

文章编号: 1006-5911(2001)05-0028-05

实数编码遗传算法的缺陷分析及其改进<sup>\*</sup>

叶晨洲, 杨杰, 黄欣, 陈念贻

(上海交通大学图象处理与模式识别研究所, 上海 200030)

**摘要:** 首先分析了一个典型的遗传算法处于某一阶段时各参数间的制约关系。指出由于实数编码遗传算法中交叉与变异操作改造个体时, 是通过改变构成个体的部分或全部实数(而不是位)实现的, 造成了个体转移矩阵(与位编码方式相比)呈现出特殊的模式, 而正是这种特殊模式, 使实数编码遗传算法更易于失效。基于这一结论, 提出了采用次级遗传算法改造实数编码遗传算法中交叉、变异算子的改进方法。该次级遗传算法将解空间各维上的实数增量组成基因串, 旨在提高交叉、变异中个体转移方向与优化对象的符合程度。实验结果表明, 改进后的算法在解决文中所涉及的优化问题时, 性能明显优于原算法。由于改进后的算法是一个纯粹的遗传算法, 因此能更好地保留遗传算法的各项优点。

**关键词:** 计算机集成制造系统; 遗传算法; 优化问题

**中图分类号:** TP18      **文献标识码:** A

流程工业 CIMS 的一大特点是生产控制技术复杂。如何使生产原料的组合(例如, 炼焦用煤炭的混合)或生产条件的控制(例如, 合成时的温度与压力调节)达到最优或在目前条件下的最优, 需要一种有效的优化算法。与传统的寻优方法(例如, 最速下降法、牛顿法<sup>[1]</sup>等)甚至模拟退火方法<sup>[2]</sup>相比, 遗传算法<sup>[3,4]</sup>能够在对象数学模型尚不十分明确, 且存在大量约束条件的情况下, 更好地排除局部最优解的干扰, 达到全局最优解, 因而它被用于与复杂或未知对象相关的参数优化过程中。为了使传统的遗传算法更便于解决连续型参数的优化问题, 人们提出了将实数作为基因, 组成基因串的实数编码遗传算法<sup>[3]</sup>。本文分析了实数编码遗传算法中常用的杂交、变异算子的缺陷, 进而提出了一种改进措施。

## 1 实数编码遗传算法的缺陷

### 1.1 实数编码遗传算法

在实数编码遗传算法中, 基因串  $X_i$  ( $i = 1, 2, \dots, N$ ) 的形式为  $(a_{i1}, a_{i2}, \dots, a_{in})$ , 其中,  $a_{ik} \in$

$D(a_k) \subset R$  ( $k = 1, 2, \dots, n$ ),  $n$  为解空间的维数,  $N$  为种群尺寸。

算法中对一对基因串  $X_i, X_j$  的交叉操作, 将根据某种方式选取若干交叉位置, 这些位置的标号组成一标记集  $L$ , 交叉产生的新个体  $X_p, X_q$  按以下方式取值: 当  $k \notin L$  时,  $a_{mk} = a_{mk}$  ( $m = p, q$ ), 当  $k \in L$  时,  $a_{pk} = r \times a_{ik} + (1-r) \times a_{jk}$ ,  $a_{qk} = r \times a_{jk} + (1-r) \times a_{ik}$ ,  $0 < r < 1$ 。对基因串  $X_i$  的变异操作, 将根据某种方式选取若干变异位置, 这些位置的标号组成一标记集  $L$ , 变异后  $X_i$  按以下方式取值: 当  $k \notin L$  时,  $a_{ik} = a_{ik}$ , 当  $k \in L$  时,  $a_{ik} = a_{ik} + \delta$ ,  $\delta$  是按某一方式获得的随机变量。算法的执行过程与传统的基于位串的遗传算法类似。相对于位编码, 实数编码避免了位编码与实参数间的转换操作, 以及由此引起的量化误差, 并在理论上能以任意精度取得结果。

### 1.2 遗传算法参数间的制约关系

为了揭示遗传算法各参数间的制约关系, 我们设计一个种群规模为  $N$ 、杂交概率为  $P_c$ 、变异概率为  $P_m$  的典型算法。其步骤如下:

