

文章编号: 1001-0920(2004)06-0683-04

一种快速压缩遗传算法及其仿真研究

李树刚, 吴智铭, 庞小红

(上海交通大学 自动化研究所, 上海 200030)

摘要: 给出了压缩遗传算法的模式定理以及收敛性和运算参数的分析, 并提出一种快速压缩遗传算法 (fcGA). 该算法用压缩遗传算法 (cGA) 运行少量代数得到的概率值及其运行代数组成一个观测样本, 借助于统计学中的最小二乘法估算几万代以后的概率值, 组成新的概率矩阵并根据该矩阵产生新的个体, 用这些新的个体更新概率矩阵. 旅行商问题 (TSP) 的仿真证明, 该算法是一种十分高效的遗传算法.

关键词: 压缩遗传算法; 最小二乘法; 旅行商问题

中图分类号: TP273 **文献标识码:** A

A kind of fast compact genetic algorithm and its simulation

LI Shu-gang, WU Zhi-ming, PANG Xiaohong

(Department of Automation, Shanghai Jiaotong University, Shanghai 200030, China. Correspondent: LI Shu-gang, E-mail: maxli@sjtu.edu.cn)

Abstract: The schema theorem of the cGA (compact genetic algorithm) and the analysis of cGA's convergence and the parameters are given. A kind of fast compact genetic algorithm (fcGA) is proposed. With the probability values that the cGA get in the beginning generations, the probability values in thousands of generations are estimated by the least square approach. The new probability matrix is composed, from which the new offspring is generated. Based on their fitness the probability matrix is updated. The simulations on the traveling salesman problem show that this algorithm is of high efficiency.

Key words: compact genetic algorithm; least square approach; traveling salesman problem

1 引言

在普通遗传算法 (GA) 中, 要获得满意的运算结果, 必须选择合适的交叉点、交叉长度、交叉概率、变异概率等, 并且需要较长的运算时间和较大的种群规模. 为克服 GA 的这些缺点, 人们提出了压缩遗传算法 (cGA)^[1,2]. 该算法的主要思想是根据概率矩阵来产生个体, 再根据产生的个体来更新这个概率矩阵. 本文给出了该算法的收敛性分析, 并给出了算法的模式定理及对算法参数的定性分析.

文献[3]用 GA 在有限几代的运算结果, 大致估算了满意解出现的代数. 受该文的启示, 本文提出一

种快速压缩遗传算法 (fcGA). 算法的基本思想是: 以 cGA 运算少量代数产生的概率值及其运行代数作为样本, 引入最小二乘法^[4]来估算几万代之后的概率值, 再用这些概率值产生新的个体, 这样便可加速 cGA 的收敛过程. 旅行商问题 (TSP) 的仿真表明, 快速压缩遗传算法具有很高的效率.

2 压缩遗传算法

本文以不对称旅行商问题为背景, 对压缩遗传算法进行仿真研究. 该算法的编码形式为 $\{c_{d_i}\}_{i=1}^N$, 其中 c_{d_i} 表示位置 d_i ($d_i \in \{1, 2, \dots, N\}$) 上城市的编

收稿日期: 2003-06-13; 修回日期: 2003-09-16.

基金项目: 国家自然科学基金资助项目 (59889505; 70071017).

作者简介: 李树刚 (1973—), 男, 黑龙江齐齐哈尔人, 博士生, 从事生产计划与调度、智能控制等研究; 吴智铭 (1936—), 男, 江苏苏州人, 教授, 博士生导师, 从事智能优化与调度等研究.

号. 目标是找到一个能走遍所有城市的最短路径, 并且各城市只访问一次.

对应一个 N -city 的旅行商问题 (TSP(N)) 的概率矩阵, 可用一个 $N \times N$ 的矩阵来表示. 矩阵的行和列分别表示城市 1 至 N , 矩阵中 i 行 j 列的元素 p_{ij}^k 表示在第 k 代产生基因块 $(c_i c_j)$ 的概率, 即在城市 i 之后访问城市 j 的概率.

