

基于正交设计模型的多目标进化算法

吴金妹¹ 王亚辉¹ 贾晨辉²

(1. 华北水利水电大学机械学院, 郑州 450011; 2. 河南科技大学机电工程学院, 洛阳 471023)

摘要: 为提高多目标进化算法在求解复杂多目标问题上的收敛性和解集多样性, 提出了一种基于正交设计模型的多目标进化算法。该算法在基于分解技术的多目标进化算法框架下, 将正交实验设计方法同分解技术相融合。利用正交实验设计方法, 有针对性地对父代个体进行重组, 并生成多个保留优良基因的子代个体, 避免了盲目性搜索以提高算法收敛性, 并应用分解技术选择优秀个体来维持全局搜索和局部寻优的动态平衡。将该算法与目前典型的优秀算法在 18 个标准测试函数集上进行对比测试, 仿真结果表明所提算法相比另外 4 种算法具有良好的竞争力, 在保持良好收敛性的同时, 所获得的 Pareto 前端分布更加均匀, 尤其在求解具有复杂 Pareto 解集的问题时, 能保持较好的搜索性能。为了测试算法在求解含有约束问题的性能, 将其应用于 I 型主梁多目标优化设计中, 获得的 Pareto 前沿较均匀, 且解集域较宽广, 对比分析表明了算法的工程实用性。

关键词: 多目标进化算法; MOEA/D; 正交设计模型; I 型主梁设计

中图分类号: TP18 **文献标识码:** A **文章编号:** 1000-1298(2017)02-0362-08

Multi-objective Evolutionary Algorithm Based on Orthogonal Designing Model

WU Jinmei¹ WANG Yahui¹ JIA Chenhui²

(1. College of Mechanical Engineering, North China University of Water Resources and Electric, Zhengzhou 450011, China

2. College of Mechanical Engineering, Henan University of Science and Technology, Luoyang 471023, China)

Abstract: Aiming to improve the convergence and diversity of multi-objective evolutionary algorithms (MOEAs) for solving complicated high dimensional multi-objective optimization problems, a multi-objective evolutionary algorithm based on orthogonal designing model (MOEA/D-OD) was proposed. Under the framework of multi-objective evolutionary algorithm with decomposition scheme as typical characteristics, the orthogonal designing model (ODM) was incorporated into decomposition mechanism. By utilizing ODM, the good genes carried by the recombinant parents were obtained by offspring to avoid blindness of searching to improve the convergence of the proposed algorithm. The decomposition mechanism was applied to selection to balance exploitation and exploration. MOEA/D-OD was compared with four state-of-the-art MOEAs on 18 benchmark testing problems. Experimental results indicated that MOEA/D-OD can obtain good convergence while having uniform distribution and wild coverage for Pareto sets. The searching performance can stay well when solving complex problems with complicated PS. To validate its performance on constraint multi-objective optimization problems, the proposed MOEA/D-OD was applied to solve the I-beam with two conflict objectives. Compared with other algorithms, the uniformly distributed Pareto sets obtained by MOEA/D-OD showed its practicability for engineering problems, which was an effective approach for solving high dimensional and complicated multi-objective optimization problems.

Key words: multi-objective evolutionary algorithm; MOEA/D; orthogonal designing model; I-beam design

引言

多目标优化问题 (Multi-objective optimization

problem, MOP) 是一类广泛存在于科学研究与工程实践中的复杂优化问题, 其存在至少两个以上的优化目标, 且多个目标之间相互冲突, 决策过程中不存

收稿日期: 2016-06-07 修回日期: 2016-07-10

基金项目: 国家自然科学基金项目(51475142)和河南省高等学校重点科研项目(17A460020、17A460019)

作者简介: 吴金妹(1988—), 女, 讲师, 主要从事机械设计制造和系统优化设计研究, E-mail: wujinmei@ncwu.edu.cn

在满足所有优化目标的单一最优解,而是一组折衷的 Pareto 解集。进化算法是一类通过模拟自然界的进化过程来求解复杂优化问题的有效算法,由于其对所求问题的性质不敏感且采用基于群体方式的多点搜索策略,因此非常适合于求解多目标问题。

自 1985 年 Schaffer 提出第一个多目标进化算法^[1]起,在 30 多年的多目标优化研究过程中,主要致力于 2 个方面的探索:①收敛性研究,即算法获得的 PF(Pareto front)和 PS(Pareto set)尽可能的逼近真实 PF 和 PS 解集。②多样性研究,即算法求解的 PF 和 PS 尽可能的均匀分布。对于上述 2 方面问题,一般采用不同评价机制可以满足,而评价机制主要区分为 3 大类^[2-3]:①基于支配关系评价机制^[4-8],主要有 NSGA-II^[4]、SPEA2^[6]、NPGA^[7]、PESA^[8]等。②基于指标评价机制^[9-12],其中典型的性能指标函数为超体积 HV(Hypervolume)^[9],主要算法有 IBEA^[10]、SMSEMOA^[11]和 HyPE^[12]等。③基于分解技术的评价机制^[13-20],此类方法的代表算法有 C-MOGA^[13]、MOEA/D^[14]和 MOEA/D 框架下的优化算法^[15-20]等。

MOEA/D 算法是一类新型的进化算法框架,其主要思想为将一个多目标问题分解为若干个子问题进行求解,同时利用邻域策略来建立子问题之间的联系,从而获得对 Pareto 解集的逼近。

正交实验设计(Orthogonal experimental designing, DOE)^[21]是一种解决多因素、多水平的实验设计方法,能以较少的实验次数找到最好或较好的实验条件。文献[22-24]将 DOE 引入进化算法中,结果均表明了 DOE 对算法性能有一定的提升。基于此,本文在 MOEA/D 算法框架下,将一种正交设计模型(Orthogonal designing model)引入算法中,提出一种基于正交设计和分解技术的多目标进化算法(MOEA/D-OD)。最后,采用 18 个复杂基准测试函数,同其他 4 种具有代表性算法进行对比仿真测试。

1 多目标优化概况

一个具有 n 个决策变量, m 个目标函数的 MOP 的基本形式为

$$\begin{cases} \min \mathbf{y} = \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \\ \text{s. t. } \mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \Omega \end{cases} \quad (1)$$

其中, Ω 为可行域空间, $f_i(x) \rightarrow \mathbf{R} (i = 1, 2, \dots, m)$ 为第 i 个目标函数。由于多个目标之间相互冲突,因此一般情况下, MOP 的最优解不是一个解而是一个 Pareto 最优解集的集合。

定义 1(Pareto 支配): 假设 $x_p, x_q \in \Omega$, 若满足

式(2), 则称 x_p 支配 x_q (记为 $x_p > x_q$)。

$$\begin{cases} \forall i \in \{1, 2, \dots, m\} & (f_i(x_p) \leq f_i(x_q)) \\ \exists j = 1, 2, \dots, m & (f_j(x_p) < f_j(x_q)) \end{cases} \quad (2)$$

定义 2(Pareto 最优解): 一个解 $x^* \in X_f$ 被称为 Pareto 最优解, 当且仅当满足如下条件

$$\neg \exists \mathbf{x} \in X_f: \mathbf{x} > \mathbf{x}^* \quad (3)$$

即在可行解集当中, 若不存在任何一个可行解 x 相对于 x^* 是 Pareto 占优的, 则称 x^* 为 Pareto 最优解。

定义 3(Pareto 最优解集): 对于 MOP, 其 Pareto 最优解集(Pareto set, PS)是所有 Pareto 最优解的集合

$$PS = \{x^* \mid \neg \exists x \in X_f: x > x^*\} \quad (4)$$

定义 4(Pareto 前端): Pareto 最优解集 PS 在目标空间的投影集合称为 Pareto 前端(Pareto front, PF), 即

$$PF = \{F(x) \mid x \in PS\} \quad (5)$$

2 正交设计模型的多目标进化算法

2.1 MOEA/D 算法框架

MOEA/D^[14]算法将 MOP 分解为一组单目标优化问题, 每个单目标问题称为子问题; 各子问题之间相互协作同时优化这组子问题, 从而获得对 Pareto 最优解集的逼近。