<sup>\*</sup> 收稿日期: 2000-04-17; 修订日期: 2000-07-13

基金项目: 国家 863/CIMS 主题资助项目(863-511-945-005; 863-306-ZD13-05-6)

作者简介: 叶晨洲(1974-), 男(汉族), 上海人, 上海交通大学图象处理与模式识别研究所博士研究生, 主要从事数据挖掘等研究。

(1) 计算种群中每一个体的适应度  $F_i$  ( $i = 1, 2, \dots, N$ ); 计算当前种群的平均适应度  $F$ , 获得每一个体的归一化适应度  $F_i/F$ ;

(2) 赋给每一个体一生存概率(进入交配池的概率), 其值为该个体的归一化适应度, 按这一概率选择  $M$  ( $M \geq N$ ) 个个体, 组成一交配池;

(3) 从交配池中随机取  $P_c \times N$  个父本个体, 对每个被选中的父本个体, 从交配池中随机选择一个母本个体进行杂交, 它们产生一个子代个体;

(4) 用所有子代个体取代父本个体;

(5) 从交配池中随机取  $P_m \times N$  个个体, 对每个被选中的个体进行变异, 产生一个子代个体;

(6) 子代个体依据适应度, 或者取代原种群中的个体, 或者被丢弃;

(7) 若满足中止条件, 则停止运行, 否则进入下一代。

遗传算法运行时, 每一代种群中总存在一个或多个相同, 且适应度最高的个体  $I_x$  ( $x$  是能够唯一标识该个体形态的值, 并且仅取决于该个体的构成)。若在算法连续若干代运行中  $I_x$  不变, 我们定义此时算法处于阶段  $S_x$ , 包括该  $I_x$  首次产生至最后一次保持适应度最高时算法所经历的运行代数。一个遗传算法的运行过程可表示为队列  $S_a, S_b, \dots, S_x, \dots$ 。若算法在不可接受的长时间内停留在阶段  $S_x$ , 且  $I_x$  不是问题的最优解或可接受的次优解时, 我们认为遗传算法失效。

下面针对上述算法的某一阶段  $S_x$  进行分析。假设此阶段内任一代种群中适应度最高的个体都为  $I_x$ , 其适应度为  $f_{\max}$ , 当前为第  $k$  代且不是  $S_x$  包括的最后代。定义此时平均适应度为  $F$ , 最高适应度与平均适应度之差  $\varepsilon = f_{\max} - F$ , 种群中  $I_x$  的数目为  $n_{xk}$ , 下一代种群中  $I_x$  的数目为  $n_{xk+1}$ 。若不计截留误差, 可获得以下三个结论。

**结论 1:** 当  $\varepsilon > (1 - ((1 - P_c N/M)(1 - P_m N/M))^{-1} - \forall(n_{xk}))^{-1} M/N) f_{\max}$  时,  $n_{xk+1} > n_{xk}$  成为可能, 其中,  $\forall(n_{xk}) > 0$ , 且随  $n_{xk}$  的增加而以一递减的速率增大, 当  $n_{xk}$  较大时,  $\forall(n_{xk})$  增大的速率接近于 0。

**证明:** 令  $z = n_{xk} \frac{f_{\max}}{f_{\max} - \varepsilon} \times \frac{M}{N}$ , 它是交配池中  $I_x$

数目的期望值,  $P'_c = P_c \frac{N}{M}$ ,  $P'_m = P_m \frac{N}{M}$ , 它们是交配池中实际的交叉与变异概率。对杂交操作而言, 若

父、母本个体完全相同, 则它们将产生一个与它们完全相同的子代个体, 若父、母本个体同为  $I_x$ , 则它们的子代也为  $I_x$ , 并且根据假设该子代必定进入下一

代, 所以:  $n_{xk+1} \geq P'_c z (\frac{z}{M}) (1 - P'_m) + (1 - P'_c) z (1 - P'_m)$ , 为使  $n_{xk+1} > n_{xk}$ , 须使  $P'_c (1 - P'_m) \frac{1}{M} z^2 + (1 - P'_c) (1 - P'_m) z - n_{xk} > 0$ 。

