doi:10.6041/j.issn.1000-1298.2015.04.046

多邻域结构多目标遗传算法*

朱大林[1]　詹　腾[1,2]　张　屹[1]　郑小东[1]　张灯皇[1]　余竹玛[1]
(1.三峡大学机械与动力学院, 宜昌 443002; 2.株洲联成集团有限责任公司, 株洲 412001)

摘要: 为了解决应力约束类桁架结构的尺寸优化多目标问题,提出一种多领域结构的多目标遗传算法应用于尺寸优化设计。利用个体之间欧氏距离信息,将种群划分为多个领域以形成多个小生境种群。该算法为每个个体提供一定数量的邻居个体,并规定只能同邻居个体进行交叉变异操作,通过实验分析了不同邻居规模对算法性能的影响。将新算法与其他经典算法在18个标准测试函数上进行了仿真分析,结果表明,所得到的Pareto前端分布更加均匀且更加逼近真实Pareto前端,具有良好的收敛性和多样性。将该算法应用于经典的25杆空间桁架结构优化的求解,获得Pareto前端更均匀,收敛性更好,相对于其他的优化算法具有更好的优化效果。该算法在程序设计、求解空间及其方法通用性等方面表现出良好的性能,并且简单、实用,更加适合于工程实际应用。

关键词: 多邻域结构　邻居规模　桁架结构　多目标优化
中图分类号: TP301.6　**文献标识码**: A　**文章编号**: 1000-1298(2015)04-0309-07

Multi-neighborhood Structure Based Multi-objective Genetic Algorithm

Zhu Dalin[1]　Zhan Teng[1,2]　Zhang Yi[1]　Zheng Xiaodong[1]　Zhang Denghuang[1]　Yu Zhuma[1]
(1. *College of Mechanical & Power Engineering*, *China Three Gorges University*, *Yichang* 443002, *China*
2. *Zhuzhou Lince Group Co.*, *Ltd.*, *Zhuzhou* 412001, *China*)

Abstract: In order to solve the problem of multi-objective size optimization of truss structures with stress constraints, a multi-objective optimization algorithm with multi-neighborhood was proposed. Based on the Euclidean distance between individuals, the population was divided into multi-neighborhood to form several niche populations. A number of individuals were assigned to each cell as neighborhood by the proposed algorithm. The individuals were only allowed interacting with each other within its neighborhood and generating offspring. The influence of different sizes of neighbors on the performance was analyzed through simulation experiments. The test results on 18 benchmarks revealed that the proposed algorithm outperformed some state-of-the-art algorithm in terms of covered area and diversity, which showed good uniformity and diversity. The obtained Pareto front showed good uniformity and diversity when solving the classic multi-objective optimization problem of 25-bar truss structure. The algorithm showed good performance in program design, solution space and generality and so on, which was very simple, practical and suitable for engineering practice.

Key words: Multi-neighborhood structure　Neighbor size　Truss structure　Multi-objective optimization

引言

多目标优化问题是指需要同时优化几个相互冲突目标的问题,广泛存在于现实生活和科学研究中,这类问题的解通常是一组折衷解的集合而非唯一最优解。为有效求解这类问题,人们提出了多目标进化算法(Multi-objective evolutionary algorithm, MOEA)。在过去的20多年中,多目标进化算法在

收稿日期: 2014-06-18　修回日期: 2014-07-19
* 国家自然科学基金资助项目(51275274)
作者简介: 朱大林,教授,主要从事系统优化设计研究,E-mail: dlzhu@ctgu.edu.cn

解决多目标优化问题上取得了令人瞩目的成果，国际上涌现出了许多经典的多目标进化算法，如NSGA-II[1]、SPEA2[2]、MOEA/D[3]和NNIA[4]等。

Alba等[5]将进化算法按照种群结构的分布方式，可以分为两类：单种群进化算法和结构化种群进化算法。将种群分布于不同的拓扑结构中，并约束个体进行遗传操作的范围，对算法的性能有较大的影响[6-10]。本文提出一种多邻域结构的多目标遗传算法（Multi-objective genetic algorithm for multi-neighborhood structure, MNS-MOGA），该算法采用一种简单的种群划分规则，将种群划分为多个邻域，每个邻域之间存在交叉重叠，且每个个体只能同规定的邻域中的个体进行交叉、变异操作，并分析不同的邻居大小对算法性能的影响。通过14个无约束问题和4个有约束问题对MNS-MOGA算法进行性能测试并与目前性能优异的NSGA-II、SPEA2和PEAS算法进行对比分析，为了进一步验证算法的性能，将MNS-MOGA算法应用于求解桁架结构多目标优化设计问题，并对其进行对比分析。

1 多邻域结构的多目标遗传算法

1.1 多邻域结构划分

进化算法按照种群中个体的位置划分，可分为单种群（随机交配种群结构，图1a）和结构化种群，其中结构化种群分为分布式种群结构（图1b）和元胞种群结构（图1c）。对于大多数遗传算法都可以归为单种群结构，即每个个体在单种群中选择父代个体，再进行交叉变异操作；对于结构化种群的算法，种群将会被划分为多个子种群，每个子种群之间通过某种关系进行交流，根据种群划分的不同分为粗晶粒结构和细晶粒结构，或称为分布式遗传算法和元胞遗传算法。

