

## 免疫遗传算法及在新产品投入计划中的应用\*

汪定伟, 于海斌

(中国科学院 沈阳自动化研究所, 沈阳 110015)

**摘要:** 提出一种新的求解复杂约束优化问题的免疫遗传算法. 算法首先产生大量抗原来训练抗体, 从而建立起一个具有自体和非自体识别能力的免疫系统. 将该免疫系统嵌入到遗传算法中, 即可在遗传过程中不经解码就能识别非法或不可行的染色体. 算法有效地改进了遗传算法求解复杂约束优化问题的性能. 算法用于求解用半无限规划模型描述的新产品投入计划问题, 获得了满意的结果.

**关键词:** 免疫系统; 遗传算法; 机器学习; 新产品投入计划; 半无限规划; 约束优化

**中图分类号:** TP301

**文献标识码:** A

### Immune-genetic algorithm and its application to introduction planning of new products

WANG Ding-wei, YU Hai-bin

(Shenyang Institute of Automation, Academia Sinica, Shenyang 110015, China)

**Abstract:** To solve complex constrained optimization problems, we propose a new immune-genetic algorithm. It randomly produces a lot of antigens for production and training of antibodies. Then, an efficient immune system with the capability to recognize self and non-self antigens is consisted by these trained antibodies. We embed the immune system into genetic algorithm, and use it to identify the illegal and infeasible chromosomes in the genetic iterations. The recommended algorithm is able to improve the performance of GAs for complex constrained optimization problems. It has been applied into the new product introduction problem presented by a semi-infinite programming model. The satisfactory results have been achieved.

**Key words:** immune system; genetic algorithms; machine learning; introduction planning of new products; semi-infinite programming; constrained optimization

#### 1 引言(Introduction)

自 Holland 1975 年提出遗传算法(GA)以来<sup>[1]</sup>, 遗传算法一直是一个十分活跃的研究领域<sup>[2,3]</sup>. 然而, 如何在遗传过程中保持种群中染色体的合法性和可行性始终是一个难题, 特别是用非二进制编码的遗传算法解复杂约束优化问题时, 这个问题更为严重. 这是因为传统的 GA 缺少一个抵抗非自体(非法和不可行染色体)入侵的免疫系统.

免疫系统是生命体防止病毒、细菌和其他生物入侵的防卫系统<sup>[4,5]</sup>. 通过产生新的抗体和免疫网络的通讯, 免疫系统具有很强的学习、记忆和识别能力. 因此, 人们一直试图用计算机模仿这些功能来解决多种问题<sup>[6]</sup>. 1986 年, Farmer 和 Packard 首先提出免疫系统的人工智能模型<sup>[5]</sup>. 20 世纪 90 年代中, 免疫系统逐渐被应用到多种工业问题中, 包括机器故

障诊断、存储控制、噪声控制、车辆行为判断、电磁装置形状优化, 等等<sup>[6-9]</sup>.

本文提出一种新的免疫-遗传算法(I-GA). 算法首先随机地产生大量训练抗体的抗原, 然后用抗体组成一个免疫系统. 将免疫系统嵌入到遗传算法中, 用来在遗传迭代中识别非法的或不可行的染色体. 以上算法求解复杂的约束优化问题, 可以大大改进遗传算法的计算性能. 算法首先应用于新产品投入时间计划问题, 取得了满意的结果.

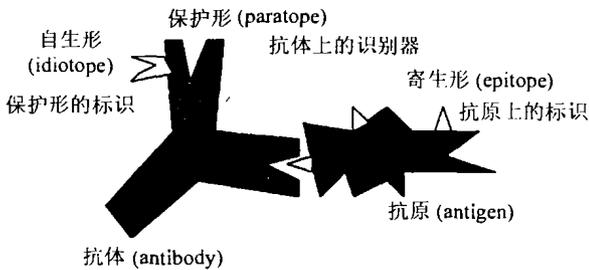
#### 2 免疫系统与免疫遗传算法(Immune system and immune-genetic algorithm)

按免疫学理论, 免疫系统主要是由 B-淋巴和 T-淋巴组成的<sup>[4,5]</sup>. 每个 B-淋巴有不同的分子结构, 在其表面产生“Y”型抗体. 抗体能识别特定的抗原. 抗原是入侵生命体的外来物质. 为了适应环境的不

\* 基金项目: 国家自然科学基金(60084003, 70171056)和沈阳市自然科学基金共同资助项目.