由一元二次方程的性质得:

$$z > \left( \sqrt{\left( \frac{M}{P'_c} (1 - P'_c) \right)^2 + 4 \frac{n_{xk} M}{(1 - P'_m) P'_c}} - \sqrt{\left( \frac{M}{P'_c} (1 - P'_c) \right)^2} \right) / 2$$

设  $f(y) = \sqrt{y}$ , 令

$$\forall(n_{xk}) = \left[ f' \left( \left( \frac{M}{P'_c} (1 - P'_c) \right)^2 \right) \frac{4 n_{xk} M}{(1 - P'_m) P'_c} - \left( f' \left( \left( \frac{M}{P'_c} (1 - P'_c) \right)^2 + \frac{4 n_{xk} M}{(1 - P'_m) P'_c} \right) - f' \left( \left( \frac{M}{P'_c} (1 - P'_c) \right)^2 \right) \right] / (2 n_{xk})$$

则当  $\varepsilon > (1 - \frac{1}{1/(1 - P'_c)(1 - P'_m) - \forall(n_{xk})} \times \frac{M}{N}) f_{\max}$  时,  $n_{xk+1} > n_{xk}$ 。由  $f(y)$  的性质可知,  $\forall(n_{xk})$  随  $n_{xk}$  的增加而以一递减的速率增大, 当  $n_{xk}$  较大时,  $\forall(n_{xk})$  增大的速率接近于 0。

**结论 2:** 杂交操作产生新个体的概率  $P_{cv}$  的上限, 随  $n_{xk}$  的增加而加速减小。

**证明:**  $P_{cv} \leq \frac{MP'_c - P'_c z^2/M}{MP_c} \leq 1 - (\frac{n_{xk}}{N} \times \frac{f_{\max}}{f_{\max} - \varepsilon})^2 \leq 1 - (\frac{n_{xk}}{N})^2$ ,  $\partial(1 - (\frac{n_{xk}}{N})^2) / \partial n_{xk} = -\frac{2n_{xk}}{N^2}$ , (其中部分符号定义参见结论 1 的证明)。

**结论 3:** 子代个体要进入下一代种群, 其适应度的下限为:

$$\theta = \frac{(N - n_k) f_{\max} - N \varepsilon}{(N - n_k) + P_{cv} P_c N (1 - P_m N/M) + P_m N - 1}$$

**证明:** 设当前种群中与  $I_x$  不同的个体总数为  $n_a$ ,  $\Sigma a$  为它们的适应度之和, 交配池中与  $I_x$  不同的子代个体总数为  $n_b$ ,  $\Sigma b$  为它们的适应度之和, 欲使  $I_i$  进入下一代种群, 须有  $F_i \geq \frac{\Sigma a + \Sigma b}{(n_a + n_b)}$  成立, 因为  $n_a = N - n_k$ ,  $\Sigma a = (N - n_k) f_{\max} - N \varepsilon$ ,  $n_b \leq P_c P_{cv} N + P_m N - P_c P_{cv} P_m N^2/M$ , 而  $\Sigma b \geq F_i$ 。

根据上述三个结论, 对处于阶段  $S_x$  的算法不难构造这样一个“消极”过程: 第  $k$  代时  $n_{xk+1} > n_{xk}$  (结论 1), 这导致第  $k+1$  代的  $P_{cv}$  呈下降趋势 (结论 2), 进一步限制了第  $k+1$  代交配池中产生与  $I_x$  不

同的新个体的数量, 这些个体或者全部被丢弃, 或者部分或全部进入下一代(结论3), 若前者发生, 由于  $n_{xk+1} > n_{xk}$ , 则会加强  $I_x$  个体增加的趋势(结论一), 若后者发生,  $\varepsilon$  必然减小从而加强了交配池中与  $I_x$  不同的个体进入下一代种群的限制(结论三)。这样一个过程总是阻止当前群体引入新个体。