(1) 分解技术

将一个 MOP 问题, 分解为一系列的子问题 $\text{ming}^i(x) (i = 1, 2, \dots, N)$, 每个子问题的最优解对应于 MOP 问题的一个 Pareto 最优解。其中, 每个子问题的目标值是各个目标分量的聚合函数, 每个子问题对应一个权重向量 $\lambda^i = (\lambda_1^i, \lambda_2^i, \dots, \lambda_m^i) (i = 1, 2, \dots, N)$, 通过权重向量来获取每个子问题的邻居, 每个子问题的优化被限定为只能通过与其邻域内的子问题来进行进化操作完成。以切比雪夫法应用最为广泛, 其分解机制为:

设种群规模为 N , 目标个数为 m , $\Lambda = \{\lambda^1, \lambda^2, \dots, \lambda^N\}$ 为一个权重向量集合, 其中 $\lambda^i = \{\lambda_1^i, \lambda_2^i, \dots, \lambda_m^i\}$ 且满足 $\sum_{j=1}^m \lambda_j^i = 1$, 则第 i 个子问题的表达式为

$$\begin{cases} g^{te}(x \mid \lambda^i, z^*) = \min_{1 \leq j \leq m} \{\lambda_j^i |f_j(x) - z_j^*|\} \\ \text{subject to } x \in \Omega \end{cases} \quad (6)$$

其中, $z^* = (z_1^*, z_2^*, \dots, z_m^*)^T$ 为最优参考点, 通常很难获取精确的最优参考点, 一般用 $z_i^* = \min\{f_i(x) \mid x \in \Omega\} (i = 1, 2, \dots, m)$ 来作为最优参考点的近似参考点。对于 MOP 中的每一个 Pareto 最优解 x^* , 一定存在一个对应的权重向量 λ , 使得 x^* 对应单目标问

题 $g^{te}(x|\lambda, z^*)$ 的一个最优解。反之,单目标问题 $g^{te}(x|\lambda, z^*)$ 的每个最优解也对应 MOP 的一个 Pareto 最优解。本文中 N 个权重向量是事先在权重向量空间中均匀选择的,具体算法见文献[14]。

(2) 邻居子问题

由于每个子问题对应一个权重向量,子问题的邻居是根据每个权重向量与其欧式距离最小的 T 个权重向量的编号对应的个体来确定的。MOEA/D 算法框架的重组算子的父代个体选择源于领域子,并且新个体会替换邻域中个体,从而使新个体尽可能地参与进化。

2.2 正交设计模型

(1) 正交实验设计

采用 LEUNG 等^[22]提出的 $L_M(Q^K)$ 构造方法,具体方法见表 1。 $L_M(Q^K)$ 正交表表示该实验系统具有 K 个因素,每个因素具有 Q 个水平, M 是构造的组合数量。

(2) 正交设计交叉算子

父代个体之间的重组以交换个体之间的优秀基因,同时引入更多的启发式信息对解空间进行有效的搜索,能够增强子代个体在解空间的采样范围,维持进化过程中种群的多样性,避免算法陷入局部收敛。遗传操作算子就是通过父代个体的组合生成潜在的优秀解,该过程可以被视为是实验组合。因此,正交设计来重新组合父代个体进而生成优秀的子代个体是合理的。本文采用正交实验设计在子代个体的样本空间内组合出具有代表性的子代个体,以提高搜索效率。

考虑任意 2 个具有实数编码的父代个体 $e = (e_1, e_2, \dots, e_D)$ 和 $g = (g_1, g_2, \dots, g_D)$,正交设计交叉的首要步骤是对这 2 个个体进行离散化得到 Q 个水平,具体方式为

$$l_{i,j} = \min(e_i, g_i) + \frac{j-1}{Q-1}(\max(e_i, g_i) - \min(e_i, g_i))$$

$$(i=1, 2, \dots, D; j=1, 2, \dots, Q) \quad (7)$$

由于维数 D 通常大于因素 K ,如果直接应用正交表,则构造出的正交表会比较庞大,同时对于不同的维数 D ,需要重新构造正交表。因此考虑将决策向量个体随机分成 K 个子向量

$$\begin{cases} \mathbf{H}_1 = (x_1, x_2, \dots, x_{t_1}) \\ \mathbf{H}_2 = (x_{t_1+1}, x_{t_1+2}, \dots, x_{t_2}) \\ \vdots \\ \mathbf{H}_K = (x_{t_{K-1}+1}, x_{t_{K-1}+2}, \dots, x_D) \end{cases} \quad (8)$$

其中, t_1, t_2, \dots, t_{K-1} 是随机生成的正整数且满足 $1 < t_1 < t_2 < \dots < t_{K-1} < D$ 。

在得到 K 个子向量后,将每个子向量 \mathbf{H}_i 视为一个因素,同时定义其 Q 个水平为

$$\begin{cases} \mathbf{L}_{i1} = (l_{t_{i-1}+1,1}, l_{t_{i-1}+2,1}, \dots, l_{t_i,1}) \\ \mathbf{L}_{i2} = (l_{t_{i-1}+1,2}, l_{t_{i-1}+2,2}, \dots, l_{t_i,2}) \\ \vdots \\ \mathbf{L}_{iQ} = (l_{t_{i-1}+1,Q}, l_{t_{i-1}+2,Q}, \dots, l_{t_i,Q}) \end{cases} \quad (9)$$

在 D 维的实数空间内,将每个子向量看作是正交实验设计中的一个因素,即 K 个因素,同时结合父代个体离散化得到 Q 个水平,这样就将对父代个体的重组操作转换为 K 个因素、 Q 个水平的正交实验问题。此时,就可以应用正交表 $L_M(Q^K)$ 来对其每个因素 H_i 以及因素对应的 Q 个水平组合生成 M 子代个体,其具体操作见以下算法。

(1) 将参与重组的父代个体进行离散化,将决策向量随机分成 K 个子向量 $\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_K$,定义因素 \mathbf{H}_i 的 Q 个水平为 $\mathbf{L}(\mathbf{H}_i) = \{\mathbf{L}_{i1}, \mathbf{L}_{i2}, \dots, \mathbf{L}_{iQ}\}$,具体操作方法如下:随机生成 t_1, t_2, \dots, t_{K-1} 的正整数且满足 $1 < t_1 < t_2 < \dots < t_{K-1} < D$,把父代个体分成 K 个片段,每个片段表示一个因素,其 K 个子向量生成如式(9)。根据 \mathbf{H}_i ,得到 Q 个水平,如式(9)。

(2) 构造正交表 $L_M(Q^K)$,根据正交表 $L_M(Q^K)$ 安排实验,产生 M 个子代个体。

(3) 根据 $g^{te}(x|\lambda^i, z^*) = \min_{1 \leq j \leq m} \{\lambda_j^i |f_j(x) - z_j^*|\}$ 计算每个个体的适应度,选择该值最小的个体作为最终的子代个体。

在得到子代个体中,需要利用选择算子从子代个体中选择优秀的个体参与到下一代的进化,多目标中常用基于 Pareto 方法或者辅以拥挤距离来评价个体的优劣。然而,由于得到的子代群体较小,子代个体大部分都是互不支配的并且拥挤距离的区分也不太明显,难以从子代群体中选择到优秀的个体。分解技术将一个多目标问题分解为若干子问题,每一个个体对应着一个具体的适应度,该适应度表征了该个体与其最优参考点的接近程度,该值越小则该个体与最优参考个体越接近,因此该个体就越优秀。根据式(6)计算每个子代个体的适应度,选择出 $g^{te}(x|\lambda^i, z^*)$ 值最小的子代个体作为最终的子代个体参与到种群的后续进化中。

