

文章编号:1001-9081(2009)10-2719-04

一种带有随机变异的动态差分进化算法

高岳林,刘俊梅

(北方民族大学 信息与系统科学研究所,银川 750021)

(gaoyuelin@263.net)

摘要:提出一种带有随机变异的动态差分进化算法。在这个算法中,两种不同的变异策略 DE/rand/1 和 DE/best/1 通过线性递减加权组合策略产生新的变异策略,以便动态利用 DE/rand/1 和 DE/best/1 的优点,并且引入一种指数递增交叉概率算子、线性递减缩放因子和一种随机变异机制以进一步提高算法的全局寻优能力。通过四个标准测试函数的测试结果表明,该算法是一种收敛速度快、求解精度高、鲁棒性较强,更适合求解高维复杂的全局优化问题。

关键词:全局优化;差分进化算法;加权策略;指数递增交叉概率;随机变异

中图分类号: TP18 **文献标志码:** A

Dynamic differential evolution algorithm with random mutation

GAO Yue-lin, LIU Jun-mei

(Institute of Information and System Science, North National University, Yinchuan Ningxia 750021, China)

Abstract: The dynamic Differential Evolution (DE) algorithm with random mutation was proposed. In this algorithm, the mutation strategies of DE/rand/1 and DE/best/1 were combined by linear decreased weight convex combination strategy to produce a new mutation strategy so as to dynamically use the advantages of DE/rand/1 and DE/best/1. In order to improve the global optimization ability of DE algorithm, exponent increased crossover probability operator, linear decreasing scaling factor and random mutation mechanism were introduced. The test results on the four standard test functions show that the new algorithm has fast convergence, high accuracy and more robustness, more suitable to solve high-dimensional complex global optimization problems.

Key words: global optimization; Differential Evolution (DE) algorithm; weighted strategy; exponent increased crossover probability; random mutation

0 引言

差分进化(Differential Evolution, DE)算法是^[1]一种基于种群并行随机搜索的新型进化算法。该算法从原始种群开始,通过变异、杂交、选择几种遗传操作来衍生出新的种群,不断地迭代计算,并根据每一个个体的适应度值,保留优良个体,淘汰劣质个体,引导搜索过程向最优解逼近,可实现全局最优解的搜索。近年来 DE 在函数优化、实际工程、化工等诸多领域得到广泛应用。文献[2]作者把差分进化算法用于求解混合整数非线性规划,文献[3]作者把差分进化算法用于求解非线性约束优化问题,可广泛应用于工程实践中。

近年来 DE 多用于求解非线性约束优化问题。但是差分进化算法作为一种较新的进化算法也有其缺陷,即优化迭代后期接近最优解时收敛速度缓慢,易陷入局部最优,而且参数对算法的结果有很大的影响。为了克服 DE 算法的缺陷,国内许多学者对 DE 算法进行改进。文献[4]作者提出一种新的差分进化算法,文献[5]作者采用差分进化算法与粒子群优化算法相结合的混合算法来求解全局优化问题,文献[6]作者对 DE 算法的参数进行自适应的调整。数值模拟结果表明这些改进后的算法在一定程度上改善了 DE 算法在迭代后期收敛速度慢,易陷入局部最优的缺陷。

为了克服 DE 算法在求解全局优化问题时的缺陷,我们对 Price 和 Storn 提出了十种变异策略进行实验分析,结果表

明,有两个变异策略(DE/best/1 和 DE/rand/1^[7])效果尤为明显,其中 DE/rand/1 版本全局搜索能力强,是用差分进化算法来解决问题的首选方法。DE/best/1 版本局部搜索能力强,可以提高算法的收敛精度和收敛速度。结合两个版本的优点,本文提出一种改进的差分算法,即动态变异差分进化(Dynamic Mutation's Differential Evolution, DMDE)算法,把 DE/best/1 和 DE/rand/1 通过线性模拟退火加权策略组合成新的变异操作,用线性退火因子作为加权因子^[3],使得算法初始阶段有较强的全局搜索能力,尽可能多地发现可能的全局最优点,而在后阶段则应有较强的局部搜索能力,以提高算法的收敛精度和收敛速率。使用一种指数递增算子^[4]和线性递减算子来动态调整交叉概率和收缩因子,以此平衡全局搜索能力和局部搜索能力,使算法快速收敛到全局最优解。同时采用一种变异机制,对陷入停滞现象的个体进行随机变异^[5],增强种群的多样性。通过以上改进,使得本文给出的新算法收敛速度快,求解精度高,鲁棒性强,更适合求解高维复杂优化问题。