算法步骤如下:

1) 概率矩阵的初始化: 初始时将矩阵对角线上的元素赋值为 0, 其他元素赋值为 0.5.

2) 新个体的产生规则: 在每一行中随机找一个元素 (对角线元素除外), 并以该元素的值作为选择概率进行选择. 如果没有选中, 则再找其他的元素, 直到选定一个为止. 如果找遍一行但没有一个元素被选中, 则在该行中随机选择一个元素. 将每一行中选中的元素作为一个基因块, 并将这些基因块组合起来组成新的个体. 在目标中引入惩罚函数来惩罚违反约束的个体.

3) 概率矩阵的更新: 压缩遗传算法中概率矩阵根据下式进行更新:

$$p_{ij}^{k+1} = \begin{cases} p_{ij}^k + \frac{1}{n}, & (c_i c_j) \in W(k), (c_i c_j) \notin L(k); \\ p_{ij}^k + \frac{1}{n}, & (c_i c_j) \in L(k), (c_i c_j) \notin W(k); \\ p_{ij}^k, & \text{otherwise.} \end{cases} \quad (1)$$

其中: n 是模拟种群的规模, $W(k)$ 和 $L(k)$ 分别是第 k 代产生的最好个体和最差个体.

如果 $p_{ij}^{k+1} < 0$, 则 $p_{ij}^{k+1} = 0$; 如果 $p_{ij}^{k+1} > 1$, 则 $p_{ij}^{k+1} = 1$.

3 压缩遗传算法的理论分析

算法运行时会导致 $p_{ij}^{k+1} = 0$, 使得算法不能达到某些状态. 由于算法运算的随机性, 不能保证这些不能到达的状态中不含有全局最优解, 也就不能保证算法收敛于全局最优解. 为此, 对算法进行如下改进:

- 1) 保留到当前代为止产生的最好个体;
- 2) 在概率矩阵更新中, 当 $p_{ij}^{k+1} < \partial$ 时, $p_{ij}^{k+1} = \partial$. 这里 ∂ 是一个很小的正数, 本文仿真中取 $\partial = 0.1$.

3.1 算法分析定理

定理 1 改进的压缩遗传算法收敛于全局最优解的概率为 1.

证明 压缩遗传算法的子代状态 (个体及其更

新的概率矩阵) 只与其父代状态有关, 而与以前各代无关, 因此可描述成一个有限的 Markov 链^[5]: $\{q(t), t \geq 0\}$, 这里 $q(t)$ 是算法运行到 t 代时的状态. 用改进的遗传算法构造新的 Markov 链: $q(t) = \{A(t), q(t)\}$, 其中 $A(t)$ 是到当前代为止产生的最好个体. 在改进的压缩遗传算法中, 概率矩阵的任何一个元素都大于零, 即算法可从一种状态向任意状态转变 (是可达的), 因此该 Markov 链是遍历的.

用 r_{hl} 表示算法从状态 h 转移到状态 l 的概率, 并设所有的状态空间 H 可分成两大类: 包含最优个体的状态集合 (用 H_n 表示) 和不包含最优个体的状态集合 (用 H_0 表示). 于是整个状态空间可表示成 $H = H_n \cup H_0$. 算法在 $t + 1$ 代处于 l 状态的概率为

$$R_l(t + 1) = \sum_{h \in H} R_h(t) r_{hl}, \quad t = 0, 1, \dots \quad (2)$$

式 (2) 可写成

$$R_l(t) = \sum_{h \in H} R_h(t) r_{hl}. \quad (3)$$

由上述描述可知, 子代个体的产生与时间起点无关, 因此 $R_l(t)$ 的稳态概率与其初始状态无关. 因为保存了到当前代为止产生的最好个体, 所以从状态 H_n 转移到状态 H_0 的概率为零. 由于改进的算法是可达的, 算法可以找到全局最优解, 也就是说, 稳态时状态 H 中一定含有状态 H_n , 即 $r_{hl} = 0, t \rightarrow \infty, \forall h \in H, \forall l \in H_0$. 因此 $R_l(t) = 0, l \in H_0$, 即算法收敛于不含有最优解的状态的概率为 0. 所以改进的遗传算法收敛于最优解的概率为 1.