同时, 还存在一“积极”过程: 从当前群体中产生一个新的个体取代  $I_x$ , 从而使算法进入新阶段。当算法处于阶段  $S_x$  时存在着这两种过程的竞争。如果“消极”过程长期占主导地位, 则算法将可能在不可接受的长时间内停留在阶段  $S_x$ , 若  $I_x$  不是问题的最优解或可接受的次优解时, 算法被认为失效。

### 1.3 实数编码遗传算法交叉变异算子的缺陷

对第  $k$  代的交配池  $M_k$ , 每个个体  $I_i$  接受某种操作  $OP$  (为  $C$  时表示杂交操作, 为  $M$  时表示变异操作, 为  $CM$  时表示先杂交后变异操作) 后, 向所有可能存在的个体转化的概率是确定的, 因此可获得一状态转移矩阵  $T_{OPki} = (P_{OPki1}, P_{OPki2}, \dots, P_{OPkiL})^T$ ,  $P_{OPkij}$  是第  $k$  代交配池中的个体  $I_i$  接受操作  $OP$  后, 向  $I_j$  转变的概率,  $L$  是不同个体的总数(事实上, 即使是实数编码的遗传算法, 它产生的个体种类也是有限的, 取决于运行环境对实数的表示方式),  $T_{OPki}$  的构成取决于杂交、变异算子产生子代的方式及当前种群的构成。个体  $I_i$  接受某种操作  $OP$  后, 产生适应度大于  $\theta$  的后代概率为  $P_{\theta OPki} = \sum_{j=1}^L P_{OPkij} p_{\theta object}(j)$ ,  $p_{\theta object}(j)$  是优化对象在  $I_j$  表示的位置处适应度大于  $\theta$  的概率分布。易知当  $T_{OPki}$  的构成与  $p_{\theta object}$  一致时,  $p_{\theta OPki}$  取最大值。

相对于位编码遗传算法, 实数编码遗传算法中交叉与变异操作改造个体时, 是通过改变构成个体的部分或全部实数(而不是位)实现的。这样, 造成了  $T_{OPki}$  的构成(与位编码方式的相比)呈现出特殊的模式。例如 1.1 算法中杂交算子产生的后代局限在父母本个体张成的最小凸空间内(位编码遗传算法当父母本个体不同时, 不存在这一问题), 并且当交叉或变异的点数小于个体长度时,  $T_{OPki}$  的构造中存在着大量不可到达的“死区”(位编码遗传算法在交叉或变异点数合理时, 存在这一问题的风险较小)。相对于位编码遗传算法, 实数编码遗传算法中与某一  $I_i$  对应的  $T_{OPki}$  的构成, 更易与  $p_{\theta object}(j)$  严重不一致, 这时,  $p_{\theta OPki}$  取值往往很小, 甚至为 0。

综合 1.2 中的分析可知, 当算法停留在某一阶

段  $S_x$  时, 若交配池中与  $I_x$  相同的个体, 经遗传算子变化后, 产生适应度更高的个体的概率很微小, 并且交配池中与  $I_x$  不同的个体经遗传算子变化后, 产生适应度大于适应度下限(结论3)的个体的概率也很微小时, “消极”过程得到了加强。这可能导致种群趋于单调(种群中存在着大量相同或相似的个体<sup>[6]</sup>)。这时, 所有个体经遗传操作向其他个体转移的状态转移矩阵也趋于一致。由于实数编码遗传算法交叉、变异算子的特殊性, 更容易造成种群中所有个体的状态转移矩阵与优化对象的构造严重不符合的情况。而这又加剧了“消极”过程, 增大了算法失效的风险。这正是文献[5]中报告的坐标旋转后, 实数编码遗传算法性能大幅降低的原因。

