

群体启发进化规划^{*}

窦全胜¹ 周春光¹ 徐中宇² 潘冠宇¹

(吉林大学计算机科学与技术学院 长春130012¹ (长春工业大学计算机科学与工程学院 长春130012)²)

摘要 本文提出了一种群体启发进化规划(Population Heuristic Evolutionary Programming, PHEP)方法,在进化过程中,通过群体的四个参数,把握群体中个体的分布情况,并通过这些信息有效地调整个体的变异步长,使用标准测试问题对 PHEP 方法进行了测试和比较,试验数据表明:PHEP 方法能够较好地平衡群体压力和选择压力,搜索到的结果要明显优于先前的一些方法。

关键词 进化规划,优化

Population Heuristic Evolutionary Programming

DOU Quan-Sheng¹ ZHOU Chun-Guang¹ XU Zhong-Yu² PAN Guan-Yu¹

(Department of Computer Science and Technology, Jilin University, Changchun 130012)¹

(School of Computer Science and Engineering, Changchun University of Technology, Changchun 130012)²

Abstract This paper presents a new method for optimization problem which named population heuristic evolutionary programming (PHEP), it can grasp the information of distribution-status of population by control four parameters of population in the evolution process. PHEP adjusts the mutation size of individual according to such information. Experiments for benchmark problem imply that the PHEP can keep the balance of the population press and the selection press very well, the results are great better than other version EP methods.

Keywords Evolutionary programming, Optimization

1 前言

许多科学和工程问题都可以归结为求解最优化问题,并可通过如下数学公式表达:

$$\min_{x \in S} f(x)$$

满足:

$$s_i(x) \geq 0 \quad i=1, 2, \dots, m$$

$$h_l(x) = 0 \quad j=1, 2, \dots, l(l \leq n)$$

其中 S 是解域, f 为被优化函数, s_i, h_j 是约束条件集合。本文只考虑无约束优化问题。

进化规划(Evolutionary Programming, EP)方法是由 Lawrence Fogel^[1]提出的进化计算技术,近几年,有很多关于进化规划技术的研究,一些优秀算法成功地应用于各类优化问题。

在求解优化问题的进化规划方法中,CEP(Classical Evolutionary Programming)、FEP(Fast Evolutionary Programming)、IFEP(Improved Fast Evolutionary Programming)、REP(Robust Evolutionary Programming)等方法先后在文[2~7]中被提出和应用。以下算法1是 CEP(标准进化规划)的执行过程:

算法1: 标准进化规划(CEP)方法

Step1. 在解域 $S \subseteq R^n$ 内随机地产生由 μ 个个体组成的群体。每个个体由一对实值向量构成:

$$(x_i, \eta_i) \quad \forall i \in (1, \dots, \mu)$$

其中, x_i 为个体目标向量, η_i 是变异向量;

Step2. 根据评价函数 $f(x)$ 计算每个个体的评价函数值;

Step3. 对群体中的每一个个体 (x_i, η_i) 通过如下变异产生一个后代 $(x'_i, \eta'_i) i=1, \dots, \mu$

$$x'_i(j) = x_i(j) + \eta_i(j)N_j(0, 1)$$

$$\eta'_i(j) = \eta_i(j) \exp(\tau' N(0, 1) + \tau N_j(0, 1))$$

其中, $x_i(j), x'_i(j), \eta_i(j), \eta'_i(j)$ 分别表示向量 $x_i, x'_i, \eta_i, \eta'_i$ 的第 j 个分量 $j=1, 2, \dots, n, N(0, 1)$ 是满足正态分布的随机数, τ 和 τ' 通常被设置成 $(\sqrt{2\sqrt{n}})^{-1}$ 和 $(\sqrt{2n})^{-1}$;

Step4. 计算每一个后代 $(x'_i, \eta'_i) i=1, \dots, \mu$ 的评价函数值;

Step5. 所有父个体和子个体构成一个由 2μ 个个体组成的群体。对于该群体中的每一个个体,从群体中随机等概率地选取出 q 个个体和它进行比较,在每次比较中,如果该个体的评价值不大于与之比较的个体的评价值(对于所研究的最小化问题),则称该个体获得一次胜利,其中 q 为事先指定的正整数;

