Journal of System Simulation

Volume 35 | Issue 1 Article 4

1-18-2023

Large-scale Multi-objective Natural Computation Based on Dimensionality Reduction and Clustering

Weidong Ji
1.College of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China;, kingjwd@126.com

Yuqi Yue
1.College of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China;

Xu Wang
1.College of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China;

Ping Lin
2.Harbin Medical Sciences University, Harbin 150081, China;

Large-scale Multi-objective Natural Computation Based on Dimensionality Reduction and Clustering

Abstract

Abstract: In multi-objective optimization problems, as the number of decision variables increases, the optimization ability decreases significantly. To solve "dimension disaster", *a large-scale multi-objective natural computation method based on dimensionality reduction and clustering is proposed. The decision variables are optimized by locally linear embedding(LLE) to obtain the representation of high-dimensional variables in the low-dimensional space, then the individuals are grouped through K-means to select the appropriate guide individuals for the population to strengthen the convergence and diversity.* To verify the effectiveness, the method is applied to the multi-objective particle swarm optimization algorithm and the non-dominated sorting genetic algorithm. The convergence is analyzed to prove that the algorithm converges with probability 1. Experiments is carried out through 8 functions of ZDT and DTLZ series, compared with 6 representative algorithms, and its comprehensive performance is verified through the evaluation results of PF, IGD and HV, and applied to the water pump scheduling problem. Comprehensive experimental results show that the proposed method has better performance.

Keywords

dimension reduction, multi-objective optimization, locally linear embedding(LLE), natural computation, K-means

Recommended Citation

Weidong Ji, Yuqi Yue, Xu Wang, Ping Lin. Large-scale Multi-objective Natural Computation Based on Dimensionality Reduction and Clustering[J]. Journal of System Simulation, 2023, 35(1): 41-56.

第 35 卷第 1 期 系统仿真学报© Vol. 35 No. 1
2023 年 1 月 Journal of System Simulation Jan. 2023

基于降维和聚类的大规模多目标自然计算方法

季伟东[1*]，岳玉麒[1]，王旭[1]，林平[2]
(1. 哈尔滨师范大学 计算机科学与信息工程学院，黑龙江 哈尔滨 150025；2. 哈尔滨医科大学，黑龙江 哈尔滨 150081)

摘要：在多目标优化问题中，随着决策变量数目增多，算法的寻优能力会显著下降，针对这种“维数灾难”的问题，*提出基于LLE降维思想和K-means聚类策略的大规模多目标自然计算方法。首先通过LLE降维思想对决策变量进行优化，得到高维变量在低维空间中的表示，再通过K-means策略对个体分组，为种群选择合适的引导个体，提高算法的收敛性和多样性。*为验证算法有效性，将该方法应用于多目标粒子群优化算法和非支配排序遗传算法中，对收敛性进行了分析，证明该算法以概率1收敛。通过ZDT、DTLZ系列8个测试问题进行仿真试验，与6个代表性算法进行对比，通过PF、IGD指标、HV指标的评价结果验证其综合性能，并将其应用于水泵调度问题中。综合实验结果表明，所提方法具有较好性能。

关键词：降维；多目标优化；LLE；自然计算方法；K-means

中图分类号：TP391.9 文献标志码：A 文章编号：1004-731X(2023)01-0041-16

DOI: 10.16182/j.issn1004731x.joss.21-0667

引用格式: 季伟东, 岳玉麒, 王旭, 等. 基于降维和聚类的大规模多目标自然计算方法[J]. 系统仿真学报, 2023, 35(1):41-56.

Reference format: Ji Weidong, Yue Yuqi, Wang Xu, et al. Large-scale Multi-objective Natural Computation Based on Dimensionality Reduction and Clustering[J]. Journal of System Simulation, 2023, 35(1): 41-56.

Large-scale Multi-objective Natural Computation Based on Dimensionality Reduction and Clustering

Ji Weidong[1*], *Yue Yuqi*[1], *Wang Xu*[1], *Lin Ping*[2]

(1. College of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China;
2. Harbin Medical Sciences University, Harbin 150081, China)

Abstract: In multi-objective optimization problems, as the number of decision variables increases, the optimization ability decreases significantly. To solve "dimension disaster", *a large-scale multi-objective natural computation method based on dimensionality reduction and clustering is proposed. The decision variables are optimized by locally linear embedding(LLE) to obtain the representation of high-dimensional variables in the low-dimensional space, then the individuals are grouped through K-means to select the appropriate guide individuals for the population to strengthen the convergence and diversity.* To verify the effectiveness, the method is applied to the multi-objective particle swarm optimization algorithm and the non-dominated sorting genetic algorithm. The convergence is analyzed to prove that the algorithm converges with probability 1. Experiments is carried out through 8 functions of ZDT and DTLZ series, compared with 6 representative algorithms, and its comprehensive performance is verified through the evaluation results of PF, IGD and HV, and applied to the water pump scheduling problem. Comprehensive experimental results show that the proposed method has better performance.

收稿日期：2021-07-13 修回日期：2021-10-20

基金项目：国家自然科学基金(31971015)；黑龙江省自然科学基金(LH2021F037)；哈尔滨市科技局科技创新人才研究专项(2017RAQXJ050)；哈尔滨师范大学计算机科学与信息工程学院科研项目(JKYKYY202001)

第一作者：季伟东(1978-)，男，教授，博士，研究方向为大数据、群体智能。E-mail：kingjwd@126.com

第 35 卷第 1 期　　系统仿真学报　　Vol. 35 No. 1
2023 年 1 月　　Journal of System Simulation　　Jan. 2023

Keywords: dimension reduction; multi-objective optimization; locally linear embedding(LLE); natural computation; K-means

0　引言

现实中存在的问题大多需要同时进行多个目标的优化，这些问题通常被称作多目标优化问题(multi-objective problems，MOPs)[1]。伴随着实际应用问题的逐渐复杂化，一些更加复杂的多目标优化问题不断出现，这些问题主要被划分为两类：一类是目标数目较多的高维多目标优化问题(many-objective optimization problem，MaOP)；另一类是决策变量数目较多的大规模多目标优化问题(large-scale multi-objective optimization problem，LSMOP)[2]。

现有求解多目标优化问题的算法大多更加关注于高维多目标优化问题，而对于决策变量数目较多的大规模多目标优化问题的研究相对较少[3]。迄今为止，国内外学者以不同的视角和问题背景为基础针对 LSMOP 提出了若干解决方法，Potter 等[4-5]提出了协同进化算法，用遗传算法对子问题进行求解，但是大多问题的决策变量之间存在关联，并非独立存在，该算法考虑到变量与变量之间的相关性，因此在处理一些变量关系复杂的问题时无法获得较好的效果。Antonio 等[6]提出 CCGDE3，将协同进化与广义差分进化策略融合以解决 500 个决策变量的大规模多目标优化问题，但该算法同样对于变量关系没有较好的分析处理，缺乏发现变量间关联性的有效方法。Basu 等[7]提出 MOEA/D(s&ns)，将 MOEA/D-DE 与协同进化方法相结合，该算法针对决策变量之间的关系进行了改进，通过判断决策变量之间是否可分离将其划分成子种群，然后对各个子种群进行优化，显著增强了算法的性能。

与上述通过基于协同进化的方法不同，Zhang 等[8]提出了 LMEA，该算法利用基于角度的聚类分析方法分析变量属性，该方法对于决策变量的分析是一种值得参考的思路，且计算代价相对较低，但当决策变量增多时，由于需要将其划分为收敛性相关变量和多样性相关变量，并分别采用不同策略进行优化，计算资源将会急剧增加，影响算法性能。Chen 等[9]提出了 S^3-CMA-ES，该方法将协方差矩阵自适应策略 CMA-ES 进一步改进，提出可扩展小种群，将划分后的收敛性相关变量分为多个子类，多个子类同时进化得到一组 Pareto 最优解，因此该方法中对于变量的交互分析与分类会严重影响其时间复杂度，其计算开销也相对较大，如何更好分析处理变量并更合理有效地分配计算资源仍待解决。

近年来，学者们还提出了基于问题重构方法的算法来解决问题，Zille 等[10]提出了加权优化框架(weighted optimization framework，WOF)，该方法对优化问题的转化即问题重构更好地解决了 LSMOP 问题，该算法将变量进行分组并关联权重，同一组内的决策变量有同样的权重，从而将对高维决策变量的优化转化成为对相对较低维度的权重向量的优化，有效地降低决策空间的维度，在重构过程中较依赖分组技术，仍需对问题转换策略做进一步研究。He 等[11]基于 WOF 策略提出了利用问题重构方法直接跟踪 Pareto 最优解的方法 LSMOF，将候选解与决策空间相关联，再将 LSMOP 转化成为单目标问题，利用基于指标的多目标优化策略求解，与基于问题重构的其他方法比较，LSMOF 对计算资源的消耗相对较少，且收敛速度较快。

除此之外，学者们还提出了其他方法求解大规模多目标优化问题。Cheng 等[12]提出了基于社会学习的求解大规模多目标问题粒子群优化算法，该方法使粒子的每一维向较优的粒子对应维度进行学习，从而得到问题的最优解；Zhang 等[13]将竞争机制引入粒子群算法，该方法在生成子

https://dc-china-simulation.researchcommons.org/journal/vol35/iss1/4
DOI: 10.16182/j.issn1004731x.joss.21-0667