1 标准差分进化算法及指数递增交叉概率因子

1.1 标准差分进化算法

对于一个最小化问题 $\min F(x)$, DE 从包含 N 个候选解的初始种群 x_i^t ($i = 1, 2, \dots, N$) 开始,其中 i 是个体在种群中的序号, t 为当前代。

收稿日期:2009-04-07;修回日期:2009-06-16。

基金项目:国家自然科学基金资助项目(60962023);国家社会科学基金资助项目(60962006);宁夏自然科学基金资助项目(NZ0848)。

作者简介:高岳林(1963-),男,陕西榆林人,教授,博士,主要研究方向:最优化理论方法、智能计算、金融数学与金融工程;刘俊梅(1980-),女,陕西榆林人,硕士研究生,主要研究方向:最优化理论方法、智能计算。

变异操作

1) DE/rand/1 版本:

$$v_i^t = x_{r1}^t + F(x_{r2}^t - x_{r3}^t) \quad (1)$$

2) DE/best/1 版本

$$v_i^t = x_{\text{best}}^t + F(x_{r2}^t - x_{r3}^t) \quad (2)$$

其中 v_i^t 为根据式(1)和(2)产生的任一随机向量。 $r1, r2, r3 \in \{1, 2, \dots, N\}$ 是随机整数, $r1, r2, r3$ 互不相等且不等于 i , x_{best}^t 为种群中适应值最好的个体, $F \in [0, 1]$ 为缩放因子。

交叉操作

新种群 $y_i^t = [y_{i1}^t, y_{i2}^t, \dots, y_{id}^t]$ 由随机向量 $v_i = [v_{i1}, v_{i2}, \dots, v_{id}]$ 和目标向量 $x_i = [x_{i1}, x_{i2}, \dots, x_{id}]$ 共同产生。

$$y_{ij} = \begin{cases} v_{ij}, & \text{rand}(b(j)) \leq CR \text{ 或 } j = \text{rand}(r(i)) \\ x_{ij}, & \text{rand}(b(j)) > CR \text{ 且 } j \neq \text{rand}(r(i)) \end{cases} \quad (3)$$

其中: $j \in [1, D]$, $\text{rand}(b(j)) \in [0, 1]$ 是同一随机数发生器的第 j 个值。 $CR \in [0, 1]$ 是变异概率。 $\text{rand}(r(i)) \in [1, 2, \dots, D]$ 是随机选择指数, 它确保 x_i^t 能从 v_i^t 中得到至少一个参数。

选择操作

$$x_i^{t+1} = \begin{cases} x_i^t, & \phi(x_i^t) < \phi(x_i^t) \\ x_i^t, & \text{其他} \end{cases} \quad (4)$$

采用贪婪策略, 其中 $\phi(x)$ 代表适应函数。

1.2 指数递增交叉概率因子

为了提高算法中 DE 算法的优化性能, 我们加入了一种指数递增交叉概率策略。交叉概率因子的更新状态如下^[4]:

$$CR = CR_{\min} + (CR_{\max} - CR_{\min}) \exp(-a(1 - t/T_{\max})^b) \quad (5)$$

其中: $a = 30$, $b = 3$, t 为当前迭代次数, T_{\max} 为最大迭代次数, 参数 $CR_{\min} = 0.1$, $CR_{\max} = 0.9$ 。该交叉概率能良好地平衡全局搜索能力和局部搜索能力, 使算法快速收敛到最优解。

