第 27 卷第 4 期
2010 年 4 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 27 No. 4
Apr. 2010

文章编号: 1000−8152(2010)04−0473−08

字典序进化算法用于组合优化问题

肖赤心[1,2]，蔡自兴[1], 王 勇[1]

(1. 中南大学 信息科学与工程学院, 湖南 长沙 410083; 2. 湘潭大学 信息工程学院, 湖南 湘潭 411105)

摘要: 为了寻求快速、高效的算法在合理的计算时间内解决大规模组合优化问题以克服目前许多算法的不足, 本文提出了一种新的编码方法, 将离散的组合空间一一映射到连续的整数区间, 结合进化策略的成熟搜索机制提高新算法的性能. 整数编码与问题的组合向量一一对应,所有编码均为可行方案, 有效避免了以往算法中的冗余运算, 进一步缩小了问题的搜索空间. 其次, 进化策略中加入了一个精英队列, 并且建立了相应的精英学习策略.在整个群体进化的同时, 精英个体也按照相应的策略不断优化, 从而有效吸收以往算法在组合优化问题上的成功经验, 有利于保留好的基因段. 最后证明了新算法以概率1收敛到全局最优. 基于旅行商问题测试库的仿真实验结果表明了算法的有效性.

关键词: 字典序; 组合问题; 进化策略; 旅行商问题

中图分类号: TP301　　**文献标识码**: A

Evolutionary strategy of lexicographic order for combinational problem

XIAO Chi-xin[1,2], CAI Zi-xing[1], WANG Yong[1]

(1. School of Information Science and Engineering, Central South University, Changsha Hunan 410083, China;
2. Information Engineering College, Xiangtan University, Xiangtan Hunan 411105, China)

Abstract: In order to construct a fast and effective algorithm to solve large-scale combinational problems in desirable computational time rather than be trapped in weakness as some existing algorithms, a novel encoding approach is proposed in this paper which applies an one to one mapping from a discrete space to a continuous integer section. Assembled with successful exploration and exploitation mechanism of evolutionary strategy, the performances of the algorithm are largely promoted. Since the one to one mapping between codes and combinational vectors, the new scheme only provides feasible solutions, which can help to avoid redundant computation existing in some algorithms effectively and the search space is further reduced. Secondly, a queue of elites is added in evolutionary mechanism combined with some particular learning strategy. The queue is refreshed frequently in evolution. This can help the algorithm to maintain better gene blocks. Finally, its convergence to global optimal solution with probability one is proved. The numerical experiments based on the Benchmarks of traveling salesman problem library(TSPLIB) show the effectiveness of algorithm proposed.

Key words: lexicographic order; combinational problem; evolutionary strategy; traveling salesman problem

1 引言(Introduction)

旅行商问题(traveling salesman problem, TSP)是一个典型的组合优化问题(combinatorial optimization problem, COP)[1], 即在n个城市中, 寻找一条遍历每个城市一次且仅一次的汉密尔顿(Hamilton)回路. 它描述简单却难以求解, 因而一直作为衡量各种优化算法性能的标准. 求解TSP问题已有了很多较好的算法, 本文将其大致分为两类：1)与问题本身特征相关的局部启发式搜索算法, 如2-Opt, 3-Opt和LK等算法[1,2]；2)不依赖于具体问题的通用算法, 如模拟退火法、蚂蚁算法以及遗传算法. 第一类算法过于依靠问题本身特征, 且容易陷入局部最优. 第二类算法具有强大的全局搜索能力, 成为近年来的研究热点. 同时也涌现出许多较好的路径杂交操作, 如order crossover, cycle crossover, partially matched crossover, edge recombination crossover, matrix crossover, distance preserving crossover edge assembly crossover, natural crossover[3~6].

进化算法(evolutionary algorithms, 简称EAs)是一种模拟自然进化过程的全局优化方法[7,8]. 目前实数编码进化算法多用于非离散事件的优化. 在组合优化领域, 很多问题是NP难问题, 这种算法还没有广泛的应用. 寻找组合问题的最优解往往与事件的排列有关, 使得许多用于连续空间搜索的且十分成熟

收稿日期: 2008−09−25, 收修改稿日期：2009−04−02.
基金项目: 国家自科基金重大专项(90820302); 国家自科基金(青年)项目 (60805027); 国家博士点基金项目(200805330005).