收稿日期: 2000-11-21; 收修改稿日期: 2001-05-31.

断变化,生物能不断产生大量的抗体.抗原上供抗体识别的部分称为寄生形(epitope),而抗体上识别抗原的部分则称为保护形(paratope).最近的研究表明,抗体上还有标明其所能识别抗原的自生形(idiotope).抗体与抗原的结构见图1.



抗体 (antibody)

图1 抗体与抗原的结构

Fig. 1 Structure of antigen and antibody

著名免疫学家 N.K. Jerne 提出“自生形网络假说”,又称为“免疫网络假说”<sup>[5]</sup>.即认为,抗体/淋巴不是孤立的,不同的抗体相互之间是能够通讯的.抗体间的刺激与抑制的链接关系组成了一个大型的网路,成为自体与非自体的识别器.

免疫网络的概念启发我们考虑将其作为遗传算法中合法性与可行性的识别器.

由于遗传运算通常会导致非可行的染色体,如何满足约束成了应用 GA 解约束优化问题的关键.现有的解决约束的方法主要有:拒绝策略、修复策略、改进遗传算子、惩罚策略<sup>[2,3]</sup>.然而,这些方法都必须将编码空间中的染色体解码映射到解空间,才能检查可行性.由于在每步遗传迭代中都必须检查所有染色体的可行性,这个过程是非常费时的.因此,用遗传算法解约束优化问题效率往往不高.

免疫-遗传算法中建立免疫系统的基本步骤分以下3个阶段:

- 1) 通过测试在编码空间中随机产生的大量的抗原的可行性,生成并训练抗体;
- 2) 基于免疫网络假设,通过抗体间的通讯加速抗体的学习过程;
- 3) 将训练后的抗体嵌入到遗传算法中,作为可行与不可行的识别器.

免疫系统中,抗体可以是一组阈值或对染色体的限制条件.这样,可行与非可行的识别可以直接在编码空间进行,而不必解码到解空间.这使得每步迭代中对染色体的可行性检查变得较为容易.由于抗体的训练过程是在遗传迭代之前离线进行的,免疫遗传算法比一般遗传算法更省时间.

### 3 新产品投入计划 (Introduction planning of new products)

新产品开发是任何企业赢得市场份额的关键因素.早期的研究集中在新产品开发的案例研究上<sup>[11]</sup>.由于制造资源和市场容量的限制,企业不可能一次投入多个新产品.因此,适时投入新产品对企业效益十分重要.如何在正确的时间投入正确的产品是一个十分重要的约束优化问题.

设企业在计划期  $[1, T]$  内,有  $n$  个待投入的新产品.产品  $i$  的收益函数为  $f_i(t)$ ,  $t \in [1, T]$ ,投入费用为  $c_i$ ,单位收益的资源需求为  $w_i$ ,  $i = 1, 2, \dots, n$ .企业的目标收益和可用资源量表达为函数  $P(t)$  和  $R(t)$ ,  $t \in [1, T]$ .一般我们认为收益和资源函数都是随时间单调增加的.于是,用指数函数表达为:

$$P(t) = P_0 e^{pt} \text{ 和 } R(t) = R_0 e^{rt}, t \in [0, T]. \quad (1)$$

其中,  $P_0, R_0$  和  $p, r$  是代表初值和增长速度的参数.

产品的生命周期分为4个阶段,即投入期、增长期、成熟期和衰退期.对于产品  $i$ ,其收益函数可表达为:

$$f_i(t) = a_i t^2 e^{-t/b}, t \in [0, T], i = 1, 2, \dots, n. \quad (2)$$

其中  $a_i$  和  $b_i$ ,  $i = 1, 2, \dots, n$ , 表示峰值和达到峰值时间的参数.如图2所示,该函数较好地表达了产品生命周期的4个阶段.选择不同参数就可获得不同类型的产品.

