

文章编号: 1001-0920(2007)10-1155-05

一种基于相似度的新型粒子群算法

刘建华^{1,2}, 樊晓平¹, 瞿志华¹

(1. 中南大学 信息科学与工程学院, 长沙 410083; 2. 福建师范大学 数学与计算机科学学院, 福州 350007)

摘要: 分析了基本粒子群算法(PSO)全局搜索能力与收敛速度的矛盾,提出了粒子群相似度的概念. 根据每个粒子与全局最优粒子的不同相似度,对基本 PSO 算法的惯性权重进行动态调整. 同时提出一种根据相似度计算聚集度的方法,并根据聚集度的大小随机地对粒子重新赋值,控制粒子群的多样性,提高了全局搜索能力. 典型优化问题的实例仿真验证了该算法的有效性.

关键词: 粒子群算法; 全局最优性; 相似度; 聚集度

中图分类号: TP18 **文献标识码:** A

A new particle swarm optimization algorithm based on similarity

LIU Jian-hua^{1,2}, FAN Xiaoping¹, QU Zhi-hua¹

(1. College of Information Science and Engineering, Central South University, Changsha 410083, China; 2. College of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China. Correspondent: FAN Xiaoping, E-mail: xpfan@ieee.org)

Abstract: The contradiction of the global exploration and convergence speed of particle swarm optimization (PSO) algorithm is analyzed. A new PSO algorithm is proposed, in which the inertia weight of every particle will be changed dynamically with the similarity between the particle and the current optimal position. And based on similarity, a method of collection calculation is proposed, which is used to randomly initialize the position of the particle, control the diversification of PSO, and improve the ability of global exploration. Experiments on benchmark functions show the effectiveness of the new algorithm.

Key words: Particle swarm optimization algorithm; Global optimality; Similarity; Collection

1 引言

粒子群优化(PSO)算法是 Eberhart 和 Kennedy 于 1995 年开发的一种进化计算技术^[1,2],它源于对鸟群捕食行为的研究. 后来 Shi 等人^[3]引入惯性权重 w ,更好地控制收敛性和探索能力,形成了基本的 PSO 算法. 与其他进化算法相比,PSO 算法概念简单,收敛较快,容易实现,且只有少数参数需要调整. 目前,PSO 算法获得了很大的发展,并广泛应用于许多领域.

基本 PSO 算法也存在一些问题,主要是早熟收敛和收敛较慢两个难题. 随着代数的增加,各粒子变得越来越相似,基本 PSO 算法不能对解进行持续优化,而是在局部最优解附近徘徊. 对基本 PSO 算法改进的方法主要有两种:一种是对权重的变化进行改进,权重变化对算法的全局探索和局部探测有很大影响,为此把权重线性变化变成非线性变化或模

糊化^[1];另一种是通过控制种群的多样性来提高算法性能,通过解决微粒间冲突^[4],引入速度变异和位置变异^[5],小概率重新初始化^[6]等方法,增加粒子的多样性,提高全局搜索能力,避免陷入局部最优.

本文对基本粒子群优化算法的性能及问题进行分析,定义粒子之间相似度的概念,提出对权重进行动态调整的方法. 根据相似性计算粒子群的聚集度,根据聚集度以概率形式对粒子重新赋值,以此来增加粒子群的多样性. 将该算法应用于多峰标准函数的优化问题,仿真结果表明,在克服收敛性与探索性这对矛盾时,新算法比基本 PSO 算法有了明显的改进. 为了叙述方便,把本文提出的 PSO 算法称为 SPSO 算法.

2 基本粒子群优化算法

假设搜索空间为 D 维,粒子群中第 i 个粒子的位置表示为 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$,第 i 个粒子的速

收稿日期: 2006-08-22; 修回日期: 2007-01-10.

作者简介: 刘建华(1967—),男,江西安福人,讲师,博士生,从事智能计算、数据挖掘等研究;樊晓平(1961—),男,浙江绍兴人,教授,博士生导师,从事智能控制、优化算法等研究.