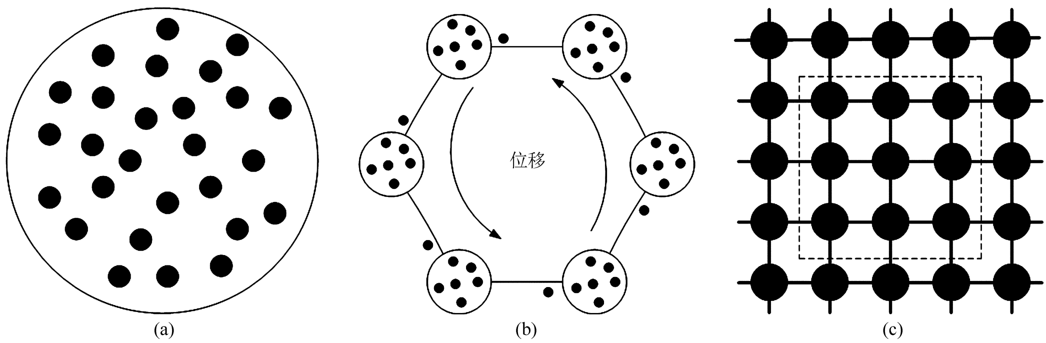

图1 种群结构的示意图

Fig. 1 Population structure

（a）随机交配种群结构 （b）分布式种群结构 （c）元胞种群结构

在研究结构化种群遗传算法的基础上，提出一种简单的种群划分规则，即将种群划分为多个邻域，每个个体都有专属的邻域，且每个个体只能同规定邻域中的个体进行交叉、变异操作。设群体规模为N_p，则第$i(i=1,2,\cdots,N_p)$个个体可以表示为$x_{i,D}$，D为决策空间，$f_K(x_{i,D})$为个体的适应度函数，邻域划分规则

$$D_{i,j}=\sum_{k=1}^{K}(f_k(x_{i,D})-f_{k+1}(x_{j,D}))^2 \tag{1}$$

$$N(x_{i,D})=\mathrm{Sorting}(D_{i,1},D_{i,2},\cdots,D_{i,i-1},D_{i,i+1},\cdots,D_{i,N_p-1}) \tag{2}$$

$$N_{si}=\{x_{n,D},\cdots,x_{m,D}\}\quad(n、m\in[1,N_p]) \tag{3}$$

$$T=\mathrm{Size}(\mathrm{Neighbors}_i) \tag{4}$$

式中 $D_{i,j}$——个体i与个体j之间的欧氏距离函数

$N(x_{i,D})$——个体i的邻域划分函数

$\mathrm{Sorting}(D_{i,1},\cdots,D_{i,N_p-1})$——个体与种群中其他个体之间的欧氏距离排列函数

N_{si}——个体i的邻居个体集合

T——邻居集合大小

在种群初始化后，对于每个个体i通过与其他个体之间的欧氏距离来划分邻居，每个个体都有T个邻居，因此每个邻居之间存在交叉重叠，在进化过程中，会存在一种隐性迁移机制，可以有效地避免遗传漂移现象。本文通过欧氏距离划分邻居，使用欧氏距离从小到大选择T个邻居，有利于进化过程中优秀基因的保留，提高算法的收敛性。

1.2 外部存档及算法性能度量

算法在进化的过程中，对于产生的非劣解，使用外部存档进行存储，随着进化代数的增加，外部存档的规模会随之增加，当外部存档的大小超过设定值时，采用拥挤距离方法计算每个非劣解的距离，然后对超出个体进行剔除。本文采用收敛性（ε）[11]、多样性（Δ）[1]和分布性（H_V）[12]3个性能指标进行度量。

1.2.1 ε评价指标

假设Pareto前端集合为A，那么ε指标是移除A

中每个个体所需要的最小距离的一个尺度。其具体形式是：假设 $\boldsymbol{Z}^1=(z_1^1,\cdots,z_n^1)$，$\boldsymbol{Z}^2=(z_1^2,\cdots,z_n^2)$，其中 n 是问题的目标维数，则

$$\varepsilon(A)=\inf\{\varepsilon\in R\mid\forall z^2\in PF,\exists z^1\in A:z^1\prec_{\varepsilon}z^2\}\tag{5}$$

其中，仅当 $\forall 1\leqslant i\leqslant n:z_i^1<\varepsilon+z_i^2$，则 $z^1\prec_{\varepsilon}z^2$ 成立。

1.2.2　Δ 评价指标

Deb 提出的分布指标是衡量所得的 Pareto 前端解集的分布情况。其计算公式为

$$\Delta=\frac{d_f+d_l+\sum_{i=1}^{n-1}\mid d_i-\bar{d}\mid}{d_f+d_l+(n-1)\bar{d}}\tag{6}$$

式中　d_i——所得 Pareto 前端上每两个连续解点的欧氏距离

$\bar{d}$——欧氏距离的平均距离

d_f、d_l——所得 Pareto 前端的边界点与 Pareto 最优边界点的欧氏距离

n——所得 Pareto 前端个体的数目

对于分布均匀的解来说，该指标取零。因此，该指标的值越小，表明分布程度越均匀。

1.2.3　H_V 评价指标

超体积是用来计算获得的 Pareto 解集个体在目标域所覆盖的体积。其计算公式为

$$H_V=\text{volume}\left(\bigcup_{i=1}^{|Q|}v_i\right)\tag{7}$$

式中　Q——获得的 Pareto 前端的个数

对于这个 Pareto 前端中的每一个体 i，v_i 是由参考点 $w=(0,0,\cdots,0)$ 和成员 i 所形成的超体积，此指标越大表明所得的 Pareto 解越能宽广地覆盖在其前端上。

1.3　约束处理方法

对于含有约束条件的多目标优化问题，常见的处理方式为罚函数法和多目标法[13]。本文采用 Deb[14] 的多目标法处理违反约束的个体。具体如下：①当在两个比较的个体中，一个个体为可行解，另外一个个体为不可行解时，选择可行解。②当两个比较的个体均为可行解时，选择非支配的个体占优。③当两个比较的个体均为不可行解时，选择违反约束条件程度小的个体。

