

基于最小二乘法的高维目标减少算法

郑金华, 周 聪, 李 珂, 吕 卉

(湘潭大学 信息工程学院, 湖南 湘潭 411105)

摘要: 多目标进化算法在许多领域有广泛的应用, 大部分文献都只针对二维与三维的测试问题, 目标减少成为高维优化的热点之一. 本文从决策者角度考虑冗余目标问题, 提出了基于最小二乘法的目标减少算法(ORLSM), 该方法将每个目标函数分段拟合为若干条直线段, 然后比较各直线段之间的斜率来确定最冗余目标对, 进而确定冗余目标. 同时针对目标减少前后个体支配关系的变化情况, 提出了支配关系改变率的评价方法. 通过3个测试函数, 分别用逆世代距离(IGD)、支配关系改变率(CDR)和时间效率3个方面, 对同类的两个算法进行了性能测试. 结果表明, ORLSM在总体上具有最好的性能; CDR和IGD具有基本一致的评价结果.

关键词: 高维目标进化; 目标减少算法; 最小二乘法; 评价方法

中图分类号: TP18 **文献标识码:** A

Objective-reduction using the least squares method

ZHENG Jin-hua, ZHOU Cong, LI Ke, LÜ Hui

(Institute of Information and Engineering, Xiangtan University, Xiangtan Hunan 411105, China)

Abstract: Multi-objective evolutionary algorithms are widely applied to many real world problems; however, most of the papers merely focus on the problems with two or three objectives, in which objective-reduction has become a research focus for many multi-objective optimization. From the views of decision makers, this paper proposes a new objective-reduction using the least squares method(ORLSM). This algorithm fits each objective into multi-straight lines and determines the most redundant objective couples between each two slope vectors for searching the most redundant objective. Moreover, in view of the variety of individual dominance relation after the number of objectives is decreased, a performance assessment metric based on the changed Pareto dominance ratio(CDR) is also proposed. From an extensive comparative study with two similar algorithms in terms of inverted generational distance(IGD), CDR, and running time on 3 test problems, ORLSM indicates its superiority in overall performances; CDR and IGD almost have the evaluation in assessments.

Key words: multi-objective optimizations; objective-reduction algorithm; the least squares method; metric

1 引言(Introduction)

多目标进化算法(multi-objective evolutionary algorithm, MOEA)已经广泛应用到工程、科学等许多领域^[1], 但是大部分情况只考虑了对两个目标或3个目标的优化, 应用MOEA求解高维目标(比如10个目标或者更多)的Pareto解集是一项十分困难的工作: 1) 将高维目标空间可视化是一个难题; 2) 随着目标维数的增加, 计算复杂度也随之增加, 特别是算法的时间复杂度依赖于目标维数的情况; 3) 当目标维数增加后, 因非支配集规模过大, 完整描绘Pareto最优面几乎不可能; 4) 决策者要分析数量如此多的目标, 然后从中找出满意解集的区域, 也是一个十分复杂的事情. 这些都是制约MOEA应用到高维目标领域的原因.

目前, 处理高维问题主要有两种方法: 一种方法是放宽Pareto支配关系. 如Francesco等提出了有效序

的概念(efficiency of order)^[2], 对Pareto解集中的每个解的目标向量依次向 $k(k = m - 1, m - 2, \dots, 1)$ 维空间进行投影. 在目标空间中, 找到最小的 k , 使得向量的 $(k - 1)$ 个分量组成的子向量至少被种群中一个目标向量的这些分量组成的子向量支配, 而目标向量的任意 k 个分量组成的子向量不被种群中任意目标向量的这些分量组成的子向量支配. 基于有效序的思想排序非支配解的算法能找到更逼近Pareto前沿面且分布均匀的非支配解. 将其应用到四维、五维和六维问题, 取得了很好的效果; 另一种方法是减少目标个数. 一般情况下, 目标个数增多, 将给决策者和搜索过程带来困难^[3]. 有学者提出减少冗余目标, 即将完全一致或冲突较小的目标去掉, 从而在一定程度上减少目标的维数, 降低高维问题的求解难度^[4~11]. 当然, 冗余目标减少的方法并不能用随机法, 也不能采用穷举法. 而是要考虑如何从目标空间入手, 首先确定是否存在冗余, 然后采用有效方法减

少冗余目标,最终得到一个非冗余目标集.

本文从决策者角度出发,采用最小二乘法将目标空间中每个目标函数拟合为多条直线段,然后两两比较各直线段的斜率,确定最冗余目标对,并将冗余目标从目标集中删除.此外,本文还提出一个新的评价机制,检验不同目标减少算法的性能.最后,从算法时间效率方面分析同类算法的优劣.

2 相关工作(Related work)

近年来,采用MOEA实现对高维目标的优化成为学者们研究的一个热点问题,冗余目标减少作为处理高维目标问题的方法被众多学者关注.比较有代表性的方法主要有3类:基于主分量分析的目标减少算法,基于最小子集覆盖的目标减少算法,以及基于特征选择的目标减少算法.

2.1 基于主分量分析的目标减少算法(Objective-reduction using principal component analysis)

Deb和Saxena提出了基于主分量分析(principal component analysis, PCA)的目标减少算法^[4~6].在执行PCA分析主分量之前,首先将目标空间标准化,并求出它的协方差矩阵与相关系数矩阵 R ,然后求解 R 的特征值及其对应的特征向量.最后将特征值按从大到小排序,对应的特征向量分别为第一主分量、第二主分量,等等.然后分析各分量以减少冗余目标.该算法一次运行的时间复杂度为 $O(ms^2 + s^3)$,其中: m 为种群中非支配解的个数, s 为目标个数.