## 2 采用遗传算法改进实数编码遗传算法的交叉、变异算子

### 2.1 改进依据

由前述分析可知, 为减小实数编码遗传算法在未取得最优解之前, 长时间停留在某一阶段  $S_x$  的风险, 可采取两项改进措施: ①改进算法的控制参数(种群规模、复制、交叉及变异的概率等), 推迟种群单调化发展的开始时间; ④改进交叉变异算子, 使个体的转化方向与优化对象的构造尽可能一致(文献[6]的实验结果也证明了这一点)。一些混合遗传算法<sup>[7, 8]</sup>正是采用了这些手段。然而, 许多优化对象尚无明确的数学模型, 也无法得知其各个局部特性, 因此实施这些改进存在着困难和限制。我们引入了次级遗传算法提高原算法中交叉、变异算子的效率, 从而使遗传算法能够保持和利用自身的优势, 提高性能。

### 2.2 次级遗传算法

改进的交叉与变异算子采用了相同的次级遗传算法。该算法描述如下:

(1) 在某一初始点  $X = (x_1, x_2, \dots, x_n)$  (解空间中的点)附近, 随机产生规模为  $N'$  的种群。种群中的基因串可由  $\hat{X}_i = (\Delta x_{i1}, \Delta x_{i2}, \dots, \Delta x_{in})$  ( $i = 1, 2, \dots, N'$ ) 表示, 其中  $\Delta x_{im}$  ( $m = 1, 2, \dots, n$ ) 表示相对于  $x_m$  的增量且  $\Delta x_{im} + x_{im} \in D(x_{im})$ ;

(2) 按 1.1 中的方法执行遗传算法, 并设定  $r = 0.5$ , 交叉点、变异点的数目和位置在  $1 \sim n$  上均匀选取, 种群中基因串的适应度是该基因串与  $X$  叠加后, 在主遗传算法中的适应度, 当执行的代数达到预设值或基因串的适应度满足要求时, 停止执行算

法, 与 1. 1 的不同之处在于每次杂交只产生一个后代, 变异时对处于变异点上的  $\Delta x_{im}$  重新初始化;

(3) 将最优基因串叠加在  $X$  上, 作为结果返回。

2.3 基于遗传算法的交叉、变异算子及改进的实数编码遗传算法

主遗传算法仿照 1. 2 中设计的遗传算法。交叉操作中对父本个体  $X_f$  与母本个体  $X_m$ , 令  $X = (X_f + X_m) / 2$ , 根据解空间尺寸决定  $D(\Delta x_{im}) (m = 1, 2, \dots, n)$ , 据此, 采用次级遗传算法产生一个后代, 取代  $X_f$ 。变异操作选择一  $X_f$ , 根据解空间尺寸决定  $D(\Delta x_{im}) (m = 1, 2, \dots, n)$ , 采用次级遗传算法产生一个后代, 取代  $X_f$ 。变异点的数目和位置在  $1 \sim n$  上均匀选取。我们称采用上述交叉及变异算子的实数遗传算法为改进的实数编码遗传算法。

3 实验结果

在以下实验中, 改进的遗传算法中主遗传算法与次遗传算法的群体规模都为 200, 杂交概率都为 0. 8, 变异概率都为 0. 1,  $D(\Delta x_{im}) (m = 1, 2, \dots, n)$  为  $x_m$  定义域长度的  $\pm 12. 5\%$  (对实验一、三) 或  $x_m$  定义域长度的  $\pm 50\%$  (对实验二), 次级遗传算法的

最大执行代数 为 10。算法采用的适应度函数为  $1 / (1 + f(X) - f_{\min})$ 。用于对照的算法是从改进算法中去掉次级算法获得的(它是一个典型的基于实数编码的遗传算法)。

为体现典型实数编码遗传算法中交叉、变异算子的缺陷带来的影响, 实验一和实验二(实验三无需旋转即能体现这种影响) 中对最后两维坐标进行了如下旋转变换:  $x_{n-1} = x_{n-1} \cos \theta - x_n \sin \theta, x_n = x_n \cos \theta + x_{n-1} \sin \theta$ , 然后使  $x_{n-1} = x_{n-1}, x_n = x_n$  (可以证明该变换在实数向量空间中是一等距同构映照<sup>[9]</sup>, 不改变函数的面貌)。实验对比了连续 10 次执行改进前后的算法对目标函数进行寻优的结果。每次运行中两种算法对目标函数的采样次数相同。