1.4　邻居规模选择分析

邻居规模会对算法的收敛速度和多样性有一定的影响，为了使算法在现有的条件下具有指导性，加快算法的收敛速度和种群多样性，因此对不同的邻居规模 T 进行探讨。

设定种群大小为 100，外部存档的最大容量为 100，最大进化代数为 250 代，模拟二进制交叉算子的交叉概率为 0.9，多项式变异的变异概率为 $1/l_{en}$，l_{en} 为变量维数。采用 ZDT 系列、WFG 系列和 4 个含有约束的测试函数，其中 4 个含有约束的函数为：Golinski、Srinivas、Tanaka 和 Osyczka2，对上述 18 个测试函数各独自运行 30 次，使用 Friedman 进行统计分析，其中对于分布性指标 H_V，数值越大说明分布性越好，ε 和 Δ 指标，数值越小说明性能越好。图 2 是 3 个性能指标的 Friedman 排名柱状图，其中邻居规模 T 为横坐标。由图 2a 可以看出，邻居规模 $T=10$ 时，算法的收敛性最优，且在后续增加邻居个数，算法的收敛性变差。由此说明当邻居个数太小或者太大都不利于算法的收敛性。由图 2b 可得，在 $T=20$ 时，算法的多样性最好。由图 2c 表明，在 $T=10$ 时，算法的分布性最佳，且远优于 $T=15$ 及以上。综合考虑，选择邻居规模 $T=10$ 时，算法的性能相对较优。

1.5　MNS-MOGA 算法流程

（1）初始化种群并进行算法参数设计。

（2）对种群个体进行适应度评估，并将非劣解存入外部存档，设定进化代数 $g=0$。

（3）对每个个体进行 1.2 节中的多邻域结构邻居划分。

（4）在邻居中选择 1 个个体与当前个体组成父代个体，进行模拟二进制交叉操作，对生成的子代个体进行多项式变异操作。对子代个体进行适应度评估，如果优于当前个体，则替换，否则不替换。

（5）在进化过程中，将非劣解存入外部存档中。

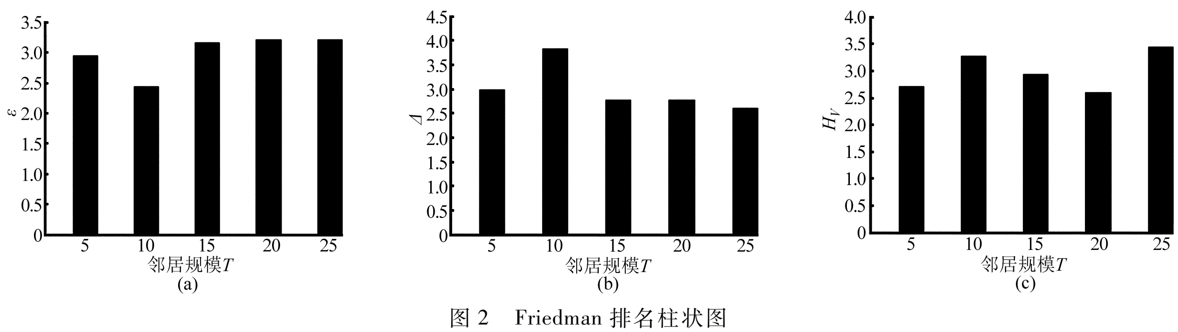

图 2　Friedman 排名柱状图

Fig. 2　Ranking histogram of Friedman number

（6）如果 $g<100$，则转至步骤（4）；否则，每隔20代，转至步骤（3），更换每个个体的邻居个体，增加对种群的扰动。如果 $g\geqslant 250$，输出外部存档，获得 Pareto 最优解。

2 仿真实验

为了充分地测试算法的性能，将本文提出的算法同其他经典的多目标优化算法 NSGA-II[1]、SPEA2[2]、PAES[15]在 ZDT[16]、WFG[17]系列测试函数及其他4个约束测试函数上进行对比实验。这些测试函数所具有的特性既有凸的也有非凸的，既有连续的也有不连续的，还有多峰的，可以充分地测试算法的性能。本文的所有对比实验均在同一台计算机上进行，所用编程语言为 Java。待测试的4种算法均采用实数编码，交叉算子为 SBX 交叉算子，变异算子为多项式变异算子，交叉概率和变异概率的设置同1.4节，种群大小均为100，最大迭代次数为250代。4种算法对每个测试函数均独立运行30次，然后统计指标 H_V、Δ、ε 的平均值和标准差，所得结果如表1～3所示。

从表1的 H_V 的平均值和标准差来看，在18个测试函数中，MNS-MOGA 获得了12个最优值，NSGA-Ⅱ获得了4个最优值，SPEA2 获得了2个最优值，PAES 没有获得最优值。这表明 MNS-MOGA 算法所获得的 Pareto 前端具有良好的分布性与覆盖性。并且 MNS-MOGA 获得了 ZDT 系列测试函数的全部5个最优值，表明该算法非常适合求解 ZDT 系列测试函数。