代个体过程中通过失败者向成功者学习来进行粒子更新。这些方法基于不同视角分析解决了大规模多目标问题，且在一些问题测试中得到了较好的结果，但这些方法大多针对某一种进化算法进行优化，并且方法的实现必须严格依赖于该算法的具体细节[14]，针对此问题，本文提出了一种基于降维思想和聚类策略的大规模多目标自然计算方法(large-scale multi-objective nature computation based on dimension reduction and clustering strategy，DRC-LMNC)，使其能够适用于多种不同的自然计算方法且不需要考虑算法的具体操作细节，更加具有普适性，其创新点主要体现在以下2个方面：

(1) 将LLE(local linear embedding)降维思想应用于大规模多目标优化领域；

(2) 提出基于K-means聚类的引导个体选取方法。

将该方法应用在2种不同的自然计算方法中，分别采用ZDT和DTLZ两组测试函数与其他6个代表性的算法进行对比，并应用于水泵调度问题中，验证了本文中所提策略的普适性和有效性。

1 多目标优化问题(MOP)

一个具有n个决策变量，m个目标变量的多目标优化问题可描述为

$$\begin{cases} \min & F(x)=(f_1(x), f_2(x), \cdots, f_m(x)) \\ \text{s.t.} & g_i(x)\leqslant 0,\ i=1, 2, \cdots, q \\ & h_j(x)=0,\ j=1, 2, \cdots, p \end{cases} \tag{1}$$

式中：$x=(x_1, x_2, \cdots, x_n)\in \Omega^n$为$n$维的决策变量，$\Omega^n$是决策变量的可行解空间；$f_k(x)(k=1, 2, \cdots, m)$为目标函数，$m$为目标空间维数。$g_i(x)\leqslant 0,\ i=1, 2, \cdots, q$和$h_j(x)=0,\ j=1, 2, \cdots, p$分别为MOP的第$i$个不等式约束和第$j$个等式约束。

有关求解LSMOP的Pareto最优定义如下[15]：

(1) Pareto支配：当且仅当

$$\forall i=1, 2, \cdots, m,\ \ f_i(x_v)\leqslant f_i(x_u) \tag{2}$$

$$\wedge \exists j=1, 2, \cdots, m,\ \ f_j(x_v)<f_j(x_u) \tag{3}$$

称决策变量$x_v=(v_1, v_2, \cdots, v_n)$对$x_u=(u_1, u_2, \cdots, u_n)$是Pareto支配的，记作$x_v \prec x_u$。

(2) Pareto最优解：对于任意$x'\in \Omega^n$为问题的可行解，当且仅当

$$\neg\exists x\in \Omega^n,\ x \prec x' \tag{4}$$

称x'为Pareto最优解。

(3) Pareto最优解集：如果

$$P^*=\{x\in \Omega^n | \neg\exists x'\in \Omega^n,\ x' \prec x\} \tag{5}$$

则称P^*为Pareto最优解集。

(4) Pareto前沿：根据P^*的定义，其对应的目标向量的集合被称为Pareto前沿(PF)

$$PF=\{y=(f_1(x), f_2(x), \cdots, f_m(x)) | x\in P^*\} \tag{6}$$

2 基于降维思想和聚类策略的大规模多目标自然计算方法(DRC-LMNC)

2.1 局部线性嵌入(LLE)降维方法

LLE是一种流形学习算法[16]，其运用局部的线性化构建超平面，通过局部的线性关系相结合来表示全局非线性结构[17-19]，从而将高维数据映射到低维空间中，使数据保持局部线性结构不变。设高维数据表示为$\boldsymbol{X}=(x_1, x_2, \cdots, x_m)\in \mathbb{R}^{n\times m}$，得到的低维数据为$\boldsymbol{Y}=(y_1, y_2, \cdots, y_m)\in \mathbb{R}^{n\times m}$，LLE算法降维具体步骤如下：

(1) 假设在较小局部中数据是线性的，即每个数据可以通过它的几个邻近样本进行线性表示。在算法中，采用欧氏距离来确定每个样本x_i的h个近邻。

(2) 计算样本x_i与其h个近邻之间的线性关系即权重矩阵$\boldsymbol{w}$，采用均方差作为损失函数，最小化误差函数为

$$J(\boldsymbol{w})=\sum_{i=1}^{m}\left\| \boldsymbol{x}_i-\sum_{j\in Q(i)} w_{ij}\boldsymbol{x}_j \right\|_2^2 \tag{7}$$

式中：$Q(i)$表示i的m个近邻样本，且需要对w_{ij}进行归一化限制，即要满足

$$\sum_{j\in Q(i)} w_{ij}=1 \tag{8}$$

(3) 计算降维后的矩阵 $\boldsymbol{Y}$，通过步骤(2)求解出高维权重系数，将高维数据$\{x_1, x_2, \cdots, x_m\}$映射到低维数据空间$\{y_1, y_2, \cdots, y_m\}$，并使其保持线性关系，最小化损失函数$J(\boldsymbol{Y})$如下：

$$J(\boldsymbol{Y})=\sum_{i=1}^{m}\left\|\boldsymbol{y}_i-\sum_{j=1}^{m}w_{ij}\boldsymbol{y}_i\right\|_2^2 \tag{9}$$

为了得到标准化的低维数据需满足约束条件：

$$\sum_{i=1}^{m}\boldsymbol{y}_i=0;\ \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{y}_i\boldsymbol{y}_i^{\mathrm{T}}=I \tag{10}$$

令$\boldsymbol{M}=(\boldsymbol{I}-\boldsymbol{W})(\boldsymbol{I}-\boldsymbol{W})^{\mathrm{T}}$，进一步求解：

$$J(\boldsymbol{Y})=tr(\boldsymbol{YMY})^{\mathrm{T}} \tag{11}$$

可得$\boldsymbol{Y}$即$\boldsymbol{M}$的特征向量构成的矩阵[20]。

2.2 基于K-means聚类的全局引导个体选取策略

全局引导个体的选取能够极大影响算法的收敛性与多样性，目前的全局引导选取方法中大多更加偏重于考虑个体间的距离，对于整个PF上解的分布关注较少，解的均匀性较差时不易选择较优的全局引导个体。合适的引导个体应当对整个PF的分布情况进行考虑，并且对于较稀疏的区域应当更加关注，从而增强算法的多样性。因此本文采用K-means聚类算法对解进行分类，使每一个类追逐自己的引导个体，使引导个体能够带领种群逐步移动至Pareto前沿并能够均匀覆盖，以保持种群的多样性和均匀性。

K-means是依赖欧氏距离的方法[21]，随机选取初始的聚类中心个体，将其他个体划分到最近子类中。以两目标问题为例，如图1所示，图中f_1为目标1，f_2为目标2，根据目标空间的值将外部档案中的种群划分K个子类，并对子类中个体的非支配等级进行排序，每一个子类中随机选择一个等级最高的个体，最终得到K个个体作为种群的全局引导个体。

K-means策略具体步骤如下。

输入：外部档案中的n个个体，聚类数K。

输出：K个全局引导个体。

(1) 在外部档案中随机选择K个个体，分别代表K个子类的中心个体；

(2) 将其余个体划分到与其欧氏距离最小的中心个体所在的类中；

(3) 对每个类的中心重新进行计算；

(4) 重复操作步骤(2)和(3)直到达到迭代次数；

(5) 得到最终K个子类，对子类中个体进行非支配排序；

(6) 随机选取每个类中一个非支配等级最高的个体作为该类的引导个体。

将外部档案中的个体进行聚类后，不同类间的个体密度较小，可以引导种群在不同的区域搜索，从而保证Pareto前沿的均匀性，同一类中的个体密度较大，能够加强局部探索能力，进一步调整种群的探索与开发能力，在加强搜索的同时也能够平衡算法的收敛性与多样性。

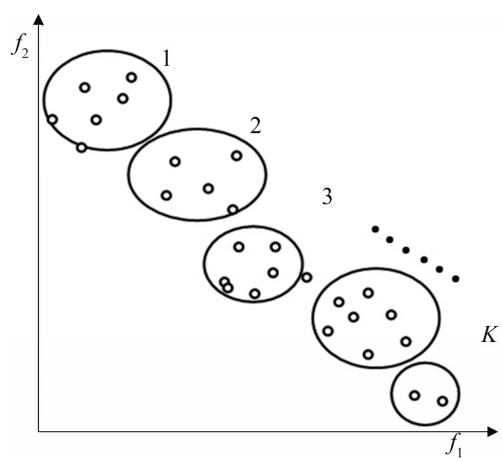


图1　基于K-means的种群分组及引导个体选择策略
Fig. 1　Population grouping and guide individual selection based on K-means

2.3 基于降维思想和聚类策略的大规模多目标自然计算方法(DRC-LMNC)

在多目标优化问题中，由于不存在真正意义上的最优解，并且多目标粒子群优化算法(MOPSO)算法的收敛速度较快，很容易导致粒子聚集，陷入局部收敛搜索停滞，而在决策变量维度增加时，又存在着寻优效率低，无法找到真实

https://dc-china-simulation.researchcommons.org/journal/vol35/iss1/4
DOI: 10.16182/j.issn1004731x.joss.21-0667

