位置  $P$  与旧位置  $OP$  进行交叉操作, 得到粒子位置  $P_1$  和  $P_2$ .

**Step 2:** 计算粒子位置  $P$ 、 $P_1$  和  $P_2$  的适应值, 从中选出适应度最高的粒子位置, 记为  $P_3$ .

**Step 3:** 比较粒子位置  $P_3$  与  $OP$ , 如果  $P_3$  优于  $OP$ , 则取  $P_3$  作为粒子的新位置以促进算法的收敛, 并转至 **Step 7**, 否则, 转至 **Step 4** 进行粒子旧位置  $OP$  对全局最优  $g_{best}$  的学习.

**Step 4:** 根据采取的交叉原则, 将粒子旧位置  $OP$  中的相应变量替换为全局最优  $g_{best}$  中的对应变量, 得到粒子位置  $P_4$ .

**Step 5:** 计算粒子位置  $P_3$  与  $P_4$  的适应值, 从中选出适应度最高的粒子位置, 记为  $NP$ .

**Step 6:** 比较粒子位置  $NP$  与  $OP$ , 如果  $NP$  优于  $OP$ , 则取  $NP$  作为粒子的新位置以促进算法的收敛; 如果  $NP$  和  $OP$  的适应值相等或无法比较, 则同样保存  $NP$  作为粒子的新位置以提高算法的多样性; 如果  $NP$  劣于  $OP$ , 则取  $OP$  作为粒子的位置以防止优解的丢失.

**Step 7:** 粒子进化及最佳学习结束.

## 2.3 外部档案信息交换

在多目标粒子群优化算法中, 全局最优  $g_{best}$  的选择会对算法的性能产生直接影响. 在优化过程中, 全局最优  $g_{best}$  通常会在外部档案更新完成后, 根据一定策略从外部档案存储的非支配解中进行选择. 在 MLII-MOPSO 算法中, 为了为粒子群优化和粒子进化与学习提供更好的全局最优, 外部档案更新完成后, 在外部档案存储的非支配解间进行信息交换, 以提高外部档案中非支配解的收敛性和多样性.

外部档案更新完成后, 根据此时档案中非支配解的数目, 按照一定比例随机选取要进行信息交换的非支配解集合  $\Phi$ , 同时生成一个随机数  $rand$ , 根据随机数  $rand$  在外部档案剩余的非支配解中选取集合  $\Phi$

的对应集合  $\Psi$ . 根据分维合作的思想, 对于集合  $\Phi$  中的第  $i$  个非支配解  $\Phi(i)$ , 依次使用集合  $\Psi$  中第  $i$  个非支配解  $\Psi(i)$  的变量替换其对应的变量, 并计算新解  $\Phi'(i)$  的适应值. 如果  $\Phi'(i)$  优于  $\Phi(i)$ , 则令  $\Phi(i) = \Phi'(i)$ ; 如果  $\Phi'(i)$  和  $\Phi(i)$  的适应值相等或无法比较, 则根据一定概率  $P_s$ , 令  $\Phi(i) = \Phi'(i)$ ; 否则, 保持  $\Phi(i)$  的值不变.

假设集合  $\Phi$  和  $\Psi$  中非支配解的个数为  $n$ , 每个非支配解的维数为  $m$ , 则集合  $\Phi$  和  $\Psi$  中的非支配解可分别表示为

$$\Phi(i) = (x_1^i, x_2^i, \dots, x_m^i), \Psi(i) = (y_1^i, y_2^i, \dots, y_m^i),$$

外部档案信息交换的伪代码如下:

For  $i = 1$  to  $n$

For  $j = 1$  to  $m$

将  $\Phi(i)$  中第  $j$  个元素替换为  $\Psi(i)$  中第  $j$  个元素, 得

$$\Phi'(i) = (x_1^i, x_2^i, \dots, y_j^i, \dots, x_m^i);$$

If  $\Phi'(i) \prec \Phi(i)$ ,  $\Phi(i) = \Phi'(i)$ ;

Else if  $\Phi'(i)$  和  $\Phi(i)$  的适应值相等或无法比较, 则根据概率  $p_s$ , 令  $\Phi(i) = \Phi'(i)$ ;

Else 保持  $\Phi(i)$  不变;

End if

End for

End for

在非支配解集合  $\Phi$  信息交换完成后, 重新进行外部档案更新.