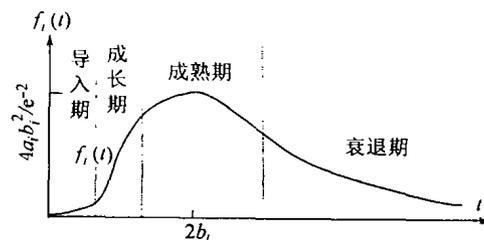


图2 新产品的收益曲线

Fig. 2 Profit curve of a new product

现有产品的收益可用函数  $Q(t)$  表达

$$Q(t) = Q_0 e^{-qt}. \quad (3)$$

这里,  $Q_0$  和  $q$  是代表现值 and 衰减率的参数.

$$x_i = \begin{cases} t \in [1, T], & \text{产品 } i \text{ 在 } t \text{ 时刻投入,} \\ 0, & \text{产品 } i \text{ 不在计划期内投入.} \end{cases} \quad (4)$$

定义变量  $x_i$  为产品  $i$  的投入时间,则新产品投入计划模型可表达为:

$$\max \sum_{i=1}^n \text{sgn}(x_i) \left[ \int_{x_i}^T f_i(t - x_i) dt - c_i \right], \quad (5)$$

$$\text{s.t. } \sum_{i=1}^n \text{sgn}(x_i) f_i(x-x_i) \geq p(t) - Q(t), \quad t \in [1, T], \quad (6)$$

$$\sum_{i=1}^n \text{sgn}(x_i) w_i f_i(t-x_i) \leq R(t) - w_0 Q(t), \quad t \in [1, T], \quad (7)$$

$x_i$  为 0 或  $[1, T]$  间的整数,  $i = 1, 2, \dots, n$ . (8)

其中,  $w_0$  是现行产品的单位资源需求.  $\text{sgn}(x)$  为符号函数, 即  $\text{sgn}(x)$  在  $x$  大于、等于和小于 0 时, 分别为 1, 0 和 -1.

式(5)~(8)是一个非线性的半无限规划模型. 当  $t$  在区间  $[1, T]$  中取无限个值时, 式(6)和式(7)有无限个约束. Fang 和 Wu 提出的非精确法提供了一条处理无限约束的途径<sup>[11]</sup>. 文献[12]介绍了将非精确法与智能算法结合解非线性半无限规划的方法.

#### 4 免疫系统的建立(Building immune system)

本问题用投入时间向量作基因表达方式是最自然的, 即用  $x = [x_1, x_2, \dots, x_n]$  代表一个染色体. 其中,  $x_i$  由公式(4)定义. 如一个计划期为  $[1, 10]$  的 8 个产品的例子:

$$x = [3, 8, 0, 0, 1, 7, 8, 10]$$

表示产品 1 在时间 3 投入, 产品 2 在时间 1, 产品 3 和 4 未被选中,  $\dots$ , 依此类推.

在以上整数表达中, 重复是允许的, 只要  $x_i$  在 0 和  $T$  之间. 当采用常用的两点交叉和换位变异时, 编码的合法性能够保持. 但是, 可行性却不易达到. 例如, 以上例子中, 在区间  $[4, 6]$  中由于没有产品投入目标收益约束可能不能满足, 而  $[7, 8]$  中由于投入产品过多, 又违反了资源约束. 因此, 如何保持可行性是对本问题的最大挑战.

##### 1) 抗体设计的基本思想.

抗体设计的基本思想是: 新产品的投入在计划期  $[1, T]$  中应是大体均匀的, 任何时间片段中都没有产品的堆积和短缺. 因为产品堆积会造成资源不足, 而产品短缺就达不到目标收益. 这样, 所有时间片段的产品的数值的上下限就可作为免疫系统的抗体. 所有时间片段的阈值/抗体连接成网络, 就构成了可行/非可行的识别器.

由于产品的峰值对收益和资源约束起到关键作用, 以时间段中的峰值数代表产品数是合理的. 峰值时间  $q_i$  可以按下式方便地计算:

$$q_i = x_i + 2b_i, \quad i = 1, 2, \dots, n. \quad (9)$$

##### 2) 窗口技术.

将计划期划分为一系列相互重叠的时间片段,

称为窗口技术. 设窗口的宽度为  $W$ , 计划期划分为  $T - W + 1$  个窗口, 即:

$$[1, W], [2, W+1], \dots, [k, W+k-1], \dots, [T-W+1, T].$$

##### 3) 学习规则.

由于收益和资源约束函数都是单调增的, 可行性和不可行性能够向相邻的窗口传递. 当抗体定义为窗口中的峰值数时, 这种相邻抗体间的可行性信息的传递就称为学习(其他类型的函数也可以根据具体特点确定学习规则).