Step6. 从 2μ 个个体中选择获胜次数最多的 μ 个个体作为下一代的父个体;

Step7. 若满足终止条件,则算法停止。否则,转到 Step2。

FEP 执行步骤与 CEP 相同,只是用 Cauchy 分布函数代替 CEP 中的 Gaussian 分布函数^[2],以增大跳出局部最优的概

^{*} 基金项目:国家自然科学基金项目资助(批准号:60175024);教育部“符号计算与知识工程”重点实验室资助。窦全胜 博士研究生,研究方向:软计算理论。周春光 教授,博士生导师,研究方向:计算智能。徐中宇 讲师,博士研究生,研究方向:图形、图像识别。潘冠宇 博士研究生,研究方向:数据挖掘。

率。IFEP 同时使用了 Cauchy 分布函数和 Gaussian 分布函数进行变异操作,并选出最好的个体。REP 采用了与 CEP 不同的编码方式,并使用三个变异算子 O_{dep} 、 O_{del} 和 O_{ins} 来控制下一代的产生。

不难看出,如何在算法的运行过程中有效地调整变异向量,是这类方法的关键所在。一些定性的理论分析可以在文 [2~11] 中得到。在 EP 方法中,对变异向量的调整与群体在运行过程中的状态没有关系,在通常情况下,当个体进入某个局部最优解的吸引域后,用小方差变异得到改进解的概率一般要大得多。因此,在这种机制下,个体的变异向量逐步减小。这意味着算法变异能力的逐步降低,使算法执行到一定程度时,很难摆脱局部最优的吸引,造成算法“早熟”。本文提出了一种基于群体启发的进化规划方法 (Population Heuristic Evolutionary Programming, PHEP),该方法在执行过程中,能够掌握群体中个体的分布情况,并根据这一情况来调整下一步变异向量的大小。目的是在算法的搜索过程中,尽可能地保持群体的多样性。测试结果证明了 PHEP 方法的有效性和可靠性。

2 群体启发进化规划

在进化规划中,群体产生过程,可用下式描述:

$$P_t = S\{f[M(P_{t-1})]\}$$

其中, P_t 表示第 t 代群体,最初 P_0 是随机产生的, M 、 f 和 S 分别表示变异、评价和选择操作。

算法早熟是进化规划遇到的最大问题,算法早熟的原因很多,求解问题的复杂性和欺骗性是造成算法早熟的一个原因,从问题入手,对问题进行改造和变换是一种解决方法,但这需要对问题有深入的理解,而且有些实际问题复杂得根本无法进行处理,这方面本文不作更多的讨论。我们仅从进化规划算法本身出发,通过改进算法的选择机制和变异机制,保持群体多样性,克服早熟。

在个体变异过程中,新个体由下式产生:

$$x'_i = x_i + \eta_i N_i(0, 1)$$

上式中的变异向量 η_i 起着“步长”的作用,当 $|\eta_i|$ 的值较大时,个体的搜索范围较大,有很强的“开拓”能力,相反当 $|\eta_i|$ 值较小时,个体的搜索范围变小,局部“探索”能力得到增强。通常情况下, η_i 有几种自适应方式,有的与迭代的次数 n 有关,或者与变异成功的概率有关。一些方法采用了所谓 1/5 的成功法则,取得了较好的效果。即当变异成功的概率大于 1/5 时,增大变异向量,而在变异成功概率小于 1/5 时缩小变异向量。在算法执行的后半期,变异成功的难度不断增大,变异成功的概率不断减小。因此,算法在后来的执行过程中很难跳出局部最优的吸引,造成“早熟”;另一方面,进化规划算法多是按某一概率进行选择个体来组成下一代群体,这样使评价较高的个体及其后代存活下来的概率远大于评价函数较低的个体,从而使群体的多样性在一定程度上遭到破坏,加快了“早熟”的发生。