## 2.4 算法流程

MLII-MOPSO 算法流程可以归纳如下.

**Step 1:** 初始化粒子种群  $Pop$ , For  $i = 0$  to  $N$  (种群粒子数目), 随机初始化粒子  $Pop(i)$ .

**Step 2:** 初始化粒子速度, For  $i = 0$  to  $N$ ,  $Vel(i) = 0$ .

**Step 3:** 计算粒子适应度值.

**Step 4:** 初始化粒子个体最优, For  $i = 0$  to  $N$ ,  $p_{best}(i) = Pop(i)$ .

**Step 5:** 寻找初始种群中的非支配解, 并将其存储到外部档案.

**Step 6:** While (未达到最大循环次数):

**Step 6.1:** 由式 (5) 和 (6) 更新粒子速度和位置;

**Step 6.2:** 判断粒子新位置是否优于旧位置, 如果新位置优于旧位置, 则转至 **Step 6.4**, 否则, 转至 **Step 6.3**;

**Step 6.3:** 粒子进化和最佳学习;

**Step 6.4:** 计算粒子适应度值;

Step 6.5: 外部档案更新, 将新产生的非支配解加入外部档案, 并删除外部档案中的支配解, 同时, 判断外部档案中非支配解的数目, 如果其超过了外部档案上限, 则进行外部档案删减;

Step 6.6: 外部档案信息交换;

Step 6.7: 从外部档案中为粒子群优化和粒子进化与学习分配全局最优  $g_{best}$ ;

Step 6.8: 更新粒子个体最优  $p_{best}(i)$ ;

Step 6.9: 增加循环计数.

### 3 实验分析

#### 3.1 性能评价标准

对于一个多目标优化算法, 通常通过其解集逼近待优化问题真实 Pareto 前沿的程度和解集分布的均匀性对其性能进行评价, 因此, 本文选取 GD (generation distance) 和 SP (spacing) 两个性能指标对多目标优化算法进行评价. GD 主要用来衡量优化算法产生的 Pareto 前沿 ( $PF_{known}$ ) 到待优化问题真实 Pareto 前沿 ( $PF_{true}$ ) 的距离, 定义如下:

$$GD = \left( \frac{1}{n_{pf}} \sum_{i=1}^{n_{pf}} d_i^2 \right)^{1/2}. \quad (7)$$

其中:  $n_{pf}$  为解集  $PF_{known}$  中解的数目,  $d_i$  为解集  $PF_{known}$  中第  $i$  个解与解集  $PF_{true}$  中其最靠近的解的欧氏距离. GD 的值越小表明算法的收敛性越好.

SP 主要表示优化算法产生的 Pareto 前沿 ( $PF_{known}$ ) 中解分布的均匀性, 定义如下:

$$SP = \left[ \frac{1}{n_{pf}} \sum_{i=1}^{n_{pf}} (d_i - \bar{d})^2 \right]^{1/2}. \quad (8)$$

其中:  $d_i = \min_j \left( \sum_{k=1}^m |f_k^i(\mathbf{x}) - f_k^j(\mathbf{x})| \right)$ ,  $i, j = 1, 2, \dots, n_{pf}$ ,  $i \neq j$ ,  $n_{pf}$  为解集  $PF_{known}$  中解的数目,  $m$  为待优化的目标的个数,  $\bar{d}$  为所有  $d_i$  的均值. SP 的值越小表明解集  $PF_{known}$  的均匀性越好.

#### 3.2 实验分析

为了验证本文优化算法的有效性, 将其与 NSGA-II 算法和 MOPSO 算法进行比较. 选择 3 个有代表性的测试函数 ZDT1、KUR 和 FON 进行算法性能测试, 其 Pareto 前沿分别为凸的、非连续的和非凸的.

在实验中, NSGA-II 算法种群大小为 100, 交叉概率为 0.8, 采用锦标赛选择方法, 变异概率为  $1/u$  ( $u$  为决策向量维数). MOPSO 和 MLII-MOPSO 算法种群粒子数为 100, 外部档案大小为 100, 在进行外部档案更新时, 格网间隔为 30. 测试时, 当适应度函数评价次数达到 25 000 次时, 算法停止运行.