对于窗口  $k, k = 1, 2, \dots, T - W + 1$ .

**规则 1** 如峰值数  $p$  对收益约束在窗口  $k$  可行, 则在窗口 1 到  $k - 1$  都可行.

**规则 2** 如峰值数  $p$  对收益约束在窗口  $k$  不可行, 则在窗口  $k + 1$  到  $T - W + 1$  都不可行.

**规则 3** 如峰值数  $p$  对资源约束在窗口  $k$  可行, 则在窗口  $k + 1$  到  $T - W + 1$  都可行.

**规则 4** 如峰值数  $p$  对资源约束在窗口  $k$  不可行, 则在窗口 1 到  $k - 1$  都不可行.

##### 4) 学习过程.

抗体的学习过程表达为步骤 LP.

##### 步骤 LP

步 1 给定收益与资源约束的满足率的可接受水平  $\alpha$  和  $\beta$ .

步 2 随机产生  $N$  个抗原(染色体),  $N$  是一个大的正整数.

步 3 对抗原  $j(j = 1, 2, \dots, N)$  和窗口  $k(k = 1, 2, \dots, T - W + 1)$ , 数得其峰数为  $p$ ;

子步 3.1 检查收益约束可行性.

如可行, 令峰数  $p$  在窗口  $k$  的收益可行数  $FP(k, p) = FP(k, p) + 1$ , 总的收益测试数  $TP(k, p) = TP(k, p) + 1$ . 由规则 1, 学习得到

$$FP(l, p) = FP(l, p) + 1,$$

$$TP(l, p) = TP(l, p) + 1, \quad l = 1, 2, \dots, k - 1.$$

如不可行, 则令  $TP(k, p) = TP(k, p) + 1$ , 并由规则 2 得到

$$TP(l, p) = TP(l, p) + 1,$$

$$l = k + 1, \dots, T - W + 1.$$

子步 3.2 检查资源约束可行性.

如可行, 令峰数  $p$  在窗口  $k$  的资源可行数  $FR(k, p) = FR(k, p) + 1$ , 总的资源测试数  $TR(k, p) = TR(k, p) + 1$ . 由规则 3, 学习得到

$$FR(l, p) = FR(l, p) + 1, \quad TR(l, p) = TR(l, p) + 1,$$

$l = k + 1, K = 2, \dots, T - W + 1$ . 如不可行, 则令  $TR(k, p) = TR(k, p) + 1$ , 并由规

则4得到

$$TP(l, p) = TP(l, p) + 1, l = 1, 2, \dots, k - 1.$$

步4 对窗口  $k(k = 1, 2, \dots, T - W + 1)$  和峰数  $p(p = 0, 1, \dots, n)$ , 计算收益和资源约束的满足率:

$$\mu_P(k, p) = FP(k, p) / TP(k, p)$$

和  $\mu_R(k, p) = FR(k, p) / TR(k, p)$ .

步5 对窗口  $k(k = 1, 2, \dots, T - W + 1)$ , 按下式计算峰数的上下限阈值:

$$N^{\text{up}}(k) = \min\{l: \mu_P(l, p) > \alpha\}$$

和  $N^{\text{low}}(k) = \max\{l: \mu_R(l, p) > \beta\}$ .

所有窗口的峰数的上下限阈值就构成了可行与非可行的识别器. 应该说明的是, 以上识别器能够剔出大量不可行解, 但不是全部. 这样, 对本问题可行性检查仍是需要的. 显然, 由于大部分非可行解也被找出, 余下的可行性检查变得较为容易. 对于免疫系统检查不可行的染色体, 可以在编码空间中简单地将一些产品从多余的窗口中移到短缺的窗口中来修复.

#### 5) 可行性检查.

虽然不能检查无限多个约束, 但只要投入产品收益和资源的函数的极小值和极大值能分别满足收益和资源约束, 则可确定解的可行性.