本文提出的 PHEP 方法,通过改进进化规划的选择策略和变异策略,在试图保证算法“局部搜索”能力的同时保持群体的多样性,下面简单地阐述 PHEP 方法的主要思想:设 $x_i, i = 1, \dots, \mu$ 是第 t 代群体 $POP(t)$ 中的元素,我们按如下方式选择其中元素构成第 $t+1$ 代群体 $POP(t+1)$,计算 $f_{avg} =$

$$\frac{\sum_{i=1}^{\mu} f(x_i)}{\mu}, \text{把 } POP(t) \text{ 中所有满足 } f(x_i) < f_{avg} \text{ 的 } i = 1, \dots, \mu \text{ 的 } x_i$$

构成的集合记为 P_1 , P_1 中元素的个数记为 $Count(P_1)$,然后对所有 $y_j \in POP(t) - P_1, j = 1, \dots, \mu - Count(P_1)$ 进行变异,设 \hat{y}_j 是 y_j 变异后产生的新个体,我们把所有满足 $f(\hat{y}_j) \leq f(y_j)$ 的 \hat{y}_j 构成的集合记为 P_2 ,若 $Count(P_1 \cup P_2) < \mu$,则从集合 $P_1 \cup P_2$ 中随机选择 $\mu - Count(P_1 \cup P_2)$ 个个体进行变异,设 \hat{z}_k 是集合 $P_1 \cup P_2$ 中个体 z_k 变异后产生的新个体,把所有满足 $f(\hat{z}_k) \leq f(z_k)$ 的 \hat{z}_k 构成的集合记为 P_3 ,若 $Count(P_1 \cup P_2 \cup P_3) < \mu$,在 $P_1 \cup P_2 \cup P_3$ 中随机选择 $\mu - Count(P_1 \cup P_2 \cup P_3)$ 个个体直接构成 P_4 。PHEP 通过四个集合元素的个数占个体总数的比例来掌握群体中个体的分布情况,进而调整变异向量的值。下一代群体 $POP(t+1)$ 等于上述四个集合的并集或者保持不变, PHEP 具体由算法 2 给出。

算法 2: 群体启发进化规划 (PHEP) 方法

Step1. 令 $k=1$, 在解域 $S \subseteq R^n$ 内随机的产生由 μ 个个体组成的群体 $P(1)$, 其中, $P(k)$ 表示第 k 代群体;

Step2. 利用指定的评价函数 f 计算:

$$f_{avg} = \frac{\sum_{i=1}^{\mu} f(x_i)}{\mu};$$

Step3. 用 $\hat{P}_{avg}(k)$ 表示 $P(k)$ 中所有满足如下条件: $f(x_i) < f_{avg}, i \in (1, \dots, \mu)$ 的个体的集合。用如下方法构造集合:

P_{pre_sel} :

a. 令 $P_{pre_sel} = \hat{P}_{avg}(k)$, 设 $\hat{P}_{avg}(k)$ 中有 m 个元素, 记

$$O_{normal} = \frac{m}{\mu}.$$

b. 规定变异次数, 对所有 $x_l \in P(k) - \hat{P}_{avg}(k), l = 1, \dots, \mu - m$ 按如下方法进行变异:

$$x'_l = x_l + \eta_l N(0, 1)$$

得到 x'_l , 若 $f(x'_l) < f(x_l)$, 则把 x'_l 加到 P_{pre_sel} 中。假设满足这样条件 x'_l 的有 n 个, 记 $O_{mut} = \frac{n}{\mu}$ 。

c. 若 $m+n < \mu$, 则在 P_{pre_sel} 中随机地选择 $\mu - m - n$ 个个体, 规定变异次数, 按上面方法进行变异, 得到 x'_p , 若 $f(x'_p) < f_{avg}$, 则保留 x'_p 到 P_{pre_sel} 中。假设满足这样条件的 x'_p 有 q 个, 记 $O_{sel_mut} = \frac{q}{\mu}$ 。

d. 若 $m+n+q < \mu$, 则在 P_{pre_sel} 中随机地选择 $\mu - m - n - q$ 个个体补充到 P_{pre_sel} 中。记 $O_{direct_sel} = \frac{\mu - m - n - q}{\mu}$ 。

Step4. 如果 $O_{direct_sel} > \lambda$, 并且变异向量 η_i 小于某规定值, 则作如下调整: $\eta_i = \eta_i \tau_\lambda$, 其中 λ 为指定的参数, τ_λ 为大于 1 的正常数。 $P(k+1) = P(k)$, 转到步骤 (7)。