3.1 实验一

求 Ellipsoid 函数的最小值:

$$f_8(x) = \sum_{i=1}^n 10^{i-1} x_i^2 \quad -0.1 < x_i < 3.0 \quad (1)$$

其最小值为 0。表 1 为  $n = 4$  时的对比结果, 表 2 为  $n = 4$  且坐标旋转  $45^\circ$  后的对比结果, 表 3 为  $n = 6$  时的对比结果, 表 4 为  $n = 6$  且坐标旋转  $45^\circ$  后的对比结果。

表 1

算法	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次	第 6 次	第 7 次	第 8 次	第 9 次	第 10 次	胜/负
原算法	0. 006500	0. 000417	0. 000454	0. 006094	0. 000362	0. 003156	0. 000243	0. 000246	0. 001865	0. 008445	0 / 10
改进算法	0. 000003	0. 000004	0. 000001	0. 000007	0. 000010	0. 000005	0. 000005	0. 000001	0. 000002	0. 000003	10 / 0

表 2

算法	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次	第 6 次	第 7 次	第 8 次	第 9 次	第 10 次	胜/负
原算法	0. 548323	0. 254859	0. 086588	0. 680680	0. 114989	0. 091865	0. 093670	0. 416056	0. 000723	0. 204993	0 / 10
改进算法	0. 000007	0. 000007	0. 000005	0. 000004	0. 0000090	0. 000008	0. 000005	0. 000010	0. 000004	0. 000003	10 / 0

表 3

算法	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次	第 6 次	第 7 次	第 8 次	第 9 次	第 10 次	胜/负
原算法	0. 780131	1. 248409	7. 079840	2. 200982	0. 908900	0. 882691	1. 237801	12. 613442	5. 139993	0. 159640	0 / 10
改进算法	0. 000292	0. 000165	0. 000437	0. 000035	0. 000354	0. 000155	0. 000515	0. 000276	0. 000094	0. 000051	10 / 0

表 4

算法	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次	第 6 次	第 7 次	第 8 次	第 9 次	第 10 次	胜/负
原算法	31. 756510	63. 564457	58. 659451	92. 885994	22. 576706	164. 593094	33. 775970	21. 638279	30. 262918	18. 408747	0 / 10
改进算法	0. 000311	0. 002284	0. 000797	0. 001325	0. 000864	0. 002250	0. 000260	0. 001486	0. 001837	0. 000444	10 / 0

3.2 实验二

求 Schwefel 函数的最小值:

$$f_7(x) = - \sum_{i=1}^n x_i \sin(\sqrt{|x_i|}), \quad -500 < x_i < 500 \quad (2)$$

当  $n = 6$  时, 其最小值约为  $-2513. 9$ 。表 5 为  $n = 6$  时的对比结果, 表 6 为  $n = 6$  且坐标旋转  $20^\circ$  后的对比结果(定义域也作了相应裁减)。

表 5

算法	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次	第 6 次	第 7 次	第 8 次	第 9 次	第 10 次	胜/负
原算法	- 2513. 40	- 2512. 57	- 2512. 407	- 2511. 43	- 2513. 46	- 2510. 58	- 2511. 16	- 2513. 30	- 2513. 29	- 2513. 39	0/10
改进算法	- 2513. 84	- 2513. 83	- 2513. 87	- 2513. 88	- 2513. 87	- 2513. 86	- 2513. 87	- 2513. 88	- 2513. 87	- 2513. 86	10/0

表 6

算法	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次	第 6 次	第 7 次	第 8 次	第 9 次	第 10 次	胜/负
原算法	- 2387. 98	- 2513. 67	- 2361. 04	- 2391. 83	- 2393. 27	- 2394. 05	- 2512. 31	- 2394. 18	- 2391. 61	- 2390. 92	0/10
改进算法	- 2513. 88	- 2513. 81	- 2513. 88	- 2513. 86	- 2513. 86	- 2513. 86	- 2513. 88	- 2513. 85	- 2513. 87	- 2513. 88	10/0