为了能够方便地说明问题,此处给出一个简单的实例。首先将其随机分割为 Q 个水平,设随机生成的 $t_1 = 2, t_2 = 4, t_3 = 5$,则其 K 个因素为

$$\begin{cases} \mathbf{H}_1 = (x_1, x_2) \\ \mathbf{H}_2 = (x_3, x_4) \\ \mathbf{H}_3 = (x_5) \\ \mathbf{H}_4 = (x_6) \end{cases} \quad (10)$$

其次,按照式(10)得到其每个因素的水平为

$$L(H_1) = \begin{cases} L_{11} = (l_{11}, l_{21}) \\ L_{12} = (l_{12}, l_{22}) \\ L_{13} = (l_{13}, l_{23}) \end{cases} \quad L(H_2) = \begin{cases} L_{21} = (l_{31}, l_{41}) \\ L_{22} = (l_{32}, l_{42}) \\ L_{23} = (l_{33}, l_{43}) \end{cases}$$

$$L(H_3) = \begin{cases} L_{31} = (l_{51}) \\ L_{32} = (l_{52}) \\ L_{33} = (l_{53}) \end{cases} \quad L(H_4) = \begin{cases} L_{41} = (l_{61}) \\ L_{42} = (l_{62}) \\ L_{43} = (l_{63}) \end{cases}$$

在得到 4 个因素以及每个因素对应的 3 个水平后,就可以根据 $L_9(3^4)$ 正交表来进行组合实验,得到的 9 个组合实验如表 1 所示,每一行代表一次实验组合。其中单元格中括号前的数字表示对应实验组合中该因素下的第几个水平,如第一行第一列的 $1(L_{11})$ 表示第一次实验组合中因素 $L(H_1)$ 的第一个水平(L_{11}),其他依次类推。

表 1 $L_9(3^4)$ 正交表生产个体

Tab. 1 $L_9(3^4)$ orthogonal table production offspring

实验号	$L(H_1)$	$L(H_2)$	$L(H_3)$	$L(H_4)$	子代个体
1	$1(L_{11})$	$1(L_{21})$	$1(L_{31})$	$1(L_{41})$	$l_{11}, l_{21}, l_{31}, l_{41}, l_{51}, l_{61}$
2	$1(L_{11})$	$2(L_{22})$	$2(L_{32})$	$2(L_{42})$	$l_{11}, l_{21}, l_{32}, l_{42}, l_{52}, l_{62}$
3	$1(L_{11})$	$3(L_{23})$	$3(L_{33})$	$3(L_{43})$	$l_{11}, l_{21}, l_{33}, l_{43}, l_{53}, l_{63}$
4	$2(L_{12})$	$1(L_{22})$	$2(L_{31})$	$3(L_{41})$	$l_{12}, l_{22}, l_{31}, l_{41}, l_{52}, l_{63}$
5	$2(L_{12})$	$2(L_{22})$	$3(L_{33})$	$1(L_{41})$	$l_{12}, l_{22}, l_{32}, l_{42}, l_{53}, l_{61}$
6	$2(L_{12})$	$3(L_{23})$	$1(L_{31})$	$2(L_{42})$	$l_{12}, l_{22}, l_{33}, l_{43}, l_{51}, l_{62}$
7	$3(L_{13})$	$1(L_{23})$	$3(L_{33})$	$2(L_{42})$	$l_{13}, l_{23}, l_{31}, l_{41}, l_{53}, l_{62}$
8	$3(L_{13})$	$2(L_{22})$	$1(L_{31})$	$3(L_{43})$	$l_{13}, l_{23}, l_{32}, l_{42}, l_{51}, l_{63}$
9	$3(L_{13})$	$3(L_{23})$	$2(L_{32})$	$1(L_{41})$	$l_{13}, l_{23}, l_{33}, l_{43}, l_{52}, l_{61}$

2.3 算法流程

MOEA/D-OD 算法输入为:①多目标优化问题。②停止准则。③ N :MOEA/D-OD 所分解的子问题个数。④ $\lambda^1, \lambda^2, \dots, \lambda^N$:分布均匀的 N 个权重向量。⑤ T :权重向量的邻居规模。⑥ N :种群 NP 的大小。

(1) 初始化

①设置 $EP = \emptyset$ 。

②计算任何 2 个权重向量的欧氏距离,为每个权重向量选出最近的 T 个向量作为它的邻居。设 $B(i) = \{i_1, i_2, \dots, i_T\} (i = 1, 2, \dots, N)$, 其中 $\lambda^{i_1}, \lambda^{i_2}, \dots, \lambda^{i_T}$ 为距离 λ^i 最近的 T 个权重向量。

③初始化种群 x^1, x^2, \dots, x^N , 设 $FV^i = F(x^i) (i = 1, 2, \dots, N)$ 。

④采用基于问题的特点方法初始化 $z = (z_1, z_2, \dots, z_m)^T$ 。

⑤迭代计时器 $gen = 0$ 。

(2) 主循环

①如果 $rand < p_{cmin} + (p_{cmax} - p_{cmin}) e^{-2 \frac{gen}{MaxGen}}$, 进行

DE 交叉算子。

(a)从邻域中随机选择 3 个个体 $X_{r1,gen}, X_{r2,gen}, X_{r3,gen}$, 生产新的个体 $V_{i,gen}$

$$V_{i,gen} = X_{r1,gen} + F(X_{r2,gen} - X_{r3,gen})$$

$$u_{j,i,gen} = \begin{cases} v_{j,i,gen} & (rand_j \leq CR \text{ 或 } j = jrand) \\ x_{j,i,gen} & (\text{其他}) \end{cases}$$

其中, $i = 1, 2, \dots, N; j \in \{1, 2, \dots, D\}$ 。

(b)进行多项式变异操作,产生子代个体 y 。

(c)利用新解 y 更新邻域内的子问题。

②否则,从邻域中随机选择两个个体并利用正交交叉算子生成子代个体 y , 其中正交表采用 $L_9(3^4)$ 。更新邻域内的子问题,替换邻居中最差解。

(3) 停止判断

判断 $gen > MaxGen$ 。如果是,则停止算法并输出结果;否则返回步骤(2)。

3 实验仿真及分析

为测试 MOEA/D-OD 算法性能,本文选取 WFG 系列和 F9 系列共 18 个测试函数进行实验,并同 MOEA/D-DE^[20]、NSGA-II^[4]、SMPSO^[25]、AbYSS^[26] 算法对比分析。

3.1 测试函数和性能评价指标

为了检验 MOEA/D-OD 算法的有效性,本文使用 WFG 系列^[27]和 LZ09_F(1-9)^[28]系列基准函数,共 18 个测试函数。其中 WFG 系列函数的 PF 具有混合凸和凹、连续和非连续、线性、欺骗性、多模态等特性,本文 WFG 问题采用 2 目标函数,其具体函数表达式见文献[28]。LZ09_F(1-9)系列基准函数是由 ZHANG 等^[28]于 2009 年提出,这 9 个函数具有复杂的 PS,其中 F6 和 F9 为非凸 PF,其他函数为凸函数,F7 和 F8 为多峰问题,F1~F5 和 F7~F9 为 2 目标函数,F6 为 3 目标函数。

由于没有单一的评价指标能够全面地评估一种 MOEA 算法的性能,按照经验本文选用 HV^[9]和 IGD^[14]两种被广泛使用的性能指标。HV 和 IGD 均为一个综合型指标,两者均能评价算法的收敛性和多样性。对于 HV 指标,其数值越大,表明算法获得非支配解集越接近真实的 PF,而 IGD 指标,则是数值越小,算法性能越优异。