Step5. 若 $O_{normal} + O_{mut} > \gamma$, 则调整变异向量 $\eta_i = \eta_i \tau_\gamma$, 若 $v < O_{normal} + O_{mut} < \alpha$, 则调整变异向量 $\eta_i = \eta_i \tau_\alpha$, 其中 γ, α, v 为指定的参数, 且 $0 < \alpha < \gamma, \tau_\gamma > 1, 0 < \tau_\alpha < 1$ 。 $P(k+1) = P_{pre_sel}$, 转到 step7。

Step6. 在 $O_{normal} + O_{mut} < v$ 的条件下, 若 $O_{sel_mut} > \beta$, 并且变异向量 η_i 小于某规定值, 则作如下调整: $\eta_i = \eta_i \tau_\beta$, 其中 $\tau_\beta > 1$, 否则保持原变异向量不变。

Step7. 满足终止条件, 过程结束。否则, $k = k + 1$, 转到 Step2。

由步骤 4, 当 $O_{direct_sel} > \lambda$, 且变异步长较小时, 说明这时群体的变异成功率很低, 且群体多样性很差, 算法有早熟的趋势, 此时, 如果一味地缩小变异向量会造成群体的过早收敛, 因此, 算法增大变异向量, 以跳出局部最优的吸引, 并把群体直接保留到下一代中; 由步骤 5, 当 $O_{normal} + O_{mut} > \gamma$ 时, 表明群体

中只有少数较差的个体,且变异成功率较高,这时增大变异向量,以提高算法效率,当 $v < O_{normal} + O_{mut} < \alpha$ 时,说明算法变异成功率降低,应该适当缩小变异向量,提高算法变异成功率;由步骤6, $O_{normal} + O_{mut} < v$ 时,说明群体中存在少数个体,它们的评价函数值远小于其它个体(对于最小优化问题),此时,当集合 P_2 变异成功率较高时,增大变异向量,反之变异向量不变。

采用上述策略较大程度上提高了算法的搜索精度和执行

效率,下面的实验结果充分证明了这一点。

3 PHEP 测试结果

采用文[2]中提供的12个测试函数,其中既有单峰函数(只有唯一的全局最优)又有多峰函数(包括多个局部最优或全局最优),单峰函数相对简单,但是在问题的维数较高的条件下,仍然有很强的欺骗性。表1描述了本文采用的12个测试函数。

表1 文[2]提供的测试函数

函数	表达式	n	S	函数描述
f_1	$f_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]^n$	Sphere Model
f_2	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10, 10]^n$	Schwefel's Problem 2.22
f_3	$f_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	$[-10, 10]^n$	Schwefel's Problem 1.2
f_4	$f_4(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]^n$	Generalized Rosenbrock's Function
f_5	$f_5(x) = \sum_{i=1}^n x_i + 0.5 ^2$	30	$[-100, 100]^n$	Step Function
f_6	$f_6(x) = \sum_{i=1}^n ix_i^4 + random$	30	$[-1.28, 1.28]^n$	Quartic Function with Noise
f_7	$f_7(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	$[-500, 500]^n$	Generalized Schwefel's Problem 2.26
f_8	$f_8(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$[-5.2, 5.12]^n$	Generalized Rastrigin's Function
f_9	$f_9(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i) + 20 + e$	30	$[-32, 32]^n$	Ackley's Function
f_{10}	$f_{10}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600, 600]^n$	Generalized Griewank Function
f_{11}	$f_{11}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, -5]^n$	Six-hump Camel-Back Function
f_{12}	$f_{12}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]^n$	Goldstein-Price Function

取 $\mu=100, \lambda=0.3, \tau_\lambda=1.05, \gamma=0.8, \tau_\gamma=1.15, \alpha=0.65, \tau_\alpha=0.85, v=0.3, \beta=0.45, \tau_\beta=1.05$, 算法运行50次, 所得结