3.3 实验三

求 Rosenbrock 函数的最小值:

$$f_2(x) = \sum_{i=1}^3 (100(x_i^2 - x_{i+1})^2 + (1 - x_i)), -5 < x_i < 5 \quad (3)$$

其最小值为 0(这一函数对个体的转化方向有很严格的要求<sup>[5,6]</sup>)。表 7 为 n= 6 时的对比结果。

表 7

	第 1 次	第 2 次	第 3 次	第 4 次	第 5 次	第 6 次	第 7 次	第 8 次	第 9 次	第 10 次	胜/负
原算法	5. 14617	3. 32166	4. 02462	3. 52077	3. 99912	4. 37550	5. 90349	4. 47958	3. 86297	4. 44844	0/10
改进算法	0. 00153	0. 45863	0. 00930	0. 01475	0. 00697	0. 05649	0. 19690	0. 00215	0. 00160	0. 03533	10/0

3.4 结果比较

表 1、表 3 和表 5 对应的实验中,改进前后的算法都逃离了局部最优解,但由于原算法在整个解空间上采用相同的交叉、变异方法,它与全局最优点的距离大于改进算法获得的结果。表 2、表 4 和表 6 是优化对象旋转后的测试结果。由于此时原算法的交叉、变异过程中,个体转移方向与优化对象产生了严重不一致,助长了“消极”过程,因而在很长时间内远离全局最优,算法性能明显下降。改进算法所受影响不明显。表 7 的结果也反映了类似的情况。

4 结束语

本文分析了一个典型的遗传算法处于某一阶段时各参数间的制约关系。认为算法处于该阶段时存在着两种过程:一方是“消极”过程,它总是阻止当前群体引入新个体;另一方是“积极”过程,它从当前群体中产生一个新的个体,其适应度优于原有的任何个体,从而使算法进入一个新阶段。相对于位编码遗传算法,实数编码遗传算法中交叉与变异操作改造个体时,是通过改变构成个体的部分或全部实数(而不是位)实现的,造成个体转移矩阵的构成(与位编码方式相比)呈特殊的模式。这种模式使实数编码遗传算法在种群趋向单调时,更易于助长“消极”过程,从而导致算法失效。

根据分析,并结合遗传算法的特性,本文提出了采用次级遗传算法改造交叉、变异算子的方法。该次级遗传算法将解空间各维上的实数增量组成基因

串,旨在提高交叉、变异中个体转移方向与优化对象的符合程度。实验结果表明,改进后的算法在解决文中所涉及的优化问题时,性能明显优于原算法。由于改进后的算法是一个纯粹的遗传算法,因此能更好地保留算法的隐含并行性、全局寻优性、约束条件的易实现性及可并行计算等特点<sup>[3]</sup>,满足了工程设计优化的要求。

参考文献:

[1] 袁亚湘,孙文瑜. 最优化理论与方法[M]. 北京: 科学出版社, 1999.

[2] WILLIAM E H. A theoretical comparison of evolutionary algorithms and simulated annealing[A]. Evolutionary programming V: proceedings of the fifth annual conference on evolutionary programming[C]. MIT Press, 1996 : 147 - 153.

[3] 刘 勇,康立山,陈毓屏. 非数值并行算法(第二册) - 遗传算法[M]. 北京: 科学出版社, 1998.

[4] LANCE C. Practical handbook of genetic algorithms applications volume 1[M]. CRC Press, 1995.

[5] RALF S. Performance degradation of genetic algorithm under co-ordinate Rotation[A]. Evolutionary Programming V: Proceedings of the fifth annual conference on evolutionary programming[C]. MIT Press, 1996: 155 - 161.

[6] 梁艳春,周春光,李寿范. 基于遗传算法 Rosenbrock 函数优化问题的研究[J]. 软件学报, 1997, 8(9): 701 - 707.

[7] 彭 伟,卢锡城. 一种函数优化问题的混合遗传算法[J]. 软件学报, 1999, 10(8): 819 - 823.

[8] 曹先彬,郑 振,范 磊. 融合神经网络的一种改进遗传算法[J]. 模式识别与人工智能, 1999, 12(4): 486 - 492.