Pareto 前沿的问题，在求解复杂多目标问题尤其是大规模多目标优化问题(LSMOP)时收敛性和多样性明显不足，因此将 LLE 降维思想和 K-means 聚类应用于多目标粒子群优化算法和非支配排序遗传算法中。

2.3.1 基于降维思想和聚类策略的方法(DRC)实现流程

基于降维思想和聚类策略的方法(DRC)实现流程如图2所示。

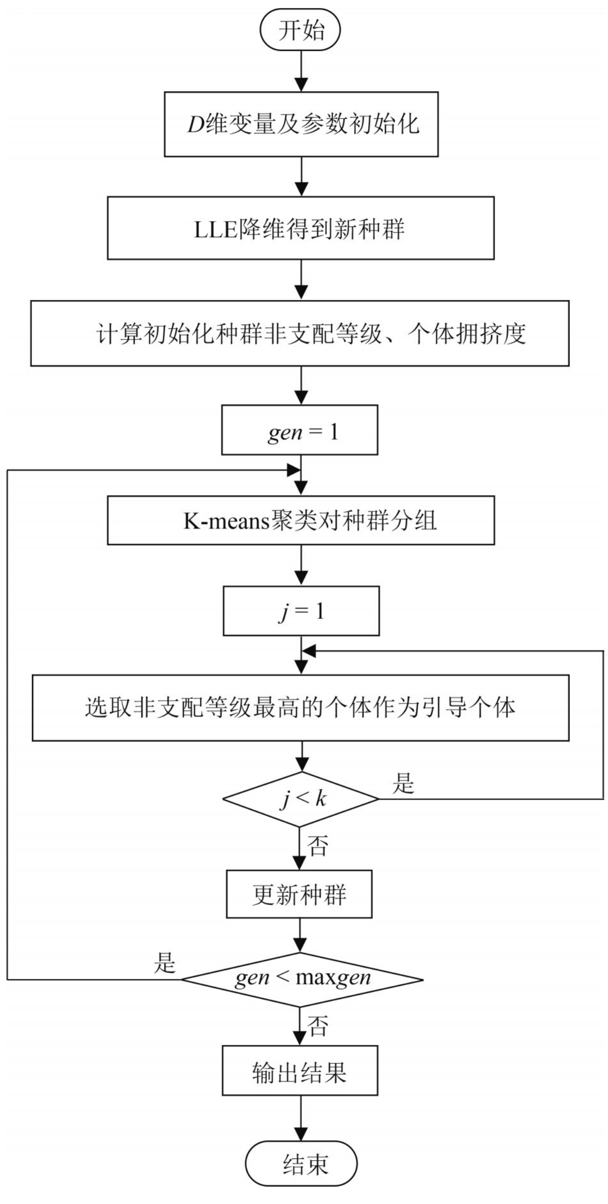


图2 DRC 方法流程
Fig. 2 DRC flow chart

2.3.2 DRC 在多目标粒子群优化算法(MOPSO)中的应用

算法实现步骤如下：

step 1：设定超参数，种群规模 N，维度 D，最大迭代次数 max*gen*，学习因子，惯性权重，近邻数 h，LLE 降维后特征维数 d，K-means 聚类分组数 K。根据种群规模 N 和种群初始维度 D，随机初始化所有粒子，生成一个 $N\times D$ 的矩阵，设定为初始种群 *pop*。

step 2：对于初始种群 *pop*，根据其每一个粒子 x_i 的 h 个邻居 x_{ij} 求解出局部权值矩阵，再根据式(7)~(11)计算得到 $\boldsymbol{Y}$，即种群降维后得到的 h 维的位置数据 *new_pop*。

step 3：对得到的新种群 *new_pop* 进行非支配排序，计算每一个粒子的非支配等级以及拥挤度距离。

step 4：使用 2.2 中的步骤(1)~(6)对种群进行聚类分组，随机选取每个子类中一个非支配等级最高的粒子作为引导个体，引导该子类其他个体。

step 5：将种群中的粒子与外部档案中的解进行比较，并判断外部档案中的解是否超过限制，是则随机删除非支配等级低的解。

step 6：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，输出非支配解。

DRC 在 MOPSO 应用的伪代码如算法 1 所示。

```
Algorithm 1: DRC-MOPSO
  pop=initialize(LLE_h, D, sizepop, x_long)
  npop=LLE(pop, x_long)
  [F1, npop_non]=non_domination_sort(npop)
  npop=crowding_distance_sort(F1, npop_non)
  For i in maxgen:
    [npop_k]=k_means(npop, k)
    leader=npop_first(npop_first_index)
    npop=MOPSO(npop)
    sort(Fi, < n)
    Pt+1=Pt+1∪Fi[1: (N-|Pt+1|)]
  End for
```

2.3.3 DRC 在非支配排序遗传算法(NSGA-II)中的应用

算法实现步骤如下。

step 1：设定超参数，种群规模 N，维度 D，最大迭代次数 max*gen*，交叉和变异概率 P_c、P_m，近邻数 h，LLE 降维后特征维数 d，K-means 聚类分组数 K。根据种群规模 N 和种群初始维度 D，随机初始化所有个体，生成一个 $N\times D$ 的矩阵，设定为初始种群 *pop*。

step 2：对于初始种群 *pop*，根据其每一个个体 x_i 的 h 个邻居 x_{ij} 求解出局部权值矩阵，再根据式(7)~(11)计算得到 $\boldsymbol{Y}$，即种群降维后得到的 h 维的位置数据 *new_pop*。

step 3：对得到的新种群 *new_pop* 进行非支配排序，计算每一个个体的非支配等级以及拥挤度距离并排序。

step 4：使用 2.2 中的步骤(1)~(6)对种群进行聚类分组，随机选取每个子类中一个非支配等级最高的粒子作为引导个体，引导该子类其他个体。

step 5：对种群进行选择、交叉、变异操作，产生下一代种群。

step 6：将种群中的个体与外部档案中的解进行比较，并判断外部档案中的解是否超过限制，是则随机删除非支配等级低的解。

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，输出非支配解。

DRC 在 NSGA-II 应用的伪代码如算法 2 所示。

```
Algorithm 2: DRC-NSGA
  pop=initialize(LLE_h, D, sizepop, x_long)
  npop=LLE(pop, x_long)
  [F1, npop_non]=non_domination_sort(npop)
  npop=crowding_distance_sort(F1, npop_non)
    For i in maxgen:
    [npop_k]=k_means(npop, k)
    leader=npop_first(npop_first_index)
    Rt=Pt ∪ Qt
    npop=NSGA-II(npop)
    Sort(Fi, < n)
    Pt+1=Pt+1 ∪ Fi[1: (N-|Pt+1|)]
    Qt+1=make_new_pop(Pt+1)
  End for
```

2.4 算法复杂度分析

DRC-LMNC 基于降维思想和聚类策略的大规模多目标自然计算方法，首先通过 LLE 思想对种群进行降维初始化，根据上文中的详细描述和运算规则推导出 LLE 算法的 3 个步骤复杂度分别为 $O(N\lg N)$, $O(DNK^3)$, $O(dN^2)$ 其中 N 为个体数目，D 为降维前的特征，d 为降维后的特征数目，K 为近邻数，在 DRC-LMNC 中，K 和 d 的取值为较小的常数，因此 LLE 降维过程的时间复杂度为 $O(N^2)$。得到新种群后每迭代一次所需的时间为 $T(K)+T(L)$，其中，$T(L)$ 是当前种群中所有个体进行一代进化操作的时间，$T(K)$ 是执 K-means 操作的时间。由于 $T(L)$ 的计算必须依据每一种算法的具体进化操作的步骤，所以主要分析本文中所提的 DRC 方法的时间复杂度，K-means 聚类时间复杂度为 $O(KDIN)$，其中 K 为聚类数，N 为个体数，I 为迭代次数，D 为特征数，因此可简化为 $O(N)$。

综上，再根据符号 O 执行条件语句的运算规则计算得到 $T(DRC-LMNC)=O(N^2)+i(O(N)+O(L))$。

3 LLE 降维思想的收敛性分析

定义 1(偏序关系与偏序集) 令 $x, y\in L$ 为定义在 x, y 上的二元关系，即存在序偶 $\langle x, y\rangle\in R$，以 xRy 表示。若对于 $\forall x\in L, xRx$，则称 R 具有自反性；若 $\forall x, y\in L, xRy, yRx$，必有 $x=y$，则称 R 具有反对称性；若 $\forall x, y\in L, xRy, yRz$，必有 xRz，则称 R 具有传递性。

若二元关系 R 是自反的、反对称的、传递的，则称 R 为偏序关系，(L, R) 为偏序集。在多目标优化问题中定义的支配关系“$\succ$”是严格的偏序关系。

定义 2(最小元素与非支配集) 若在 L 中，$\exists x \in L$，使 $x \succ x^*$，则称 x^* 是偏序集 $(L, \succ)$ 中的最小元素。所有最小元素的集合表示为 $M(L, \succ)$，这里定义的最小元素集合即为 L 的非支配集。

定义 3(“⩾”关系) 对于一个给定的MOP和2个进化群体 $\boldsymbol{P}_1$ 和 $\boldsymbol{P}_2$，若 $\forall x \in \boldsymbol{P}_1$，$\exists y \in \boldsymbol{P}_2$，使 $y \succ x$，则定义 $\boldsymbol{P}_1$ 和 $\boldsymbol{P}_2$ 的关系为 $\boldsymbol{P}_1 \geqslant \boldsymbol{P}_2$。

