RLGA: 一种基于强化学习机制的遗传算法

王本年[1,2], 高 阳[1], 陈兆乾[1], 谢俊元[1], 陈世福[1]
(1. 南京大学计算机软件新技术国家重点实验室, 江苏南京 210093; 2 铜陵学院计算机科学与技术系, 安徽铜陵 244000)

摘 要: 分析了强化学习与遗传算法工作机制, 在提出基因空间分割概念的基础上, 提出了一种将强化学习与遗传算法内在结合起来的算法 RLGA, 在遗传算法的框架下实现强化学习机制. 从理论上分析了 RLGA的收敛性, 讨论了 RLGA的时间和空间效率及其与基因空间分割的关系, 通过实验分析了 RLGA中基因空间分割的指导范围. 实验结果表明, RLGA具有良好的全局收敛性能.

关键词: 强化学习; 遗传算法; 收敛性
中图分类号: TP301.6 文献标识码: A 文章编号: 0372-2112 (2006) 05-0856-05

RLGA: A Reinforcement Learning Based Genetic Algorithm

WANG Ben-nian[1,2], GAO Yang[1], CHEN Zhao-qian[1], XIE Jun-yuan[1], CHEN Shi-fu[1]
(1. National Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu 210093, China;
2. Department of Computer Science and Technology, Tongling College, Tongling, Anhui 244000, China)

Abstract RLGA, an algorithm which implements mechanism of reinforcement learning under the framework of genetic algorithm is described by using gene space division the algorithm maps the gene space of genetic algorithm into the strategy spcaces of multi-agent The convergence theorems for the algorithm are presented, and the time and the space efficiency of the algorithm as well as the relation between them and the division granularity are discussed The experimental results show that RLGA has well global convergence performance; and the further experiments provide the guide range of the size of gene space division in RLGA.

Key words reinforcement learning genetic algorithm; convergence

1 引言

遗传算法 (Genetic Algorithm, 简称 GA)是一种模仿生物进化过程的全局优化随机搜索算法, 它通过对问题的可行解进行编码, 然后利用自然选择和交叉变异操作来进行搜索. 它的简单通用性和高度鲁棒性的特点, 使得它在许多领域都得到了广泛的应用. 一般认为, GA的理论基础是模式定理和马尔可夫链模型[1]. 模式定理从结构模式的角度考察了 GA的工作机理, 认为 GA实际上是在寻找某种结构上的相似性, 它保证了这种相似的较优的模式的样本在进化过程中呈指数级增长, 但它并不能保证找到最优解, 即这些较优的模式不一定与最优解的模式相似. 而 GA的马尔可夫链模型则从搜索行为的角度刻画了 GA的行为属性. 对 GA的理论研究表明, GA的搜索具有良好的方向性, 对 GA作某些调整可以使 GA具有某种良好的行为属性, 但是, GA的本身难以体现出明晰的目标性. 在 GA中, 选择, 交叉和变异三个算子及其相关参数体现了开采 (exploit)与探索 (explore)之间的平衡, 选择的作用是开采搜索空间以便充分利用当前群体的已有信息, 而交叉和变异则是探索搜索空间以便找寻那些可能最优的区域. GA的这种特性使人很容易联想起强化学习 (Reinforcement Learning, 简称 RL)的有关技术[2].

强化学习是一种目标驱动的自适应能力很强的机器学习技术. 一个强化学习 Agent, 在与环境交互时, 能根据环境的反馈来调整自己的行动策略, 即如果 Agent的某个行动策略导致了环境正的奖赏, 则以后产生这个行动策略的趋势便会加强, 反之, 如果 Agent的某个行动策略导致了环境负的奖赏, 则以后产生这个行动策略的趋势便会减弱. 顺序型强化学习也可用马尔可夫链建模. 在强化学习的有关算法中, Agent所采用的 ε-greedy行动选择策略, 即以 1-ε的概率选择到目前为止最好的行动策略, 以 ε的概率选择行动策略空间中的任一行动策略, 也同样体现了对

收稿日期: 2005-01-28; 修回日期: 2006-03-20
基金项目: 国家自然科学基金 (No. 60475026); 江苏省自然科学基金 (No. BK2004079); 安徽省教育厅自然科学研究重点项目 (No. 2006KJ027A)

行动策略空间的开采与探索之间的平衡.