2.2 基于最小子集覆盖的目标减少算法(Objective-reduction using minimum objective-subset)

Brockhoff和Zitzler详细地介绍了冗余目标的概念,如果在删除某些目标前后,个体之间的Pareto支配关系不发生改变,则删除的目标为冗余目标.提出了一系列减少冗余目标的方法^[7~10],并从理论上证明了目标减少算法的可行性.讨论了精确算法因时间复杂度过高,不宜用于求解高维问题,提出了利用子集覆盖原理实现目标减少的近似算法,所求得的结果与精确解差异不大. Brockhoff等将目标减少算法分为两类:第1类是通过用户参数得到非冗余目标集(δ -minimum objective-subset, δ -MOSS),复杂度为 $O(\min\{m^2s^3, m^4s^2\})$;第2类是在第1类的基础上加入满足最终非冗余目标个数 k 这个条件的目标减少算法(minimum objective-subset of size k with minimum error, k -EMOSS),时间复杂度为 $O(m^2s^3)$.

2.3 基于特征选择的目标减少算法(Objective-reduction using feature selection technique)

Jaimes和Coello提出了一种基于特征选择的目标减少算法^[11].如果两个目标是负相关(相关系数

为负值),则这两个目标函数存在非冗余关系.算法分为3步:首先按相关系数的大小以每个目标为中心,另外的 q 个目标为邻居,划分为 s 个(s 为目标个数)邻居群;然后选择最拥挤的邻居群,最后保留该邻居群的中心并删除其他目标.算法的时间复杂度为 $O(ms^2)$. Jaimes等采用传统的评价方法(inverted generational distance, IGD),说明了他们的算法与同类算法^[4,7]在冗余目标减少方面有相同的效果.

3 基于最小二乘法的目标减少算法(Objective-reduction using least square method)

根据多目标优化问题的定义,各个子问题之间是相互冲突的.对于Pareto最优,一个目标函数值的增长意味着其他至少一个目标函数值的下降.而在实际的优化问题中,因目标数目多,直觉上很难判断目标之间的冲突性和冲突程度.最小二乘法可以方便地将目标空间中的点拟合成若干条直线段,这样形成的一条曲线可以近似地代表一个目标函数的曲线.对于两个目标函数的近似曲线,如果这两个目标是相互冲突的,则在相同区域内其直线段的斜率值完全不同;相反,若在相同区域内其直线段具有相同或比较接近的斜率,则认为这两个目标函数是完全一致或基本一致的,或者说这两个目标是冗余目标对,应用MOEA优化时,可以去掉其中一个目标,从而降低优化的难度.

如图1所示,有3条目标函数曲线, F_1 和 F_2 在目标空间中的分布或趋向十分接近,而 F_1 和 F_3 (或者 F_2 和 F_3)在目标空间中的分布或趋势差异较大.直观上可以判断 F_1 和 F_2 是一对冗余目标.在具体实现时,将每个目标拟合为 l 条直线,然后比较各斜率差,如果存在冗余目标,则确定一个最冗余目标对,然后再删除冗余目标.

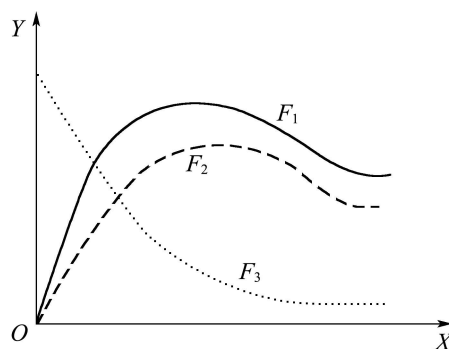


图1 目标空间冗余目标情况比较

Fig. 1 Comparison of redundancy objectives in objective space

为了便于应用最小二乘法拟合目标曲线,在此将目标空间存于矩阵 $Y = [Y_1^T \ Y_2^T \ \dots \ Y_s^T]^T$ 中,表示第 i 个目标值的向量,其长度为种群的规模 n ; s 为目

标的个数. 根据最小二乘法原理得到求斜率公式为

$$k_i = \frac{n \sum_{j=1}^n x_j y_{ij} - \sum_{j=1}^n x_j \sum_{j=1}^n y_{ij}}{n \sum_{j=1}^n x_j^2 - (\sum_{j=1}^n x_j)^2}. \quad (1)$$

式(1)中: x_j 为(0,1)区间等分为 n 段构成的向量, 即 $x = (1/n, 2/n, \dots, j/n, \dots, 1)$; y_{ij} 为种群中第 j 个个体在第 i 个目标上的函数值, $i = 1, 2, \dots, s$; $k_i = (k_{i1}, k_{i2}, \dots, k_{il})$ 为一个向量, 其长度 l 为第 i 个目标函数拟合成直线段的条数. 下面讨论算法的实现过程:

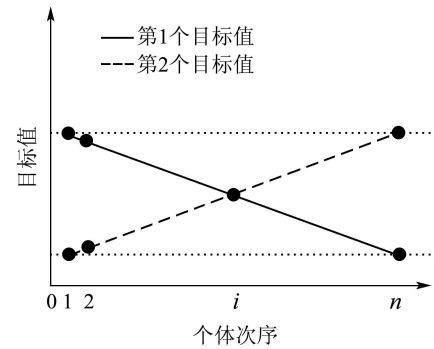
1) 将矩阵 Y (目标空间)标准化. 对矩阵 Y 按公式 $\tilde{y}_{ij} = \frac{y_{ij} - \bar{y}_i}{\sqrt{S_i}}$ 标准化, 式中 $\bar{y}_i$ 为第 i 目标的平均值, $\tilde{y}_{ij}$ 为矩阵 Y 标准化后的元素, y_{ij} 为矩阵 Y 标准化前的元素, S_i 为方差, $i = 1, 2, \dots, s$, $j = 1, 2, \dots, n$. 经标准化后, 向量 y_i 集中到一个小的区间, 并且若存在两个向量 y_i 和 y'_i , 如果它们是线性相关的, 即 $y'_i = ay_i$, 经标准化后, 它们的斜率值相同, 两者之差为0, y_i 和 y'_i 是一对冗余目标.