定义两个约束函数的极值分别为:

$$\begin{aligned} \phi(x, t_1^*) &= \\ \min_i \left\{ \sum_{i=1}^n \text{sgn}(x_i) f(t - x_i) - P(t) + Q(t) \right\}, & (10) \\ \Psi(x, t_1^*) &= \\ \max_i \left\{ \sum_{i=1}^n \text{sgn}(x_i) w_i f(t - x_i) - R(t) + w_0 Q(t) \right\}. & (11) \end{aligned}$$

那么, 只要  $\phi(x, t_1^*) \geq 0$  和  $\Psi(x, t_2^*) \leq 0$  成立, 则解可行. 于是, 关键问题成了如何找出极值点  $t_1^*$  和  $t_2^*$ . 可以证明, 约束(6)的左边有不多于  $n$  个局优点可在  $n$  个区间  $[x_i, x_i + 2b_i]$  ( $i = 1, 2, \dots, n$ ) 中用线性搜索算法找到. 然后, 全局极小值  $\phi(x, t_1^*)$  即可从中选出. 同样的方法就可找出约束(7)的极大值  $\Psi(x, t_2^*)$ .

### 5 免疫遗传算法的基本步骤 (Procedure of immune-genetic algorithm)

为满足约束, 将惩罚项加到原来的目标函数(5)中, 成为:

$$\begin{aligned} Z(x) &= \sum_{i=1}^n \text{sgn}(x_i) \left[ \int_{x_i}^T f_i(t - x_i) dt - c_i \right] - \\ &M[\phi(x, t_1^*)^- + M[\Psi(t, t_2^*)]^+]. \end{aligned} \quad (12)$$

其中,  $[x]^-$  和  $[x]^+$  分别表示只取  $x$  的负值和正值.

令种群大小为  $NP$ , 个体 / 染色体记为:  $x(j), j = 1, 2, \dots, NP$ . 第  $j$  个染色体的适值为:

$$F(j) = Z_{\max} - Z(x(j)) + d, j = 1, 2, \dots, NP. \quad (13)$$

其中,  $d$  是一个随迭代变化的动态标定参数<sup>[8]</sup>.  $Z_{\max}$  是种群中目标函数(12)的极大值.

算法采用常用的两点交叉和换位变异两种遗传算子, 采用最常用的正比选择策略指定最大代数作停止准则<sup>[8]</sup>. 即, 选一个大的正整数  $NG$  为最大的代数, 若迭代指标  $k$  大于  $NG$ , 则停止迭代并输出结果. 基于以上讨论, 免疫遗传算法的基本步骤汇总如下:

#### 步骤 I-GA

步1 指定测试抗原个数  $N$ , 种群大小  $NP$ , 最大代数  $NG$ , 交叉概率  $p_c$  和变异概率  $p_m$ .

步2 随机产生  $N$  个抗原. 调步骤 LP 训练抗体, 并组成可行/非可行识别器.

步3 随机产生有  $NP$  个染色体的初始种群,  $x(j) = [x_1(j), x_2(j), \dots, x_n(j)]$ ,  $j = 1, 2, \dots, NP$ , 其中  $x_i(j)$  为 0 与  $T$  间的整数. 设迭代指标  $k = 0$ , 初始最优解  $x^* = x(1)$ , 最优目标值  $Z^* = BP$ , 其中,  $BP$  是一个大的正数.

步4 令  $k = k + 1$ , 如  $k > NG$  转步9. 否则执行步5至步8.

步5 对染色体  $x(j), j = 1, 2, \dots, NP$ , 执行如下3个子步:

子步5.1 用可行/非可行识别器(免疫系统)检查可行性, 如不可行用移动策略修复.

子步5.2 用线性搜索计算  $\phi(x, t_1^*)$  和  $\Psi(x, t_2^*)$ , 及扩展的目标函数值  $Z(j)$ .

子步5.3 找出

$$Z_{\max} = \max\{Z(j), j = 1, 2, \dots, NP\},$$

$$Z_{\min} = \min\{Z(j), j = 1, 2, \dots, NP\},$$

令  $j^* = \arg\{Z(j) = Z_{\min}\}$ ,  $x(j^*)$  即为达到  $Z_{\min}$  的染色体.