果的平均值与文[2]提供的结果比较, 细节参见表2。

表2 PHEP 方法运行50次, 所得结果平均值与 FEP、CEP^[2] 所获结果比较

函数	f_{i_min}	PHEP		FEP		CEP	
		迭代次数	最优解平均值	迭代次数	最优解平均值	迭代次数	最优解平均值
f_1	0	1000	0	1500	5.7e-4	1500	2.2e-4
f_2	0	1000	0	2000	0.0081	2000	0.0026
f_3	0	1000	2.7280e-5	5000	0.016	5000	0.05
f_4	0	10000	2.7757	20000	5.06	20000	6.17
f_5	0	1000	0	1500	0	1500	577.76
f_6	0	3000	0.0018	3000	0.0076	3000	0.0064
f_7	-12569.5	9000	-12549.3	9000	-12554.5	9000	-7917.1
f_8	0	5000	0.079	5000	0.046	5000	89
f_9	0	1000	0	1500	0.018	1500	9.2
f_{10}	0	1000	3.7e-7	2000	0.016	2000	0.086
f_{11}	-1.0316285	100	-1.03162845	100	-1.03	100	-1.03
f_{12}	3	100	3	100	3.02	100	3

从测试结果上看, 采用 PHEP 方法, 函数 $f_1, f_2, f_5, f_9, f_{12}$ 搜索到了全局最优, 函数 f_7, f_8 的实验结果略与其他两种方法接近, 其他实验结果明显优于另外两种方法。

PHEP 使用参数: $O_{direct_sel}, O_{normal}, O_{mut}, O_{sel_mut}$ 来把握群体中个体的分布情况, 根据个体的分布调整变异步长 η , 使之在

(下转第167页)

准了的不同聚焦点图像进行融合,融合结果如图1.i所示。为了进行比较,还选择了简单加权平均法、小波变换方法和Laplacian金字塔方法分别进行融合处理。图1.f是简单平均法得到的融合结果,图1.g是采用Laplacian金字塔方法得到融合结果,图1.h是采用8层haar小波分解,对于高频层采用取极大值方法,低频层采用平均的融合方法得到的融合结果。

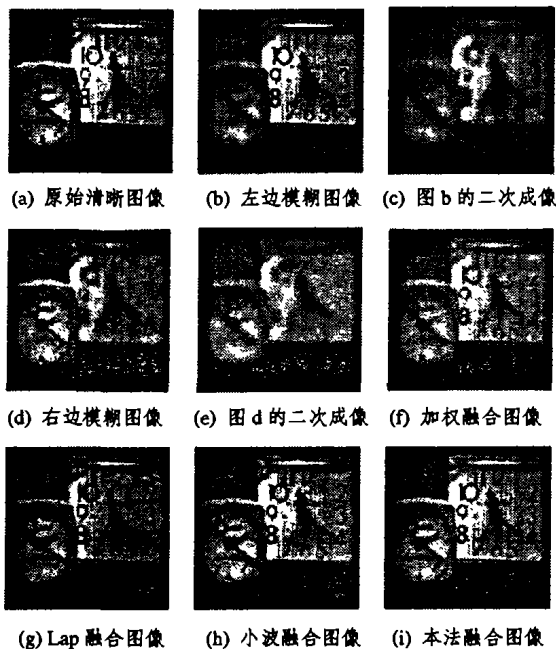


图1 原始图像和二次成像以及融合效果图

从仿真结果可以看出,本文的方法明显比其它三种方法好些,采用简单平均法和Laplacian塔型方法的结果明显不如小波方法和本文方法。

由于有标准原始图像,我们采用均方误差(MSE)作为评价融合效果的标准,融合图像 F 与理想图像 R 的均方误差为:

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [R(i,j) - F(i,j)]^2 \quad (8)$$

式(8)中, M 和 N 分别是图像的宽度和高度, $F(i,j)$ 是融合结

果图像在坐标 (i,j) 处的像素值, $R(i,j)$ 是原始清晰图像在坐标 (i,j) 处的像素值。MSE越小,说明融合效果越好,反之,则融合效果越差。上述四种融合算法的实验结果如表1所示。