表1 H_V 标准平均值及方差

Tab.1 Mean and standard deviation of H_V

	MNS-MOGA	NSGA-Ⅱ	SPEA2	PAES
ZDT1	**0.661(1.7×10⁻⁴)**	0.659(3.6×10⁻⁴)	0.660(3.0×10⁻⁴)	0.657(1.3×10⁻³)
ZDT2	**0.328(2.7×10⁻⁴)**	0.326(3.6×10⁻⁴)	0.326(4.0×10⁻⁴)	0.320(8.1×10⁻³)
ZDT3	**0.551(8.4×10⁻⁴)**	0.515(1.7×10⁻⁴)	0.514(2.6×10⁻⁴)	0.477(4.4×10⁻²)
ZDT4	**0.658(3.1×10⁻³)**	0.655(2.9×10⁻³)	0.649(1.2×10⁻²)	0.634(1.9×10⁻²)
ZDT6	**0.398(4.8×10⁻⁴)**	0.389(1.4×10⁻³)	0.379(3.0×10⁻³)	0.383(3.6×10⁻²)
WFG1	0.460(1.2×10⁻¹)	**0.516(1.0×10⁻¹)**	0.396(1.0×10⁻¹)	0.083(2.2×10⁻²)
WFG2	0.562(1.2×10⁻³)	0.562(1.2×10⁻³)	**0.563(1.3×10⁻³)**	0.434(7.7×10⁻²)
WFG3	**0.442(2.5×10⁻⁴)**	0.441(2.5×10⁻⁴)	0.441(2.8×10⁻⁴)	0.428(6.0×10⁻³)
WFG4	**0.219(1.8×10⁻⁴)**	0.217(3.3×10⁻⁴)	0.218(2.2×10⁻⁴)	0.215(6.5×10⁻⁴)
WFG5	**0.196(5.2×10⁻⁵)**	0.195(4.7×10⁻³)	0.196(1.1×10⁻²)	0.194(1.4×10⁻³)
WFG6	0.168(3.4×10⁻²)	**0.204(4.7×10⁻³)**	0.200(1.1×10⁻²)	0.202(1.4×10⁻²)
WFG7	**0.210(1.1×10⁻⁴)**	0.209(3.1×10⁻⁴)	0.210(1.8×10⁻⁴)	0.204(1.3×10⁻³)
WFG8	0.152(1.4×10⁻²)	**0.153(1.3×10⁻²)**	0.149(6.7×10⁻³)	0.084(5.4×10⁻²)
WFG9	0.236(3.6×10⁻³)	0.237(1.1×10⁻³)	**0.239(1.4×10⁻³)**	0.147(9.4×10⁻²)
Golinski	**0.969(2.1×10⁻⁴)**	0.969(1.2×10⁻⁴)	0.967(6.6×10⁻⁴)	0.840(7.4×10⁻²)
Srinivas	**0.541(4.5×10⁻⁵)**	0.538(2.5×10⁻⁴)	0.540(1.4×10⁻⁴)	0.536(8.5×10⁻⁴)
Tanaka	**0.309(2.1×10⁻⁴)**	0.308(3.1×10⁻⁴)	0.308(4.6×10⁻⁴)	0.302(2.5×10⁻³)
Osyczka2	0.588(1.7×10⁻¹)	**0.746(9.3×10⁻³)**	0.723(2.2×10⁻²)	0.411(1.1×10⁻¹)

从表2的 Δ 指标来看，MNS-MOGA 获得了17个最优值，另外 NSGA-II 获得了1个最优值，说明该算法获得的 Pareto 前端的分布性要远远优于其他算法得到的 Pareto 前端的分布性。具体来看，MNS-MOGA 算法在 ZDT1、ZDT2、ZDT6 等测试函数上得到的 Δ 比其他算法得到的 Δ 小一个数量级，在 WFG 系列测试函数上得到的 Δ 也远小于其他算法得到的 Δ。

从表3的收敛性指标 ε 来看，MNS-MOGA 获得了11个最优值，NSGA-Ⅱ获得了5个最优值，SPEA2 获得了2个最优值，这表明 MNS-MOGA 算法得到的 Pareto 前端更逼近真实的 Pareto 前端，亦即表明其具有良好的收敛性。表2和表3的分布性指标 Δ 和收敛性指标 ε 也验证了表1的 H_V 指标。同时对比表1～3可得，在 H_V、Δ、ε 3个指标中，MNS-MOGA 在4个约束测试函数中均取得了3个最优值，表明其非常适合于求解带复杂约束的多目标优化问题。

MNS-MOGA 之所以能有如此的表现，分析其原因如下：① 研究表明，结构化的优化算法相比非结构化的算法往往具有更优异的性能。结构化算法在进化过程中能够保持拓扑不变性，其交叉变异都是在邻居结构中进行的，一定程度上避免了随机交配种群中交叉变异操作的盲目性，因而相比随机交配的种群而言，其选择压力相对较小，能够有效地加快收敛速度，同时避免陷入局部收敛。② MNS-MOGA 应用了局部迁移机制，在进化到一定程度时，自动交