定义 4(多目标渐进收敛算法的收敛准则) 对于一般的优化问题 $\langle Y, f\rangle$，采用多目标随机优化算法 Z，第 t 次迭代得到的种群为 $\boldsymbol{P}_t$，下一次迭代种群则为 $\boldsymbol{P}_{t+1} = Z(\boldsymbol{P}_t, \zeta)$，其中，$Y$ 为问题的可行解空间，f 为该算法的问题函数，ζ 为该算法曾经搜索过的解集[22]，在Lebesgue测度空间中，搜索的前沿边界被定义为

$$R_\xi = \{\boldsymbol{P} \in Y \mid f(\boldsymbol{P}) < \sigma + \xi\}$$

式中：ξ 为任意大于0的数集，σ 为问题的真实Pareto解集。如果算法 Z 在迭代中寻找到了 R_ξ 中的一个子集，即认为算法 Z 找到了可以接受的全局最优Pareto解集或近似全局最优Pareto解集[23]。

条件 1 单向性 $Z(P, \zeta) \geqslant \boldsymbol{P}$，且若 $\zeta \in Y$，$Z(\boldsymbol{P}, \zeta) \geqslant \zeta$

条件 2 收敛性 对 $\forall A_\sigma \in Y,\ s.t.\ v(A_\sigma) > 0$，有

$$\prod_{t=0}^{\infty}(1 - v_t(A_\sigma)) = 0$$

式中：$v_t(A_\sigma)$ 为算法 Z 在第 t 次迭代搜索到的解集在真实解集附近空间 A_σ 上的概率测度。多目标渐进收敛算法满足条件1与条件2。

定理 1 LLE-LMNC(LLE-large-scale multi-objective natural computation)算法在使种群由 D 维降维至 d 维后，新生成的种群依然遵循收敛准则。

证明 降维前种群矩阵为 $m_D = \{\boldsymbol{P}_{1,D}, \boldsymbol{P}_{2,D}, \cdots, \boldsymbol{P}_{N,D}\}$ $\boldsymbol{P}^{(t)}_{\text{gbest},D} \geqslant, \cdots, \geqslant \boldsymbol{P}^{(2)}_{\text{gbest},D} \geqslant \boldsymbol{P}^{(1)}_{\text{gbest},D}$，其中 N 为子群个数。我们记该种群中的最佳子种群为 $\boldsymbol{P}^{(t)}_{\text{gbest},D}$。一个算法收敛的 x, y 充要条件是符合条件1与条件2。

根据条件1，最佳子群支配程度序列有，所以对 $\forall i > j$ 有 $\boldsymbol{P}^{(i)}_{\text{gbest},D} \geqslant \boldsymbol{P}^{(j)}_{\text{gbest},D}$ 成立。

根据条件2，有 $\lim\limits_{t\to\infty} \boldsymbol{P}^{(t)}_{\text{gbest},D} = \sigma$ 成立，其中 σ 为该问题的最优解集[24]。

通过LLE-LMNC算法将种群矩阵由 $m_D = \{\boldsymbol{P}_{1,D}, \boldsymbol{P}_{2,D}, \cdots, \boldsymbol{P}_{N,D}\}$ 降维至 $m_d = \{\boldsymbol{P}_{1,d}, \boldsymbol{P}_{2,d}, \cdots, \boldsymbol{P}_{N,d}\}$ 后，会产生新的最优子群 $\boldsymbol{P}^{(t)}_{\text{gbest},d}$，降维后的每个子种群依然遵循多目标渐进收敛算法进行收敛操作，那么对于降维后的种群矩阵 $m_d = \{\boldsymbol{P}_{1,d}, \boldsymbol{P}_{2,d}, \cdots, \boldsymbol{P}_{N,d}\}$，有 $\forall i > j \boldsymbol{P}^{(i)}_{\text{gbest},d} \geqslant \boldsymbol{P}^{(j)}_{\text{gbest},d}$，即降维后的种群符合条件1，生成的最佳子群支配程度序列有 $\boldsymbol{P}^{(t)}_{\text{gbest},d} \geqslant, \cdots, \geqslant \boldsymbol{P}^{(2)}_{\text{gbest},d} \geqslant \boldsymbol{P}^{(1)}_{\text{gbest},d}$，随着迭代次数的增加，$\boldsymbol{P}^{(t)}_{\text{gbest},d}$ 逐渐逼近最优Pareto解集，所以有 $\lim\limits_{t\to\infty} P(\boldsymbol{P}^{(t)}_{\text{gbest},d} = \sigma) = 1$，即降维后的种群符合条件2，所以降维后的种群在迭代过程中依然收敛。证毕。

定理 2 任意多目标渐进收敛算法在经过LLE-LMNC算法降维处理后，仍以概率为1收敛。

证明 由定理1可知，种群矩阵由 D 维降低至 d 维后，整个种群状态空间 M 中，对全部 $m_i \in M,\ (i = 1, 2, \cdots, d)$ 在迭代过程中有：

$$\prod_{i=1}^{d} \lim_{t\to\infty} P(m_i^{(t)} \in M, \boldsymbol{P}^{(t)}_{\text{gbest},i} = \sigma) = 1^{\sum_{i=1}^{d}\frac{1}{i}} = 1$$

式中：σ 为降维后种群最佳个体。LLE-LMNC算法以概率为1收敛，定理2得证。

4 实验结果分析

4.1 实验说明

为了验证DRC在LSMOP中的有效性，采用标准测试函数进行试验并应用于水泵调度问题，分别为ZDT[25]、DTLZ[26]系列共8个测试问题。ZDT系列中包括6个二目标测试函数，DTLZ系列包括7个测试函数，本文中选取DTLZ2、4、7作为三目标函数进行测试，实验仿真平台为Windows10，Matlab2020a。

本实验将DRC应用到多目标优化问题自然计算领域的NSGA-II和MOPSO中，得到DRC-NSGA算法和DRC-MOPSO算法，选取NSGA-II、MOPSO、MOEA/D、SPEA2、CCMOPSO、GDE3

第 35 卷第 1 期　　系统仿真学报　　Vol. 35 No. 1
2023 年 1 月　　Journal of System Simulation　　Jan. 2023

算法进行对比，算法性能通过Pareto前沿、反世代距离IGD指标、超体积指标HV进行评价。本文选取NSGA-II等算法进行对比的目的为：① NSGA-II[27]算法采用快速非支配排序以及拥挤距离的策略且时间复杂度低，由于其性能上的优势，多年来常被用作对比算法；② MOPSO[28]算法是一种收敛速度较快的经典多目标优化算法；③ MOEA/D[29]是基于分解的一种能够有效解决大变量优化问题的算法；④ SPEA2[30]算法是多目标经典算法之一，常与NSGA-II等算法一起被用作对比；⑤CCMOPSO[2]算法是基于变量分解的MOPSO算法，其在大规模多目标问题上有很好的表现；⑥ GDE3[31]算法是第三代进化差分算法，也是近年来解决大规模多目标问题的有效算法。测试函数描述如表1所示。

表1　标准测试函数描述
Table 1　Standard test function description

函数名称	目标函数描述	维数	说明
ZDT1	$\mathrm{ZDT1}=\begin{cases}\min f_1(x)=x_1\\ \min f_2(x)=g(1-\sqrt{f_1/g})\\ g(x)=1+9\sum_{i=2}^{d}x_i/(d-1)\\ \text{s.t. } 0\leqslant x_i\leqslant 1,\ i=1,\ 2,\cdots,\ 30\end{cases}$	D=1 000	凸，连续
ZDT2	$\mathrm{ZDT2}=\begin{cases}\min f_1(x)=x_1\\ \min f_2(x)=g(1-(f_1/g)^2)\\ g(x)=1+9\sum_{i=2}^{d}x_i/(d-1)\\ \text{s.t. } 0\leqslant x_i\leqslant 1,\ i=1,\ 2,\cdots,\ 30\end{cases}$	D=1 000	凹，连续
ZDT3	$\mathrm{ZDT3}=\begin{cases}\min f_1(x)=x_1\\ \min f_2(x)=g(1-\sqrt{f_1/g}-(f_1/g)\sin(10\pi f_1))\\ g(x)=1+9\sum_{i=2}^{d}x_i/(d-1)\\ \text{s.t. } 0\leqslant x_i\leqslant 1,\ i=1,\ 2,\cdots,\ 30\end{cases}$	D=1 000	凹，不连续
ZDT4	$\mathrm{ZDT4}=\begin{cases}\min f_1(x)=x_1\\ \min f_2(x)=g(1-\sqrt{f_1/g})\\ g(x)=1+10(d-1)+\sum_{i=2}^{d}(x_i^{\,2}-10\cos(4\pi x_i))\\ \text{s.t. } 0\leqslant x_1\leqslant 1,\ -5\leqslant x_i\leqslant 5,\ i=2,3,\cdots,10\end{cases}$	D=1 000	凸，多模态
ZDT6	$\mathrm{ZDT6}=\begin{cases}\min f_1(x)=1-\exp(-4x_i)\sin^6(6\pi x_1)\\ \min f_2(x)=g(1-(f_1/g)^2)\\ g(x)=1+9\left(\left(\sum_{i=2}^{d}x_i/(d-1)\right)^{0.25}\right)\\ \text{s.t. } 0\leqslant x_i\leqslant 1,\ i=1,\ 2,\cdots,10\end{cases}$	D=1 000	凹，不连续
DTLZ2	$\mathrm{DTLZ2}=\begin{cases}\min f_1(x)=\cos\left(\frac{\pi}{2}x_1\right)\cos\left(\frac{\pi}{2}x_2\right)(1+g(x))\\ \min f_2(x)=\cos\left(\frac{\pi}{2}x_1\right)\sin\left(\frac{\pi}{2}x_2\right)(1+g(x))\\ \min f_3(x)=\sin\left(\frac{\pi}{2}x_1\right)(1+g(x))\\ g(x)=\sum_{i=3}^{d}(x_i-0.5)^2\\ \text{s.t. } 0\leqslant x_i\leqslant 1,\ i=1,\ 2,\cdots,\ d\end{cases}$	D=1 000	凸，连续