鉴于 GA 与 RL的相似性和各自的优良特性, 自然会引出一个很有意义的问题: 即在 GA 和 RL之间是否存在一种内在的联系, 能否将它们结合起来, 从而进一步地改善它们的搜索性能? 本文对该问题进行了研究, 首先提出了基因空间分割的概念, 然后在此基础上提出了一种基于强化学习机制的遗传算法 (Genetic algorithm based on reinforcement learning 简称 RLGA), 并对它的收敛性及有关参数对算法性能的影响进行了理论分析. 实验结果表明, RLGA 具有良好的全局收敛性能.

其实, RL与 GA的结合自上世纪 80年代后期就已受到了许多研究者的关注, 已有许多这方面的研究成果. 根据 RL与 GA 结合的紧密程度可以将它们分为三种情形 (如图 1所示): 第一种是协作型, 即 RL与 GA 为共同的目标分工协作. 如: Holland提出的学习分类器系统 (LCS)[3], 该系统运用 RL来对分类器的信用度进行分配, 用 GA 来产生潜在有用的分类器并淘汰无用的分类器, 在此基础上, Dorigo 等[4]进一步提出了 Q-LCS, 他们把 Q学习运用到了 LCS 中. 而 Q 等[5]则把 RL和 GA 结合进了多 Agent系统中, 用 RL进行组成员的交互策略学习, 用 GA 来进行组的进化. RL与 GA 结合的第二种情形是主从依赖型, 即以一方为主, 另一方为辅的形式, 主方利用从方来动态修改主方运行过程中的一些参数. 如: Eriksson等[6]提出的用 GA 来进化 RL的学习步长参数的思想, Pettinger等[7]则提出用 RL来自适应地控制 GA的遗传操作算子. RL与 GA 结合的第三种情形是融合型, 它是一种深度的、内在的和本质上的结合, 本文研究的 RLGA 正是属于这种情形.

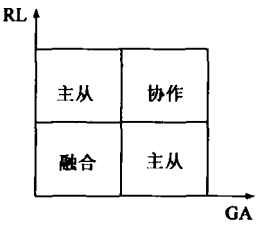


图 1　RL与 GA 的结合方式

2　强化学习

Agent强化学习的基本框架[2]如图 2所示. 在时间步 t 时刻, $t=0, 1, 2, \cdots$, Agent获得当前的环境状态 $s_t \in S$, S 为可能的环境状态集, 在此基础上, 选择一个行动 $a_i \in A(s_t)$作用于环境, $A(s_t)$是相应于状态 s_t的 Agent的可用的行动集, Agent行动的结果是在 $t+1$ 时间步, Agent收到环境的奖赏 $r_{t+1} \in R$, 同时环境变迁到一个新的状态. Agent强化学习的目的就是要获得一个行动选择策略 $p: s_t \to A(s_t)$, 即在每个时间步 t里, Agent都要实现一个从状态 s_t到可能的动作选择的概率映射, 从而使 Agent所选择的动作能够获得最大的环境奖赏.

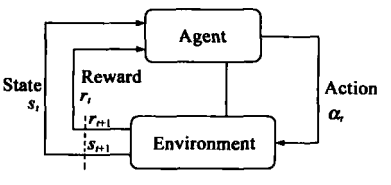


图 2　强化学习框架

在各种强化学习算法中, Q学习[8]是一种最具代表性的模型无关强化学习算法. Q学习的算法过程如下:

(1)初始化 Q 值;

(2)在状态 s_t下, 采用 ε-greedy策略确定行动 a_t, 得到经验知识和训练例 $\langle s_t, a_t, s_{t+1}, r_{t+1} \rangle$;

(3)用下列迭代公式更新 Q 值:

$$Q(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_a Q(s_{t+1}, a))$$

式中 α为学习速率, γ为折扣因子;

(4)若满足终止条件, 学习结束, 否则, 令 $t = t+1$, 转第 (2)步.

Watkins等[8]利用随机过程和不动点理论, 证明了当 α 满足一定条件时 Q学习过程的收敛性.