定理 2 (压缩遗传算法的模式定理) 低阶的优良模式在后续代中会逐渐增加, 不良模式在后续代中会逐渐减少.

证明 在任意代 k , 考虑任意 $2y$ 阶模式 $S_h: * * c_1 c_2 * * c_3 c_4 * * \dots * c_{2y-1} c_{2y} * *$. 定义: 如果该模式中含有的任何确定的基因块 $(c_i c_j) \in W(k)$, 并且 $(c_i c_j) \notin L(k), \forall i, j$, 则该模式是优良模式; 如果 $(c_i c_j) \in L(k)$, 并且 $(c_i c_j) \notin W_k$, 则该模式是不良模式. 定义产生确定的基因块 $(c_1 c_2)$ 的事件为 A_1, \dots , 产生确定的基因块 $c_{2y-1} c_{2y}$ 的事件为 A_y . 则产生模式 S_h 的概率为

$$P(S_h) = P(A_1 A_2 \dots A_y). \quad (4)$$

式 (4) 可写成

$$P(S_h) = P(A_1) P(A_2) \dots P(A_y). \quad (5)$$

由算法运算过程可知, 产生各基因块的事件是相互独立的. 根据式 (1), 如果 S_h 是优良模式, 则每

个确定位置上基因块对应的概率值在后续代中会逐渐增加到 1, 这说明式(5)右端各项的概率值会逐渐增大, 导致 $P(S_h)$ 的值也逐渐增大, 保证了在后续代中 S_h 被选择的可能性越来越大, 因此它在后续代中会逐渐增加. 同理, 如果 S_h 是不良模式, 则 $P(S_h)$ 的值会逐渐减小, 在后续代中 S_h 被选择的概率越来越小, 因此它在后续代中会逐渐减少.

由于 $P(A_i) = 1, 1 \leq i \leq y$, 则 y 越小式(5)的值越大, 产生模式 S_h 的概率也越大, 因此它在后续代中会逐渐增加. 综上所述可得出结论: 低阶的优良模式在后续代中会逐渐增加, 不良模式在后续代中会逐渐减少.

3.2 对参数 s 和 n 的分析

通常, 选择压力 s (每一代产生的个体数目) 对算法的效率影响不大. 大量的仿真实验表明, 该值一般选在 $[2, 16]$ 之间.

由概率更新式(1)可以看出, 在每一代中得到的优良模式对应的概率会至少增加 $1/n$; 同理, 不良模式对应的概率会至少减少 $1/n$. 如果 n 选得很小, 优良模式所对应的概率会很快增加到 1, 不良模式所对应的概率会很快减少到 0. 这会使算法很快收敛于一个局部最优解而停滞. 相反, 如果 n 很大, 每一代概率矩阵变化很小, 又会降低压缩算法的收敛速度.

为了验证上述观点, 本文进行了仿真实验. 以 TSP(80) 为背景, 在区间 $[10, 40]$ 内随机产生一系列的随机数作为两城市间的距离. 设定 $s = 4, n = \{10, 50, 100, 500\}$, 仿真结果如图 1 所示. 从图 1 可以看出, 当 n 很小时 (例如 $n = 10$), cGA 在 5 000 代内很快收敛, 7 000 代以后停滞; 当 n 很大时 (例如 $n = 500$), cGA 缓慢地收敛. 这证实了上面提出的观点: 小的 n 能增加收敛速度并在局部最优解停滞, 大的 n 将会降低算法的收敛速度.