2 改进的差分进化算法

2.1 新的变异操作

根据变异个体的生成方法不同, 形成了多种不同的差分进化算法方案。DE 有十种基本的变异版本, 其中 DE/rand/1 版本变异个体 v_i^t 由三个互不相同的随机个体组成, 无需任何适应值信息, 有利于保持种群的多样性, 因而全局搜索能力强, 但收敛速度慢。DE/best/1 版本变异个体 v_i^t 由 x_{best}^t 作引导, 因而局部搜索能力强, 精度高, 收敛速度快, 但会加大算法陷入局部最优点的可能性。结合这两种不同变异方式的特点, 我们更希望把这两个版本进行折中, 取长补短, 前期重视 DE/rand/1 版本, 后期重视 DE/best/1 版本, 在变异方程中同时考虑随机个体 x_{r3}^t 和最优个体 x_{best}^t 的作用。本文加入一种新的变异方案, 其变异操作^[3] 方程为:

$$v_i^t = \lambda x_{r1}^t + (1 - \lambda)x_{\text{best}}^t + F(x_{r2}^t - x_{r3}^t) \quad (6)$$

其中, $\lambda \in [0, 1]$ 。若 $\lambda = 1$, 则式(6)等价于式(1), 变成 DE/rand/1 方案; 若 $\lambda = 0$, 则式(6)等价于式(2), 变成 DE/best/1 方案。一般来说, 良好的算法要求在初始阶段有较强的全局搜索能力, 尽可能多地发现可能的全局最优点, 而在后阶段则应有较强的局部搜索能力, 以提高算法的精度和收敛速度。 λ 在搜索过程中由 1 逐渐变化为 0 的方式不同就形成不同的差分进化算法, 因此可以形成一大类差分进化算法, 通过对 λ 取值的详细测试, 本文引入线性模拟退火策略, 将 λ 设置为线性退火因子^[3], 如式(6)所示。 λ 在搜索过程中由 1 逐渐变化为 0, 使得 x_{r1} 的权重逐渐减小而 x_{best}^t 的权重逐渐增加, 从而保证算法既有较强的全局搜索又有较快的收敛速度和搜

索精度。

$$\lambda = (T_{\max} - t)/T_{\max} \quad (7)$$

其中: T_{\max} 表示最大迭代次数, t 表示当前迭代次数。式(6)中的 $F = \lambda \times 0.5 + 0.5$, 随 t 增大, F 线性递减。

2.2 停滞现象的判定和处理策略

为了避免算法中的个体在优化迭代后期出现停滞现象, 陷入早熟收敛, 引入一种变异机制, 对聚集个体进行随机变异, 进而增强种群的多样性。如果个体在预定的最大迭代次数内出现停滞现象, 那么该个体将随机变异^[5], 即被搜寻空间中的任一新位置所代替。实现如下:

$$\text{if } F(x_i^t) = F(x_i^{t+1}) = F(x_i^{t+2}) = \dots = F(x_i^{t+p}) \text{ and } F(x_i^t) \neq F^* \quad (8)$$

$$\text{then } x_i^{t+p+1} = x_{\min} + \text{rand}(0, 1) \times (x_{\max} - x_{\min})$$

其中: F^* 代表适应度函数的全局最小值, p 是允许停滞的最大迭代次数, (x_{\min}, x_{\max}) 是定义的允许搜寻范围。

2.3 新算法 MDE 描述

算法实现步骤如下。

- 1) 基本参数设置, 包括: 群体规模 N , 最大迭代次数 T_{\max} , 缩放因子 F , 交叉概率的上下界 CR_{\min}, CR_{\max} 。
- 2) 设置迭代计数器 $t = 0$ 。
- 3) 根据式(3 ~ 7)对 DE 群体中所有个体执行变异、交叉、选择操作。
- 4) 判断个体是否有停滞现象, 如果有, 按式(8)执行变异操作。
- 5) 更新迭代计数器 $t = t + 1$, 并记录当前整个群体中最佳个体, 如果满足精度要求或整个进化已达到最大迭代次数, 则终止算法, 否则转至 3)。

3 测试函数

为了测试 DMDE 的性能, 本文采用了文献[6]中常用的一系列基准测试函数, 并与两个不同版本变异策略的 DE 算法(BDE 和 RDE), 两个变异策略线性组合的 MDE 算法进行了比较。笔者测试了 10 个基准函数, 都取得较好的效果, 对于高维复杂的多峰函数效果更为明显, 限于篇幅, 笔者选取了四个高维复杂多峰函数 f_1, f_2, f_3, f_4 , 这些函数局部最优点的数量随维数指数递增, 对这四个函数进行测试, 可有效检验算法的全局搜索性能、收敛速度和收敛精度。