度表示为 $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$, 第 i 个粒子迄今为止搜索到的最好位置记为 $P_i = (p_{i1}, p_{i2}, \dots, p_{id})$, 整个粒子迄今为止搜索到的最好位置记为 $P_g = (p_{g1}, p_{g2}, \dots, p_{gd})$. 对于每个粒子, 其第 d 维(1 d

D) 根据下式变化:

$$v_{id} = wv_{id} + c_1 \text{rand}() (p_{id} - x_{id}) + c_2 \text{rand}() (p_{gd} - x_{id}), \quad (1)$$

$$x_{id} = x_{id} + v_{id}. \quad (2)$$

其中: 加速常数 c_1 和 c_2 是两个非负值; $\text{rand}()$ 是在 $[0, 1]$ 范围内取值的随机数; w 为惯性权重, 起到平衡算法全局搜索与局部搜索的作用.

文献[3]进一步提出自适应调整的策略, 即随着迭代的进行, 线性减少 w 值. 这使算法在迭代初期探索能力较强, 可以不断搜索新的区域; 然后收敛能力逐渐增强, 使算法在可能的最优解周围精细搜索.

文献[7]对权重进行改进, 把权重的线性变化改为非线性变化. 引入权重变化公式如下:

$$w_t = \{ (t_{\max} - t) / t_{\max} \}^m (w_{\max} - w_{\min}) + w_{\min}. \quad (3)$$

其中: w_{\max} 是开始运行时初始的权重, w_{\min} 是最后运行的权重, t_{\max} 是运行的最大迭代次数, t 是当前的迭代次数, w_t 是当前的权重, m 是非线性调整因子. 显然, 当 $m = 1$ 时, 权重非线性变化. 这里把这种 PSO 算法称为 NPSO 算法.

为了解决基本 PSO 算法的收敛问题, DPSO 算法^[5] 引入了随机变异操作, 使粒子有更大的几率逃出局部极小点, 保持种群的多样性. 公式描述如下:

$$\text{If } (\text{rand}() < c) \text{ Then } x_{id} = \text{rand}(l_d, u_d). \quad (4)$$

其中: c 为 $[0, 1]$ 范围内的噪声因子, $\text{rand}(l_d, u_d)$ 为 l_d 与 u_d 之间的随机数, l_d 和 u_d 为位置分量的取值范围.

3 粒子之间的相似度

由基本 PSO 算法性能分析可知, 权重对算法的收敛性和探索能力有很大影响, 随着迭代的进行, 粒子变得越来越相似, 从而影响算法的探索能力. 为了根据粒子之间的相似性来分析和改进算法, 给出粒子间相似度的概念:

定义 1 两个粒子 i 和 j 的相似度 $s(i, j)$ 必须满足下列准则:

- 1) $s(i, i) = 1$;
 - 2) 当 $d(i, j) = 0$ 时, $s(i, j) = 0$;
 - 3) 对于任何粒子 i 和 j , 有 $s(i, j) \in [0, 1]$.
- 两个粒子 i 和 j 的相似度量化公式如下:

$$s(i, j) = \begin{cases} 1, & d(i, j) < d_{\min}; \\ 1 - \left[\frac{d(i, j)}{S_{\max}} \right], & d_{\min} < d(i, j) < d_{\max}; \\ 0, & d(i, j) > d_{\max}. \end{cases} \quad (5)$$

其中: $d(i, j)$ 表示粒子 i 与 j 的空间距离, 本文采用 Euclid 距离; d_{\max} 和 d_{\min} 是固定正常数. 由概念的定义可知, 两个粒子越靠近, 粒子越相似, 因此相似度也越大.

4 基于相似度的粒子群算法^[9-12]

4.1 基于相似度的权重动态调整

在基本 PSO 算法^[3] 中, 式(1)的惯性权重 w 采用迭代线性递减的方法, 早期选择很大的惯性权重 w 值, 后期则选择较小的惯性权重 w 值. 这符合早期需要粒子的探索能力, 而后期需要保持其收敛性的要求. 这就是基本 PSO 算法能被广泛采用的原因. 但它也有不利的一面, 即若早期找到全局最优点, 则因其惯性权重过大, 有可能跳出这个最优点, 而不在其附近探索, 从而降低对最优点的搜寻能力.