3　基于强化学习机制的遗传算法 RLGA

3.1　RLGA基本思想

我们先来考察一下 GA的工作原理. GA 首先对问题的可行解进行编码, 即将可行解表示为染色体, 这里我们不妨假定采用定长二进制编码, 记编码后形成的基因空间为 X_g, $X_g = \{0, 1\}^L$, L 为 X_g 中染色体的编码长度. 然后通过对染色体执行交叉和变异来搜索 X_g 中的未知区域. 为了保持对搜索空间 X_g 的一定搜索率和加速搜索收敛速度, GA需要维持一个较大的群体规模, 并要维持群体中个体的多样性, 通过基于适应度的选择来指导 GA 向较优的区域进行搜索. 然而, GA 对好的模式的搜索是间接的, 它隐含地体现在群体的样本中.

接下来, 我们来考虑如何用 RL来处理 GA 问题. 如果我们只是简单地将 GA问题的基因空间 X_g 看作是 RL的行动策略空间, $x \in X_g$ 的适应度看作是环境对行动 x 的奖赏, 则一个 GA 问题很容易地转换成一个 RL问题. 然而, 这种方法一般是不可取的, 因为如果 L 较大的话, 则我们使用 Q学习就需要维持一个非常庞大的 Q 表, 且由于行动空间的巨大, 使得每一个行动不可能被访问足够多次. 为了解决这样的问题, 我们需要对基因空间 X_g 进行适当的分割. 下面我们给出基因空间分割的形式定义.

定义 1(基因空间分割)　令 $D = (d_1, d_2, \cdots, d_n)$, $d_i > 0$, $i = 1, 2, \cdots, n$, $\sum_{i=1}^{n} d_i = L$. 对于 $\forall x \in X_g$, 我们将 x按 D给定的长度规则进行分段, 因而形成关于基因空间的一个分割, 记为 $A = (A_1, A_2, \cdots, A_n)$, 其中 $A_i = \{0, 1\}^{d_i}$, $i = 1, 2, \cdots, n$. 我们称 $D = (d_1, d_2, \cdots, d_n)$为 X_g 的一个分割模式, n为分割度.

现在, 我们来讨论 RL与 GA 的结合机制: 给定 X_g 的一个分割 A, 相应地, 创建 n个 Agent, 并且将 A_i作为 Agent$_i$ 的行动策略空间. 这样 n个 Agent的一次并发行动 $\vec{a} = (a_1, a_2, \cdots, a_n)$, 则构成了对基因空间 X_g 的一次搜索. 记 $\vec{a}$对应的染色体为 $x(\vec{a})$. 把 $x(\vec{a})$所对应的适应度 f进行适当的定标作为 $\vec{a}$中各行动策略的联合奖赏. 这样, 在 GA 的框架下, 通过

重复 n 个 Agent的并发行动, 就构成了一个重复一阶段对策(Repeated one stage game)的 RL问题.

3.2 RLGA算法描述

根据上述 RLGA 的基本思想, 我们给出 RLGA 的伪码表示如图 3所示.

```
step 1   Initialize D = (d_1, d_2, ..., d_n);
step 2   for all i = 1, ..., n, j = 1, ..., 2^{d_i}
           Q_i(a_{ij}) = 0
step 3   k = 1;
step 4   Repeat
step 5     Choose ā = (a_{1 j_k}, a_{2 j_k}, ..., a_{n j_k}) using ε-greedy;
step 6     r = fitness(x(ā)); //计算 x 的适应度
step 7     for all i = 1, ..., n
           Q_i(a_{i j_k}) = (1-α_{ik}) Q_i(a_{i j_k}) + α_{ik}(r + γ max_j(Q_i(a_{ij})));
step 8     if best_r > r then
             best_r = r, best_x = x; //精英保留
step 9     k = k + 1;
step 10  Until terminal condition is satisfied
step 11  Output best_x;
```

图 3 RLGA算法

由图 3可以看出: (1) GA 实际上为 RL提供了一个可运行的框架和环境, RL则替代了 GA 的自然选择与遗传进化操作; (2) GA 的一些标准参数为 RL的一些标准参数及分割模式 D 所取代; (3) RLGA 中采用了精英保留策略. 我们把 RL与 GA 的运行机制再作进一步的比较, 就会发现: (1) RL发现好的行动策略对应着 GA 发现好的结构模式; (2) RL的动作选择对应着 GA 的选择算子; (3) RL中多 Agent并发行动选择对应着 GA 中的 (多点)交叉; (4) RL中的 ε行动选择对应着 GA中的(多点)变异. 因此, 在 RLGA 中, 虽然更多地采取了 RL机制, 但它却仍然保留了 GA 的本质内涵. 从这里可以看出 RLGA实现了 RL与 GA的深度融合. 融合后的 RLGA 不仅保留了 GA 的简单通用性和鲁棒性的特点以及 RL的自适应性和目标驱动性的特点, 而且较 GA 具有更好的可理解性.