3.2 算法对比实验

在此,将 MOEA/D-OD 同 MOEA/D-DE、NSGA-II、SMPSO、AbSYY 在测试函数上进行仿真分析,其中 5 种算法的每次运行最大次数为 250 次,当 2 目标时,种群大小为 300,当 3 目标时,种群大小为 500。对于 MOEA/D-OD 算法的参数如下:

$p_{\min} = 0.2$, $p_{\max} = 0.8$, $CR = 1$, $F = 0.5$, 变异概率为 $1/\eta$, η 为个体的决策变量长度, 邻域大小 $T = 20$, 邻域搜索概率 $\delta = 0.9$, 子问题更新数目 n_r 为 2 或 3。其他 4 种算法的参数, 参照文献中设置的参数进行

设置。本文算法均采用 JAVA 进行编写, 对比算法源于 Jmetal 框架, 其算法原代码来源于 <http://jmetal.sourceforge.net>。将 5 种算法同时运行 30 次, 结果如表 2、3 所示。

表 2 HV 性能指标的均值和标准差
Tab. 2 HV mean and standard deviation

	MOEAD-OD	MOEA/D-DE	NSGA-II	SMPSO	AbYSS
WFG1	$6.33 \times 10^{-1} (8.3 \times 10^{-4})^*$	$6.33 \times 10^{-1} (1.1 \times 10^{-3})$	$6.22 \times 10^{-1} (4.0 \times 10^{-3})$	$1.48 \times 10^{-1} (2.7 \times 10^{-2})$	$4.22 \times 10^{-1} (1.1 \times 10^{-1})$
WFG2	$5.65 \times 10^{-1} (1.3 \times 10^{-5})$	$5.65 \times 10^{-1} (2.3 \times 10^{-5})$	$5.65 \times 10^{-1} (9.4 \times 10^{-4})^*$	$5.64 \times 10^{-1} (1.7 \times 10^{-4})$	$5.63 \times 10^{-1} (1.9 \times 10^{-3})$
WFG3	$4.44 \times 10^{-1} (6.4 \times 10^{-6})^*$	$4.44 \times 10^{-1} (6.6 \times 10^{-6})$	$4.43 \times 10^{-1} (1.0 \times 10^{-4})$	$4.43 \times 10^{-1} (3.3 \times 10^{-5})$	$4.43 \times 10^{-1} (2.0 \times 10^{-4})$
WFG4	$2.21 \times 10^{-1} (1.9 \times 10^{-4})$	$2.21 \times 10^{-1} (1.9 \times 10^{-4})$	$2.21 \times 10^{-1} (1.2 \times 10^{-4})$	$2.10 \times 10^{-1} (1.4 \times 10^{-3})$	$2.22 \times 10^{-1} (5.7 \times 10^{-5})^*$
WFG5	$1.99 \times 10^{-1} (1.7 \times 10^{-3})$	$1.98 \times 10^{-1} (7.3 \times 10^{-5})$	$1.98 \times 10^{-1} (6.4 \times 10^{-5})$	$1.99 \times 10^{-1} (4.1 \times 10^{-4})^*$	$1.99 \times 10^{-1} (1.5 \times 10^{-5})$
WFG6	$2.13 \times 10^{-1} (7.4 \times 10^{-6})^*$	$2.13 \times 10^{-1} (8.8 \times 10^{-6})$	$2.10 \times 10^{-1} (2.5 \times 10^{-3})$	$2.13 \times 10^{-1} (6.3 \times 10^{-5})$	$1.78 \times 10^{-1} (1.9 \times 10^{-2})$
WFG7	$2.13 \times 10^{-1} (1.5 \times 10^{-5})$	$2.13 \times 10^{-1} (1.2 \times 10^{-5})$	$2.13 \times 10^{-1} (4.8 \times 10^{-5})$	$2.13 \times 10^{-1} (5.2 \times 10^{-5})$	$2.13 \times 10^{-1} (4.1 \times 10^{-5})^*$
WFG8	$1.70 \times 10^{-1} (2.3 \times 10^{-2})$	$1.70 \times 10^{-1} (2.3 \times 10^{-2})$	$1.82 \times 10^{-1} (2.2 \times 10^{-2})^*$	$1.53 \times 10^{-1} (3.5 \times 10^{-3})$	$1.60 \times 10^{-1} (1.9 \times 10^{-2})$
WFG9	$2.43 \times 10^{-1} (3.2 \times 10^{-4})$	$2.44 \times 10^{-1} (3.0 \times 10^{-4})^*$	$2.43 \times 10^{-1} (5.9 \times 10^{-4})$	$2.41 \times 10^{-1} (3.6 \times 10^{-4})$	$2.41 \times 10^{-1} (3.3 \times 10^{-3})$
F1	$6.65 \times 10^{-1} (1.2 \times 10^{-5})^*$	$6.65 \times 10^{-1} (1.5 \times 10^{-5})$	$6.62 \times 10^{-1} (1.4 \times 10^{-4})$	$6.64 \times 10^{-1} (1.8 \times 10^{-4})$	$6.61 \times 10^{-1} (3.5 \times 10^{-4})$
F2	$6.62 \times 10^{-1} (2.3 \times 10^{-4})^*$	$6.62 \times 10^{-1} (7.2 \times 10^{-4})$	$5.42 \times 10^{-1} (2.8 \times 10^{-2})$	$4.98 \times 10^{-1} (2.7 \times 10^{-2})$	$5.22 \times 10^{-1} (3.6 \times 10^{-2})$
F3	$6.53 \times 10^{-1} (1.4 \times 10^{-2})^*$	$6.52 \times 10^{-1} (1.8 \times 10^{-2})$	$6.16 \times 10^{-1} (6.1 \times 10^{-3})$	$6.04 \times 10^{-1} (5.6 \times 10^{-3})$	$5.97 \times 10^{-1} (2.0 \times 10^{-2})$
F4	$6.61 \times 10^{-1} (1.8 \times 10^{-3})^*$	$6.60 \times 10^{-1} (2.0 \times 10^{-3})$	$6.24 \times 10^{-1} (3.3 \times 10^{-3})$	$6.06 \times 10^{-1} (4.2 \times 10^{-3})$	$6.20 \times 10^{-1} (3.7 \times 10^{-3})$
F5	$6.51 \times 10^{-1} (7.1 \times 10^{-3})^*$	$6.50 \times 10^{-1} (8.8 \times 10^{-3})$	$6.27 \times 10^{-1} (6.2 \times 10^{-3})$	$6.19 \times 10^{-1} (2.5 \times 10^{-3})$	$6.14 \times 10^{-1} (1.5 \times 10^{-2})$
F6	$4.21 \times 10^{-1} (2.6 \times 10^{-3})^*$	$4.19 \times 10^{-1} (3.5 \times 10^{-3})$	$2.90 \times 10^{-1} (2.0 \times 10^{-2})$	$3.08 \times 10^{-1} (1.0 \times 10^{-2})$	$2.40 \times 10^{-1} (4.3 \times 10^{-2})$
F7	$6.40 \times 10^{-1} (3.3 \times 10^{-2})$	$6.56 \times 10^{-1} (1.1 \times 10^{-2})^*$	$5.08 \times 10^{-1} (2.3 \times 10^{-2})$	$2.92 \times 10^{-1} (7.6 \times 10^{-2})$	$3.98 \times 10^{-1} (5.0 \times 10^{-2})$
F8	$5.05 \times 10^{-1} (5.6 \times 10^{-2})^*$	$4.68 \times 10^{-1} (6.1 \times 10^{-2})$	$4.63 \times 10^{-1} (3.9 \times 10^{-2})$	$1.31 \times 10^{-1} (4.2 \times 10^{-2})$	$3.68 \times 10^{-1} (6.8 \times 10^{-2})$
F9	$3.28 \times 10^{-1} (1.3 \times 10^{-3})^*$	$3.26 \times 10^{-1} (6.1 \times 10^{-3})$	$1.92 \times 10^{-1} (5.5 \times 10^{-2})$	$1.72 \times 10^{-1} (1.8 \times 10^{-2})$	$2.90 \times 10^{-1} (4.5 \times 10^{-2})$