在 PSO 算法优化求解中, 当前最优粒子 P_g 附近有可能存在真正的全局最优解. 对于当前最优粒子 P_g , 后项为 0, $V_{gd} = w * V_{gd}$. 如果惯性权重 w 很大, 则最优粒子 P_g 有可能跳出其最优范围, 但最优粒子 P_g 有可能处于真正全局最优解的邻域. 从式(1)右边 3 项可以发现, 越靠近最优点的粒子, 其飞行速度越依赖于惯性权重 w . 为了不错过每一次的最优点, 可让靠近 P_g 的粒子飞行速度很小, 即与当前最优粒子 P_g 相似度越高的粒子, 其权重 w 应该越小. 这样可使这些粒子在 P_g 邻域内进行微小的探索, 而不承担更大范围的探索. 为此, 本文设计出一种惯性权重 w 随各粒子与当前最优粒子 P_g 相似度不同而动态变化的粒子群算法. 各不同粒子的惯性权重 w 值不仅随迭代次数的增加而递减, 并且随当前最优粒子 P_g 的相似度大小而递减.

设 $s(i, g)$ 表示第 i 个粒子与当前最优粒子 P_g 的相似度, 设置两个有关距离的参数: 最大距离 d_{\max} 和最小距离 d_{\min} , 按式(5) 计算相似度 $s(i, g)$. 当相似度 $s(i, g) = 0$ 时, 粒子 i 的权重 w_i 为最大权重 w_{\max} ; 当相似度 $s(i, g) = 1$ 时, 粒子 i 的权重 w_i 为最小权重 w_{\min} ; 当相似度 $s(i, g)$ 在 $(0, 1)$ 范围内时, 其权重 w_i 随相似度而单调递减. 其计算公式为

$$w = w_{\max} - s(i, g) (w_{\max} - w_{\min}), \quad (6)$$

$$w_i = w_{\min} + (w - w_{\min}) \frac{t_{\max} - t}{t_{\max}}. \quad (7)$$

式(6)和式(7)就是粒子 i 的权重计算方法. 式(7)使权重随迭代次数的增加而线性减小.

4.2 基于相似度的粒子群聚集分析

进化算法的改进思路是:减少相似个体的粒子数量,保持群体的多样性,从而达到同时搜索多个区域的目的.粒子群优化算法无论是早熟收敛还是全局收敛,粒子群中的粒子都会出现聚集现象.理论上已证明^[8]:假设 P_g 和 P_j 在 PSO 算法进化中保持不变,在参数满足一定的条件下,PSO 算法的 $x_i(t)$ 收敛于 P_g 与 P_j 的加权中心.可以证明^[5],当 p_i 不断变化时, $x_i(t) \rightarrow P_g$. 因此,如果粒子群聚集,最终将聚集到最优粒子 P_g 附近.根据以上分析,可以通过计算整个粒子群与最优粒子 P_g 的平均相似度来量化粒子群的聚集度.为此建立如下定义:

定义 2 第 t 代粒子群的聚类度

$$C(t) = \frac{1}{n} \sum_{i=1}^n s(i, g). \quad (8)$$

其中: $s(i, g)$ 表示粒子 i 与当前最优粒子 P_g 的相似度, $C(t)$ 用来度量第 t 代粒子群的聚集度.

为了保持群体的多样性,设计当聚集度为一定程度时,让粒子 i 依概率重新随机赋值的算法如下:

$$\text{If } \text{rand}() < \frac{c(t)}{t} s(i, g),$$

$$\text{Then } x_{id} = \text{rand}() (\text{ub}_d - \text{lb}_d) + \text{lb}_d. \quad (9)$$

其中: c 和 t 是固定常数, ub_d 和 lb_d 是第 d 维边界的最大值和最小值.

式(9)体现的是:当聚集度越大且粒子与最优粒子相似度较大时,随机重新初始化的机会就越大,增加粒子的多样性;当迭代次数增加时,随机重新初始化的机会就越小,有利于算法早期提高全局搜索能力,后期保持局部探测能力.

4.3 算法的收敛性与时间性分析

定理 1 SPSO 算法的权重 w_i 的范围仍为 $[w_{\min}, w_{\max}]$.