表 2　Δ 标准平均值及方差

Tab. 2　Mean and standard deviation of Δ

	MNS-MOGA	NSGA-Ⅱ	SPEA2	PAES
ZDT1	**$7.86\times10^{-2}(1.2\times10^{-2})$**	$0.371(3.5\times10^{-2})$	$0.150(1.3\times10^{-2})$	$0.722(7.4\times10^{-2})$
ZDT2	**$8.19\times10^{-2}(1.4\times10^{-2})$**	$0.381(2.9\times10^{-2})$	$0.156(1.7\times10^{-2})$	$0.813(1.3\times10^{-1})$
ZDT3	**$0.708(4.7\times10^{-3})$**	$0.750(1.5\times10^{-2})$	$0.710(5.0\times10^{-3})$	$1.090(1.1\times10^{-1})$
ZDT4	**$0.119(2.3\times10^{-2})$**	$0.394(2.7\times10^{-2})$	$0.277(8.7\times10^{-2})$	$1.250(1.5\times10^{-1})$
ZDT6	**$8.64\times10^{-2}(1.0\times10^{-2})$**	$0.358(2.5\times10^{-2})$	$0.232(2.2\times10^{-2})$	$0.949(2.3\times10^{-1})$
WFG1	**$0.629(8.3\times10^{-2})$**	$0.719(4.3\times10^{-2})$	$0.706(1.1\times10^{-1})$	$1.140(4.7\times10^{-2})$
WFG2	**$0.752(5.5\times10^{-3})$**	$0.787(1.0\times10^{-2})$	$0.759(7.2\times10^{-3})$	$1.090(4.7\times10^{-2})$
WFG3	**$0.368(5.1\times10^{-3})$**	$0.589(2.2\times10^{-2})$	$0.434(8.1\times10^{-3})$	$0.834(4.2\times10^{-2})$
WFG4	**$0.139(1.8\times10^{-2})$**	$0.387(2.8\times10^{-2})$	$0.270(1.7\times10^{-2})$	$0.658(4.4\times10^{-2})$
WFG5	**$0.139(1.4\times10^{-2})$**	$0.410(2.8\times10^{-2})$	$0.281(1.8\times10^{-2})$	$0.676(4.6\times10^{-2})$
WFG6	**$0.164(4.1\times10^{-2})$**	$0.378(3.3\times10^{-2})$	$0.257(1.8\times10^{-2})$	$0.699(5.7\times10^{-2})$
WFG7	**$0.120(1.5\times10^{-2})$**	$0.377(2.9\times10^{-2})$	$0.252(1.7\times10^{-2})$	$0.701(4.5\times10^{-2})$
WFG8	**$0.610(6.6\times10^{-2})$**	$0.639(2.8\times10^{-2})$	$0.622(5.1\times10^{-2})$	$1.140(1.0\times10^{-1})$
WFG9	**$0.153(1.7\times10^{-2})$**	$0.403(2.8\times10^{-2})$	$0.298(2.0\times10^{-2})$	$0.692(1.4\times10^{-1})$
Golinski	**$0.155(8.3\times10^{-2})$**	$0.437(2.9\times10^{-2})$	$0.708(3.5\times10^{-2})$	$0.918(6.2\times10^{-2})$
Srinivas	**$0.701(9.2\times10^{-3})$**	$0.400(3.1\times10^{-2})$	$1.730(1.1\times10^{-2})$	$0.616(5.5\times10^{-2})$
Tanaka	**$0.738(3.1\times10^{-2})$**	$0.803(2.7\times10^{-2})$	$0.746(3.8\times10^{-2})$	$1.250(7.0\times10^{-2})$
Osyczka2	$0.636(1.5\times10^{-1})$	**$0.578(8.7\times10^{-2})$**	$0.740(9.4\times10^{-2})$	$1.140(1.1\times10^{-1})$

表 3　ε 标准平均值及方差

Tab. 3　Mean and standard deviation of ε

	MNS-MOGA	NSGA-Ⅱ	SPEA2	PAES
ZDT1	**$6.25\times10^{-3}(2.8\times10^{-4})$**	$1.36\times10^{-2}(3.0\times10^{-3})$	$8.94\times10^{-3}(6.5\times10^{-4})$	$2.42\times10^{-2}(1.2\times10^{-2})$
ZDT2	**$5.78\times10^{-3}(2.8\times10^{-4})$**	**$1.31\times10^{-2}(2.1\times10^{-3})$**	**$1.04\times10^{-2}(6.8\times10^{-3})$**	**$7.48\times10^{-2}(8.9\times10^{-2})$**
ZDT3	**$2.76\times10^{-2}(7.7\times10^{-2})$**	**$9.29\times10^{-3}(1.6\times10^{-3})$**	$9.89\times10^{-3}(1.3\times10^{-3})$	$3.90\times10^{-1}(3.3\times10^{-1})$
ZDT4	**$8.88\times10^{-3}(2.1\times10^{-3})$**	$1.60\times10^{-2}(5.9\times10^{-3})$	$5.48\times10^{-2}(5.8\times10^{-2})$	$6.28\times10^{-2}(4.8\times10^{-2})$
ZDT6	**$6.29\times10^{-3}(3.0\times10^{-4})$**	$1.50\times10^{-2}(1.6\times10^{-3})$	$2.43\times10^{-2}(3.5\times10^{-3})$	$4.92\times10^{-2}(9.5\times10^{-2})$
WFG1	$5.91\times10^{-1}(3.0\times10^{-1})$	**$4.83\times10^{-1}(3.1\times10^{-1})$**	$9.03\times10^{-1}(3.1\times10^{-1})$	$1.36(2.0\times10^{-1})$
WFG2	$5.47\times10^{-1}(3.0\times10^{-1})$	$4.09\times10^{-1}(3.4\times10^{-1})$	**$3.17\times10^{-1}(3.4\times10^{-1})$**	$1.69(5.6\times10^{-1})$
WFG3	$2.00(1.1\times10^{-3})$	**$2.00(9.3\times10^{-4})$**	$2.00(1.9\times10^{-3})$	$2.15(1.0\times10^{-1})$
WFG4	**$1.53\times10^{-2}(6.0\times10^{-4})$**	$3.44\times10^{-2}(6.3\times10^{-3})$	$2.59\times10^{-2}(3.0\times10^{-3})$	$8.11\times10^{-2}(4.8\times10^{-2})$
WFG5	**$6.40\times10^{-2}(1.4\times10^{-3})$**	$8.67\times10^{-2}(9.2\times10^{-3})$	$7.31\times10^{-2}(1.9\times10^{-3})$	$1.33\times10^{-1}(3.6\times10^{-2})$
WFG6	$8.94\times10^{-2}(6.7\times10^{-2})$	$3.88\times10^{-2}(1.0\times10^{-2})$	**$3.74\times10^{-2}(1.7\times10^{-2})$**	**$2.46\times10^{-1}(1.9\times10^{-1})$**
WFG7	**$1.52\times10^{-2}(9.1\times10^{-4})$**	$3.48\times10^{-2}(6.8\times10^{-3})$	$2.59\times10^{-2}(2.8\times10^{-3})$	$2.56\times10^{-1}(1.6\times10^{-1})$
WFG8	$4.41\times10^{-1}(1.1\times10^{-1})$	**$3.81\times10^{-1}(1.2\times10^{-1})$**	$4.65\times10^{-1}(1.0\times10^{-1})$	$2.15(7.8\times10^{-1})$
WFG9	**$2.25\times10^{-2}(5.4\times10^{-3})$**	$3.80\times10^{-2}(5.3\times10^{-3})$	$2.92\times10^{-2}(3.6\times10^{-3})$	$3.81\times10^{-1}(7.5\times10^{-1})$
Golinski	**$5.87(1.2\times10^{0})$**	$9.37(2.2\times10^{0})$	$1.49\times10^{1}(2.9\times10^{0})$	$9.95\times10^{1}(4.5\times10^{-1})$
Srinivas	**$1.31(8.8\times10^{-2})$**	$3.36(5.7\times10^{-1})$	$1.88(1.6\times10^{-1})$	$4.55(1.1\times10^{0})$
Tanaka	**$8.02\times10^{-3}(1.7\times10^{-3})$**	$8.69\times10^{-3}(1.1\times10^{-3})$	$8.32\times10^{-3}(1.9\times10^{-3})$	$2.95\times10^{-2}(9.1\times10^{-3})$
Osyczka2	$6.27\times10^{-1}(5.7\times10^{1})$	**$8.43(9.1\times10^{0})$**	$2.32\times10^{1}(1.1\times10^{1})$	$9.67\times10^{1}(4.3\times10^{1})$