表1 四种融合算法的客观评价测度值

	简单平均法	Laplacian塔型	小波变换方法	本文方法
MSE	134.2078	84.3376	25.0868	9.4985

从表中可以看出,本文的方法均方误差最小,小波方法次之,简单平均方法最大,这说明本文方法优于其它几种方法。

结论 本文提出了一种基于图像的模拟二次成像以及二次成像前后对清晰度的差异进行多聚焦图像融合的方法。通过分析光学系统成像的基本模型和清晰度的计算方法,提出了模拟光学系统成像的二次成像的方法,并对二次成像前后清晰度的差异情况进行了分析,分析出了每种情况下如何判定清晰目标的位置。

本文方法是基于空域的处理,不需要进行任何分解,而Laplacian方法和小波方法都是基于频域的处理,必须进行多级分解才能够提取出有效的清晰区域,分解、系数分析处理以及重构都不可避免会损失一些有用信息,或者产生一些相关冗余信息,这些都是本文方法不会存在的问题。实验结果证明,通过本文方法,最大可能地保留了各幅图像中的清晰区域及其各种特征信息,具有比其它几种方法更好的融合效果。

参考文献

- Aggarwal J K. Multisensor fusion for Computer Vision [M]. Berlin: Springer-Verlag, 1993
- Varshney P K. Multisensor data fusion [J]. Electron. Commun. Eng. J., 1997, 9(12): 224~253
- Hall D L, Linares J. An introduction to multisensor data fusion [J]. Proc. IEEE, 1997, 85(1): 6~23
- 刘贵喜. 多传感器图像融合方法研究 [D]. 西安: 西安电子科技大学, 2001
- 曹茂永, 孙农亮, 郁道银. 高焦模糊图像清晰度评价函数的研究 [J]. 仪器仪表学报, 2001, 22(3. 增刊): 259~260
- 徐飞, 施晓红. MATLAB应用图像处理 [M]. 西安: 西安电子科技大学出版社, 2002
- 杨旭, 杨万海, 裴继红. 基于小波分解的不同聚焦点图像融合方法 [J]. 电子学报, 2001, 29(6): 846~848
1998. 103~112
- Yao X, Lin G, Liu Y. An Analysis of Evolutionary Algorithms Based on Neighborhood and Step Size. In: Proc. of the 6th Annual Conf. on Evolutionary Programming, Lecture Notes in Computer Science Vol. 1213, Springer-Verlag, 1997. 297~307
- Matsumura Y, Ohkura K, Ueda K. Evolutionary Programming with Non-Coding Segments for Real-valued Function Optimization. In: Proc. of IEEE Intl. Conf. on Systems, Man and Cybernetics (SMC'99), 1999, 4: 242~247
- Ohkura K, Matsumura Y, Ueda K. Robust Evolution Strategies. Applied Intelligence, Kluwer Academic Publishers, accepted for publication, 2002
- Matsumura Y, Ohkura K, Ueda K. Evolutionary Dynamics of Evolutionary Programming in Noisy Environment. In: Proc. of Congress on Evolutionary Computation (CEC2001), 2001. 17~24
- Saravanan N, Fogel D B. Multi-Operator Evolutionary Programming: A Preliminary Study on Function Optimization. In: Proc. of the 6th Annual Conf. on Evolutionary Programming, Lecture Notes in Computer Science, Springer-Verlag, 1997, 1213: 215~221
- Wrenwood G G, Zhu Q J. Convergence in Evolutionary Programs with Self-Adaptation. Evolutionary Computation, MIT Press, 2002, 2(11): 147~157
- Rudolph G. Self-adaptation and Global Convergence: a Counterexample. In: Proc. of Cong. on Evol. Comp. 1999. 646~651
- Rudolph G. Self-Adaptive Mutations May Lead to Premature Convergence. Evolutionary Computation, IEEE Transactions on Publication, 2001, 5: 410~414
- Fogel L J, Owens A J, Walsh M J. Artificial Intelligence Through Simulate Evolution. John Wiley, Chichester, UK, 1996
- Yao X, Liu Y, Lin G. Evolutionary Programming Made Faster. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82~102
- Yao X, Liu Y. Scaling up Evolutionary Programming Algorithms. In: Proc. of the 7th Annual Conf. on Evolutionary Programming,

参考文献