证明 在式(6)中, $0 \leq s(i, g) \leq 1$, 所以 $w_{\min} \leq w \leq w_{\max}$. 由此 $0 \leq w - w_{\min} \leq w_{\max} - w_{\min}$, $0 \leq \frac{t_{\max} - t}{t_{\max}} \leq 1$.

根据式(7), $w_{\min} \leq w_i \leq w_{\max}$.

由定理 1 可知,SPSO 算法的权重计算方法能保证基本 PSO 算法的权重范围,因此能保证基本 PSO 算法的收敛性.从式(7)可以看出,SPSO 算法迭代到最后 t_{\max} , w_i 值不大于 w_{\max} , 因此 SPSO 算法可能具有更快的局部收敛性.

设微粒数为 n ,问题的维数为 dim ,最大迭代数为 t_{\max} .根据前面的叙述,本文算法描述如下:

- 1) 初始化:随机产生 n 个粒子及其初始速度;
- 2) 评价每个粒子的适应度;

For $i = 1:n$

3) 确定迄今为止每个粒子找到的最好位置 P_i ;

4) 确定迄今为止整个群体找到的最好位置

P_g ;

5) 计算微粒与最优粒子 P_g 的相似度,并用式(6)和(7)计算此粒子的权重;

For $d = 1:\text{dim}$

6) 利用式(1)和(2)重新计算粒子的位置;

EndFor

7) 根据式(8)计算此时的聚集度,根据式(9)依概率随机初始化此粒子位置;

EndFor

8) 若条件未满足且未达到最大迭代 t_{\max} ,则转向 2).

在算法流程的第 5) 步,改变了基本 PSO 算法计算权重的方法,但并没有增加算法时间;在第 7) 步多了一步计算聚集度,并用式(9)变异粒子的位置,但只在内循环中线性增加了两步计算.因此,基本 PSO 算法的时间复杂度为 $O(n * \text{dim} * t_{\max})$,而 SPSO 算法的时间复杂度仍为 $O(n * \text{dim} * t_{\max})$,并没有增加基本 PSO 算法的时间复杂度.

5 实验计算与分析

为了验证本文算法的有效性,分别采用基本 PSO 算法、NPSO 算法、DPSO 算法和 SPSO 算法,对以下函数的最优问题进行实例计算并进行对比.为了检验改进效果,先选两个多峰函数进行实验.

(1) Griewank 函数

$$f_1(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \frac{x_i}{\sqrt{i}} + 1, \\ -100 < x_i < 100.$$

(2) Rastrigrin 函数

$$f_2(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10], \\ -100 < x_i < 100.$$

两个函数的最优状态都是原点,最优值为 0.

对于这两个基准函数的自变量维数及粒子个数设置如表 1 所示.两种算法的参数设置为: $c_1 = 1.8$, $c_2 = 1.8$, $w_{\max} = 1.0$, $w_{\min} = 0.4$, $v_{\max} = 10$, $d_{\min} = 10$, $d_{\max} = 50$, $\text{rand}() = 1$, $\text{rand}() = 5$.根据文献[5,7],NPSO 算法和 DPSO 算法的参数设置为: $c = 0.002$, $m = 1.2$.为使数据准确,每个函数优化迭代 1 000 次,重复 100 次,然后求得最小平均适应值和标准方差(见表 1).

从表 1 可以看出,对于每个函数的收敛效果,新算法均优于其他 3 种粒子群算法,而且解收敛的稳定性也有大幅度提升.

表 1 最小平均适应值和标准方差

维数	粒子规模	函 数	PSO	DPSO	NPSO	SPSO
10	20	Griewank	0.095 7 ±0.041 3	0.067 1 ±0.035 6	0.075 7 ±0.055 3	0.048 4 ±0.031 3
		Rastrigrin	5.824 1 ±4.563 6	4.214 3 ±3.646 1	4.625 1 ±3.513 4	3.435 6 ±2.513 7
20	50	Griewank	0.036 2 ±0.023 8	0.021 8 ±0.021 6	0.022 3 ±0.020 1	0.016 2 ±0.015 6
		Rastrigrin	21.108 2 ±14.238 6	16.441 8 ±10.318 0	15.873 2 ±11.385 0	14.478 2 ±8.584 7
30	80	Griewank	0.015 7 ±0.006 3	0.011 7 ±0.005 6	0.010 9 ±0.005 9	0.009 7 ±0.004 3
		Rastrigrin	46.745 0 ±25.648 2	40.683 2 ±21.583 0	41.174 8 ±20.655 0	34.536 7 ±19.468 0