注: * 最优解, 下同。

表 3 IGD 性能指标的均值和标准差
Tab. 3 IGD mean and standard deviation

	MOEA/D-OD	MOEA/D-DE	NSGA-II	SMPSO	AbYSS
WFG1	$3.60 \times 10^{-4} (3.5 \times 10^{-5})^*$	$3.72 \times 10^{-4} (5.1 \times 10^{-5})$	$1.22 \times 10^{-3} (2.4 \times 10^{-4})$	$1.25 \times 10^{-2} (1.6 \times 10^{-3})$	$6.74 \times 10^{-3} (3.8 \times 10^{-3})$
WFG2	$6.04 \times 10^{-4} (4.8 \times 10^{-6})$	$6.03 \times 10^{-4} (7.6 \times 10^{-6})$	$8.78 \times 10^{-4} (2.3 \times 10^{-3})$	$1.49 \times 10^{-4} (1.7 \times 10^{-5})^*$	$3.53 \times 10^{-3} (3.8 \times 10^{-3})$
WFG3	$3.01 \times 10^{-3} (1.4 \times 10^{-7})$	$3.01 \times 10^{-3} (1.5 \times 10^{-7})$	$3.01 \times 10^{-3} (4.0 \times 10^{-7})$	$3.01 \times 10^{-3} (1.5 \times 10^{-7})$	$3.00 \times 10^{-3} (5.8 \times 10^{-7})^*$
WFG4	$7.13 \times 10^{-5} (8.5 \times 10^{-6})$	$6.73 \times 10^{-5} (3.7 \times 10^{-6})$	$5.66 \times 10^{-5} (2.7 \times 10^{-6})$	$2.28 \times 10^{-4} (2.5 \times 10^{-5})$	$4.18 \times 10^{-5} (5.2 \times 10^{-7})^*$
WFG5	$9.23 \times 10^{-4} (4.9 \times 10^{-5})^*$	$9.34 \times 10^{-4} (5.3 \times 10^{-7})$	$9.34 \times 10^{-4} (7.0 \times 10^{-7})$	$9.29 \times 10^{-4} (7.4 \times 10^{-6})$	$9.32 \times 10^{-4} (7.6 \times 10^{-7})$
WFG6	$9.14 \times 10^{-5} (2.1 \times 10^{-7})$	$9.13 \times 10^{-5} (1.9 \times 10^{-7})$	$1.50 \times 10^{-4} (7.0 \times 10^{-5})$	$7.64 \times 10^{-5} (3.8 \times 10^{-6})^*$	$1.42 \times 10^{-3} (8.0 \times 10^{-4})$
WFG7	$4.06 \times 10^{-5} (3.9 \times 10^{-8})$	$4.06 \times 10^{-5} (3.6 \times 10^{-8})$	$4.02 \times 10^{-5} (1.9 \times 10^{-6})$	$2.91 \times 10^{-5} (4.3 \times 10^{-7})$	$2.82 \times 10^{-5} (3.5 \times 10^{-7})^*$
WFG8	$2.76 \times 10^{-3} (1.2 \times 10^{-3})$	$2.66 \times 10^{-3} (1.1 \times 10^{-3})$	$1.93 \times 10^{-3} (1.1 \times 10^{-3})^*$	$2.86 \times 10^{-3} (2.4 \times 10^{-4})$	$3.06 \times 10^{-3} (1.0 \times 10^{-3})$
WFG9	$4.35 \times 10^{-5} (3.2 \times 10^{-6})$	$4.26 \times 10^{-5} (2.4 \times 10^{-6})^*$	$4.52 \times 10^{-5} (3.9 \times 10^{-6})$	$6.57 \times 10^{-5} (4.4 \times 10^{-6})$	$5.96 \times 10^{-5} (4.0 \times 10^{-5})$
F1	$7.25 \times 10^{-5} (6.3 \times 10^{-7})^*$	$7.29 \times 10^{-5} (1.9 \times 10^{-6})$	$1.48 \times 10^{-4} (3.5 \times 10^{-6})$	$8.17 \times 10^{-5} (4.1 \times 10^{-6})$	$1.68 \times 10^{-4} (1.2 \times 10^{-5})$
F2	$1.43 \times 10^{-4} (1.4 \times 10^{-5})^*$	$1.66 \times 10^{-4} (5.0 \times 10^{-5})$	$4.91 \times 10^{-3} (1.2 \times 10^{-3})$	$4.65 \times 10^{-3} (8.0 \times 10^{-4})$	$6.25 \times 10^{-3} (1.7 \times 10^{-3})$
F3	$1.09 \times 10^{-3} (1.5 \times 10^{-3})^*$	$1.33 \times 10^{-3} (1.9 \times 10^{-3})$	$2.25 \times 10^{-3} (3.6 \times 10^{-4})$	$3.00 \times 10^{-3} (3.1 \times 10^{-4})$	$4.53 \times 10^{-3} (2.1 \times 10^{-3})$
F4	$3.54 \times 10^{-4} (1.5 \times 10^{-4})$	$3.77 \times 10^{-4} (1.7 \times 10^{-4})$	$3.09 \times 10^{-3} (7.8 \times 10^{-4})$	$2.86 \times 10^{-3} (2.5 \times 10^{-4})$	$3.37 \times 10^{-3} (8.1 \times 10^{-4})$
F5	$1.12 \times 10^{-3} (1.0 \times 10^{-3})$	$1.02 \times 10^{-3} (7.0 \times 10^{-4})^*$	$1.89 \times 10^{-3} (5.1 \times 10^{-4})$	$2.40 \times 10^{-3} (1.5 \times 10^{-4})$	$3.18 \times 10^{-3} (1.4 \times 10^{-3})$
F6	$1.18 \times 10^{-3} (7.2 \times 10^{-5})^*$	$1.25 \times 10^{-3} (1.1 \times 10^{-4})$	$3.72 \times 10^{-3} (5.3 \times 10^{-4})$	$4.04 \times 10^{-3} (4.3 \times 10^{-4})$	$5.60 \times 10^{-3} (9.7 \times 10^{-4})$
F7	$1.07 \times 10^{-3} (1.3 \times 10^{-3})$	$4.61 \times 10^{-4} (4.5 \times 10^{-4})^*$	$6.76 \times 10^{-3} (2.1 \times 10^{-3})$	$1.17 \times 10^{-2} (2.2 \times 10^{-3})$	$1.54 \times 10^{-2} (4.7 \times 10^{-3})$
F8	$5.44 \times 10^{-3} (1.7 \times 10^{-3})^*$	$6.25 \times 10^{-3} (1.8 \times 10^{-3})$	$7.07 \times 10^{-3} (2.3 \times 10^{-3})$	$1.55 \times 10^{-2} (1.5 \times 10^{-3})$	$1.44 \times 10^{-2} (5.6 \times 10^{-3})$
F9	$1.96 \times 10^{-4} (6.8 \times 10^{-5})^*$	$2.69 \times 10^{-4} (2.2 \times 10^{-4})$	$6.51 \times 10^{-3} (3.3 \times 10^{-3})$	$4.98 \times 10^{-3} (7.9 \times 10^{-4})$	$1.07 \times 10^{-3} (2.2 \times 10^{-3})$