续表

函数名称	目标函数描述	维数	说明
DTLZ4	$\text{DTLZ4}=\begin{cases}\min f_1(x)=\cos\left(\frac{\pi}{2}x_1^{\alpha}\right)\cos\left(\frac{\pi}{2}x_2^{\alpha}\right)(1+g(x))\\ \min f_2(x)=\cos\left(\frac{\pi}{2}x_1^{\alpha}\right)\sin\left(\frac{\pi}{2}x_2^{\alpha}\right)(1+g(x))\\ \min f_3(x)=\sin\left(\frac{\pi}{2}x_1^{\alpha}\right)(1+g(x))\\ g(x)=\sum_{i=3}^{d}(x_i-0.5)^2\\ \text{s.t. } 0\leqslant x_i\leqslant 1,\ i=1,2,\cdots,d\end{cases}$	D=1 000	凹，非均匀
DTLZ7	$\text{DTLZ7}=\begin{cases}\min f_1(x)=\frac{1}{\lfloor n/3\rfloor}\sum_{i=\lfloor(j-1)n/3\rfloor}^{\lfloor jn/3\rfloor}x_i,\ j=1,2,3\\ \text{s.t. } g_j(x)=f_3(x)+4f_j(x)-1\geqslant 0,\ j=1,2\\ g_3=2f_3(x)(x)+\min_{i,j\neq 1,i\neq j}[f_j(x)+f_j(x)]-1\geqslant 0\\ \text{s.t. } 0\leqslant x_i\leqslant 1,\ i=1,2,\cdots,n,\ n>3\end{cases}$	D=1 000	不连续，多模态

4.2 实验结果分析

4.2.1 参数设置

实验中DRC-MOPSO和MOPSO算法中的参数设置为：学习因子$c_1=1$, $c_2=2$，惯性权重$w_{\max}=0.9$，$w_{\min}=0.4$，迭代次数为1 000次，种群规模N=100，决策变量维数D=1 000。DRC-NSGA和NSGA-II参数设置为：交叉概率$P_c=0.90$，变异概率$P_m=1/n$，迭代次数1 000次，种群规模N=100，决策变量维数D=1 000，各算法对表1中的8个测试函数分别执行30次。LLE算法中近邻数h=10，降维维度d取值25，K-means策略中，聚类分组数K取值为7。

4.2.2 性能评价指标

为了更加直观有效的评价算法的有效性，本文采用反世代距离(inverted generational distance，IGD)和超体积(hypervolume，HV)分别对算法性能进行评价。

(1) IGD指标：IGD是用来评价真的Pareto前沿与算法运行得到的近似Pareto前沿之间的距离指标[32-33]，可以同时评价收敛性以及多样性且计算代价小，其表示为

$$IGD(PF,PF^*)=\frac{1}{S}\sum_{i=1}^{S}\min dis(PF_i,PF^*) \quad (12)$$

式中：PF^*为真实Pareto前沿的一组采样数据；PF为算法求得的Pareto前沿；S为Pareto前沿中非支配解的个数。IGD指标数值越小，说明算法得到的近似Pareto前沿越接近真实Pareto前沿，表明算法的收敛性和多样性越好[34]。

(2) HV指标：HV是算法得到的非支配解集与参照点所围成的目标空间中区域的体积，是一个在Pareto支配关系上严格单调的度量指标，HV值越大，说明对应算法的综合性能就越好，表示为

$$HV=\delta\left(\bigcup_{i=1}^{|S|}v_i\right) \quad (13)$$

式中：δ为Lebesgue测度，用来测量体积；$|s|$为非支配解集的数目；v_i为参照点与解集中第i个解构成的超体积。

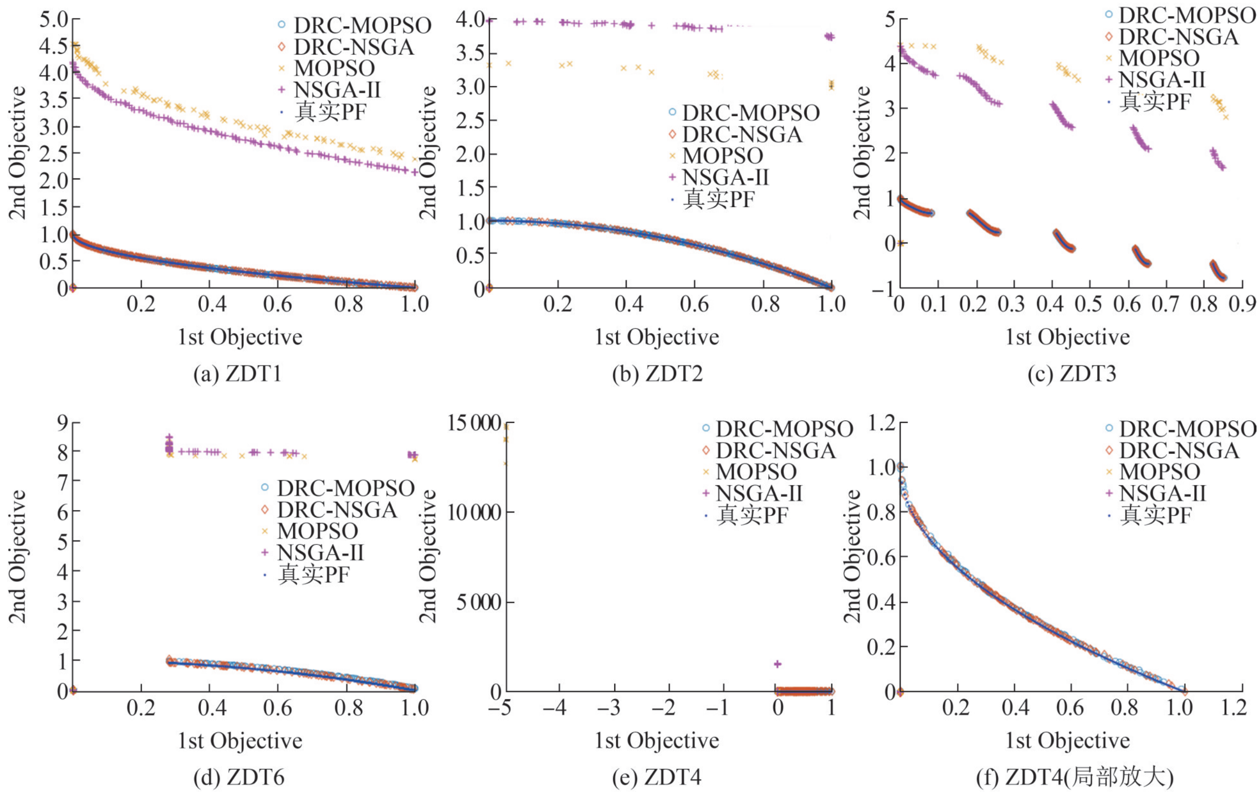
4.3 实验结果对比

图3、4分别给出了DRC-MOPSO、DRC-NSGA算法与MOPSO、NSGA-II算法在求解ZDT1-4、ZDT6、DTLZ2、DTLZ4、DTLZ7测试问题时的非支配解集和真实Pareto前沿的图像，其中图3中的(e)(f)均为测试函数ZDT4的Pareto前沿对比图，由于4种算法求解效果相差较大不易观察，故将(e)的局部放大得到(f)。由实验结果可

第35卷第1期 系统仿真学报 Vol. 35 No. 1
2023年1月 Journal of System Simulation Jan. 2023

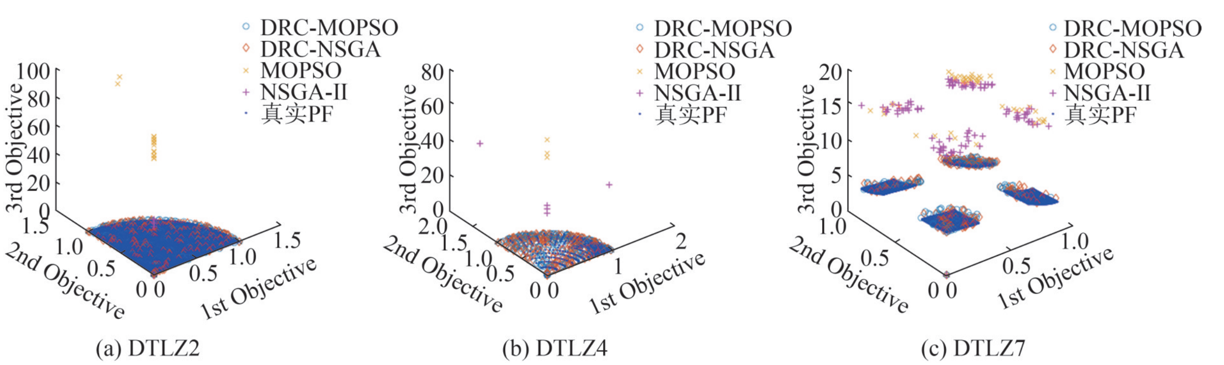
以看出，在8个测试问题中，MOPSO和NSGA-II算法在收敛性与多样性方面均缺乏良好表现，而本文所提DRC方法则能够有效提高算法性能，使其较好的收敛于真实Pareto前沿的附近，且分布较为均匀，在收敛性和多样性上均有较好的表现，表明本文提出的方法在解决大规模多目标问题中能够在很大程度上提升算法的性能。