有效的遗传算子难于直接应用于组合空间的搜索. 研究者们往往针对具体问题去设计特殊的交叉、变异等遗传算子. 本文提出一种基于空间映射思想的整数编码进化算法求解TSP问题. 通过构造一种映射关系, 使组合空间中各个排列与连续整数空间中各个整数一一对应, 依靠现有进化算法求解优化问题的成功经验, 在连续整数空间进行搜索, 使发现最优解的效率和几率大大提高. 数值实验证明了新算法的有效性和可行性.

2 基本思想及分析(Basic ideas and analysis)

单个的文字之间是离散无序的, 但在字典中定义了一定的顺序和规律, 通过索引能快速查找. 对于一个给定的组合优化问题实例(不妨设最小化TSP), 设城市的个数为n, 其城市序号集合为$C=\{c_1,c_2,\cdots,c_n\}$. 问题的每一个解对应于C的一个排列, 这样的排列共有$n!$种, 它们是离散无序的. 如果将城市的序号利用起来, 定义一种不同排列之间的字典序关系, 便有可能将无序的排列空间(设为H)映射到连续的整数空间(设为N). 相当于将每一个解的排列进行整数编码, 这种编码与排列之间是一一对应的. 在连续的整数空间上用常规的进化算法来求解. 基本思想的流程如图1所示.

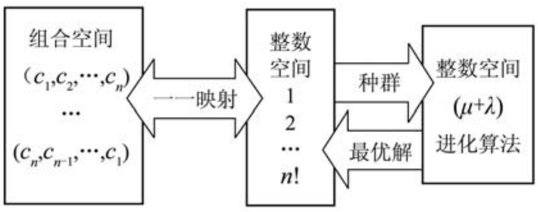


图 1 字典序进化算法

Fig. 1 Evolutionary algorithm of lexicographic order

定义 2.1 字典序关系: 设城市序号集合为$C=\{c_1,c_2,\cdots,c_n\}$, 若存在某种全序关系使得 $c_1<c_2<\cdots<c_n$. 对于任意两个由城市序号排列所组成的向量C_k, C_j, 设$C_k=(c^k_1,c^k_2,\cdots,c^k_n)$, $C_j=(c^j_1,c^j_2,\cdots,c^j_n)$, 其中$c^k_i,c^j_i\in C$, $1\leqslant i\leqslant n$, $1\leqslant k,j\leqslant n!$. 笔者称$C_k$按字典序低于$C_j$, 记为$C_k\prec C_j$, 当且仅当存在某个数$b, 1<b\leqslant n$, 对于任何$1\leqslant i<b$, 均有$c_1\leqslant c^k_i\leqslant c^j_i\leqslant c_n$且至少有一个$c^k_i<c^j_i$.

基于字典序关系, 每一个排列的起始编号必然按$c_1\prec c_2\prec\cdots\prec c_n$有序. 显然, 每一个以$c_i(1\leqslant i\leqslant n)$开头排列, 分别有$(n-1)!$个, 这样就把整个空间中$n!$个排列元素分成了$n$个区间, 每一个区间所有排列的第二位再按字典序关系排列, 如此往复, 直至最后一个元素确定. 例如：有4个城市, 设$C=\{1,2,3,4\}$, 那么其组合向量的字典序关系如表1所示.

表 1 字典序排列与整数编码($n=4$)

Table 1 Permutations of lexicographic order and integer code ($n=4$)

整数编码	字典序排列	整数编码	字典序排列
1	(1, 2, 3, 4)	13	(3, 1, 2, 4)
2	(1, 2, 4, 3)	14	(3, 1, 4, 2)
3	(1, 3, 2, 4)	15	(3, 2, 1, 4)
4	(1, 3, 4, 2)	16	(3, 2, 4, 1)
5	(1, 4, 2, 3)	17	(3, 4, 1, 2)
6	(1, 4, 3, 2)	18	(3, 4, 2, 1)
7	(2, 1, 3, 4)	19	(4, 1, 2, 3)
8	(2, 1, 4, 3)	20	(4, 1, 3, 2)
9	(2, 3, 1, 4)	21	(4, 2, 1, 3)
10	(2, 3, 4, 1)	22	(4, 2, 3, 1)
11	(2, 4, 1, 3)	23	(4, 3, 1, 2)
12	(2, 4, 3, 1)	24	(4, 3, 2, 1)