2) 将决策空间“排序”. 从决策者角度来说, 目标空间值排列的顺序并不会影响他们的判断, 他们只关心目标值在空间的分布趋势. 由于进化算法是一种随机算法, 因此, 在种群中个体的出现顺序也是随机的. 但在本文的算法中, 数据拟合与个体的排列方式相关. 比如, 在一次算法运行过程中, 得到 n 个个体, 它们出现的顺序为 $(i_1, i_2, \dots, i_n)$, 在第 i 个目标上对应的值为 $M_i = (m_{i,1}, m_{i,2}, \dots, m_{i,n})$. 另一次运行过程中, 得到一样的 n 个个体, 但是它们出现的顺序变为 $(i_n, i_{n-1}, \dots, i_1)$, 这样在第 i 个目标上对应的值变为 $M'_i = (m_{i,n}, m_{i,n-1}, \dots, m_{i,1})$. 两次运行后, 如果分别运用最小二乘法对 M_i 和 M'_i 进行拟合, 虽然是同一个目标函数, 但将得到两个不同的斜率值, 出现了矛盾的结果. 为了避免矛盾结果的出现, 可以调整 M_i 或者 M'_i 的排列顺序, 使两者一致, 这样就可以消除拟合过程由于成员出现的顺序带来的影响.

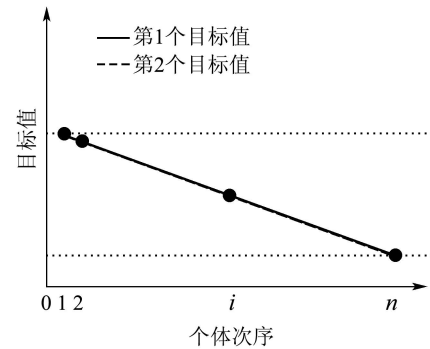
对个体排序时, 只能对决策空间进行排序. 如果目标空间排序, 将会破坏决策空间与目标空间的对应关系, 导致错误. 例如, 种群中 n 个个体对应的两个目标值分别为 $((m_{1,1}, m_{2,1}), (m_{1,2}, m_{2,2}), \dots, (m_{1,n}, m_{2,n}))$, 如果 $m_{1,j} = m_{2,n+1-j}$, $j = 1, 2, \dots, n$, 则对每个目标函数分别排序后, 虽然可以消除个体出现顺序带来的影响, 但经拟合后得到的两个斜率值一致, 如图2所示. 而事实上, 这两个目标在真实的目标空间中的分布是不相同的.

对决策空间排序时, 如果决策空间的维数是1, 排序操作十分简单, 一般的排序方法都可以实现. 但对

高维情况则比较复杂, 对任一维排序都不能全面地反映整个高维决策空间, 本文采用一种近似方法, 具体过程如算法1所示.



(a) 目标空间排序前



(b) 目标空间排序后

图2 将目标空间排序

Fig. 2 Sorting the objective space

算法1 决策空间排序算法.

Step 1 将种群中个体的决策向量(自变量) $xvar_i$ ($1 \leq i \leq n$)的每个成员 $xvar_{ij}$ ($1 \leq j \leq nvar$)转换到[0, 1]区间, 得到向量 $tvar_i$ ($1 \leq i \leq n$), $nvar_i$ 为自变量的维数.

Step 2 提取向量 $tvar_i$ 所有成员 $tvar_{ij}$ 的最高两位的数字字符分别存入 $tvar_f_{ij}$ 和 $tvar_s_{ij}$.

Step 3 以 $tvar_f_i$ 为第1顺序, $tvar_s_i$ 为第2顺序对个体排序.

准确地说, 算法1是一个固定目标值排列次序的方法. 按算法1对决策空间“排序”后, 目标空间中个体的排列顺序就可以确定下来, 确定了每次拟合的顺序, 将不会出现上述由于个体排列次序引起的差异. 同时, 这样的排序还有另外一个好处, 它将决策向量按近似的“大小”顺序排列, 使得决策空间相同或者相近的个体集中在相邻位置, 这样将增加拟合的准确性.

对决策空间排序后, 接下来的工作是: 如何确定最冗余的目标对, 以及如何从中选出冗余目标.

3) 选定最冗余目标对和确定冗余目标. 一般地, 目标函数拟合后不是一条直线段, 而是由 l 条直线段组成的曲线, 如图3所示($l = 5$). 图3中曲线为第 i 个目标函数经拟合后的分布. 每段的斜率值 k_{ij} ($j = 1, 2, \dots, 5$)组成一个向量 k_i ($i = 1, 2, \dots, s$).

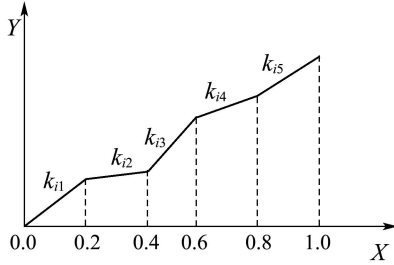


图3 将目标空间拟合为5段图

Fig. 3 Fitting the objective space to 5 strip-line

为了判断两个目标函数是否为冗余目标对, 在此引入参数 δ , δ 为用户根据经验和实际需求设定的比较小的数值. 如果两个目标曲线在相同区域中的斜率之差的绝对值为零, 则表明这两个目标之间没有冲突; 如果两个目标曲线在相同区域中的斜率之差的绝对值小于 δ , 则表明这两个目标之间的冲突程度很小, 可以忽略不计, 在高维问题目标减少过程中认定为没有冲突.