为更加直观地展示DRC-MOPSO、DRC-NSGA算法与MOPSO、NSGA-II算法在求解测试问题时的综合性能，图5给出了4种多目标优化算法在求解8个测试函数时IGD指标的箱型图。从图中可知，DRC-MOPSO和DRC-NSGA算法在所有问题上有较好的收敛性，且异常数据较少，测试结果稳定，表现均优于MOPSO和NSGA-II算法，只有DRC-NSGA算法在ZDT6测试问题上的IGD指标波动稍大，但也优于其他算法。



(a) ZDT1 (b) ZDT2 (c) ZDT3
(d) ZDT6 (e) ZDT4 (f) ZDT4(局部放大)

图3 不同多目标优化算法求解二目标测试函数的Pareto前沿对比
Fig. 3 PF of different multi-objective optimization algorithms for two-objective problem



(a) DTLZ2 (b) DTLZ4 (c) DTLZ7

图4 不同多目标优化算法求解三目标测试函数的Pareto前沿对比
Fig. 4 PF of different multi-objective optimization algorithms for three-objective problem

https://dc-china-simulation.researchcommons.org/journal/vol35/iss1/4
DOI: 10.16182/j.issn1004731x.joss.21-0667

第 35 卷第 1 期
2023 年 1 月

Vol. 35 No. 1
Jan. 2023

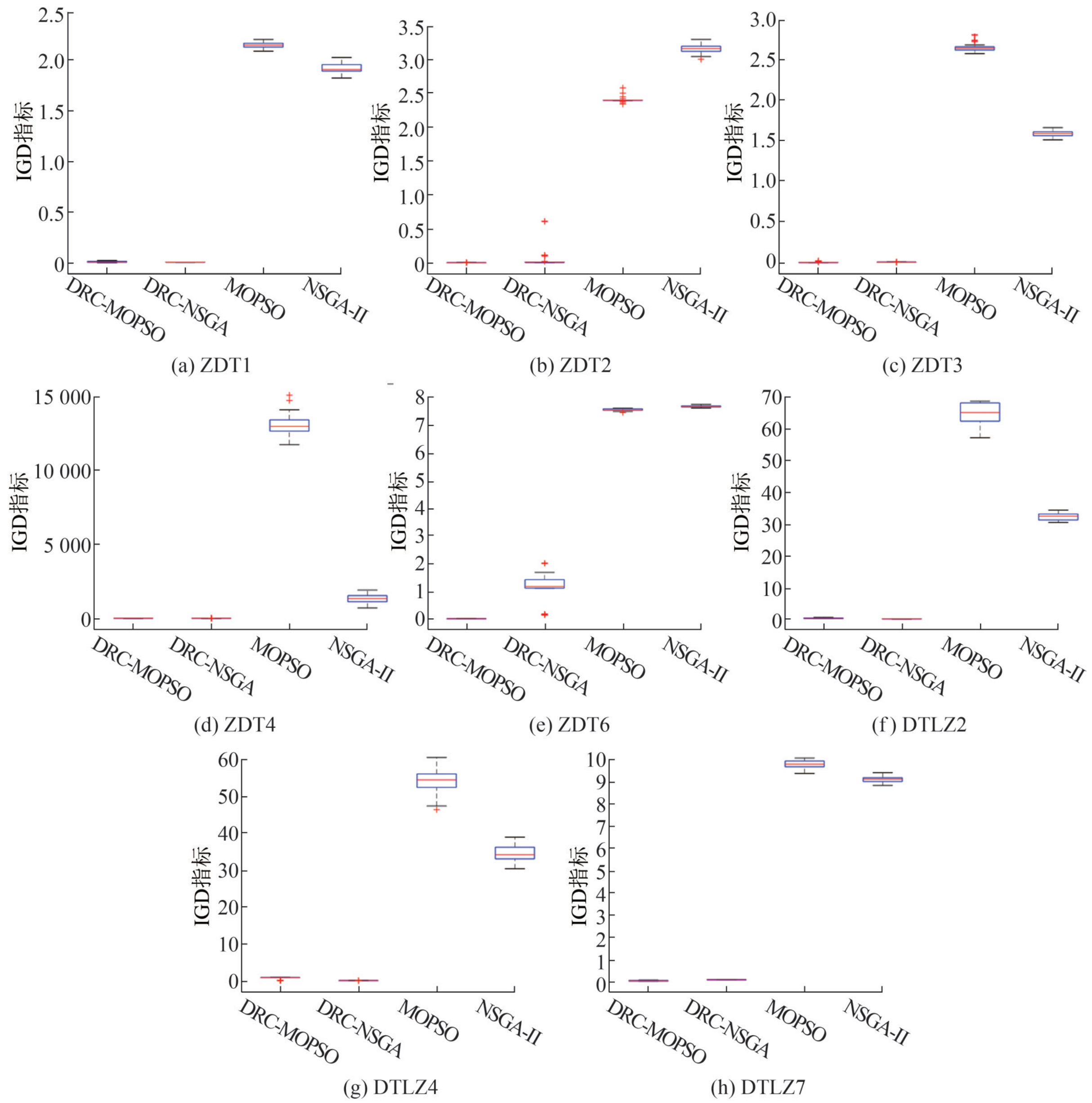


图5　不同多目标优化算法求解测试函数时IGD指标的箱型图
Fig. 5　Box plots of IGD for different multi-objective optimization algorithms

为进一步验证本文策略的有效性，分别对DRC-MOPSO算法和DRC-NSGA算法与MOPSO算法和NSGA-II算法以及经典算法MOEA/D和SPEA2进行对比实验，表2给出了6种算法在8个测试问题上IGD的最优值(Best)平均值(Mean)和标准差(std.)。

表2数据的结果显示，DRC-MOPSO和DRC-NSGA算法的IGD值优于其他经典算法，说明其求解得到的近似Pareto前沿更加接近真实Pareto前沿，这表示本文所提算法在各测试函数上均有不错的表现，具有较良好的收敛性和多样性，在提升计算效率的同时也能够获得更好的求解精度。并且和其他算法相比，DRC-MOPSO和DRC-NSGA的IGD指标标准差较低，证明应用LLE降维思想和K-means聚类策略后的算法稳定性更好。

为评估本文所提策略应用于MOPSO算法和NSGA-II算法后的综合性能，采用超体积指标(HV)对NSGA-II、MOPSO、CCMOPSO和GDE3进行对比实验，得到HV指标对比结果如表3所示。

第 35 卷第 1 期
2023 年 1 月
系统仿真学报
Journal of System Simulation
Vol. 35 No. 1
Jan. 2023

表2 DRC-NSGA算法和DRC-MOPSO算法与其它算法IGD指标对比
Table 2 IGD indicators of DRC-NSGA and DRC-MOPSO compare with other algorithms