$(1,2,3,4)\prec(1,2,4,3)\prec\cdots\prec(4,3,2,1)$. 表1列出了$n=4$时, 所有字典序排列与其整数编码的对应关系. 基于这种字典序关系, 排列空间的每个排列被映射到了一个连续的线性空间. 如果把这个连续的线性空间取为一段连续的整数空间, 则这段区间上的每一个整数就是相应排列的整数编码. 定义如下映射：

定义 2.2 设基于城市序号集合$C=\{c_1,c_2,\cdots,c_n\}$和字典序的排列区间为$H=[C_1,C_{n!}]$, 整数区间$N=[1,n!]$. 对应法则$f: H\xrightarrow{f}N$, 表示从排列区间到整数区间的一一对应关系. f和f^{-1}法则的算法流程如下:

function f 输入: 城市总数n和排列的整数编码x; 输出: 将整数编码x翻译成为城市的排列, 在数组output输出.

output$=$[ ]; Symbles$=1:n$; temp$=$Symbles;

$q=x; r=0$

for $i=1:n$

$r=\mathrm{mod}(q,\mathrm{factorial}(n-i))$ //取q除以$(n-i)$阶乘的余数

$q=\mathrm{floor}(\mathrm{q}/\mathrm{factorial}(n-i))$ //对q除以$(n-i)$阶乘的商取整

if $r==0$ //如果余数为0

output$(i)=$temp(q) //将temp第q个元素成为output第i个输出

temp$(q)=$[ ] //删除temp中第q个元素

$q=\mathrm{factorial}(n-i)$ //q被赋值为$(n-i)$的阶乘

else //如果余数不为0

output$(i)=$temp$(q+1)$ //temp第$q+1$个元素成为output第i个输出

temp$(q+1)=[\,]$ //删去temp中第q个元素

$q=r$ //q被赋值余数

end

end

disp(output).

function f^{-1}　输入: 城市总数n和任意一个排列组合combination; 输出: 排列combination在整数区间$[1,n!]$对应的整数编码x.

Symbles $=1:n$ //数组初始为顺序的整数1至n

temp = Symbles; $x=0$

for $i=1:n$

$I=$ find(temp $==$ combination(i)) //找到temp中等于combination(i)的序号

$x=x+(I-1)$*factorial$(n-i)$ //factorial$(n-i)$为$(n-i)$的阶乘

temp$(I)=[\,]$ //在temp中将第I个元素删去

end

$x=x+1$ //排列从1开始编号

disp(x).

由字典序关系的定义, 可以发现在某一区间内相邻的两个排列之间存在着不同程度的相似. 在TSP问题中每个排列实际上代表着一条路径, 这种相似会使得在一定范围内这些路径的长度也存在着某种规律的变化. 图2是基于Att48[9]48个城市的字典序排列空间中, 所有整数编码的前50个连续整数编码所代表的路径长度的关系.

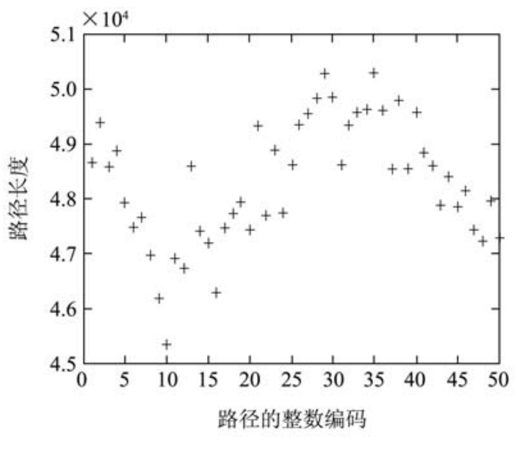


图 2　Att48中整数代码1～50所代表的路径的长度分布

Fig. 2　Distance distribution of path corresponding to the integer codes from 1 to 50 in Att48

2.1　空间舍弃(Space discard)

对于给定的城市序号集合$C=\{c_1,c_2,\cdots,c_n\}$, 在字典序关系的基础上, TSP问题的最优路径可以仅在任何以$c_i(1\leqslant i\leqslant n)$开头的$(n-1)!$ 个排列中搜索.