对于第 i 个目标和第 j 个目标, 如果

$$|k_{it} - k_{jt}| < \delta, \forall t \in \{1, 2, \dots, l\}, 1 \leq i \neq j \leq s, \quad (2)$$

则表明第 i 个目标和第 j 个目标是冗余目标对. 这样的冗余目标对可能存在多组, 将满足下列条件的冗余目标对 p 和 q 选定为最冗余目标对, 即

$$\min \left\{ \sum_{t=1}^l |k_{pt} - k_{qt}|, |k_{pt} - k_{qt}| < \delta, 1 \leq p \neq q \leq s \right\}. \quad (3)$$

在 p 和 q 中, 将它与其他目标冲突最小的选为最冗余目标, 设为 r , 即

$$r = \min \left\{ p : \sum_{i=1}^s \sum_{j=1}^l |k_{pj} - k_{ij}|, q : \sum_{i=1}^s \sum_{j=1}^l |k_{qj} - k_{ij}| \right\}. \quad (4)$$

4) 每次减少一个冗余目标. 算法限定每次只减少一个冗余目标, 这点和以往的算法有所不同, 以往的算法在一次运行后已经找出了非冗余目标集(当然Deb的算法可能要运行多次). 这样, 决策者可以更自由地选择非冗余目标集. 每次减少目标集中最冗余的目标, 因此非冗余目标集中元素个数以递减方式(每次减少一个目标)减小, 须说明一点, 目标不能减少到两个以下, 这样将不再构成Pareto支配关系.

这种操作另外还有一个好处, 由于每次只减少一个目标, 然后重复执行进化操作, 有利于防止有效信息丢失. 在前面引言部分已经介绍了MOEA不适合高维多目标问题(往往目标减小算法处理的原始目标就是高维的). 如果仅仅通过一次运行得到的结果去处理高维目标空间, 势必有许多不准确之处. 目标减少过程肯定是一个信息丢失的过程, 这里并不是一次性减少所有的冗余目标, 反复迭代有利于保持更多的有用信息, 使冗余信息反映得更加准确.

5) 非冗余目标判断. 有些测试函数, 其本身不存在冗余目标, 或者非冗余集中无法再减少目标, 这里引入一个判断机制. 在3)中, 如果不能选出冗余目标对, 则测试函数不存在冗余目标或者不再存在冗余目标.

基于最小二乘法的目标减少算法(objective-reduction using the least square method, ORLSM)如算法2所示.

算法2 基于最小二乘法的目标减少算法.

Step 1 初始化 $t = 0$, $R_t = \{1, 1, \dots, 1\}$ 以及冗余门限参数 δ .

Step 2 按 $R_t[i]$ 计算目标空间, 运行一次完整的NSGA-II并得到最终种群 P_t , 令 Y 为 P_t 的目标空间.

Step 3 将 Y 标准化, 并且将 Y_i 按决策空间排序后的顺序排序.

Step 4 将 Y_i 分为 l 段, 分别按最小二乘法拟合为直线得到斜率向量 k_i .

Step 5 如果存在冗余目标, 确定最冗余目标对, 找出最冗余目标记为 ri ; 否则转Step 7.

Step 6 $t++$, $R_t[ri] = 0$. 转Step 2.

Step 7 结束.

算法2中 R_t 为冗余目标标志向量, 其长度为目标个数 s , 为0时, 对应的目标被视为冗余.

在Step 1中, R_t 的所有成员为1, 即所有目标都是非冗余的, 并初始化参数 δ . Step 2中, 目标空间只包含 $R_t[i] = 1$ 的目标函数值. 完整地运行一次NSGA-II^[12]得到 R_t 对应的种群 P_t , 记为 Y . Step 6中, 说明已找出一个冗余目标, 并将标记删除, 返回Step 2重新开始冗余目标判定.

6) 时间复杂度分析. 在时间复杂度方面, 将 Y 标准化为 $O(m)$, 决策空间排序为 $O(m \times nvar)$, 将单个目标拟合为 l 段时间复杂度为 $O(l \times m)$, 总共有 s 个目标, 时间复杂度为 $O(l \times m \times s)$, 计算斜率向量的平均值和确定最冗余目标对都为 $O(l \times s^2)$. 因此, 在算法2中, 时间复杂度为 $O(\max\{s \times l \times m, l \times s^2\})$. 这是一次运行过程的时间复杂度, 由于目标个数在每次运行过程后都减少, 即 s 递减, 相应的时间也减少.

值得说明的是, 目标减少过程中非冗余目标个数不能少于两个, 所以算法结束有两种情况: 一是满足用户设定的参数要求, 目标集中不再存在冗余目标; 另一个是目标集中只剩下两个目标, 再减少将不再构成Pareto支配关系。

4 实验结果与分析(Experiment result and analysis)

为了测试ORLSM的性能, 在此选取两个同类算法进行实验比较. 一个是由Brockhoff和Zitzler提出的近似算法 k -Moss(命名为BZ's); 另外一个Jaimes和Coello提出的基于特征选择的算法(命名为JC's). 通过3个测试函数DTLZ5, DTLZ7和DTLZ2BZ测试3个算法的性能. 在这3个测试函数中, n 维的DTLZ5测试函数通过降维, 能收敛到一条曲线, 它的形状与二维的一致; n 维的DTLZ7测试函数, 前 $n-1$ 维的取值范围与表达形式都一致. 这两个测试函数存在冗余目标^[13]. DTLZ2BZ是对DTLZ2的修改^[9], 在DTLZ2中, 当减少一个目标后, 其他目标值会集中到一个解; 而对DTLZ2BZ, 即使减少多个目标, 剩余的目标子集仍可以保持原来的分布特征和优化效果。

下面采用两种不同的评价方法, 对以上3个算法进行性能比较. 实验参数设置为: 种群大小为200, 种群进化400代, 每个算法独立运行10次, 结果为10次运行的平均值, $\delta = 0.3$.