测试函数	IGD指标	DRC-NSGA	NSGA-II	DRC-MOPSO	MOPSO	MOEA/D	SPEA2
ZDT1	Best	4.20×10^{-3}	1.53×10^{3}	7.55×10^{-2}	2.67×10^{2}	2.02×10^{3}	2.21×10^{3}
	Mean	5.33×10^{-3}	1.70×10^{3}	9.74×10^{-2}	2.97×10^{2}	2.43×10^{3}	2.70×10^{3}
	std	2.78×10^{-3}	1.13×10^{2}	1.77×10^{-2}	1.43×10	9.98×10	1.18×10^{2}
ZDT2	Best	1.41×10^{-2}	1.57×10^{3}	7.80×10^{-2}	1.66×10^{2}	1.30×10^{3}	1.75×10^{3}
	Mean	2.73×10^{-2}	1.76×10^{3}	1.12×10^{-1}	1.95×10^{2}	2.38×10^{3}	2.67×10^{3}
	std	2.39×10^{-2}	1.12×10^{2}	2.46×10^{-2}	4.55×10	9.68×10	1.21×10^{2}
ZDT3	Best	2.78×10^{-2}	1.53×10^{3}	7.93×10^{-2}	1.67×10^{2}	2.14×10^{3}	1.14×10^{3}
	Mean	3.79×10^{-2}	1.71×10^{3}	8.91×10^{-2}	2.96×10^{2}	2.45×10^{3}	1.98×10^{3}
	std	3.16×10^{-3}	9.14×10^{1}	9.84×10^{-3}	2.95×10	8.40×10	8.65×10
ZDT4	Best	3.90×10^{-1}	5.05×10^{3}	4.12×10^{-1}	3.80×10^{3}	5.41×10^{3}	4.05×10^{3}
	Mean	6.03	5.53×10^{3}	6.57×10^{-1}	4.91×10^{3}	6.49×10^{3}	4.93×10^{3}
	std	1.34	1.95×10^{2}	1.17	2.79×10^{2}	1.99×10^{2}	1.75×10^{2}
ZDT6	Best	1.09×10^{-2}	3.03×10	3.70×10^{-2}	2.19×10	3.09×10	3.89×10
	Mean	2.32×10^{-2}	3.49×10	5.87×10^{-2}	2.40×10	3.68×10	4.29×10
	std	1.83×10^{-3}	5.13×10^{-1}	1.68×10^{-2}	1.09	6.10×10^{-1}	4.13×10^{-1}
DTLZ2	Best	4.91×10^{-2}	3.45×10^{3}	4.85×10^{-2}	2.93×10^{2}	2.85×10^{3}	4.21×10^{3}
	Mean	5.83×10^{-2}	4.76×10^{3}	5.27×10^{-2}	3.46×10^{2}	3.96×10^{3}	5.02×10^{3}
	std	1.88×10^{-3}	2.49×10^{2}	2.03×10^{-3}	2.03×10	8.74×10	9.83×10^{2}
DTLZ4	Best	3.97×10^{-2}	2.67×10^{3}	3.75×10^{-2}	1.93×10^{2}	4.86×10^{3}	3.74×10^{3}
	Mean	4.70×10^{-2}	3.01×10^{3}	4.68×10^{-2}	2.94×10^{2}	6.83×10^{3}	4.19×10^{3}
	std	2.89×10^{-3}	2.07×10^{2}	3.93×10^{-3}	4.53×10	9.93×10	2.98×10^{2}
DTLZ7	Best	4.67×10^{-2}	2.45×10^{3}	4.78×10^{-2}	1.96×10^{2}	3.85×10^{3}	2.47×10^{3}
	Mean	5.86×10^{-2}	4.18×10^{3}	5.98×10^{-2}	2.27×10^{2}	4.73×10^{3}	4.03×10^{3}
	std	7.98×10^{-3}	8.02×10	7.53×10^{-2}	2.86×10	8.37×10	8.92×10

表3 DRC-NSGA算法和DRC-MOPSO算法与其他算法HV指标对比
Table 3 HV indicators of DRC-NSGA and DRC-MOPSO compare with other algorithms

测试函数	HV指标	DRC-NSGA	NSGA-II	DRC-MOPSO	MOPSO	CCMOPSO	GDE3
ZDT1	Mean	0.858	0.309	0.762	0.407	0.850	0.407
ZDT2	Mean	0.780	0.310	0.650	0.603	0.850	0.603
ZDT3	Mean	0.957	0.305	0.925	0.437	0.864	0.437
DTLZ2	Mean	0.991	0.567	0.974	0.853	0.587	0.759

对比结果显示，通过LLE降维思想和K-means聚类策略，DRC-NSGA和DRC-MOPSO算法所得到的HV指标值均优于NSGA-II和MOPSO算法，除ZDT2测试函数以外，在大部分问题上HV值优于CCMOPSO算法。

综合实验结果表明，本文中基于LLE降维思想的大规模多目标自然计算方法得到的非支配解集能够较好的收敛于真实Pareto前沿附近，这是由于算法通过LLE降维策略对决策变量进行了重构同时提升了寻优效率，并且通过K-means聚类加强了算法的局部勘探能力和开发能力，有效平衡了算法的收敛性与多样性，使算法在处理大规模多目标问题时能够较好的收敛同时降低计算复杂度。

4.4 DRC-MOPSO、DRC-NSGA 水泵调度问题测试

为测试DRC在解决实际优化问题中的有效性，将DRC-MOPSO、DRC-NSGA应用于水泵调度问题中，在该优化问题的测试中，采用NSGA-II、MOPSO和SPEA2作为对比算法，5种算法的初始种群规模设定为800，最大迭代次数为50，决策变量维度为864(详细描述见后文)。

如图6所示，该模型是一个简易供水模型，包括水源地、水泵组系统和蓄水池，水从水源地取出至取水泵组系统，经过水处理后进入蓄水池。

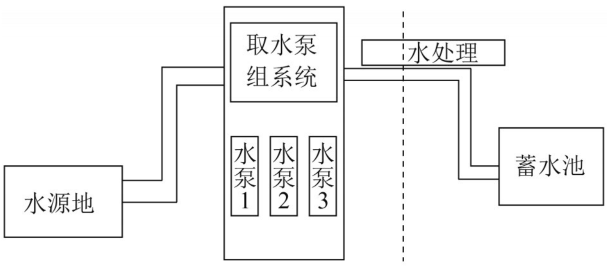


图6　简易供水模型

Fig. 6　Simple water supply model

水泵调度问题是在一定的约束条件下，通过平衡多个目标之间的冲突，寻求全局最优解的大规模多目标问题，是本文方法所解决的问题的实际应用。本文以满足约束条件下，最小化电力成本及维护成本为例，求解T时间内，L个水泵的水泵调度问题的数学描述如下：

$$\begin{cases} \min E_1 = M_L \sum_{i=1}^{204} c(p_i) + M_H \sum_{i=205}^{264} c(p_i) + \\ \qquad M_L \sum_{i=265}^{288} c(p_i) \\ \min E_2 = \sum_{i=2}^{288} \left\| \max\{0, (p_i - p_{i-1})\} \right\| \\ \text{s.t.}\quad h_{\min} \leqslant h_i \leqslant h_{\max} \end{cases} \tag{14}$$

式中：E_1为电力成本，即工作时间内产生的电力消耗；M_L为低电力成本指数；M_H为高电力成本指数；p_i为在间隔时间i内的水泵组合；$c(p_i)$为p_i的电力成本；E_2为维护成本，主要包括水泵的开关磨损；在该问题中，蓄水池中的水位至少保持$h_{\min}$高度，但不可超过蓄水池最大高度$h_{\max}$。

本问题中电力成本采用不同时段不同电价的方式计费，为了计算方便，本文将低成本记为0，高成本记为1，其成本结构如下：

低成本(0)：0:00—17:00；22:00—24:00

高成本(1)：17:00—22:00

本文以24 h内，3个水泵的调度问题为例，时间以5 min设置为一组时间间隔，共288组，变量维度为864，设置$h_{\min}=36$，$h_{\max}=50$，水泵调度问题的编码如表4所示。

表4　水泵调度问题的编码实现

Table 4　Coding implementation of pump scheduling problem

时间间隔 (5 min)	1			…	288		
水泵号	1	2	3	…	1	2	3
位串	0	1	0	…	1	0	1
状态	关闭	开启	关闭	…	开启	关闭	开启

最优解对比如图7所示，水泵调度方案的成本如表5所示。

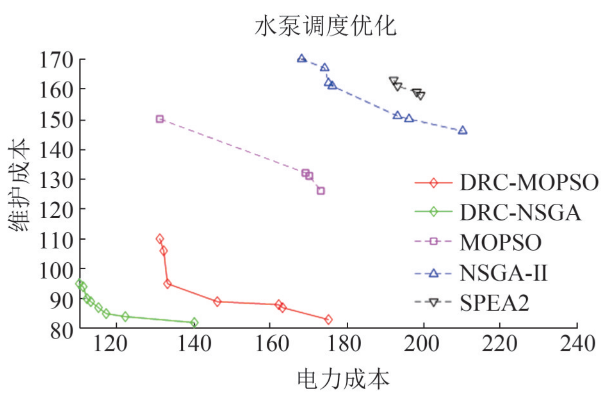


图7　水泵调度优化解集

Fig. 7　Pump scheduling optimizes the solution

从图7可以看出，采用DRC策略对水泵调度问题进行求解，所得的优化解集分布性较好，对于两类供应成本均能得到合理的调度方案。其中DRC-NSGA和DRC-MOPSO在该问题上的收敛性与多样性均优于其他算法能获得较为经济的调度方案，且消耗计算资源较低。

第 35 卷第 1 期 系统仿真学报 Vol. 35 No. 1
2023 年 1 月 Journal of System Simulation Jan. 2023

表5 水泵调度方案的成本
Table 5 Cost of pump scheduling scheme

DRC-MOPSO		DRC-NSGA		MOPSO		NSGA-II		SPEA2	
电力成本	维护成本	电力成本	维护成本	电力成本	维护成本	电力成本	维护成本	电力成本	维护成本
131	110	110	95	131	150	168	170	192	163
132	106	111	94	169	132	174	167	193	161
133	95	112	90	170	131	175	162	198	159
146	89	113	89	173	126	176	161	199	158
162	88	115	87			193	151		
163	87	117	85			196	150		
175	83	122	84			210	146		
		140	82						

5 结论

本文提出了一种基于LLE降维思想和K-means聚类策略的大规模多目标自然计算方法，算法通过LLE思想对决策变量进行优化，得到高维变量在低维空间中的表示，极大地降低了寻优的难度，有效地提升了种群中个体的搜索效率；通过K-means聚类策略对个体进行分组，为子种群选择合适的引导个体，避免因为引导个体不能有效指导种群而陷入局部收敛，搜索停滞。

为验证算法的有效性，本文通过ZDT系列5个测试问题和DTLZ系列3个测试问题进行仿真实验，并与6个具有代表性的多目标优化算法进行对比，通过Pareto前沿、IGD指标、HV指标的评价结果验证了该策略的综合性能，综合实验结果表明，本文策略具有较好的收敛性和多样性，且对于个体的均匀分布也有良好的效果，在平衡算法的局部探索与全局开发的同时保证算法拥有较均匀的Pareto前沿。并且在实际工程应用中的水泵调度问题上也有较好表现，但是本文的不足之处在于没有充分考虑变量之间的关联性，在后续研究中可以对其进一步讨论并分类，在优化过程中使用不同的进化策略进行改进，使得算法能够得到更好的收敛速度和均匀Pareto前沿。

参考文献：

[1] 乔俊飞, 李霏, 杨翠丽. 一种基于均匀分布策略的NSGA II算法[J]. 自动化学报, 2019, 45(7): 1325-1334.
Qiao Junfei, Li Fei, Yang Cuili. An NSGA II Algorithm Based on Uniform Distribution Strategy [J]. Acta Automatica Sinica, 2019, 45(7): 1325-1334.