[9] 程其襄,张奠宙,魏国强,等. 实变函数与泛函分析基础[M]. 北京: 高等教育出版社, 1983.

(下转第 41 页)

- ufacturing environment for rapid prototyping and education [J]. International Journal of Computer Integrated Manufacturing, 1999, 12(1): 54 – 60.
- [3] URICH G, FRANZ J, HAUCK. Promondia: a java – based framework for real – time group communication in the web[EB/OL]. <http://www4.informatik.uni-erlangen.de/promondia/>.
- [4] 马晓龙, 冯 键, 林守勋, 等. WCEEC – 一个基于 Web 的电子会议系统[J]. 计算机辅助设计与图形学学报, 1999, 11(3): 259 – 261.
- [5] APPELT W, BUSBACH U. The BSCW system: a WWW based application to support cooperation of distributed groups[EB/OL]. <http://orgwis.gind.de/busbach/wetice.ps>, 1996.
- [6] LESLIE M. An integrated CSCW architecture for integrated product / process design and development[J]. Robotics and Computer – Integrated Manufacturing, 1999, 15: 145 – 153.
- [7] KINDBERG T. Mushroom: a framework for collaboration and interaction across the Internet [A]. Proc. of the ERCIM Workshop on CSCW and the Web [C]. Germany: 1996.
- [8] NIKOLA B. SERBEDZIJ. Web computing framework [J]. Journal of System Architecture, 1999, 45: 1293 – 1306.
- [9] 曹 健, 张友良, 等. 基于 Agent 的产品并行开发过程管理系统 [J]. 中国机械工程, 1999, 10(4): 414 – 417.
- [10] 王 茜, 沈 璐, 等. 企业 CIMS 集成中的 Internet 技术[J]. 计算机集成制造系统 – CIMS, 2000, 6(1): 70 – 73.
- [11] 慎 健. 基于 Java 平台的多层应用软件体系结构[N]. 计算机世界, 1999 – 9 – 13(C5), (C9).

## Framework and Realization of Inter – enterprise Cooperative Work in the Web

HA Jin – bing, ZHANG You – liang, LI Zhou – zhou

(CIMS Institute, Nanjing University of Science & Technology, Nanjing 210094, China)

**Abstract:** With the increasing demands of agility, a cooperative design and manufacturing environment must be built between the enterprises. The widely used web solutions show poor performance in interactivity and logic computing. They are mainly focus on the static information sharing or limited dynamic databases sharing. The paper is concerned with employing the web technology for the cooperative product development system. A five – layer architecture is proposed to built an inter – enterprise cooperative work environment. The implementation and characteristic of the system are also discussed.

**Key words:** cooperative work; web; enterprise; Internet

(上接第 32 页)

## The Drawback of Real – Valued GA and Its Remedy

YE Chen – zhou, YANG Jie, HUANG Xin, CHEN Nian – yi

(Inst. of Image Processing & Recognition, Shanghai Jiaotong University, Shanghai 200030, China)

**Abstract:** This paper analyses the interaction among the parameters of a typical GA running at a specific stage, and points out: because the cross – over and mutation operators of real – valued GA modify individual by digits instead of bits, the transition matrix of each individual gets a special pattern; it is this special pattern that hurts the efficiency of real – valued GA. Based on this conclusion the paper puts forward a new way to remedy the drawback. It replaces the original operators with new ones that adopt a second level real – valued GA, in which the gene stand for increment of each dimension, to improve the consistency of the operators with the optimization target. The testing results prove that the modification can improve the performance of real – valued GA in solving the optimization problems concerned in this paper. Because the modified algorithm is a “pure” GA, it can reserve the GA’s advantages entirely.

**Key words:** CIMS; genetic algorithm; optimization problem

凡投 CIMS 期刊的稿件, 请给出作者简介, 内容包括姓名、出生年、性别、民族、籍贯、学历、职称、工作单位及主要研究方向等。为保证印刷质量, 稿件插图请尽量不用彩色图、灰度图。谢谢合作!

另外, 为方便联系, 请作者在投稿时给出联系电话或 E – mail 地址。谢谢!