4.1 IGD评价方法(IGD metric)

IGD(inverted generational distance)评价方法主要用于测试算法所求得最终Pareto面到真实Pareto面的收敛程度^[11], 其定义为

$$IGD = \sqrt{\sum_{i=1}^n d_i^2 / n}. \quad (5)$$

n 为真实Pareto面上点的个数, d_i 为真实Pareto面上的解到最终Pareto解的最小的欧氏距离. IGD值越小越好。

表1为DTLZ5(3, M)问题^[6]的IGD评价结果, 通过参数控制, 3个算法将从 M 个目标减少到最终3个非冗余目标, ORLSM将目标空间拟合为5段直线。

从表1可知, 对6个测试项, ORLSM明显地优于其他两个算法. BZ's与JC's的测试结果相差不大, BZ's对目标数分别为5, 10, 20和30共4项比JC's好, 而对目标数为15和25这2项比JC's差。

表2列示了3个算法, 对DTLZ5原始目标为5个和10个两种, 分别减少到最终的非冗余目标是两个或者3个. 表2中前两行为5个目标的情形, 分别减少到最终两个或者3个目标, “*”后面对应的数字为目标编号, 如*5表示目标空间中第5个子目标函数. 同样, 表2中的后两行是10个目标的情况。

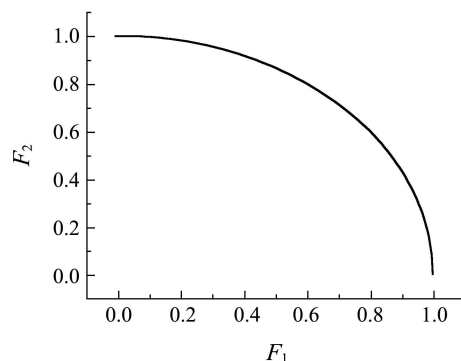
表1 DTLZ5的IGD平均值
Table 1 DTLZ5 IGD average value

目标个数	JC's	BZ's	ORLSM(5段)
5	0.273	0.144	0.0216
10	0.561	0.323	0.0166
15	0.191	0.385	0.0218
20	0.306	0.3	0.0152
25	0.195	0.507	0.0165
30	0.379	0.313	0.0177

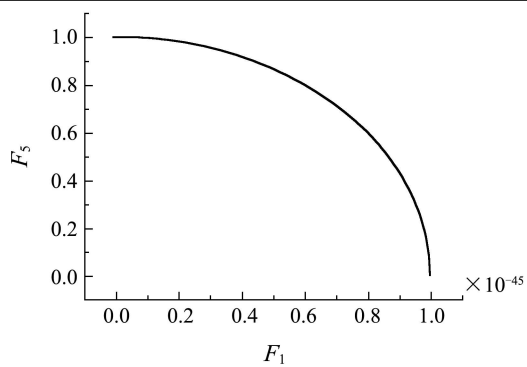
表2 DTLZ5的非冗余目标集
Table 2 DTLZ5 non-redundancy objectives set

目标数	JC's和BZ's	图示结果	ORLSM(5段)	图示结果
5	*1, *5	图4 (b)	*4, *5	图4 (c)
5	*1, *4, *5	图4 (g)	*3, *4, *5	图4 (h)
10	*1, *10	图4 (d)	*9, *10	图4 (e)
10	*1, *9, *10	图4 (i)	*8, *9, *10	图4 (j)

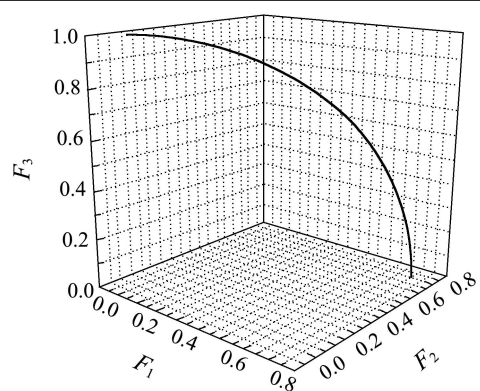
对照表2和图4所示结果, 可以直观地观察3个算法得到的非冗余目标集及其分布特征. 先分析目标减少到二维目标情况, 对应表2的第1行和第3行, 如果图4(a)~(e)所示. 图4(a)为原始目标集是二维时的真实Pareto面的分布; 图4(b)为由JC's和BZ's两个算法得到的非冗余目标集的最终种群的分布, 图4(c)为ORLSM得到的结果. 从图4(a)~(c)这3个图中可以看出: 1) DTLZ5测试函数存在冗余, 从5维目标可以减少到二维, 目标减少后的Pareto面基本保持不变; 2) ORLSM比另外两个算法得到的非冗余目标集具有更好的分布特征, 在图4(b)中, 横坐标(F_1)的取值范围基本上在0附近, 而对照图4(a)可知, DTLZ5原始的二维目标函数值分布在区间[0,1], 图4(c)的分布区间比较接近[0,1], 因此图4(c)比图4(b)具有更好的分布特征. 对于从10维目标减少到二维目标的情况, 从图4(d)和图4(e)可以得到与上述5个目标一致的结论. 接下来分析目标减少到3个的情况, 图4(f)是原始3维目标真实Pareto面的分布. 比较图4(g)和图4(h)以及图4(i)和图4(j), 可以看出, ORLSM比JC's和BZ's两个算法具有更好的分布特征, 与二维情况下的结论一致。