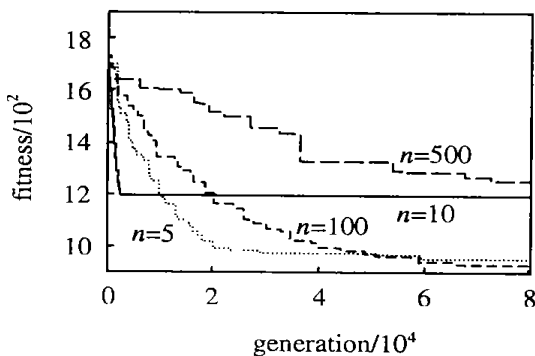


图 1 TSP 仿真结果

4 快速压缩遗传算法

矩阵中元素值随着代数的增加而变化, 可近似地把它看成运行代数的函数, 写作 $p_{ij}^t = f(t)$, 其中 t 是运行代数. 由于算法的随机性, 这些概率值与运行代数之间是一种不确定的关系, 很难确切知道它的表达式. 由式(1)可知, 随着代数的增加, 对应优良基因的概率值不断增大, 对应不良基因的概率值不断减少, 这表明概率值与运行代数在一定程度上是准线性关系. 因此可采用回归分析中的最小二乘法来估算若干代之后的概率值.

规定每隔一定代数, 选择数据对 (p_{ij}^k, k_l) 作为观测值. 其中: k_l 表示运行代数, p_{ij}^k 表示运行至 k_l 时的概率值. 用这些观测值组成一个样本, 估计几万代后的概率值. 最小二乘法可描述如下^[4]:

$$p_{ij}^x = 1 \cdot x + 0. \tag{6}$$

其中

$$1 = \frac{\sum_{l=1}^L (k_l - \bar{k}_l) (p_{ij}^k - \bar{p}_{ij})}{\sum_{l=1}^L (k_l - \bar{k}_l) (k_l - \bar{k}_l)},$$

$$0 = \bar{p}_{ij} - 1 \bar{k}_l.$$

式中

$$\bar{k}_l = \frac{1}{L} \sum_{l=1}^L k_l, \bar{p}_{ij} = \frac{1}{L} \sum_{l=1}^L p_{ij}^k,$$

L 是数据对 (p_{ij}^k, k_l) 的数量, x 是运行代数.

由这些估计值组成新的概率矩阵, 再由这个概率矩阵产生新的个体. 这样会省去几万代的选择、产生新个体、更新等操作, 极大地提高算法的效率.

快速压缩遗传算法描述如下:

Step1: 初始化概率矩阵;

Step2: 根据概率矩阵产生 s 个个体, s 为选择压力;

Step3: 计算这些个体的适应度值, 并根据式(1)更新概率矩阵;

Step4: 重复 Step2 和 Step3, 直到获得一定数量的数据对 (p_{ij}^k, k_l) , 并根据式(4)估算几万代之后矩阵的元素值, 用这些元素值组成新的概率矩阵;

Step5: 用上一步产生的概率矩阵来产生新的个体, 再根据式(1)更新这个概率矩阵;

Step6: 重复 Step5, 直至达到最大代数.

5 fcGA, GA 和 cGA 的仿真比较

利用第 3 节提供的 TSP(80) 问题进行仿真, 任务是找到适应度值不大于 1 000 的个体并计算所需

的最短时间. 在 fcGA 中, 每 10 代取一对观测数据 (p_{ij}^k, k_l) , 共得到 100 对这样的数据. 以这些数据作为样本, 用最小二乘法估计 50 000 代时的概率值. fcGA, cGA 和 GA^[6] (WALL 用 C++ 开发的关于 TSP 问题的 GA 程序库) 分别运行 10 次.