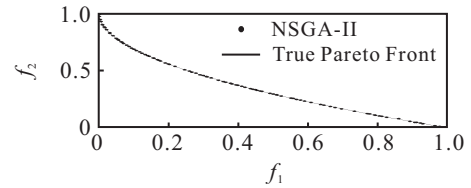
##### 1) 测试函数 ZDT1.

测试函数 ZDT1 具有凸的 Pareto 前沿, 定义为

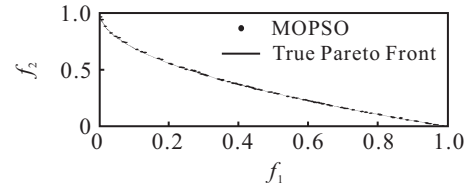
$$\min \begin{cases} f_1(\mathbf{x}) = x_1, \\ f_2(\mathbf{x}) = g(\mathbf{x})[1 - \sqrt{x_1/g(\mathbf{x})}], \\ g(\mathbf{x}) = 1 + 9 \left( \sum_{i=2}^n x_i \right) / (n-1). \end{cases} \quad (9)$$

其中:  $x_i \in [0, 1]$ ,  $i = 1, 2, \dots, 30$ .

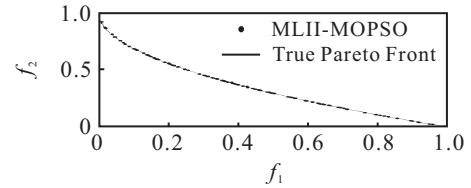
NSGA-II、MOPSO 和 MLII-MOPSO 算法生成的 Pareto 前沿与 ZDT1 函数真实的 Pareto 前沿如图 2 所示, 相应的性能评价指标如表 1 所示. 由图 2 可见, 对于 ZDT1 函数, 3 个优化算法的解集均能完全覆盖真实的 Pareto 前沿. 由表 1 可见, 在 3 个优化算法中, MLII-MOPSO 算法的收敛性是最好的 (GD 值最小), 其解集分布的均匀性仅次于 NSGA-II 算法, 好于 MOPSO 算法.



(a) NSGA-II 算法生成的 Pareto 前沿



(b) MOPSO 算法生成的 Pareto 前沿



(c) MLII-MOPSO 算法生成的 Pareto 前沿

图 2 不同优化算法生成的 ZDT1 函数的 Pareto 前沿

表 1 各优化算法对 ZDT1 函数的优化性能指标

	NSGA-II	MOPSO	MLII-MOPSO
GD(mean)	0.0024	0.0028	0.0014
GD(best)	0.0012	0.0017	0.0008
SP(mean)	0.0076	0.0094	0.0092
SP(best)	0.0066	0.0079	0.0075

##### 2) 测试函数 KUR.

测试函数 KUR 的 Pareto 前沿为非连续的, 定义为

$$\min \begin{cases} f_1(\mathbf{x}) = \sum_{i=1}^{n-1} (-10 \exp(-0.2 \sqrt{x_i^2 + x_{i+1}^2})), \\ f_2(\mathbf{x}) = \sum_{i=1}^n (|x_i|^{0.8} + 5 \sin x_i^3). \end{cases} \quad (10)$$

其中:  $x_i \in [-5, 5], i = 1, 2, 3$ .

NSGA-II、MOPSO 和 MLII-MOPSO 算法生成的 Pareto 前沿与 KUR 函数真实的 Pareto 前沿如图 3 所示, 相应的性能评价指标如表 2 所示. 由图 3 和表 2 可见, 对于 KUR 函数, MOPSO 和 MLII-MOPSO 算法的解集均能完全覆盖真实的 Pareto 前沿, MLII-MOPSO 算法的收敛性好于 MOPSO 算法. 对于 NSGA-II 算法, 其部分解集偏离真实的 Pareto 前沿较远, 收敛性较差. 在解集分布均匀性方面, MLII-MOPSO 算法好于 NSGA-II 算法和 MOPSO 算法.