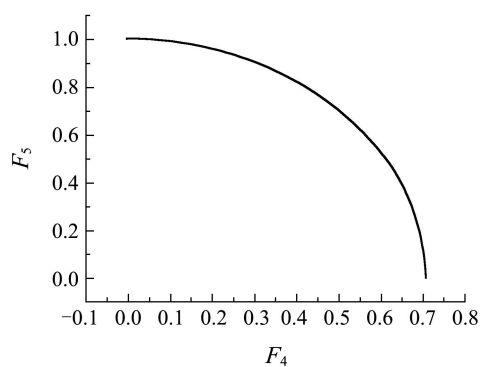
(a) 二维原始目标集



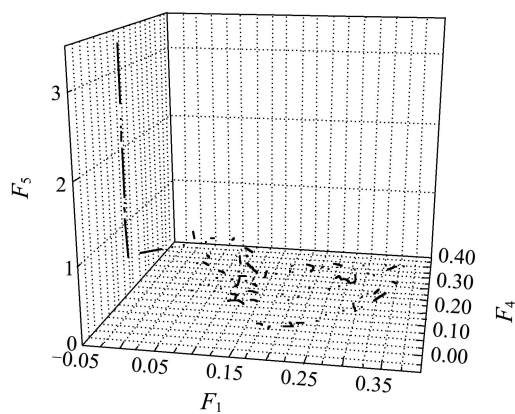
(b) JC's和BZ's



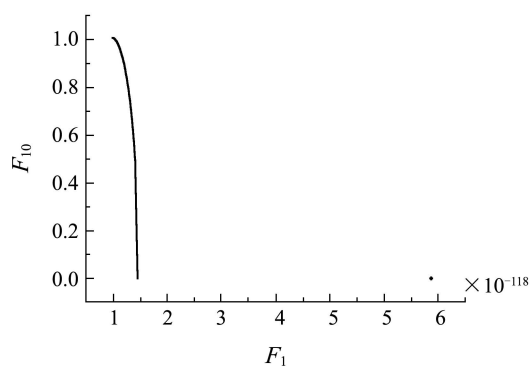
(f) 三维原始目标集



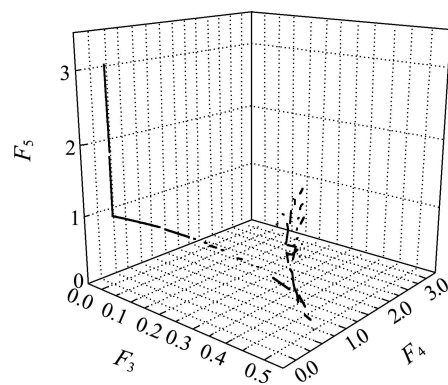
(c) ORLSM



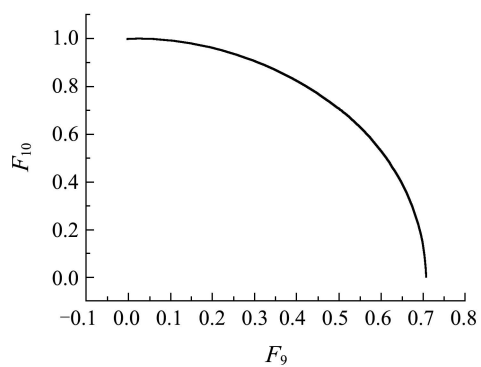
(g) JC's和BZ's



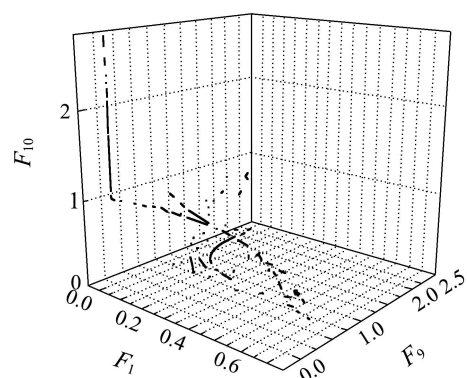
(d) JC's和BZ's



(h) ORLSM



(e) ORLSM



(i) JC's和BZ's

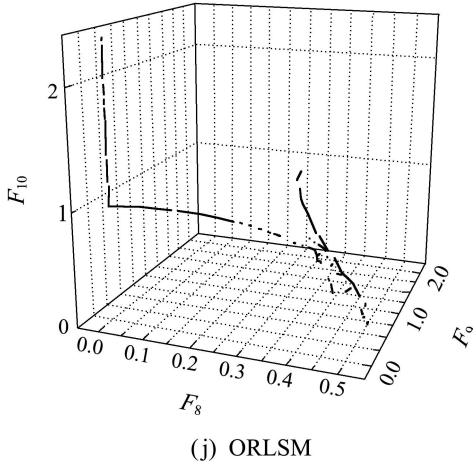


图 4 非冗余目标空间分布

Fig. 4 Non-redundancy objectives-distribution

在时间复杂度方面, 3个算法的运算时间如图5所示. 容易发现, JC's在运算时间上比另外两个算法占优, ORLSM与BZ's在运算时间方面基本相同. 当然, ORLSM由于每减少一个目标后都要输出目标空间与自变量空间等大量的信息, 占用了过多的时间, 如果根据用户的需要只输出某一个非冗余目标集信息, 总的运算时间将会减少. 另外反复迭代, 多次运行NSGA-II占据了绝大部分的运行时间.

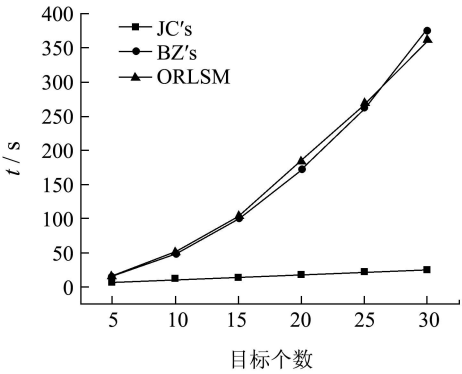


图 5 求解DTLZ5时各算法的运算时间

Fig. 5 DTLZ5 computation cost of every algorithm

对于测试函数DTLZ7, 表3列示了对DTLZ7(3, M), 3个算法的IGD评价结果.