[2] 邱飞岳, 莫雷平, 江波, 等. 基于大规模变量分解的多目标粒子群优化算法研究[J]. 计算机学报, 2016, 39(12): 2598-2613.
Qiu Feiyue, Mo Leiping, Jiang Bo, et al. Research on Multi-objective Particle Swarm Optimization Algorithm Based on Large-Scale Variable Decomposition[J]. Chinese Journal of Computers, 2016, 39(12): 2598-2613.

[3] Zhang X, Ye T, Ran C, et al. A Decision Variable Clustering-Based Evolutionary Algorithm for Large-Scale Many-objective Optimization[J]. IEEE Transactions on Evolutionary Computation (S1941-0026), 2018, 22(99): 97-112.

[4] Potter M A. A Cooperative Co-Evolutionary Approach to Function[J]. Third Parallel Problem Solving Form Nature (S0302-9743), 1994, 866: 249-257.

[5] Potter M A, Jong K. Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents[J]. Evolutionary Computation. IEEE (S1530-9304), 2014, 8 (1): 1-29.

[6] Antonio L M, Coello C. Use of Cooperative Coevolution for Solving Large Scale Multiobjective Optimization Problems[C]// 2013 IEEE Congress on Evolutionary Computation. Cancun, Mexico: IEEE, 2013: 2758-2765.

[7] Basu S, Mondal A, Basu A. A Cooperative Coevolutionary Approach for Multi-objective Optimization[M]. Recent Trends in Signal and Image Processing. Springer, Singapore, 2019: 57-65.

[8] Zhang X, Ye T, Ran C, et al. A Decision Variable Clustering-Based Evolutionary Algorithm for Large-Scale Many-objective Optimization[J]. IEEE Transactions on Evolutionary Computation (S1941-0026), 2018, 22(99):

97-112.

[9] Chen H, Zhu X, Pedrycz W, et al. PEA: Parallel Evolutionary Algorithm by Separating Convergence and Diversity for Large-Scale Multi-objective Optimization [C]// 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2018: 223-232.

[10] Zille H, Ishibuchi H, Mostaghim S, et al. A Framework for Large-Scale Multiobjective Optimization Based on Problem Transformation[J]. IEEE Transactions on Evolutionary Computation (S1941-0026), 2018, 22(2): 260-275.

[11] He C, Li L, Tian Y, et al. Accelerating Large-Scale Multi-objective Optimization via Problem Reformulation [J]. IEEE Transactions on Evolutionary Computation (S1941-0026), 2019, 23(6): 949-961.

[12] Cheng R, Jin Y. A Competitive Swarm Optimizer for Large Scale Optimization[J]. IEEE Transactions on Cybernetics (S2168-2275), 2015, 45(2): 191-204.

[13] Zhang X, Zheng X, Cheng R, et al. A Competitive Mechanism Based Multi-objective Particle Swarm Optimizer with Fast Convergence[J]. Information Sciences (S0020-0255), 2018, 427: 63-76.

[14] 王蓉芳, 焦李成, 刘芳, 等. 自适应动态控制种群规模的自然计算方法[J]. 软件学报, 2012, 23(7): 1760-1772.
Wang Rongfang, Jiao Licheng, Liu Fang, et al. Nature Computation with Self-adaptive Dynamic Control Strategy of Population Size[J]. Journal of Software, 2012, 23(7): 1760-1772.

[15] Kumar R S, Kondapaneni K, Dixit V, et al. Multi-objective Modeling of Productionand Pollution Routing Problem with Time Window: A Self-learning Particle Swarm Optimization Approach[J]. Computersand Industrial Engineering (S0360-8352), 2016, 99: 29-40.

[16] Wang Jing. Real Local-linearity Preserving Embedding [J]. Neurocomputing (S0925-2312), 2014, 136(20): 7-13.

[17] 张潞瑶, 季伟东, 程昊. 基于LLE降维思想的自然计算方法[J]. 系统仿真学报, 2020, 32(10): 1943-1955.
Zhang Luyao, Ji Weidong, Cheng Hao. Natural Calculation Method Based on LLE Dimensionality Reduction Idea [J]. Journal of System Simulation, 2020, 32(10): 1943-1955.

[18] 马瑞, 王家廞, 宋亦旭, 等. 基于局部线性嵌入(LLE)非线性降维的多流形学习[J]. 清华大学学报(自然科学版), 2008, 48(4): 582-585.
Ma Rui, Wang Jiaxin, Song Yixu, et al. Multi-manifold Learning using Locally Linear Embedding(LLE) Nonlinear Dimensionality Reduction[J]. Journal of Tsinghua University (Science and Technology), 2008, 48 (4): 582-585.

[19] Ridder D, Kouropteva O, Okun O. Supervised Locally Linear Embedding[C]// Artificial Neural Networks and Neural Information Processing. Istanbul, Turkey: Springer, 2003: 333-341.

[20] Roweis S T, Saul L K. Nonlinear Dimensionality Reduction by Locally Linear Embedding[J]. Science (S1095-9203), 2000, 290(5500): 2323-2326.

[21] Li Yuan, Bai Yansong. Research on KNN fault Detection with Improved Principal Component Analysis [J]. Journal of Shenyang University of Chemical Technology (S2095-5198), 2018, 32(4): 366-371.

[22] SOLIS F J, WETS J B. Minimization by Random Search Techniques[J]. Mathematics of Operations Research (S0364-765X), 1981, 6(1): 19-30.

[23] 张孟健, 龙道银, 王霄, 等. 基于马尔科夫链的灰狼优化算法收敛性研究[J]. 电子学报, 2020, 48(8): 1587-1595.
Zhang Mengjian, Long Daoyin, Wang Xiao, et al. Research on the Convergence of Grey Wolf Optimization Algorithm Based on Markov Chain[J]. Chinese Journal of Electronics, 2020, 48(8): 1587-1595.

[24] 潘峰, 周倩, 李位星, 等. 标准粒子群优化算法的马尔科夫链分析[J]. 自动化学报, 2013, 39(4): 381-389.
Pan Feng, Zhou Qian, Li Weixing, et al. Markov Chain Analysis of Standard Particle Swarm Optimization Algorithm[J]. Acta Automatica Sinica, 2013, 39(4): 381-389.

[25] Zitzler E, Deb K, Thiele L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results[J]. Evolutionary Computation (S1530-9304), 2000, 8(2): 173-195.

[26] Deb K, Thiele L, Laumanns M, et al. Scalable Test Problemsfor Evolutionary Multiobjective Optimization [C]// IEEE Congress on Evolutionary Computation. Piscataway: IEEE, 2002: 26-33.

[27] Deb K, Pratap A, Agarwal S, et al. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation (S1941-0026), 2002, 6(2): 182-197.

[28] Coello C A C, Pulido G T, Lechuga M S. Handling Multiple Objectives with Particle Swarm Optimization[J]. IEEE Transon Evolutionary Computation (S1941-0026), 2004, 8(3): 256-279.

[29] Zhang Q, Hui L. MOEA/D: A Multio-bjective Evolutionary Algorithm Based on Decomposition[J]. IEEE Transactions on Evolutionary Computation (S1941-0026), 2008, 11(6):712-731.

[30] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm[C]// Athens.

Greece, Technical Report Gloriastrasse, 2001: 103.

[31] Kukkonen S, Lampinen J. GDE3: The Third Evolution Step of generalized Differential Evolution[C]// 2005 IEEE Congress on Evolutionary Computation. IEEE, 2005: 443-450.

[32] Hu W, Yen G G. Adaptive Multiobjective Particle Swarmoptimization Based on Parallel Cell Coordinate System[J]. IEEETransactions on Evolutionary Computation (S1941-0026), 2015, 19(1): 1-18.

[33] Czyzak P, Jaszkiewicz A. Pareto Simulated Annealing—A Metaheuristic Technique for Multiple-objective Combinatorial Optimization[J]. Journal of Multi-criteria Decision Analysis (S1099-1360), 1998, 7(1): 34-47.

[34] 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 Maoguo, Jiao Licheng, Yang Dongdong, et al. Research on Evolutionary Multi-objective Optimization Algorithm[J]. Journal of Software, 2009, 20(2): 271-289.