表 2 测试函数的平均寻优结果比较

函数	群体规模	进化代数	收敛误差	平均收敛率				平均收敛代数			
				PSO	DPSO	NPSO	SPSO	PSO	DPSO	NPSO	SPSO
f_3	20	100	0.001	20	21	24	28	56.711	52.547	51.643	49.364
f_3	50	200	0.001	70	74	73	82	68.869	53.578	53.944	50.467

为了进一步验证本文算法的有效性,利用基本 PSO 算法、NPSO 算法、DPSO 算法和 SPSO 算法,对 Schaffer 函数的最小化问题进行计算和对比。

(3) Schaffer 函数

$$f_3(x) = \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2} + 0.5,$$

$$-100 < x_1, x_2 < 100.$$

函数 f_3 的全局极小点是 (0, 0), 最优值为 0. 在距全局最优点大约 3.14 附近, 有许多局部最优点, 函数强烈振荡, 因此一般算法难以得到最优解。

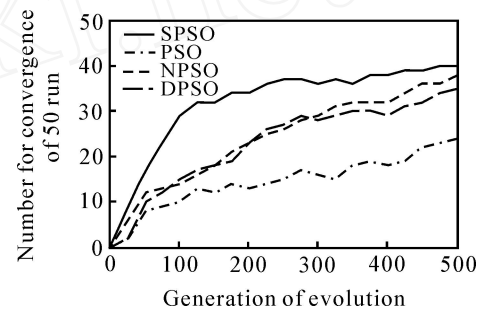
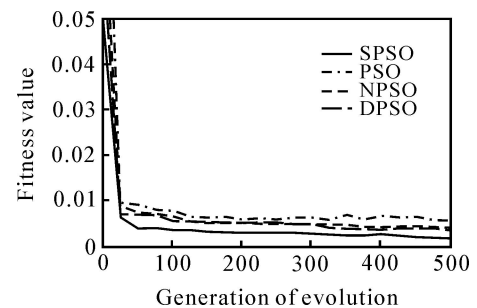
实验方法是在相同的条件下, 比较 4 种算法收敛最优的百分比, 以及收敛时平均收敛代数. 对上述问题进行 100 次仿真计算, 计算结果如表 2 所示。

从表 2 可以看出, 不论平均收敛率还是平均收敛代数, SPSO 算法比基本 PSO 算法、DPSO 算法和 NPSO 算法都有所提高, 平均收敛率比平均收敛代数改善程度更好。

为了进一步比较算法的性能, 在相同的参数设置下, 对多峰函数 f_3 进行如下实验: 每增加 25 代进化数时, 记录 50 次计算平均最优解和 50 次计算平均收敛次数. 即在误差为 0.001 时, 50 次计算中能够收敛的次数. 测试的最大进化代数为 500, 其结果如图 1 和图 2 所示。

从图 1 可以看出, 无论在多少个进化代数情况下, SPSO 的收敛次数比另外 3 种算法都高. 从图 2 可以看出, SPSO 算法的最优值在不同代数情况下, 所求出最优点比另外 3 种算法计算效果明显提高, 并且速度更快. 表 2 数据也显示, SPSO 的平均收敛代数和收敛率比另外 3 种算法都好。

SPSO 算法权重动态变化的思想主要是改进算法早期遗失最优点, 有利于提高收敛代数. SPSO 算法中各个粒子群都划分为三部分: 离最优粒子较近

图 1 f_3 进化代数收敛的变化图 2 f_3 进化代数最优值的变化

的一部分, 其权重很小, 主要负责在全局最优点附近进行探索; 离粒子较远的一部分, 其权重始终是最大值, 主要负责在全局范围内进行探索; 中间部分的粒子, 其权重随着代数和距离的变化线性变小, 早期主要是探索, 后期主要是收敛于全局最优点. 基于相似度和聚集度的粒子重新赋予新位置, 有利于粒子的全局探索能力, 提高算法的收敛率。