表 3 DTLZ7的IGD平均值

Table 3 DTLZ7 IGD average value

目标个数	JC's	BZ's	ORLSM(5段)
5	0.558	0.45	0.466
10	17.63	17.5	17.57
15	40.45	40.52	39.52
20	57.78	57.47	57.27
25	73.17	72.93	72.97
30	87.16	85	84.16

从表3可知, BZ's和ORLSM比JC's的分布特征略微好一些; BZ's和ORLSM在6个测试方案中各占一半的优势. 总体来说, 对于DTLZ7, 3个算法的分布特征差异不大. 从图6可知, 与DTLZ5类似, JC's的运行时间占优, 其他的两个算法基本相同.

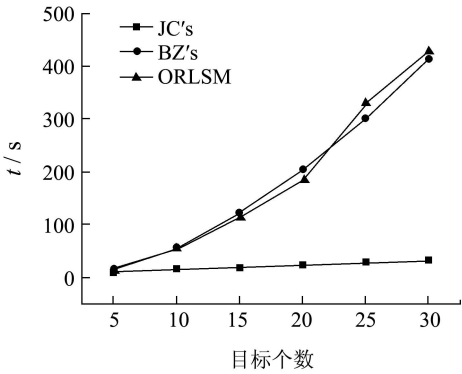


图 6 求解DTLZ7时各算法的运算时间

Fig. 6 DTLZ7 computation cost of every algorithm

表4列示了测试函数DTLZ2BZ的IGD平均值. 从表4可知, ORLSM和JC's比较接近, 它们在6项测试用例中各占一半优势, BZ's的结果比较起来差一些.

在运算时间方面, JC's最小, 其次是ORLSM, BZ's最大, 如图7所示.

表 4 DTLZ2BZ的IGD平均值

Table 4 DTLZ2BZ IGD average value

目标个数	JC's	BZ's	ORLSM(5段)
5	0.168	0.375	0.178
10	0.233	0.498	0.283
15	0.391	0.364	0.328
20	0.352	0.425	0.37
25	0.383	0.395	0.355
30	0.6	0.472	0.371

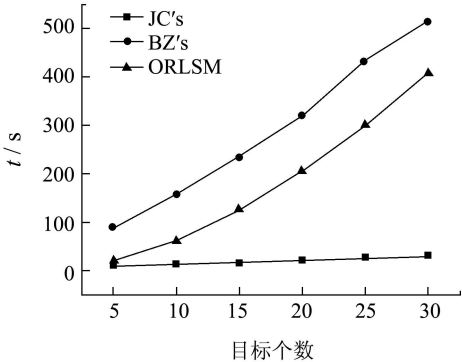


图 7 求解DTLZ2BZ时各算法的运算时间

Fig. 7 DTLZ2BZ computation cost of every algorithm

4.2 支配关系改变率评价方法(Changed Pareto domination ratio metric)

用IGD评价目标减少算法的效果时存在着3个方面的不足:

1) IGD评价方法的出发点是评价算法的收敛性能,它并不能准确地反映目标减少算法的优劣程度.

2) IGD评价方法必须先确定真实Pareto面的表达式,并求出Pareto面均匀的分点.这对某些测试函数或者实际优化问题来说是比较困难的.

3) 在目标维数比较高的情况下,真实Pareto面上的点的个数太多,规模大,计算量大,效率会比较低.

为此,本文提出一种基于支配关系的评价方法,即目标减少前后 Pareto 支配关系改变的比率(changed Pareto domination ratio, CDR).

一种理想的情况是,如果一个目标减少方法使用后,没有改变优化问题在目标减少前的个体之间的支配关系,那么这个方法是最好的,即要求所减少的目标是完全没有冲突的.然而,实际情况通常是,需要减少完全没有冲突或冲突比较小的目标.因此,一般来说,目标减少前后,个体的支配关系可能会有一些变化.

目标减少前后Pareto支配关系改变的比率,就是在使用目标减少算法之后, Pareto支配关系改变的总次数在群体规模中的比率, CDR的计算公式定义如下:

$$CDR = (\sum_{i=1}^n \sum_{j=1, j \neq i}^n f(x_i, x_j)) / n^2. \quad (6)$$

式(6)中: n 为种群大小,若目标减少前后 x_i 和 x_j 的支配关系发生了改变, $f(x_i, x_j)$ 的值为1; 否则,为0. CDR得到的是一个百分比,它的值域为 $[0, 1]$,该值越小越好.如果为0,说明减少的目标完全不会影响Pareto支配关系.

表5列示了CDR测试DTLZ5(3,M)的平均值.从表5可知,ORLSM比JC's和BZ's两个算法在DTLZ5(3,M)上具有明显的优势,它在6个测试项中都明显优于另外两个算法.

表 5 DTLZ5的CDR平均值
Table 5 DTLZ5 CDR average value

目标个数	JC's	BZ's	ORLSM(5段)
5	0.233	0.0478	0.0022
10	0.261	0.138	0.00047
15	0.0374	0.106	0.0261
20	0.118	0.121	0
25	0.0546	0.349	0.00551
30	0.143	0.111	0.00012

表6列示了测试函数DTLZ7(3,M)评价结果.从

表6可知, BZ's比另外两个算法略占优势,占6个测试项的一半, ORLSM和JC's两个算法基本上差不多,且各占2项优势.