从表 2 可知, MOEA/D-OD 算法占优的问题个数为 11 个, MOEA/D-DE、NSGA-II 和 AbYSS 算法获得 2 个, SMPSO 算法仅获得 1 个。对于 WFG

系列问题, MOEA/D-OD 和 MOEA/D-DE 求解获得数值都较接近, 处于一个数量级, 但是 MOEA/D-OD 整体上优于 MOEA/D-OD, 由此表明 MOEA/D-

OD 性能较 MOEA/D-DE 更稳定,收敛性也更好。在 WFG1 问题上,MOEA/D-OD 算法较 NSGA-II、SMP SO 和 AbsYY 更优;对于 WFG(2,3,5,7,9)问题,5 种算法性能相当;对于 WFG4 问题,AbSYY 算法最优,SMP SO 算法相对最差;对于 WFG6 问题,MOEA/D-OD 算法远优于 AbSYY 算法;对于 WFG8 问题,NSGA-II 算法性能最优,MOEA/D-OD 性能排名 2。对于 LZ09 系列函数,MOEA/D-OD 算法较其他 4 种算法而言占有绝对优势,其中除了 F7 问题,性能稍差于 MOEA/D-DE 算法。

由表 3 可知,MOEA/D-OD 算法占优的问题为 9/18,MOEA/D-DE 和 AbYSS 算法为 3/18,SMP SO 算法获得了 2/18,NSGA-II 仅获得 1 个最优值。在 WFG1 问题上,MOEA/D-OD 算法较另外 4 种更优;对于 WFG2 问题,SMP SO 算法性能最优,MOEA/D-OD、MOEA/D-DE 和 NSGA-II 性能相当,AbSYY 性能最差,相差一个数量级;对于 WFG(3,5,7,9)问题,5 种算法性能差距较小;对于 WFG4 问题,AbSYY 算法最优;对于 WFG6 问题,MOEA/D-OD、MOEA/D-DE 和 SMP SO 算法性能在一个数量级,其中 SMP SO 稍占优,AbSYY 性能最差;对于 WFG8 问题,NSGA-II 算法性能最优,MOEA/D-OD、MOEA/D-DE 和 SMP SO 性能相当。对于 LZ09 系列函数,MOEA/D-OD 算法较其他 4 种算法而言占有绝对优势,其中除了 F(7,5)问题,性能稍差于 MOEA/D-DE 算法。对于 LZ09 系列函数,MOEA/D-OD 算法整体上优于 MOEA/D-DE 算法,表明在 MOEA/D 框架下采用 OD 模型能够有效提高算法的整体性能。对于 F(2,4,7,9)问题,MOEA/D-OD 较 NSGA-II、SMP SO 和 AbSYY 数值上基本都提高了 1 个量级。

对 MOEA/D-OD 与 MOEA/D-DE、NSGA-II、SMP SO 和 AbsYY,5 种算法进行 Friedman 测试,测试结果如表 4 所示。从测试数据可以看出,MOEA/D-OD 算法性能排名第 1,MOEA/D-DE 算法排名第 2,MOEA/D-OD 算法比 MOEA/D-DE 算法整体性能平均提高 10% 左右。对于 HV 指标,MOEA/D-OD 算法相对 NSGA-II、SMP SO 和 AbsYY 算法,统计数值分别是 1.916 倍、2.3 倍和 2.19 倍;对于 IGD 指标,MOEA/D-OD 算法相对 NSGA-II、SMP SO 和 AbsYY 算法分别提高了 36%、38% 和 51%。综合以上分析可以得出,MOEA/D-OD 算法相比另外 4 种算法,具有较强的竞争力。

为了评估算法的收敛速度,绘制了 MOEA/D-OD、MOEA/D-DE、NSGA-II、SMP SO 和 AbsYY 算法随着进化代数的 IGD 值,如图 1 所示。从收敛速

表 4 HV 和 IGD 上的 Friedman 排名

Tab.4 Friedman ranking of HV and IGD

算法	数值(排名)	
	HV	IGD
MOEA/D-OD	5.111 1(1)	2.333 3(1)
MOEA/D-DE	4.555 5(2)	2.555 6(2)
NSGA-II	2.666 7(3)	3.666 7(3)
SMP SO	2.222 2(5)	3.777 8(4)
AbYSS	2.333 4(4)	4.777 8(5)

度看,MOEA/D-OD 算法在 F2 问题上相对于其他算法具有较快的收敛速度。SMP SO 算法对于 F(1,4,5,7)问题,在进化初期都有较快的收敛速度,也表明了 SMP SO 算法的钳制粒子飞行速度的方法具有快速的全局寻优能力,但其后期进化能力较弱。MOEA/D-OD 算法在问题 F(3,4,8,9)上,后期寻优能力较强。对于 F(5,7)问题,MOEA/D-DE 算法后期具有很好的寻优效果。

4 工程实例

鉴于 MOEA/D-OD 算法在求解无约束多目标问题方面的良好性能,为了进一步验证其在求解多目标约束问题的能力,将其应用于某主梁的多目标设计问题。主梁的示意图如图 2 所示,主梁尺寸在满足几何和强度约束条件下,梁的截面积和静载弯曲力最小,其中已知 $E = 2 \times 10^4 \text{ kN/cm}^2$, $\sigma_\alpha = 16 \text{ kN/cm}^2$, $P = 600 \text{ kN}$, $Q = 50 \text{ kN}$, $L = 200 \text{ cm}$ 。

由于问题存在约束,本文采用 DEB^[29]提出的多目标法处理违反约束的个体。MOEA/D-OD 算法的基本参数为:最大进化代数 300 次,种群大小为 500, $p_{\min} = 0.2$, $p_{\max} = 0.8$, $CR = 1$, $F = 0.5$,变异概率为 $1/\eta$, η 为个体的决策变量长度,邻域大小 $T = 20$,邻域搜索概率 $\delta = 0.9$,子问题更新数目 $n_r = 3$ 。根据文献[30],可得到数学模型为

$$\min f_1(x) = 2x_2x_4 + x_3(x_1 - 2x_4)$$

$$\min f_2(x) = \frac{PL^3}{48EI}$$

约束条件

$$g = \frac{M_y}{Z_y} + \frac{M_z}{Z_z} - \sigma_\alpha \leq 0$$

其中

$$10 \leq x_1 \leq 80 \quad 10 \leq x_2 \leq 50$$

$$0.9 \leq x_3 \leq 5 \quad 0.9 \leq x_4 \leq 5$$

$$M_y = \frac{PL}{4} \quad M_z = \frac{QL}{4}$$

$$I = \frac{1}{12} \{ x_3 (x_1 - 2x_4)^3 + 2x_2x_4 [4x_4^2 + 3x_1(x_1 - 2x_4)] \}$$

$$Z_y = \frac{1}{6x_1} \{ x_3 (x_1 - 2x_4)^3 + 2x_2x_4 [4x_4^2 + 3x_1(x_1 - 2x_4)] \}$$