表 1 fcGA, cGA 和 GA 的仿真结果

n	S	fcGA		cGA		GA	
		Fitness	t/s	Fitness	t/s	Fitness	t/s
300	4	997	74	997	148	$C_r = 4$	400
	8	999	63	997	89	994	
	16	994	71	1 000	108		
500	4	994	100	997	124	$C_r = 8$	390
	8	974	105	993	150	970	
	16	998	124	970	180		
800	4	993	191	1 000	250	$C_r = 10$	401
	8	970	179	983	274	1 000	
	16	1 000	234	998	329		

表 1 中给出了执行一次所需的平均时间. 在 cGA 和 fcGA 中, 模拟种群规模 $n = \{300, 500, 800\}$, 选择压力 $s = \{4, 8, 16\}$; 在 GA 中, 种群规模 $n = \{300, 500, 800\}$, 交叉长度 $C_r = \{4, 8, 10\}$, 交叉概率 $P_c = 0.8$, 变异概率 $P_m = 0.2$.

从表 1 可以看出, 在参数选取相同的情况下, fcGA 所用的时间最少, 因此效率最高. 这说明引入

最小二乘法进行估算是有效的.

6 结 语

本文首先给出了压缩遗传算法的收敛性分析以及相关参数的分析, 并给出了该算法的模式定理; 然后提出一种快速压缩遗传算法, 即用算法在最初几代内得到的概率值及其对应的运行代数作为样本, 引入最小二乘方法对几万代以后的概率值进行估算, 并用估算值组成一个概率矩阵, 再用这个概率矩阵产生新的个体; 最后利用旅行商问题的仿真证明了该算法的有效性.

参考文献 (References):

- [1] Baraglia R, Hidalgo J I, Perego R. A hybrid heuristic for the traveling salesman problem[J]. *IEEE Trans on Evolutionary Computation*, 2001, 5(6): 613-622.
- [2] Harik G, Lobo F, Goldberg D. The compact genetic algorithm[J]. *IEEE Trans on Evolutionary Computation*, 1999, 3(4): 287-297.
- [3] Wu Zhiming, Zhao Chunwei. Genetic algorithm approach to job shop scheduling and its use in real-time cases[J]. *Int J of Computer Intergrated Manufacturing*, 2000, 13(5): 422-429.
- [4] 吴翌. 应用统计学[M]. 长沙: 国防科技大学出版社, 1995. 135-180.
- [5] 汪荣鑫. 随机过程[M]. 西安: 西安交通大学出版社, 1987. 183-201.

(上接第 682 页)

- [3] Magdi S M. Robust H^∞ control of discrete systems with uncertain parameters and unknown delays[J]. *Automatica*, 2000, 36(4): 627-635.
- [4] Carlos E S, Minyue F, Lihua Xie. H^∞ analysis and synthesis of discrete-time systems with time-varying uncertainty[J]. *IEEE Trans on Automatic Control*, 1993, 38(3): 459-462.
- [5] Guan X, Lin Z, Duan G. Robust guaranteed cost control for discrete-time uncertain systems with delay[J]. *IEE Proc Control Theory Application*, 1999, 146(6): 598-602.
- [6] Oucheriah S. Robust tracking and model following of uncertain dynamic delay systems by memoryless linear con-

trollers[J]. *IEEE Trans on Automatic Control*, 1999, 44(7): 1473-1477.

- [7] Cohen A, Shaked U. Linear discrete-time H^∞ -optimal tracking with preview [J]. *IEEE Trans on Automatic Control*, 1997, 42(2): 270-276.
- [8] 姜培刚, 李春文, 龙图景, 等. 参数和时延不确定性离散时间系统的 H^∞ 鲁棒控制 [J]. *控制与决策*, 2003, 18(2): 190-194.
(Jiang P G, Li C W, Long T J, et al. Robust H^∞ control for discrete-time systems with parameter uncertainty and time-delayed uncertainty[J]. *Control and Decision*, 2003, 18(2): 190-194.)