Griewank 函数:

$$f_1(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

自变量的取值范围为 $[-600, 600]$, 维数为 30, 函数理论最优值 0。

Ackley 函数:

$$f_2(x) = -20 \exp\left[-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right] - \exp\left(\sum_{i=1}^n \cos(2\pi x_i)/n\right) + 20 + e$$

自变量的取值范围为 $[-32, 32]$, 维数为 30, 函数理论最优值 0。

Generalized Penalized 函数:

$$f_3(x) = \frac{\pi}{n} \{10 \sin^2(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1}) + (y_n - 1)^2]\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$$

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$$

其中 $y_i = 1 + \frac{1}{4}(x_i + 1)$ 。

$$f_4(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 \right\} + \sum_{i=1}^n u(x, 5, 100, 4)$$

$f_3(x)$ 和 $f_4(x)$ 共用 $u(x_i, a, k, m)$ 函数, 并且自变量的取值范围为 $[-50, 50]$, 维数为 30, 函数理论最优值 0。

4 数值实验

为了说明 DMDE 算法的性能, 下面我们将 DMDE 算法与 MDE 算法、RDE 算法、BDE 算法进行比较。MDE 算法表示采

用 DE/rand/1 和 DE/best/1 的线性组合进行变异, 其中 CR 、 F 取为常数的算法。RDE 算法表示采用 DE/rand/1 版本变异, 其中 CR 、 F 取为常数的算法。BDE 算法表示采用 DE/best/1 版本变异, 其中 CR 、 F 取为常数的算法。其中 DMDE 算法中的参数设置为 $CR_{\min} = 0.1, CR_{\max} = 0.9$ 。

其他三个算法的参数设置 $CR = 0.9, F = 0.5$ 。最大允许停滞次数 $P = 20$, 四种算法群体规模大小皆为 60, 最大迭代次数取 1500。每个实验设置运行 20 次, 取 20 次的平均值作为最终结果。

表 1 给出了四种算法对 4 个测试函数在上面参数设置时的测试结果, 包括最好值、最差值、平均适应值以及结果运行 20 次的标准差。图 1~4 分别给出了 4 个基准测试函数的平均最好适应值(实验重复 20 次的最优适应值的平均值)进化曲线。

表 1 所有算法对四个标准测试函数的测试结果

函数	算法	最好值	最差值	平均值	标准差
f_1	MDE	0	0	0	0
	BDE	0	0.9683	0.0981	0.0491
	RDE	0	0	0	0
	DMDE	0	0	0	0
f_2	MDE	3.3478E-010	0.0639	2.2130E-008	1.1374E-008
	BDE	6.2314	14.4369	7.5934	3.7998
	RDE	2.6334E-012	19.8664	6.5734E-012	1.3178E-012
	DMDE	3.4007E-015	4.4409E-015	4.2188E-015	2.1111E-015
f_3	MDE	4.0251E-025	0.7320	1.5384E-021	5.7912E-022
	BDE	0.1037	8.3947	1.5757	0.7885
	RDE	2.1793E-022	3.7316	2.2602E-020	1.1310E-020
	DMDE	1.5705E-032	4.1521E-032	1.9825E-032	9.9205E-033
f_4	MDE	2.7917E-032	2.8743E-025	1.5697E-020	7.3452E-021
	BDE	1.7044E-030	839.0652	61.3437	30.6966
	RDE	1.6683E-022	4.4924E-021	8.7014E-022	4.3542E-022
	DMDE	1.0098E-032	3.8150E-032	1.2823E-032	6.4166E-033