3.3 RLGA算法分析

图 3所示的 RLGA 算法我们把它叫做精英保留 RLGA 算法, 如果我们去掉它的第 8步, 并把第 11步改为:

Choose $\vec{a} = (a_{1,j_k}, a_{2,j_k}, \cdots, a_{n,j_k})$ using greedy policy;

Output $x(\vec{a})$;

则它就变为非精英保留 RLGA 算法, 它们的区别仅在于在算法的运行过程中是否保留了精英个体.

下面我们来讨论 RLGA 算法的收敛性. 记 $k^m(a_{ij})$为行动 a_{ij}第 m 次被试的索引.

定理 1(非精英保留 RLGA 算法的收敛性) 如果奖赏 r有界, 学习速率 $0 \leqslant \alpha_{ik} < 1$, 且

$$\sum_{m=1}^{\infty} \alpha_{ik^m(a_{ij})} = \infty, \text{且} \sum_{m=1}^{\infty} \alpha^2_{ik^m(a_{ij})} < \infty, \text{对} \forall i, j$$

则非精英保留 RLGA 算法以概率 1-ε收敛于最优解.

证明: 不考虑算法中的 i的具体含义, 仅把它看作环境状态变量, 由定理的条件可知, Watkins等[18]的 Q-学习过程收敛性定理的条件能被满足, 因此, 当 $k \to \infty$时, $Q_i(a_{ij})$以概率 1-ε趋于最优 $Q_i^*(a_{ij})$, 取 $a_i^* = \arg\max_{a_{ij}}(Q_i^*(a_{ij}))$, 由分割模式和奖赏 r的定义可知, $x(a_1^*, \cdots, a_n^*)$为最优个体, 因此, 非精英保留 RLGA 算法以概率 1-ε收敛于最优解. 证毕.

定理 2(精英保留 RLGA算法的收敛性) 在与定理 1相同的条件下, 精英保留 RLGA算法以概率 1收敛于最优解.

证明: 考虑到精英保留, 由定理 1立即可得. 证毕.

现在, 我们来讨论 RLGA 中的参数对 RLGA 性能的影响. 在 RLGA 中, 能直接影响 RLGA 的全局收敛性和收敛速度的因素主要有两个, 一个是参数 ε和 α_{ik}, 另一个是基因空间分割模式 D. ε是行动探索概率, 由 RL的理论可知, ε体现了 Agent行动探索(explore)和开发(exploit)之间的平衡, ε大则倾向于探索, ε小则倾向于开发, 因而, 我们可以在 RLGA 的运行初期通过增大 ε来提高 Agent对 X_g 的全局搜索力度, 积累行动策略的经验知识, 而在 RLGA 的运行后期, 通过减小 ε来充分发掘行动策略的经验知识, 提高 RLGA 的收敛性能. α_{ik}是学习速率, 它的大小可以显式地控制 RLGA 的收敛速度. 基因空间分割模式 D 则直接决定着基因空间的分割粒度, 进而反应出不同的结构搜索模式, 从而影响 RLGA 的搜索结果; 基因空间分割模式 D 也直接决定着 Agent行动策略空间的大小, 从而影响算法的空间和时间效率.

在 RLGA 中, RLGA 的空间效率主要体现在 Q-表所占用的空间的大小上. 定义 $d_{\max} = \max(d_1, d_2, \cdots, d_n)$, 则有:

$$2L \leqslant \sum_{i=1}^{n} 2^{d_i} \leqslant n 2^{d_{\max}} \leqslant L^2$$

由上式可以看出, 如果 $d_{\max}$越小, 则 RLGA 所占用的空间就越少. 它的两个极端情况是 $d_{\max} = 1$和 $d_{\max} = L$, 与此相对应, RLGA 占用的空间分别约为 $2L$和 L^2. 考虑穷尽搜索法, 若各个 Agent以 ε概率执行行动探索, 则搜遍整个基因空间 X_g 所需的时间为 $O\left(\prod_{i=1}^{n} 2^{d_i}/\varepsilon\right) = O(L^2/\varepsilon^n)$, 即分割度越小, 所需的运行时间就越短, 当 $n = 1$时, 运行时间最短, 为 $O(L^2/\varepsilon)$. 然而, RLGA 不是穷尽搜索法, 它是一种目标驱动的启发式搜索方法. 由 RL的特性可知, 若行动空间越小, 则探索越充分, 因而 d_i的大小与运行时间呈现出 $2^{\Delta d_i}$倍关系, 即若 d_i减少 $|\Delta d_i|$, 则运行时间减少 $2^{|\Delta d_i|}$倍, 所以, 分割模式 D 越细越好. 这就要求 RLGA 在分割模式 D 上有一个折中.

4 实验结果与分析

4.1 实验方法

为了验证 RLGA 算法的有效性, 我们选用 6个常用的

函数优化测试函数的例子来进行测试. 这 6个测试函数见表 1所示. 其中: 函数 f_2 中的 integer(·)是取下整函数, 函数 f_3 中的 $N(0, 1)$表示一个满足均值为 0方差为 1的正态分布随机数. 上述所有函数均是求全局极小值.

我们在这 6个测试函数上分别运行 RLGA 和 SGA, 将它们的运行结果进行比较. SGA 的运行步骤如下:

(1)指定种群规模 N, 交叉概率 Pc, 变异概率 Pm, 最大运行代数 T, 随机确定初始种群;

(2)依据个体适应度, 用赌轮选择法从当前种群中选择 $N/2$个个体对;

(3)依概率 Pc对上述 $N/2$个个体对执行单点交叉, 产生 N 个中间个体;

(4)依概率 Pm 对上述 N 个中间个体执行变异, 从而产生新一代种群;

(5)计算新一代种群个体的适应度, 保留精英个体;

(6)如果未达到最大运行代数, 转(2), 否则, 输出精英个体, 结束.

实验中 SGA 的有关参数设置为: 种群规模 $N = 100$, 交叉概率 $Pc = 0.8$, 变异概率 $Pm = 0.1$, 最大运行代数 $T = 100$. RLGA 的有关参数设置为: 贪心选择概率 $\varepsilon = 0.3$, 学习速率 $\alpha_{i,k} = 0.9$, $\gamma = 0.9$, 分割模式, $d_i = 6$, $i = 1, \cdots, n$, 为了使 RLGA 与 SGA 在总的搜索规模上保持一致, RLGA 的最大运行代数设置为 SGA 的运行代数 $T \times$ SGA 的种群规模 $N = 10000$.

4.2 实验结果及分析

RLGA 与 SGA 在表 1给定的 6个测试函数上的运行结果见表 2和图 4-9所示, 其中表 2列出的是相应算法获得的最小值和最小值出现的代数以及相应的变量 x 的取值, 图 4-9反映的是在 6个测试函数上相应算法所搜索到的最好值的改进过程, 图中的 x 轴都进行了规范化处理. 从表 2和图 4-9可以看出, 除在 f_2 上 RLGA 与 SGA 获得了同样的最小值外, 在其他测试函数上, RLGA 都较 SGA 的最终结果好, 而且, 如果考虑到搜索规模, 在除 f_3 之外的其他测试函数上, RLGA 获得最好值的时间都比 SGA 早, 而在 f_3 上, 从图 6可以看出, SGA 基本上处于一种停滞状态, 而 RLGA 的搜索结果则在不断地改进, 而且最后的结果更接近于真正的函数最优值. 因此, RLGA 具有良好的寻优和全局收敛性能. 对于图 4-9一个需要解释的现象是在它们的开始部分, SGA 基本上都是比 RLGA 表现得好, 其主要原因是因为 SGA 在开始点取的值是 100个初始个体中最好的一个, 而 RLGA 则是随机取的一个, 还有就是 RLGA 采用了固定的贪心选择概率 ε, 它在开始时太小, 因而减少了行动探测的机会.