步6 若  $Z_{\min} < Z^*$ , 令  $Z^* = Z_{\min}$  和  $x^* = x(j^*)$ .

步7 对  $j = 1, 2, \dots, NP$ , 计算  $x(j)$  的适值, 并按正比选择策略计算选择概率.

步8 选择  $NP_c = p_c \times NP$  个染色体作两点交叉,  $NP_m = p_m \times NP$  个染色体作换位变异, 并另选  $NP - NP_c - NP_m$  个染色体直接进入下一代. 然后转步4.

步9 输出  $Z^*$  和  $x^*$  为取得的最好解.

这里, 步2调步骤 LP 训练抗体. 步4至步8为

遗传算法的主干.

### 6 数值分析(Numerical analysis)

本例是根据某电机厂产品投入计划问题改编的.设有 20 个新产品考虑在未来 16 个季度(4 年)中投入.产品的具体参数略.

窗口宽度、抗原个数、费用和资源约束的可接受水平为:  $W = 3, N = 1000, \alpha = 0.025$  和  $\beta = 0.025$ . 通过训练,得到所有窗口的峰数的阈值如表 1 所示.

表 1 各窗口峰数的阈值

Table 1 The thresholds of peak numbers in windows

窗口序号	1	2	3	4	5	6	7
时间段	1-3	2-4	3-5	4-6	5-7	6-8	7-9
上限	0	1	1	1	1	1	2
下限	0	0	0	0	0	0	0
$x^*$ 中的峰数	0	1	1	1	1	1	2
窗口序号	8	9	10	11	12	13	14
时间段	8-10	9-11	10-12	11-13	12-14	13-15	14-16
上限	2	3	3	4	4	5	5
下限	0	0	0	0	0	0	1
$x^*$ 中的峰数	2	1	1	4	3	2	3

遗传算法的参数设定为:  $NP = 100, NG = 100, p_c = 0.45, p_m = 0.45$ . 取得的结果见表 2. 最好解的目标值为 6754.90 百万元,运行时间为 112.51 秒.各窗口中的峰数见表 1 的最后一行.由以上结果可见,资源消耗较大的产品 9, 19 和 20, 以及产品 4 落选.限于问题的复杂性,尚不能确认上述解是最优的,但决策人对此解感到满意.

表 3 I-GA 和 GA 解不同规模问题的性能比较

Table 3 The performances comparison for the problems with different sizes

规模 $n \times T$	免疫遗传算法(I-GA)			一般遗传算法(GA)		
	运行时间 (秒)	最高可行 百分比	最好解 的目标值	运行时间 (秒)	最高可行 百分比	最好解的 目标值
10×10	32.85	82%	3164.93	40.37	11%	2786.93
15×12	44.71	82%	4301.96	73.44	5%	3765.08
20×16	127.43	73%	6612.90	156.71	3%	5635.53
25×18	208.71	76%	8597.59	211.36	3%	8339.99
30×20	324.94	77%	10398.79	364.76	2%	9881.39
40×24	700.96	73%	15852.98	667.24	2%	15075.78

由表 3 可见,传统 GA 的可行染色体的百分比随问题规模增大下降很快,而 I-GA 的下降却不明显.此外,I-GA 取得的解通常好于 GA. 并且多数情况下 I-GA 的运行少于 GA,原因是 GA 要花比免疫检查更多的时间来处理不可行的染色体.

表 2 免疫遗传算法取得的最好解

Table 2 The best solution of immune-genetic algorithm

$i$	1	2	3	4	5	6	7	8	9	10
$x_i$	8	15	6	0	14	13	15	11	0	6
峰值 时间	9.6	16.5	8.0	—	16.5	14.3	17.6	12.5	—	7.2
$i$	11	12	13	14	15	16	17	18	19	20
$x_i$	4	10	15	13	1	11	10	14	0	0
峰值 时间	5.8	11.2	16.8	14.6	3.5	12.3	12.6	15.5	—	—

为测试免疫遗传算法的性能,在完全相同条件下将其与不含免疫系统的遗传算法作比较.可行染色体的百分比随迭代的变化情况见图 3.