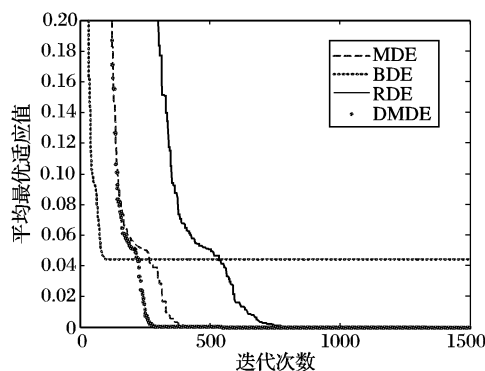


图 1 f_1 的进化曲线

通过表 1 和图 1~4 可以看出: 对于 f_1 , BDE 算法陷入早熟, MDE 算法在收敛精度上优于 BDE 算法, 在收敛速度上优于 RDE 算法, 而 DMDE 算法在收敛精度、收敛速度上明显优于 MDE 算法。对于 f_2 , BDE 陷入早熟, MDE 算法在收敛精度上优于 BDE 算法, 在收敛速度上优于 RDE 算法, 而 DMDE 算法在收敛精度上明显优于 MDE 算法。对于 f_3 , BDE 陷入早熟, MDE 算法在收敛精度收敛速度上优于 BDE 算法、RDE 算法, 而 DMDE 算法在收敛精度上明显优于 MDE 算法。

对于 f_4 , RDE 算法陷入早熟, MDE 算法在收敛精度上优于 RDE 算法, 在收敛速度上优于 BDE 算法, 而 DMDE 算法在

收敛速度上明显优于 MDE 算法。在收敛精度上通过数据实验表明线性加权策略弥补了两个版本各自的缺陷; 而加入动态交叉策略, 又能使得算法收敛精度增强, 故 DMDE 算法在求解精度、收敛速度、鲁棒性方面优于另外三种算法。特别是针对高维多峰函数, 本文提出的算法显示出了一定的优势。

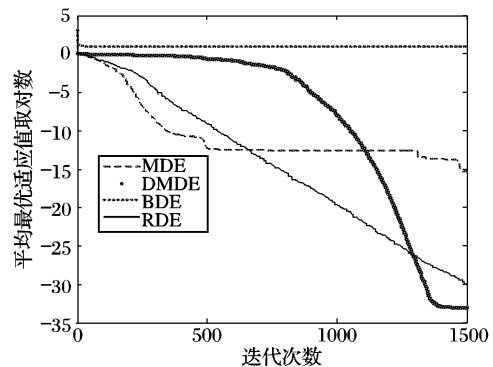


图 2 f_2 的进化曲线

5 结语

本文提出一种带有随机变异的动态差分进化算法——DMDE 算法, 我们采用四个标准测试函数来测试 DMDE 算法的性能, 并于 MDE 算法、BDE 算法、RDE 算法相比较。所有

的实验证明了本文提到的算法针对高维多峰函数具有较好的收敛速度、求解精度和鲁棒性。

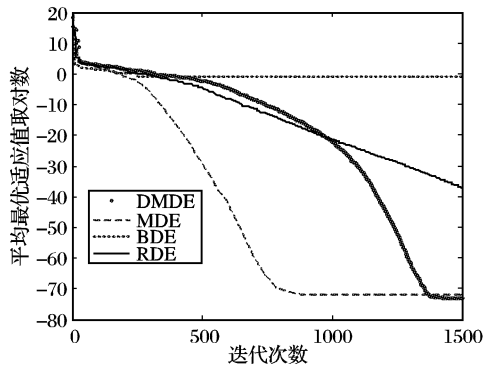


图 3 f_3 的进化曲线

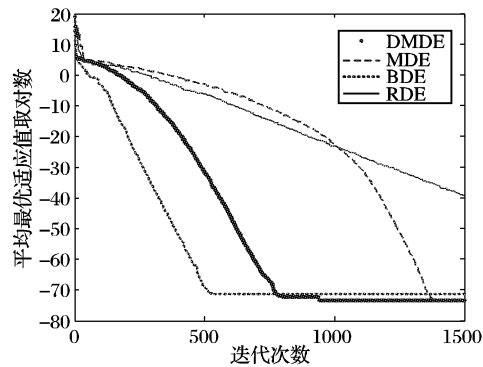


图 4 f_4 的进化曲线

(上接第 2718 页)