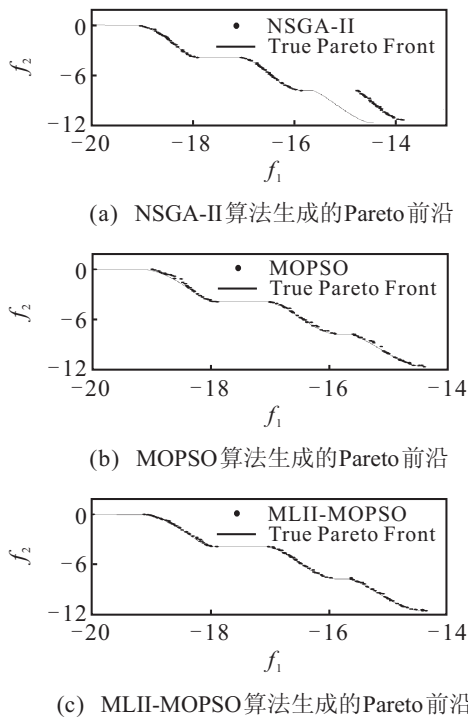


图 3 不同优化算法生成的 KUR 函数的 Pareto 前沿

表 2 各优化算法对 KUR 函数的优化性能指标

	NSGA-II	MOPSO	MLII-MOPSO
GD(mean)	0.5219	0.0461	0.0296
GD(best)	0.2182	0.0265	0.0129
SP(mean)	0.1541	0.0777	0.0479
SP(best)	0.0415	0.0626	0.0287

### 3) 测试函数 FON.

测试函数 FON 的 Pareto 前沿为非凸的, 定义为

$$\min \begin{cases} f_1(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^n \left(x_i - \frac{1}{\sqrt{n}}\right)^2\right), \\ f_2(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^n \left(x_i + \frac{1}{\sqrt{n}}\right)^2\right). \end{cases} \quad (11)$$

其中:  $x_i \in [-4, 4], i = 1, 2, 3$ .

NSGA-II、MOPSO 和 MLII-MOPSO 算法生成的 Pareto 前沿与 FON 函数真实的 Pareto 前沿如图 4 所示, 相应的性能评价指标如表 3 所示. 由图 4 可见, 对于 FON 函数, 3 个优化算法的解集均能完全覆盖真实

的 Pareto 前沿. 由表 3 可见, NSGA-II 算法的解集分布比较均匀, 但收敛性较差; MOPSO 算法的收敛性较好, 但解集分布的均匀性较差. 相比于 NSGA-II 算法和 MOPSO 算法, MLII-MOPSO 算法具有更好的收敛性, 其解集分布也更均匀.

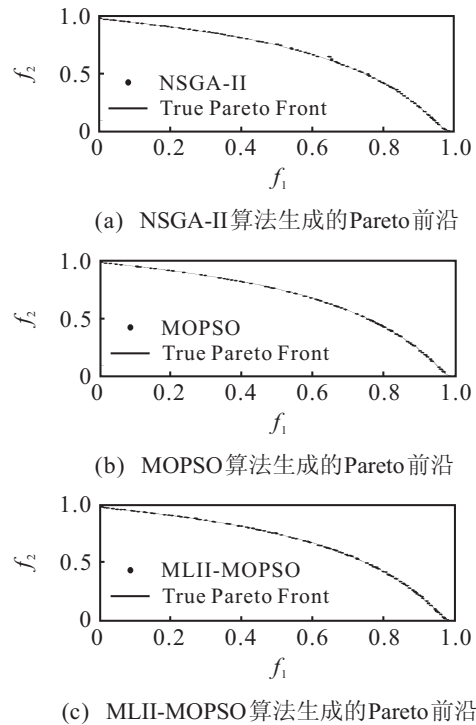


图 4 不同优化算法生成的 FON 函数的 Pareto 前沿

表 3 各优化算法对 FON 函数的优化性能指标

	NSGA-II	MOPSO	MLII-MOPSO
GD(mean)	0.0021	0.0011	0.0010
GD(best)	0.0017	0.0009	0.0009
SP(mean)	0.0071	0.0095	0.0069
SP(best)	0.0060	0.0081	0.0054

通过对 3 个典型测试函数的优化实验可知, 本文提出的 MLII-MOPSO 算法能够更好地收敛到待优化问题真实的 Pareto 前沿, 并且解集分布的均匀性更好.