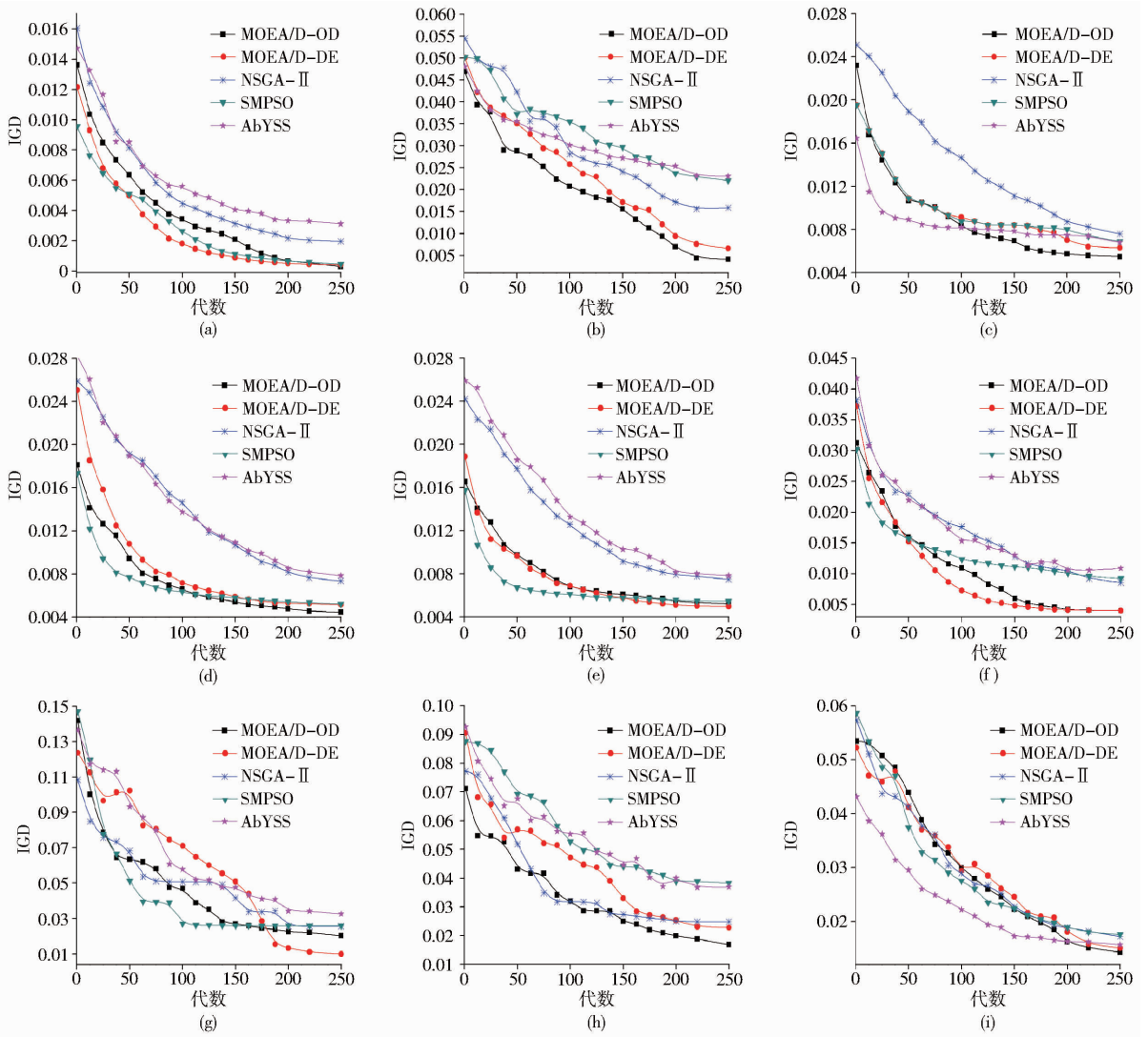


图1 F(1-9)系列进化过程中IGD值

Fig.1 Evolution of mean IGD metric values versus generations

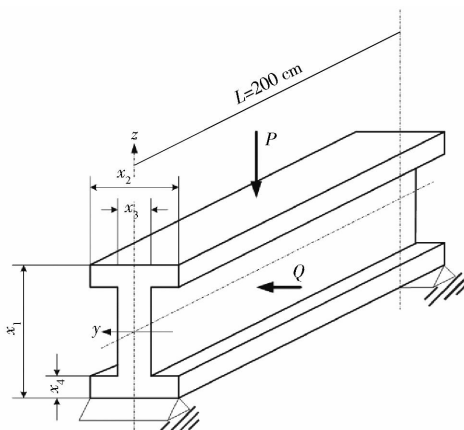


图2 I型梁设计示意图

Fig.2 Schematic diagram of I-beam design

$$Z_z = \frac{1}{6x_2} [(x_1 - 2x_4)x_3^3 + 2x_4x_2^3]$$

图3为MOEA/D-OD算法求解I型梁多目标优化设计问题获得的Pareto前沿,由图可知获得PF较均匀,解集域较广。为了对比算法的性能,将

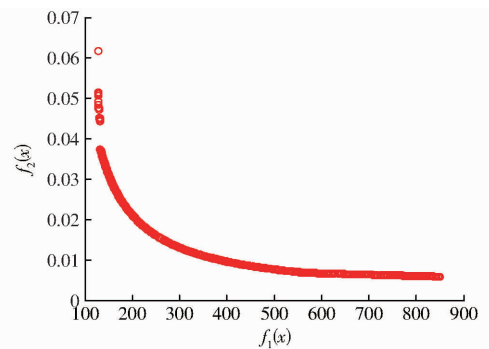


图3 Pareto 前沿图

Fig.3 Diagram of Pareto front

MOEA/D-OD算法与文献[30]中的结果进行对比,其中本文算法获得极端解为(126.7047, 0.061523)和(850.000, 0.005903),文献[30]中,NSGA-II获得的极端解为(135.555, 0.03974)和(850.000, 0.005903),MOPSO-CD获得的极端解为(128.1708, 0.049401)和(850.000, 0.005903),RM-MEDA获得的极端解为(127.391, 0.054075)

和(850.000,0.005 903),MOTLBO获得的极端解为(126.705 1,0.052 304)和(850.000,0.005 903)。由对比可知,本文算法求得的解集更宽广,与文献[30]中提出的MOTLBO算法性能相当。

5 结束语

提出了一种正交设计模型,并采用差分进化策略与其协同进化。基于此,提出了基于正交设计模型MOEA/D算法的MOEA/D-OD算法。将MOEA/D-OD算法与MOEA/D-DE、NSGA-II、

SMPSO、AbSSYY算法在18个基本测试数上进行仿真对比实验,分析结果表明:正交实验模型能够有效地找到种群的结构信息,算法获得Pareto解集更逼近真实的PF,尤其是在求解具有复杂PS的LZ09系列函数。相对于仅采用DE算子,MOEA/D-OD算法较MOEA/D-DE算法整体性能更优,鲁棒性更强。将MOEA/D-OD算法应用于某主梁多目标设计问题求解时,算法能获得较均匀的PF,对比验证表明了工程实用性。