6 结 语

本文提出了粒子之间相似度的概念, 并使每个粒子的惯性权重随其与最优粒子之间相似度动态变化的算法. 计算每个时刻的粒子群聚集度, 根据聚集度随机初始化每个粒子的位置. 这样可以增加粒子群的多样性, 有利于粒子群算法的全局搜索. 仿真结果表明, 所提出的粒子群算法在求解多峰函数时, 算

法收敛效果和收敛速度都有较大提高. 本文是根据不同粒子与最优粒子的相似度来改变权重参数, 并由此计算粒子群的聚集度, 随机初始化粒子的位置, 增加粒子群的多样性, 提高全局搜索能力. 但本文算法新增加了一些参数, 对这些参数如何设置还需进一步讨论.

参考文献(References)

- [1] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory[C]. The 6th Int Symp on Micro Machine and Human Science. Nagoya, 1995: 39-43.
- [2] Kennedy J, Eberhart R C. Particle swarm optimization [C]. Proc of IEEE Int Conf on Neural Networks. Piscataway: IEEE Service Center, 1995: 1942-1948.
- [3] Shi Y, Eberhart R C. A modified particle swarm optimizer [C]. Proc of the IEEE Int Conf on Evolutionary Computation. Piscataway: IEEE Press, 1998: 69-73.
- [4] Krink T, Vesterstrom J S, Riget J. Particle swarm optimization with spatial particle extension[C]. Proc of the IEEE Int Conf on Evolutionary Computation. Honolulu: IEEE Inc, 2002: 1474-1497.
- [5] Xie X F, Zhang W J, Yang Z L. A dissipative particle swarm optimization[C]. Proc of the IEEE Int Conf on Evolutionary Computation. Honolulu: IEEE Inc, 2002: 1666-1670.
- [6] Van Den Bergh. An analysis of particle swarm optimizers [D]. Pretoria: University of Pretoria, 2002: 81 - 83.
- [7] Chatterjee A, Siarry P. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization[J]. Computers and Operations Research, 2006, 33(3): 859-871.
- [8] Ratnaweera A, Halgamuge S K, Watson H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients [J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 240-255.
- [9] Zhang L P, Yu H J, Hu S X. Optimal choice of parameters for particle swarm optimization [J]. J of Zhejiang University, 2005, 6(6): 528-534.
- [10] He S, Wu Q H, Wen J Y, et al. A particle swarm optimizer with passive congregation[J]. Bio-systems, 2004, 78(1-3): 135-147.
- [11] 王凌. 智能优化算法及其应用[M]. 北京: 清华大学出版社, 2001.
(Wang Ling. Intelligent optimization algorithms with applications[M]. Beijing: Tsinghua University Press, 2001.)
- [12] 窦全胜, 周胜光, 马铭. 粒子群优化的两种改进策略 [J]. 计算机研究与发展, 2005, 42(5): 897-904.
(Dou Quan-sheng, Zhou Sheng-guang, Ma Ming. Two improvement strategies for particle swarm optimization [J]. J of Computer Research and Development, 2005, 42(5): 897-904.)

(上接第 1154 页)

- [6] 李建更, 涂攀生. 一类 Flow shop 调度问题最优调度区间摄动鲁棒性[J]. 控制理论与应用, 2004, 21(1): 25-29.
(Li Jian-geng, Tu Peng-sheng. Interval perturbation robustness of optimal schedules for a class of flow shop problems[J]. Control Theory and Applications, 2004, 21(1): 25-29.)
- [7] Kouvelis P, Yu G. Robust discrete optimization and its applications [M]. Dordrecht: Kluwer Academic Publishers, 1997.
- [8] Lenstra J, Rinnooy K A, Brucker P. Complexity of machine scheduling problems [J]. Annals of Discrete Mathematics, 1977, 1: 343-362.
- [9] 玄光男, 程润伟. 遗传算法与工程优化 [M]. 于歆杰, 等译. 北京: 清华大学出版社, 2004.
(Gen Mitsuo, Cheng Run-wei. Genetic algorithm and engineering optimization [M]. Yu Xin-jie, et al (translators). Beijing: Tsinghua University Press, 2004.)
- [10] Michalewicz Z. Genetic algorithm + data structure = evolution programs[M]. New York: Springer-Verlag, 1994.