证　TSP问题是求一条闭合Hamilton回路. 设存在一条最短Hamilton回路$(c_1,c_2,\cdots,c_n,c_1)$, 简写为$(c_1c_2Ac_nc_1)$,其中A表示回路中省略的路径节点. 那么在以c_2开头的回路中, 必然存在一条回路$(c_2Ac_nc_1c_2)$, 与前者有着相同的距离长度. 显然, 有着同样最短距离的回路共有n条. 它们分别存在于以c_i（$1\leqslant i\leqslant n$）开头的各个字典序排列区间，均可以通过循环移位操作变成$c_1c_2Ac_nc_1\in[1,(n-1)!]$.　证毕.

2.2　空间转移(Space shift)

本文进行整数编码的区间是$[1,n!]$因为计算机计算精度的限制, 在大数区间进行高精度运算比较困难, 即使利用大数运算工具其精度也是有限的. 当组合成员大于500时, 采用空间转移技术. 正如一本很厚的字典, 可以分卷, 每一卷重新开始编号. 设任意整数编码$k(1\leqslant k\leqslant n!)$, 它对应的字典序$I$的排列为$C_k$. 如果以排列$C_k$中的城市顺序为新起始重新进行另一种字典序$I'$编号, 那么原来的整数编码$k$就成为新的$I'$序关系下的起点. 如此往复, 只要$n!$有界, 就可以用现有的计算机精确搜索.

3　字典序算法(Lexicographic order algorithm)

本文采用$(\mu+\lambda)$进化策略. 其中, 父代种群数目为μ, 子代种群数目为λ. 由2.1节分析可知, 对于n个城市的TSP问题可以仅在字典序关系的前$(n-1)!$个排列(即以c_1开头的排列)中搜索最优解. 初始种群在区间$[1,(n-1)!]$随机产生, 当n很大时, 搜索空间依然很大. 在这么大的搜索空间中如果不加入启发式信息, 算法效果不会很好. 为了加快收缩搜索空间, 增加搜索的针对性, 在算法中设置了一个精英队列.

3.1　初始与精英(Initialization and elites)

对于TSP问题, 在最佳路径里必然包括而且在很大程度上包括了相邻城市间距离最短(或者较短)的那边. 利用贪心法每次取与出发点相邻的最短边求得n条回路, 例如：$c_1c_2Ac_nc_1$, $c_2c_iA'c_jc_2,\cdots$, 它们分别从$c_i(1\leqslant i\leqslant n)$出发最后回到$c_i$. 全局最优路径的距离一定会小于或者等于贪心法所得的结果. 将每一个回路全部循环左移至c_1开头的回路. 便得到n个以c_1开头的排列, 根据f^{-1}法则可以进一步得到这n个排列的整数编码. 例如：以36号城市为起始节点的贪心法最优路径为：(36, 44, 31, 38, 9, 1, 8, 22, 16, 3, 23, 11, 12, 15, 33, 46, 40, 20, 47, 21, 13, 25, 14, 34, 41, 29, 5, 48, 39, 32, 24, 10, 42, 26, 4, 35, 45, 2, 30, 6, 37, 19, 27, 43, 17, 28, 7, 18). 调整后得 (1, 8, 22, 16, 3, 23, 11, 12, 15, 33, 46, 40, 20, 47, 21, 13, 25, 14, 34, 41, 29, 5, 48, 39, 32, 24, 10, 42, 26, 4, 35, 45, 2,

30, 6, 37, 19, 27, 43, 17, 28, 7, 18, 36, 44, 31, 38, 9)转换得到其整数编码为35323197226231168789383325297315096488667585081870454173390. 选取其中距离较短的前10个加入初始精英队列. 在精英队列的指导下, 算法可以很快地抛弃一些不必要的搜索区域, 增加了搜索的针对性和有效性.