换每个个体的邻居个体，避免了随着进化过程的进行种群中的个体趋于一致的情况，因而能够更好地保持种群的多样性，从而使算法得到的 Pareto 前端具有良好的分布性。③ 对于多目标优化算法而言，合理的精英策略能够保证算法具有较强的收敛性。外部文档作为一种常见的精英策略，也被证明是一种十分有效的精英策略。MNS-MOGA 应用外部文档存储进化过程中得到的非劣解，因而能够有效地保持收敛性。

3　桁架结构优化设计

3.1　桁架结构数学模型

对桁架结构进行优化设计时，一般来说结构的质量越轻，消耗的材料就越少，从而工程造价就越低，但是桁架截面太小，结构变形量会增大，因此考虑将结构的节点位移作为一个目标。以 25 杆桁架

结构系统为研究对象,在规定的静力载荷条件下,针对具体的研究对象,其基本参数(材料密度、弹性模量、最大许用应力、节点坐标等)已知,在满足结构给定的所有约束条件下,使得结构的质量最轻和变形最小[18-19]。桁架截面优化的数学模型为

$$\begin{cases} \boldsymbol{A}=(A_1,A_2,\cdots,A_n) \\ \min f_1=W=\sum_{i=1}^{n}\rho_i A_i L_i \\ \min f_2=\max(u_{1l},u_{2l},\cdots,u_{ml}) \quad (l=3) \\ \text{s.t.}\begin{cases} g_k^{\delta}(\boldsymbol{A})=[\delta_k]-\delta_k\geqslant 0 \quad (k=1,2,\cdots,K) \\ \boldsymbol{A}\in\{q_1,q_2,\cdots,q_n\} \text{或} \boldsymbol{A}_{\min}\leqslant\boldsymbol{A}\leqslant\boldsymbol{A}_{\max} \end{cases} \end{cases} \tag{8}$$

式中 $\boldsymbol{A}$——设计变量　　n——杆件组数
W——桁架结构质量
ρ_i——第 i 组杆件密度
A_i——第 i 组杆件截面积
L_i——第 i 组杆件长度
$[\delta_k]$——第 k 个杆件的许用应力
δ_k——第 k 个杆件在各个工况下的最大应力
u_{ml}——各工况下节点 m 在给定方向 l 上的位移
l——3维空间桁架坐标方向
K——杆件数　　f_1、f_2——子目标函数

当设计变量为连续变量时,$\boldsymbol{A}_{\min}\leqslant\boldsymbol{A}\leqslant\boldsymbol{A}_{\max}$,其中 $\boldsymbol{A}_{\min}$ 为边界最小值,$\boldsymbol{A}_{\max}$ 为边界最大值;当设计变量为离散变量时,$\boldsymbol{A}\in\{q_1,q_2,\cdots,q_n\}$,其中 $\{q_1,q_2,\cdots,q_n\}$ 为杆件截面可选择的截面集合。

3.2 工程实例

25杆桁架为空间桁架结构[20],平面示意图如图3所示,其中 $L=635$ mm。25杆桁架算例,包括2个子目标,分别为桁架结构质量 W 最小和节点1、2、3、4、5、6的最大位移最小。25杆桁架属于离散变量优化问题,杆件截面积分为8组,因此其自变量的维数为8,其中约束条件为每个杆件的应力约束条件,

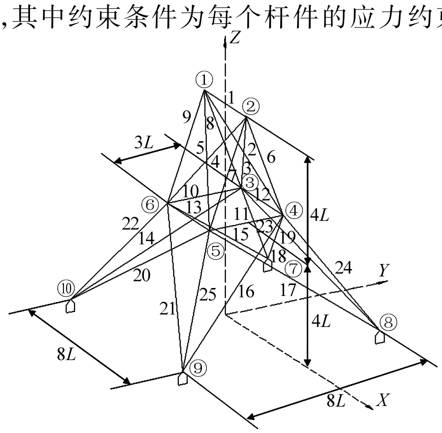

图3 25杆空间桁架

Fig.3 25-bar space truss

其基本参数见表4,载荷参数如表5所示,杆件分类如表6所示[21]。

表4 25杆桁架的参数

Tab.4 Parameters of 25-bar space truss

参数	数值
弹性模量 E/Pa	6.895×10^{10}
密度 ρ/(kg·m^{-3})	2 768.0
许用应力 σ/MPa	±275.8
离散设计变量 D/mm^2	645.16×{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2, 3.3, 3.4}

表5 25杆桁架节点载荷

Tab.5 Loading conditions of 25-bar space truss

节点号	F_x/kN	F_y/kN	F_z/kN
1	4.448	-44.48	-44.48
2	0	-44.48	-44.48
3	2.224	0	0
6	2.669	0	0

表6 25杆件分组

Tab.6 Group members of 25-bar space truss

组号	杆件号	组号	杆件号
X_1	1	X_5	12,13
X_2	2,3,4,5	X_6	14,15,16,17
X_3	6,7,8,9	X_7	18,19,20,21
X_4	10,11	X_8	22,23,24,25