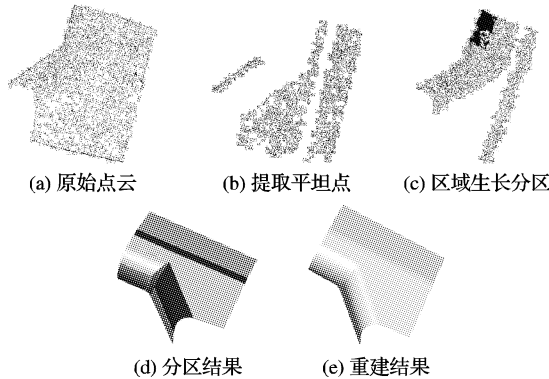


图 5 基于区域生长的点云分区

4 结语

本文依照表面曲率的特征粗略分割点云,用多项式逼近 K -邻近和二次提取法提高了计算效率。与 Leonardis 等提出的重复选择区域生长法^[10]比较,本文的方法更强调种子区域的选择,不允许区域重叠,特别是使用曲面曲率和棱边检测去选择种子区域以利于生长。这样可以创建较少的种子区域以减少区域生长的工作量,在面对海量数据时这一优点尤为重要。最后,本文引入了点与曲面 G^0 和 G^1 兼容的概念以确保更加精确的区域生长。基于区域生长的点云分区方案通过循环生长并重建使分区更为精确。图 5 的钣金件点云分区与重建证明提出的分区方法是必要和可行的。将来的工作可进一步放在如何预加工最初的点数据、精炼区域的粗糙分割、减少噪声对生长的影响上。

参考文献:

- [1] YE B, ZHU C Z, GUO C X, *et al.* Generating extended fuzzy basis function networks using hybrid algorithm [C]// Proceedings of the 2nd International Conference on Fuzzy Systems and knowledge Discovery. Berlin: Springer-Verlag, 2006: 79 - 88.
- [2] 吴亮红,王耀南,陈正龙. 求解混合整数非线性规划问题的改进差分进化算法[J]. 小型微型计算机系统, 2007, 28(4): 666 - 669.
- [3] 吴亮红,王耀南,周少武,等. 采用非固定多段映射罚函数的非线性约束优化差分进化算法[J]. 系统工程理论与实践, 2007, 27(3): 128 - 133.
- [4] 邓泽喜,曹敦虔,刘晓冀,等. 一种新的差分进化算法[J]. 计算机工程与应用, 2008, 44(24): 40 - 42.
- [5] 栾丽君,谭立静,牛奔. 一种基于粒子群优化算法和差分进化算法的新型混合全局优化算法[J]. 信息与控制, 2007, 36(6): 708 - 714.
- [6] BREST J, GREINER S, BOSKOVIC B, *et al.* Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(6): 646 - 657.
- [7] STORN R. On the usage of differential evolution for function optimization [C]// Proceedings of the Biennial Conference of the North American Fuzzy Information Processing Society. Piscataway: IEEE Press, 1996: 519 - 523.

参考文献:

- [1] 贺美芳. 基于散乱点云数据的曲面重建关键技术研究[D]. 南京: 南京航空航天大学, 2006.
- [2] YANG M, LEE E. Segmentation of measured point data using a parametric quadric surface approximation [J]. Computer Aided Design, 1999, 31(7): 449 - 457.
- [3] HUANG J, MENQ C H. Automatic data segmentation for geometric feature extraction from unorganized 3-D coordinate points [J]. IEEE Transactions on Robotics and Automation, 2001, 17(3): 268 - 279.
- [4] BESL P J, JAIIN R C. Segmentation through variable-order surface fitting [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1988, 10(2): 167 - 192.
- [5] YOKOYA N, LEVINE M D. Range image segmentation based on differential geometry: A hybrid approach [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989, 11(6): 644 - 649.
- [6] SAPIDIS N S, BESL P J. Direct construction of polynomial surfaces from dense range images through region growing [J]. ACM Transactions on Graphics, 1995, 14(2): 171 - 200.
- [7] 周儒荣,张丽艳,苏旭,等. 海量散乱点的曲面重建算法研究 [J]. 软件学报, 2001, 12(2): 249 - 255.
- [8] 金涛,童水光. 逆向工程技术 [M]. 北京: 机械工业出版社, 2003: 187 - 189.
- [9] HE MEI-FANG, ZHOU LAI-SHU. Segmentation of scattered point data through a new curvature analysis algorithm [J]. Journal of Southeast University: English Edition, 2004, 20(1): 91 - 92.
- [10] LEONARDIS A, JAKLIC A, SOLINA F. Superquadrics for segmenting and modeling range data [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19(11): 1289 - 1295.