参 考 文 献

- SCHAFFER J D. Multiple objective optimization with vector evaluated genetic algorithms[C]//Proceedings of the 1st International Conference on Genetic Algorithms, Mahwah, NJ: Lawrence Erlbaum Associates Inc., 1985: 93-100.
- LI K, SAM Kwong, ZHANG Q F, DEB K. Interrelationship-based selection for decomposition multi-objective optimization[J]. IEEE Transactions on Cybernetics, 2015, 45(10): 2076-2088.
- 周爱民,张青富,张桂成.一种基于混合高斯模型的多目标进化算法[J].软件学报,2014,25(5):913-928.
ZHOU A M, ZHANG Q F, ZHANG G X. Multi-objective evolutionary algorithm based on mixture Gaussian models[J]. Journal of Software, 2014, 25(5): 913-928. (in Chinese)
- DEB K, PRATAP A, AGARWAL S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.
- 杨咚咚,马晶晶,焦李成,等.一种改进 ε 支配的等度规映射方法[J].软件学报,2011,22(10):2291-2304.
YANG D D, MA J J, JIAO L C, et al. Improved ε dominance by isomap[J]. Journal of Software, 2011, 22(10): 2291-2304. (in Chinese)
- ZITZLER E, LAUMANN S, THIELE L. SPEA2: improving the strength Pareto evolutionary algorithm[C]//Evolutionary Methods for Design, Optimization and Control with Application to Industrial Problems (EUROGEN 2001), International Centre for Numerical Methods in Engineering (CIMNE), Athens, 2002: 95-100.
- HORN J, NAFPLIOTIS N, GOLDBERG D E. A niched Pareto genetic algorithm for multi-objective optimization[C]//Proceedings of the 1st IEEE Conference on Evolutionary Computation, 1994: 82-87.
- CORNE D W, KNOWLES J D, OATES M J. The Pareto envelope-based selection algorithm for multi-objective optimization[C]//Proceedings of the 6th Work-shop on Parallel Problem Solving from Nature (PPSN VI), Springer, Berlin, Heidelberg, 2000: 839-848.
- ZITZLER E, THIELE L. Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(4): 257-271.
- ZITZLER E, KÜNZLI S. Indicator-based selection in multi-objective search[C]//Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII), Springer, Berlin, Heidelberg, 2004: 832-842.
- BEUME N, NAUJOKS B, EMMERICH M. SMS-EMOA: multi-objective selection based on dominated hypervolume[J]. European Journal of Operational Research, 2007, 181(3): 1653-1669.
- BADER J, ZITZLER E. HypE: an algorithm for fast hypervolume-based many-objective optimization[J]. Evolutionary Computation, 2011, 19(1): 45-76.
- MURATA T, ISHIBUCHI H, GEN M. Specification of genetic search directions in cellular multi-objective genetic algorithms[C]//Proceedings of the 1st International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001), Springer, Berlin, Heidelberg, 2001: 82-95.
- ZHANG Q F, LI H. MOEA/D: a multi-objective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731.
- LI H, LANDA S D. An adaptive evolutionary multi-objective approach based on simulated annealing[J]. Evolutionary Computation, 2011, 19(4): 561-595.
- 王亚辉,贾晨辉,赵仁鹏.基于分解机制的多目标蝙蝠算法及应用[J/OL].农业机械学报,2015,46(4):316-324. http://www.j-csam.org/jcsam/ch/reader/view_abstract.aspx. DOI: 10.6041/j.issn.1000-1298.2015.04.047.
WANG Y H, JIA C H, ZHAO R P. Multi-objective bat algorithm based on decomposition and its application[J/OL]. Transactions of the Chinese Society for Agricultural Machinery, 2015, 46(4): 316-324. (in Chinese)
- ZHANG Q, LI H, DARINGER M, et al. MOEA/D with NBI-style tchebycheff approach for portfolio management[C]//IEEE Congress on Evolutionary Computation, 2010: 1-8.
- MA X, QI Y, LI L, et al. MOEA/D with uniform decomposition measurement for many-objective problems[J]. Soft Compute, 2014, 18: 2541-2564.
- QI Y, MA X, LIU F, et al. MOEA/D with adaptive weight adjustment[J]. Evolutionary Computation, 2014, 22(2): 231-264.

- Shanghai Jiao Tong University, 2007. (in Chinese)
- 13 SUBRAMANIAN S C, DARBHA S, RAJAGOPAL K R. Modeling the pneumatic subsystem of a s-cam air brake system[J]. *Journal of Dynamic Systems Measurement & Control*, 2003, 126(1):1416-1421.
 - 14 BOWLIN C L, SUBRAMANIAN S C, DARBHA S, et al. Pressure control scheme for air brakes in commercial vehicles[J]. *IEE Proceedings-Intelligent Transport Systems*, 2006, 153(1):21-32.
 - 15 鲍官军, 王志恒, 杨庆华, 等. 气动柔性扭转关节动态特性研究[J]. *农业机械学报*, 2010, 41(6):204-207.
BAO Guanjun, WANG Zhiheng, YANG Qinghua, et al. Dynamic characteristics of flexible pneumatic torsion joint[J]. *Transactions of the Chinese Society for Agricultural Machinery*, 2010, 41(6):204-207. (in Chinese)
 - 16 施岩, 陆艺, 王学影, 等. 基于 AMEsim 的汽车制动阀动态特性研究[J]. *仪表技术与传感器*, 2014(8):23-26.
SHI Yan, LU Yi, WANG Xueying, et al. Analysis of brake valve dynamic characteristics based on AMEsim[J]. *Instrument Technique and Sensor*, 2014(8):23-26. (in Chinese)
 - 17 贺志凯. 立柱液控单向阀的仿真分析与流场可视化研究[D]. 太原: 太原理工大学, 2012.
HE Zhikai. Simulation analysis of prop pilot operated check valve and research on visualization of flow filed[D]. Taiyuan: Taiyuan University of Technology, 2012. (in Chinese)
 - 18 张晓艳. 客车制动系统的建模和仿真研究[D]. 成都: 西南交通大学, 2011.
ZHANG Xiaoyan. Modeling and simulation research on passenger train breaking system[D]. Chengdu: Southwest Jiao Tong University, 2011. (in Chinese)
 - 19 廖湘平, 龚国芳, 孙辰晨, 等. 基于 AMESim 的液粘调速离合器动态接合特性研究[J/OL]. *农业机械学报*, 2016, 47(6):324-332. http://www.j-csam.org/jcsam/ch/reader/view_abstract.aspx?flag=1&file_no=20160643&journal_id=jcsam. DOI:10.6041/j.issn.1000-1298.2016.06.043.
LIAO Xiangping, GONG Guofang, SUN Chenchen, et al. Dynamic engagement performance of hydro-viscous clutch based on AMESim[J/OL]. *Transactions of the Chinese Society for Agricultural Machinery*, 2016, 47(6):324-332. (in Chinese)
 - 20 徐志鹏, 王宣银, 罗语溪. 高压气动比例减压阀设计与仿真[J]. *农业机械学报*, 2011, 42(1):209-212.
XU Zhipeng, WANG Xuanyin, LUO Yuxi. Design and simulation of a high pressure proportional pneumatic pressure reducing valve[J]. *Transactions of the Chinese Society for Agricultural Machinery*, 2011, 42(1):209-212. (in Chinese)

~~~~~  
(上接第 369 页)

- 20 LI H, ZHANG Q. Multi-objective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II[J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(2):284-302.
- 21 赵选民. 实验设计方法[M]. 北京: 科学出版社, 2006.
- 22 LEUNG Y W, WANG Y. An orthogonal genetic algorithm with quantization for global numerical optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2001, 5(1):41-53.
- 23 蔡自兴, 江中央, 王勇, 等. 一种新的基于正交实验设计的约束优化进化算法[J]. *软件学报*, 2010, 33(5):855-864.  
CAI Z X, JIANG Z Y, WANG Y, et al. A novel constrained optimization evolutionary algorithm based on orthogonal experimental design[J]. *Journal of Software*, 2010, 33(5):855-864. (in Chinese)
- 24 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]. *Journal of Software*, 2009, 20(2):271-289. (in Chinese)
- 25 NEBRO A J, DURILLO J J, GARCÍA J, et al. SMPSO: a new PSO-based metaheuristic for multiobjective optimization[C]// *IEEE Symposium on Computational Intelligence in Multi-criteria Decision-making*, 2009:66-73.
- 26 NEBRO A J, LUNA F, ABLE E, et al. AbYSS: adapting scatter search to multi-objective optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2008, 12(4):439-457.
- 27 HUBAND S, BARONE L, WHILE L, et al. A scalable multi-objective test problem toolkit[C]// *Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, Springer, Berlin, Heidelberg, 2005:280-295.
- 28 ZHANG Q, LIU W, LI H. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances[C]// *IEEE Congress on Evolutionary Computation*, 2009:203-208.
- 29 DEB K. An efficient constraint handling method for genetic algorithms[J]. *Computation Methods in Applied Mechanics and Engineering*, 2000, 86(2-4):311-338.
- 30 ZOU F, WANG L, HEI X, et al. Multi-objective optimization using teaching-learning-based optimization algorithm[J]. *Engineering Applications of Artificial Intelligence*, 2013, 26(4):1291-1300.