为了进一步验证MNS-MOGA算法的性能,同时也与当前最流行的多目标遗传算法NSGA-Ⅱ对桁架结构多目标优化问题进行计算对比。这2种算法的参数设置如下:种群规模大小 $N_p=100$,外部存档容量为100,交叉概率 $p_c=0.9$,变异概率 $p_m=1/l_{en}$,其中 l_{en} 为维变量维数。图4为MNS-MOGA算法获得的Pareto前端,图5为NSGA-Ⅱ算法获得的Pareto前端。由图4和图5,可以直观地得出MNS-MOGA

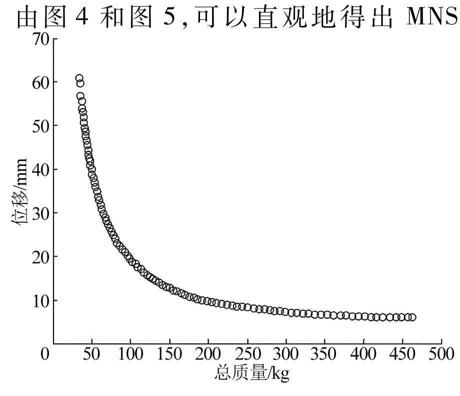

图4 MNS-MOGA获得的Pareto前端

Fig.4 Pareto front of MNS-MOGA

算法的 Pareto 前端分布更加均匀,说明算法的分布性更好,种群的多样性较优。图 6 为 2 种算法获得的 Pareto 前端比较,2 种算法获得极端值相同,但是在 Pareto 前端的中部,MNS-MOGA 算法获得解能够支配 NSGA-Ⅱ的部分解,而中间部分是决策者选取方案的重要依据,由此可以说明 MNS-MOGA 算法的收敛性更好,且能够为决策提供更多有效的方案。

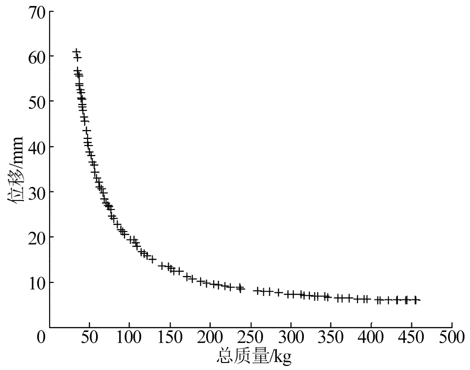

图 5　NSGA-Ⅱ获得的 Pareto 前端

Fig. 5　Pareto front of NSGA-Ⅱ

4　结论

(1) 提出了一种多邻域结构的多目标遗传算法(MNS-MOGA)。该算法依据个体之间的欧氏距离

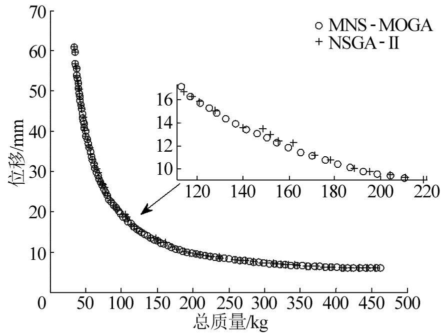

图 6　MNS-MOGA 和 NSGA-Ⅱ获得的 Pareto 前端比较

Fig. 6　Comparison of Pareto front between MNS-MOGA and NSGA-Ⅱ

信息,将种群划分为多邻域结构,即每个个体拥有固定的邻居个体,且每个个体只能与其邻居个体进行交叉、变异操作。通过 Friedman 实验分析,综合得出在邻居规模为 10 时,算法的综合性能最优。将提出的算法与 NSGA-Ⅱ、SPEA2 和 PEAS 算法,在 18 个基准测试函数上仿真实验对比分析,结果表明 MNS-MOGA 算法收敛性和分布性较好,尤其在多样性方面具有较明显的优势。

(2) 将 MNS-MOGA 算法应用于桁架结构多目标优化设计问题中,并与 NSGA-Ⅱ相比较,结果表明,该算法获得的 Pareto 前端更加均匀,且收敛性更好,能够为决策提供更多有效的方案。

参　考　文　献

1　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.

2　Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm [M]// Giannakoglou K, Tsahalis D T, Périaux J, et al. Evolutionary methods for design, optimization and control with applications to industrial problems. Berlin: Springer-Verlag, 2002:95 – 100.

3　Gong M G, Jiao L C, Du H F, et al. Multiobjective immune algorithm with nondominated neighbor-based selection [J]. Evolutionary Computation, 2008, 16(2):225 – 255.

4　Li Hui, Zhang Qingfu. Multi-objective optimization problems with complicated Pareto sets, MOEA/D and NSGA-Ⅱ [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 284 – 302.

5　Enrique Alba, Bernabé Dorronsoro. Cellular genetic algorithms[M]. New York: Springer-Verlag, 2008.

6　Alba E, Dorronsoro B. The exploration/exploitation tradeoff in dynamic cellular genetic algorithms [J]. IEEE Transactions on Evolutionary Computation, 2005, 9(2):126 – 141.

7　Nebro A J, Durillo J J, Luna F, et al. A cellular genetic algorithm for multi-objective optimization [J]. International Journal of Intelligent Systems, 2009, 24(7):726 – 746.

8　Al-Naqi A, Erdogan A T, Arslan T. Adaptive three-dimensional cellular genetic algorithm for balancing exploration and exploitation processes[J]. Soft Computing, 2013, 17(7): 1145 – 1157.

9　Dorronsoro B, Bouvry P. Cellular genetic algorithms without additional parameters [J]. Journal of Supercomputing, 2013, 65(3): 816 – 835.