表 1　测试函数

函数	变量定义域
$f_1(x) = \sum_{i=1}^{3} x_i^2,$	$5.12 \leqslant x_i \leqslant 5.12$
$f_2(x) = \sum_{i=1}^{5} \text{integer}(x_i),$	$5.12 \leqslant x_i \leqslant 5.12$
$f_3(x) = \sum_{i=1}^{30} ix_i^4 + N(0, 1),$	$1.28 \leqslant x_i \leqslant 1.28$
$f_4(x) = (x_1^2 + x_2^2)^{0.25}(\sin^2(50(x_1^2 + x_2^2)^{0.1}) + 1),$	$100 \leqslant x_i \leqslant 100$
$f_5(x) = 0.5 + \frac{\sin^2\sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	$100 \leqslant x_i \leqslant 100$
$f_6(x) = (4 - 2.1x_1^2 + x_1^4/3)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2,$	$3 \leqslant x_i \leqslant 3$

RLGA 之所以能够很好地工作, 除了它具有 RL和 GA 各自的优点外, 更重要的是 RLGA 不象 GA, 把较好的模式隐含地包含在群体的染色体样本中, 虽然 GA 在选择算子的作用下, 使它们能以较高的几率生存下来, 但却因为随机的交叉和变异而很容易遭到破坏, 而RLGA由于采用了

表 2　RLGA 与 SGA 在 6个测试函数上的运行结果比较

函数	SGA(种群规模 = 100)			RLGA		
	代数	最小值	x	代数	最小值	x
f_1	73	0.43383	(-0.33659, -0.54376, -0.1577)	1560	0.008892	(0.01502, -0.05505, -0.07507)
f_2	79	-28	(-5.0925, -5.0175, -4.5793, -4.2219, -5.0695)	6457	-28	(-5.12, -4.6324, -5.12, -4.6324, -5.12)
f_3	32	87.824	(-0.15455, 1.0457, 0.92004, -0.68245, -0.7273, -0.10332, -0.09543, -0.08136, 0.14626, 0.32347, 0.19664, -0.3601, 0.81046, -0.12326, -0.81669, 0.16676, -0.421, 0.055464, 0.49983, 0.13183, -0.35214, 0.11046, 0.90106, 0.16032, -0.84307, 0.23527, -0.38603, -0.10441, 0.31785, 1.069)	7713	25.456	(-0.18286, 0.18286, 0.91429, 0.18286, 0.18286, 0.54857, 0.18286, -0.91429, -0.54857, -0.54857, -0.18286, 0.18286, 0.18286, -0.54857, -0.54857, -0.18286, 0.54857, -0.18286, 0.54857, 0.54857, -0.54857, -0.18286, 0.18286, 0.18286, 0.54857, 0.18286, -0.18286, 0.18286, -0.18286, 0.54857)
f_4	61	2.7474e-5	(-38.371, -39.691)	5408	8.4397e-7	(-92.517, -17.527)
f_5	98	-0.95594	(3.1273, 0.21209)	6478	0.9921	(-0.009156, 0.088504)
f_6	56	-0.10495	(0.63841, -0.98902)	5017	-1.0268	(0.06473, -0.72796)