表 6 DTLZ7的CDR平均值
Table 6 DTLZ7 CDR average value

目标个数	JC's	BZ's	ORLSM(5段)
5	0.103	0.127	0.113
10	0.178	0.186	0.176
15	0.197	0.199	0.197
20	0.248	0.195	0.202
25	0.204	0.143	0.204
30	0.216	0.187	0.215

表7列出了测试函数DTLZ2BZ的CDR平均值.从表7可知, ORLSM在6项测试中4项占优, BZ's的CDR值与它比较接近, JC's与BZ's各有1项占优.总的来说, ORLSM在测试函数DTLZ2BZ上比JC's和BZ's两个算法更有优势.

表 7 DTLZ2BZ的CDR平均值
Table 7 DTLZ2BZ CDR average value

目标个数	JC's	BZ's	ORLSM(5段)
5	0.0972	0.0333	0.0709
10	0.156	0.0877	0.0852
15	0.118	0.0922	0.0773
20	0.106	0.0824	0.0644
25	0.0589	0.154	0.117
30	0.079	0.0917	0.0673

本节使用CDR评价方法,分别针对3个测试函数DTLZ5, DTLZ7和DTLZ2BZ,对ORLSM, JC's和BZ's 3个算法进行了性能测试,测试结果与IGD方法基本一致. IGD方法侧重于评价算法的收敛性能,而CDR侧重于评价目标减少前后支配关系改变的比率, CDR更适合用于评价目标减少算法;此外, CDR不需要真实Pareto面,它的计算机量比较小,使用将会更广泛.

5 结论(Conclusion)

本文从决策者的角度考虑目标冗余问题,提出了一种基于最小二乘法的目标减少算法(ORLSM),该方法将各子目标函数分别拟合为若干条直线段,通过比较直线段之间的斜率之差确定最冗余目标对,进而选定冗余目标;同时针对目标减少前后个体支配关系的变化情况,提出了一种基于支配关系改变率的评价方法(CDR).通过3个测试函数DTLZ5, DTLZ7和DTLZ2BZ,采用两种不同的评价方法IGD和CDR,分别从收敛性和支配关系改变率2项指标对ORLSM, JC's和BZ's 3个算法进行了性能测试;同时,对3个算法的运行效率进行对比测试.结果表明,

ORLSM在总体上具有最好的性能, CDR和IGD评价结果基本一致。

参考文献(References):

- [1] COELLO C A C, VANVELDHUIZEN D A, LAMONT G. *Evolutionary Algorithms for Solving Multi-objective Problems*[M]. Boston, MA: Kluwer Academic Publishers, 2002.
- [2] DI PIERO F. *Many objectives evolutionary algorithms and applications to water resources engineering*[D]. Exeter university, UK: Computer Science and Mathematics, 2006.
- [3] FLEMING P J, PURSHOUSE R C, LYGOE R J. Many objective optimization: an engineering design perspective[C] // *Evolutionary Multi-criterion Optimization*. Berlin: Springer-Verlag, 2005, 3410: 14 – 32.
- [4] SAXENA D K, DEB K. Dimensionality reduction of objectives and constraints in multiobjective optimization problems, a system design perspective[C] // *2008 IEEE World Congress on Computational Intelligence*. New York: IEEE, 2008: 3204 – 3211.
- [5] DEB K, SAXENA D. Search for pareto optimal solutions through dimensionality reduction for certain large dimensional multi objective optimization problems[C] // *Proceeding of the IEEE Congress on Evolutionary Computation(CEC2006)*. Los Alamitos: IEEE Computer Society Press, 2006: 3352 – 3360.
- [6] SAXENA D K, DEB K. Non-line dimensionality reduction procedures for certain large dimensional multiobjective optimization problem: employing correntropy and a novel maximum variance unfolding[M] // ODAYASHI S, et al. *Lecture Notes in Computer Science: Evolutionary Multi Criterion Optimization*. Berlin: Springer-Verlag, 2007, 4403: 272 – 787.
- [7] BROCKHOFF D, ZITZLER E. Dimensionality reduction in multiobjective optimization, the minimum objective subject problem[M] // *Operations Research Proceedings*. Berlin: Springer, 2007: 423 – 429.
- [8] BROCKHOFF D, ZITZLER E. Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization[M] // *Parallel Problem Solving from Nature*. Berlin: Springer, 2006: 533 – 542.
- [9] BROCKHOFF D, ZITZLER E. Objective reduction in evolutionary multiobjective optimization: theory and applications[J]. *Evolutionary Computation*, 2009, 17(2): 135 – 166.
- [10] BROCKHOFF D, ZITZLER E. Improving hypervolume based multi-objective evolutionary algorithms by using objective reduction methods[C] // *IEEE Congress on Evolutionary Computation*. New York: IEEE, 2007: 2086 – 2093.
- [11] JAINES A L, COELLO C A C, CHAKRABORTY D, et al. Objective reduction using a feature selection technique[C] // *Proceedings of the Genetic and Evolutionary Computation Conference(GECCO 08)*. Atlanta: ACM Press, 2008: 673 – 680.
- [12] K DEB, S AGRAWAL, A PRATAP, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182 – 197.
- [13] DEB K, LOTHAR T, MARCO L, et al. Scalable multiobjective optimization test problems[C] // *Congress on Evolutionary Computation(CEC02)*. Piscataway, NJ: IEEE, 2002, 1: 25 – 830.

作者简介:

郑金华 (1963—), 男, 教授, 博士生导师, 从事进化计算、并行处理的研究, E-mail: jhzheng@xtu.edu.cn;

周 聪 (1984—), 男, 硕士研究生, 从事进化计算的研究, E-mail: cleverchou23@yahoo.com.cn;

李 珂 (1985—), 男, 硕士研究生, 从事进化计算的研究, E-mail: Jerryi00@yahoo.com.cn;

吕 卉 (1983—), 女, 硕士研究生, 从事进化计算的研究, E-mail: lvyishan@126.com.