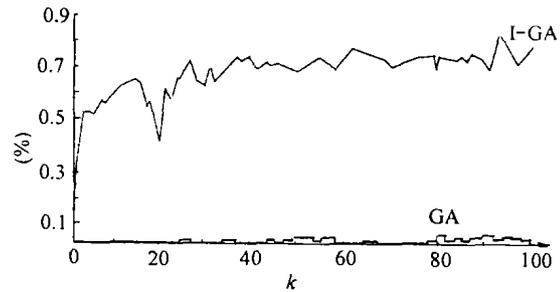


图 3 I-GA 和 GA 可行染色体的百分比的对比  
Fig. 3 The feasible rates comparison of I-GA and GA

由图 3 可见,GA 可行染色体的百分比低于 6%,且不少迭代找不到可行解.而 I-GA 的可行染色体的百分比通常高于 65%,最高达到 89%.因此 I-GA 获得的有效的染色体数远多于 GA,这样,它肯定能找到好于 GA 的解.I-GA 和 GA 解不同规模问题的性能对比见表 3.

### 7 结论(Conclusions)

根据免疫遗传算法的研究可以获得如下几点结论:

1) 推荐的免疫遗传算法将免疫系统的可行性识别能力和遗传算法的进化选择能力结合起来,从

而形成了求解复杂约束优化问题的能力.

2) 模型(5)~(8)为新产品投入计划问题提供了一种量化的描述方法.

3) 虽然 I-GA 能获得比 GA 高的可行百分比和更好的解,但仍不能保证解的最优性.

### 参考文献(References)

- [1] Holland J H. *Adaptation in Natural and Artificial Systems* [M]. Ann Arbor, Michigan, USA: University of Michigan, 1975
- [2] Gen M, Cheng R. *Genetic Algorithms and Engineering Design* [M]. New York: John Wiley & Son Press, 1996
- [3] Michalewicz Z. A survey of constraint handling techniques in evolutionary computation methods [A]. In J. McDonnell, et al (Ed). *Evolutionary Programming IV*, Combridge [M]. MA: MIT Press, 1995, 135 - 155
- [4] Jerne N K. The immune system [J]. *Scientific American*, 1973, 229 (1): 52 - 60
- [5] Farmer J D, Packard N H. The immune system, adaptation, and machine learning [J]. *Physica*, 1986, 22(2): 187 - 204
- [6] Ishida Y, Adachi N. Active noise control by an immune algorithm: adaptation in immune system as an evolution [A]. *Proc. of 1996 IEEE Int. Conference on Evolutionary Computation* [C]. Nagoya, Japan, 1996, 150 - 154
- [7] Ishiguro A, Watanabe Y, Uchikawa Y. Fault diagnosis of plant systems using immune networks [A]. *Proc. of 1994 IEEE Int. Confer-*

- ence on Multisensor Fusion and Intelligent Systems* [C]. Las Vegas, USA, 1994, 34 - 50
- [8] Joshi R R. Immune network memory: an inventory approaches [J]. *Computer and Operations Research*, 1995, 22(6): 575 - 591
- [9] Chun J S, Kim M K, Jung H K, et al. Shape optimization of electromagnetic devices using immune algorithm [J]. *IEEE Trans. on Magnetics*, 1997, 33(2): 1876 - 1899
- [10] Datar S, Jordan C, Kekre S, et al. New product development structures and time-to-market [J]. *Management Science*, 1997, 43(3): 452 - 464
- [11] Fang S C, Wu S Y. An inexact approach to solving linear semi-infinite programming problems [J]. *Optimization*, 1994, 28(8): 291 - 299
- [12] Wang D, Fang S C. A semi-infinite programming model for earliness/tardiness production planning with a genetic algorithm [J]. *Computers & Mathematics with Applications*, 1996, 31(8): 95 - 106

### 本文作者简介

汪定伟 1948年生.分别于华中科技大学和东北大学获硕士和博士学位.东北大学教授,博士生导师,现为中国科学院沈阳自动化研究所客座研究员.研究领域包括:建模与优化,软计算方法,生产计划与调度方法等. Email: dwwang@mail.neu.edu.cn

于海斌 1964年生.东北大学获硕士和博士学位.中国科学院沈阳自动化研究所研究员,博士生导师.研究领域包括:智能生产调度,分布式控制系统,离散事件动态系统等.