## 4 结 论

本文提出了一种基于多层次信息交互的多目标粒子群优化算法, 该算法将整个优化过程分为标准粒子群优化层、粒子进化与学习层以及档案信息交换层 3 个层次. 在粒子进化与学习层中, 充分利用每一次迭代得到的粒子信息, 通过粒子进化和粒子全局最优学习, 保证每一次迭代都能得到更好的粒子位置. 在档案信息交换层中, 通过档案中非支配解之间的信息交换提高非支配解的质量, 从而为粒子群优化以及粒子进化与学习提供更好的全局最优. 通过与 NSGA-II 和 MOPSO 算法的实验比较, 表明 MLII-MOPSO 算法具有良好的性能, 能够有效解决多目标优化问题.

MLII-MOPSO 算法通过粒子进化与学习和档案信息交换提高了多目标优化算法的收敛性和多样性,但也增加了算法的时间复杂度,因此,如何进一步提高 MLII-MOPSO 算法的时间效率是今后将要解决的问题.

### 参考文献(References)

- [1] 王辉, 钱锋. 基于拥挤度与变异的动态微粒群多目标优化算法[J]. 控制与决策, 2008, 23(11): 1238-1242.  
(Wang H, Qian F. Improved PSO-based multi-objective optimization by crowding with mutation and particle swarm optimization dynamic changing[J]. Control and Decision, 2008, 23(11): 1238-1242.)
- [2] Kaveh A, Laknejadi K. A novel hybrid charge system search and particle swarm optimization method for multi-objective optimization[J]. Expert Systems with Applications, 2011, 38(12): 15475-15488.
- [3] 张利彪, 周春光, 马铭, 等. 基于粒子群算法求解多目标优化问题[J]. 计算机研究与发展, 2004, 41(7): 1286-1291.  
(Zhang L B, Zhou C G, Ma M, et al. Solutions of multi-objective optimization problems based on particle swarm optimization[J]. J of Computer Research and Development, 2004, 41(7): 1286-1291.)
- [4] Srinivas N, Deb K. Multi objective function optimization using nondominated sorting genetic algorithms[J]. Evolutionary Computation, 1995, 2(3): 221-248.
- [5] 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.
- [6] Zitzler E, Thiele L. Multi objective evolutionary algorithms: A comparative case study and the strength Pareto approach[J]. IEEE Trans on Evolutionary Computation, 1999, 3(4): 257-271.
- [7] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm[R]. Zurich, Switzerland: Swiss Federal Institute Technology, 2001.
- [8] Knowles J D, Corne D W. Approximating the nondominated front using the Pareto archived evolution strategy[J]. Evolutionary Computation, 2000, 8(2): 149-172.
- [9] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proc of the IEEE Int Conf on Neural Networks. Piscataway: IEEE, 1995: 1942-1948.
- [10] 徐鹤鸣. 多目标粒子群优化算法的研究[D]. 上海: 上海交通大学电子信息与电气工程学院, 2013.  
(Xu H M. Research on multiobjective particle swarm optimization algorithms[D]. Shanghai: School of Electronics and Electric Engineering, Shanghai Jiaotong University, 2013.)
- [11] Coello C A, Pulido G T, Lechuga M S. Handling multiple objectives with particle swarm optimization[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 256-279.
- [12] Tsai S J, Sun T Y, Liu C C, et al. An improved multi-objective particle swarm optimizer for multi-objective problems[J]. Expert Systems with Applications, 2010, 37(8): 5872-5886.
- [13] 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.
- [14] 张利彪, 周春光, 刘小华, 等. 求解多目标优化问题的一种多子群体进化算法[J]. 控制与决策, 2007, 22(11): 1313-1316.  
(Zhang L B, Zhou C G, Liu X H, et al. A multiple subswarms evolutionary algorithm for multi-objective optimization problems[J]. Control and Decision, 2007, 22(11): 1313-1316.)
- [15] 唐贤伦. 混沌粒子群优化算法理论及应用[D]. 重庆: 重庆大学自动化学院, 2007.  
(Tang X L. The theory and application of particle swarm optimization algorithm based on chaos[D]. Chongqing: College of Automation, Chongqing University, 2007.)

(责任编辑: 郑晓蕾)