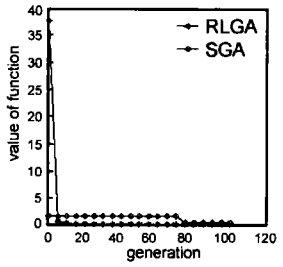


图4 RLGA与SGA在f_1上的结果

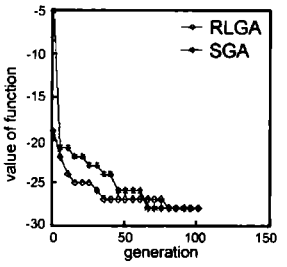


图5 RLGA与SGA在f_2上的结果

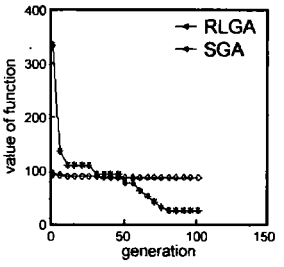


图6 RLGA与SGA在f_3上的结果

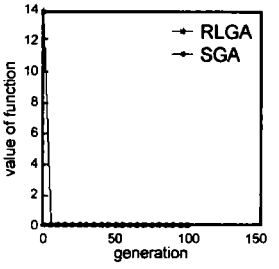


图7 RLGA与SGA在f_4上的结果

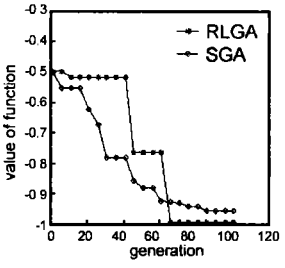


图8 RLGA与SGA在f_5上的结果

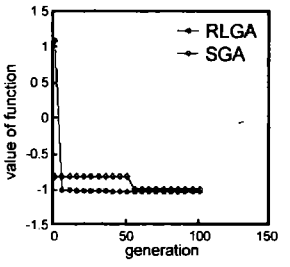


图9 RLGA与SGA在f_6上的结果

基因分割, 将染色体样本划分为很小的块, 从而使得较优的模式很容易得以保持.

4.3 分割模式 D 对 RLGA 性能的影响

在上述实验中, 我们仅仅只是简单地设置所有的 $d_i = 6$, 这样的选择是不是合适呢? 在第 3.3节关于 RLGA 的算法分析中, 我们已经分析了分割模式 D 对 RLGA 性能的影响. 为了验证我们的分析, 并对 d_i 的选择范围提供一个适当的指导, 我们分别在 f_2 和 f_5 上进行了进一步的实验. 在保持精度和最大运行代数不变的前提下, 分别取 $d_i = 1, 2, \cdots, 11$, 对每个 d_i 的取值运行 10次, 实验结果取 10次运行结果 (每次的最好值) 的平均值, 图 10和 11是它们的实验结果. 图 10和 11表明, 当 $1 \leqslant d_i \leqslant 7$时, RLGA 都能表现出很好的性能, 而当 $d_i > 7$时, RLGA 的性能则急骤下降. 这除了可以从行动空间的大小来解释外, 我们也可以从基因模式的角度来加以解释, 因为较小的 d_i 则对应着较短的基因模式块, 它使得当个体进化到一定的代数后, 那种较好且较短的基因模式块能得到很好的保持.

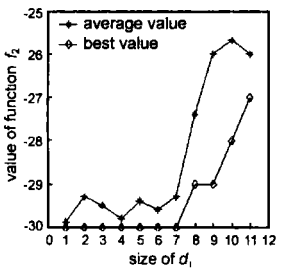


图 10 f_2上分割模式对RLGA的影响

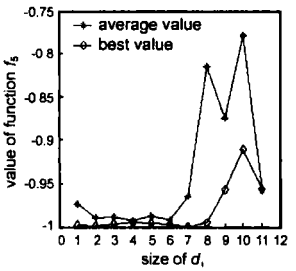


图 11 f_5上分割模式对RLGA的影响

5 总结

本文在深入分析和比较 RL与 GA 的基础上, 提出了一种基于强化学习机制的遗传算法: RLGA, 该算法通过基因空间分割模式 D 把 RL和 GA 内在地融合在一起, 从而使得 RLGA 既具有 GA 的简单通用性和鲁棒性, 又具有 RL的自适应性和目标驱动性. 此外, 本文还对 RLGA 的收敛性以及时间和空间效率进行了理论分析. 实验表明, RLGA 具有良好的全局收敛性能, 通过实验, 我们还获得了基因空间分割模式 D 的一个指导取值范围. 进一步地, 从结构模式的角度来考察 RLGA, 我们可以对基因空间作不同的分割, 使得 RLGA 在群体规模上运行, 执行并行搜索. 对于 RLGA在更弱的条件下的收敛性和基因空间分割模式 D 对 RLGA 的性能影响的严格理论分析, 则是以后需进一步研究的问题.

作者简介:

王本年 男, 1967年生于安徽, 南京大学计算机科学与技术系博士研究生, 铜陵学院计算机科学与技术系副教授, 主要研究方向为多Agent系统、机器学习和计算智能. E-mail: bnwang@mail.tl.ah163.net

高 阳 男, 1972年生, 南京大学计算机科学与技术系博士, 副教授, 主要研究方向为 Agent理论、强化学习、信念修正和网格计算.

陈兆乾 女, 1940年生, 南京大学计算机科学与技术系教授, 博士生导师, 主要研究领域为人工智能、知识工程、机器学习、神经网络和数据挖掘.

谢俊元 男, 1961年生, 南京大学计算机科学与技术系教授, 博士生导师, 主要研究方向为人工智能、计算机网络与信息安全.