10　朱大林,詹腾,张屹,等. 元胞多目标粒子群优化算法与其应用[J]. 农业机械学报,2013,44(12):280 – 287.
Zhu Dalin, Zhan Teng, Zhang Yi, et al. Algorithm and application of cellular multi-objective particle swarm optimization [J]. Transactions of the Chinese Society for Agricultural Machinery, 2013, 44(12):280 – 287. (in Chinese)

11　Durillo J J, Nebro A J. jMetal: a java framework for multi-objective optimization [J]. Advances in Engineering Software,2011, 42(10): 760 – 771.

12　Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach [J]. IEEE Transactions on Evolutionary Computation, 1999, 3(4):257 – 271.

(下转第 324 页)

13 Messac A, Ismail-Yahaya A, Mattson C. The normalized constraint method for generating the Pareto frontier [J]. Structural and Multidisciplinary Optimization, 2003, 25(2):86 - 98.
14 谭艳艳.几种改进的分解类多目标进化算法及其应用[D].西安:西安电子科技大学,2013.
Tan Yanyan. Several modified decomposition-based multi-objective evolutionary algorithms and their applications [D]. Xi'an: Xidian University,2013. (in Chinese)
15 Price K, Storn R, Lampinen J. Differential evolution: a practical approach for global optimization [M]. Berline: Springer Verlag, 2005.
16 Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results [J]. Evolutionary Computation, 2000, 8(2):173 - 195.
17 Fonseca C M, Knowles J D, Thiele L, et al. A tutorial on the performance assessment of stochastic multiobjective optimizers[C]// Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005), 2005, 216: 240.
18 Corne D W, Jerran N R, Knowles J D, et al. PESA-II: region-based selection in evolutionary multi-objective optimization[C]// Proceedings of the Genetic and Evolutionary Computation Conference, 2001:283 - 290.
19 Zitzler E, Thiele L, Laumanns M, et al. Performance assessment of multi-objective optimizers: an analysis and review [J]. IEEE Transactions on Evolutionary Computation, 2003, 7(2):117 - 132.
20 Ishibuchi H, Akedo N, Nojima Y. Relation between neighborhood size and MOEA/D performance on many-objective problems [C]// Evolutionary Multi-Criterion Optimization. Berlin, Heidelberg: Springer, 2013: 459 - 474.
21 Deb K, Agrawal R B. Simulated binary crossover for continuous search space [J]. Complex Systems, 1995, 9(2):115 - 148.
22 卢青波,张学良,温淑花,等 基于差异演化算法的动压滑动轴承多目标优化[J]. 农业机械学报,2013,44(3):230 - 236.
Lu Qingbo, Zhang Xueliang, Wen Shuhua, et al. Multi-objective optimization of hydrodynamic sliding bearing based on differential evolution algorithm [J]. Transactions of the Chinese Society for Agricultural Machinery, 2013, 44(3):230 - 236. (in Chinese)
23 张鄂,蒙娟,贾焕如.液体动压径向滑动轴承的概率多目标优化设计[J].机械科学与技术,2001,20(2):224 - 226.
Zhang E, Meng Juan, Jia Huanru. The multi-aim probability optimal optimization of hydrodynamic sliding bearing [J]. Mechanical Science and Technology, 2001, 20(2):224 - 226. (in Chinese)
24 王增胜,刘保国,吴磊,等.齿轮箱流体动压滑动轴承的多目标优化[J].机械传动,2007,31(5):74 - 75.
Wang Zengsheng, Liu Baoguo, Wu Lei, et al. Multi-objective optimization of hydrodynamic sliding bearing used in gearbox [J]. Journal of Mechanical Transmission, 2007, 31(5):74 - 75. (in Chinese)
25 Deb K. An efficient constraint handling method for genetic algorithms [J]. Computation Methods in Applied Mechanics and Engineering, 2000, 86(2 - 4):311 - 338.

(上接第 315 页)

13 王勇,蔡自兴,周育人,等. 约束优化进化算法[J].软件学报,2009, 20(1):11 - 29.
Wang Yong, Cai Zixing, Zhou Yuren, et al. Constrained optimization evolutionary algorithms [J]. Journal of Software, 2009, 20(1):11 - 29. (in Chinese)
14 Deb K. An efficient constraint handling method for genetic algorithms [J]. Computation Methods in Applied Mechanics and Engineering, 2000, 86(2 - 4):311 - 338.
15 Knowles J D, Corne D W. Approximating the nondominated front using the Pareto archived evolution strategy [J]. Evolutionary Computation, 2000, 8(2):149 - 172.
16 Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results [J]. Evolutionary Computation, 2000,8(2):173 - 195.
17 Huband S, Hingston P, Barone L, et al. A review of multiobjective test problems and a scalable test problem toolkit [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(5):477 - 506.
18 Pholdee N, Bureerat S. Hybrid real-code population-based incremental learning and approximate gradients for multi-objective truss design [J]. Engineering Optimization, 2014, 46(8):1032 - 1051.
19 Gandomi A H, Talatahari S, Yang X S, et al. Design optimization of truss structures using cuckoo search algorithm [J]. Structural Design of Tall and Special Buildings, 2013,22(17):1330 - 1349.
20 唐和生,胡长远,薛松涛.桁架结构多目标优化的免疫克隆选择算法[J].湖南大学学报:自然科学版,2013,40(5):18 - 23.
Tang Hesheng, Hu Changyuan, Xue Songtao. Immune clone selection algorithm for truss structure multi-objective optimization [J]. Journal of Hunan University: Natural Sciences, 2013,40(5):18 - 23. (in Chinese)
21 张卓群,李宏男.基于蚁群算法的桁架结构布局离散变量优化方法[J].计算力学学报,2014,30(3):336 - 342.
Zhang Zhuoqun, Li Hongnan. The method of truss structure layout discrete variable optimization based on ant colony optimization [J]. Chinese Journal of Computational Mechanics,2014,30(3):336 - 342. (in Chinese)