陈世福 男, 1938年生, 南京大学计算机科学与技术系教授, 博士生导师, 主要研究方向为人工智能、机器学习和图像处理.

参考文献:

[1] Nix A E, Vose M D. Modeling genetic algorithms with Markov chains[J]. Annals Mathematics and Artificial Intelligence, 1992, 5(1): 77-88.

[2] Sutton R S, Butro A G. Reinforcement Learning: Introduction [M]. Cambridge, MA: MIT Press, 1998.

[3] Holland J H. A mathematical framework for studying learning in classifier systems[J]. Physica, 1986, 22D(1-3): 307-317.

[4] Dorigo M, Bersini H. A comparison of Q-learning and classifier systems[A]. Proceedings of From Animals to Animats,

(下转第 866页)

参考文献:

[1] 冈萨雷斯, 等. 数字图像处理 [M]. 北京: 电子工业出版社, 2004. 50-51.

[2] Y Jin, L Fayad, A F Lainel. Contrast enhancement by multi-scale histogram equalization[A]. Proceedings of SPIE[C]. Wavelets Applications in Signal and Image Processing IX, 2001. 206-213.

[3] Z Yu, C Bajaj. A fast and adaptive method for image contrast enhancement[A]. Proceedings of 2004 IEEE International Conference on Image Processing[C]. Singapore, 2004. 1001-1004.

[4] A Iranli, H Fatemi, M Pedram. HEBS: Histogram equalization for backlight scaling[A]. Proceedings of the 42nd Annual Conference on Design Automation[C]. New York: ACM Press, 2005. 346-351.

[5] Y T Kim. Contrast enhancement using brightness preserving bi-histogram equalization[J]. IEEE Transactions on Consumer Electronics, 1997, 43(1): 1-8.

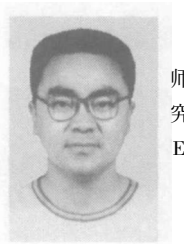
[6] S D Chen, A R Ramli. Contrast enhancement using recursive mean-separate histogram equalization for scalable brightness preservation[J]. IEEE Transactions on Consumer Electronics, 2003, 49(4): 1301-1309.

[7] S Pizer, et al. Adaptive histogram equalization and its variations[J]. Computer Vision Graphics & Image Processing, 1987, 39(3): 355-368.

[8] V Caselles, et al. Shape preserving local contrast enhancement[A]. Proceedings of the 1997 International Conference on Image Processing (ICIP'97)[C]. Washington: IEEE Computer Society, 1997. 314-317.

[9] J Y Kim, L S Kim, S H Hwang. An advanced contrast enhancement using partially overlapped sub-block histogram equalization[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2001, 11(4): 475-484.

作者简介:



江巨浪 男, 1967年生于安徽潜山县, 安庆师范学院副教授, 合肥工业大学博士研究生, 研究方向为计算机图形学与图像处理.
E-mail: jiangjulang@126.com

张佑生 男, 1941年生于湖南浏阳, 合肥工业大学教授, 博士生导师, 研究方向为计算机图形学、图像识别与理解、智能 CAD.

薛 峰 男, 1978年生于安徽舒城县, 合肥工业大学讲师, 博士研究生, 研究方向为计算机图形学.

胡 敏 女, 1967年生于安徽淮北市, 合肥工业大学副教授, 博士, 研究方向为图像处理、人工智能.

(上接第 860页)

Third International Conference on SIMULATION of Adaptive Behavior[C]. MA: MIT Press, 1994. 248-255.

[5] Qi D, Sun R. A multi-agent system integrating reinforcement learning, bidding and genetic algorithms[J]. Web Intelligence and Agent Systems, 2003, 1(3-4): 187-202.

[6] Eriksson A, Capi G, Doya K. Evolution of metaparamters in reinforcement learning algorithm[A]. Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems[C]. Piscataway NJ: IEEE Press, 2003. 412-417.

[7] Pettinger J E, Everson R M. Controlling genetic algorithms with reinforcement learning[A]. Proceedings of the Genetic and Evolutionary Computation Conference[C]. San Francisco, CA: Morgan Kaufmann, 2002. 692-692.

[8] Watkins C J C H, Dayan P. Technical note: Q-learning[J]. Machine Learning, 1992, 8(3-4): 279-292.