3.2 交叉(Crossover)[10]

本文的进化策略采用加权平均交叉$X_k' = X_{S,i} + \theta * (X_{T,j} - X_{S,i})$, 这里$X_j'$表示新产生的子代个体, 下标$S$ 和T指从μ个父代个体或者精英群体中随机选取的两个个体, $\theta \in [0,1]$为均匀随机变量, 本文取0.5. 如果让精英群体每次都参与交叉就可能使算法收敛过快, 从而陷入局部最优. 所以采用如下策略: 每10次交叉中有1次从精英群体中选择1个个体参与交叉, 其它均从父代群体中选择. 交叉算子如下：

```
function crossover( )
设P和E分别代表父代群体和精英群体;
for k = 1 to 子代群体个数λ do
  if (flip(10%))
    从E选择一个精英个体X_{S,i}
  else
    从P中选择一个父代个体X_{S,i}
  end if
    从P中选择一另个父代个体X_{T,j}
    X'_k = X_{S,i} + θ*(X_{T,j} − X_{S,i})
end for
end
```

说明 flip(P)函数功能是, 如果随即获得的概率小于或等于P则返回true.

3.3 变异(Mutation)

Mühlenbein 和 Schlierkamp-Voosen[11,12] 提出了BGA变异算子. 本文对其进行一些改写, 假设对子种群$\vec{o}$进行BGA变异操作得到新群体$\vec{o}' = (x_1', \cdots, x_r', \cdots, x_s', \cdots, x_\lambda')$ 如下式：

$$x_j' = \begin{cases} x_j \pm \mathrm{rang}_j \sum\limits_{k=0}^{15} \alpha_k 2^{-k}, & U(0,1) < 0.97, \\ x_j, & j = 1, \cdots, \lambda. \end{cases}$$

式中, $\pm$以50%的概率取“+”或者“−”, α_k以1/16的概率取1, 否则为0. $U(0,1)$为0至1的均匀分布随机数产生器. rang_j是变异的范围, 这里定义为$(Ub-Lb) \cdot (1 - \mathrm{current_gen}/\mathrm{total_gen})^6$, 其中$Ub$, Lb分别是个体整数编码的上、下边界. current_gen, total_gen分别代表当前的进化代数与总的进化代数. 用这个变异算子能可靠地在$[x_j - \mathrm{rang}_j, x_j + \mathrm{rang}_j]$ 区间产生与个体x_j相邻的个体, 并且变异的幅度随着进化代数的增大而减小.

3.4 精英优化(Elites optimized)

贪心法求得最优路径的策略是每次在剩余的城市中选取与出发城市最近的城市作为下一个路径节点. 从以下几个方面对这些路径进行优化调整.

策略 1(替换长边) 1) 随机选择精英群体中的个体;

2) 从中选取距离较长的边, 设两个端点分别为a, b；

3) 分别求距它们最近的前$n \cdot 30\%$个城市集合设为A, B(n为城市总个数);

4) 若$A \cap B \neq \varnothing$,即$A$和$B$相交为非空集时, 则尝试交集中是否存在有节点$c$使得$d_{ac} + d_{cb} < ab$；

5) 将c 在路径中的原位置删除, 局部路径ab变成acb.

策略 2(消除交叉边) 1) 建立$n \times n$ 邻接矩阵, 每行的元素k_{ij}表示距离城市i第j近的城市.

2) 遵循精英个体对应的路径序列寻找该路径中是否有交叉边. 设存在局部路径$\cdots a, b \cdots c, d \cdots$, 它们的结点坐标分别为$(x_a, y_a)$, (x_b, y_b), (x_c, y_c), (x_d, y_d). 若

$$\begin{aligned} &[\min(x_c, x_d), \max(x_c, x_d)] \cap \\ &[\min(x_a, x_b), \max(x_a, x_b)] \neq \varnothing, \\ &[\min(y_c, y_d), \max(y_c, y_d)] \cap \\ &[\min(y_a, y_b), \max(y_a, y_b)] \neq \varnothing, \end{aligned}$$

则边a, b 和c, d一定是相交的两条边. 这里设$x_a \leqslant x_c \leqslant x_b \leqslant x_d, y_d \leqslant y_a \leqslant y_c \leqslant y_b$, 根据向外扩展原则, 参照距离邻接矩阵将原局部路径调整为无交叉的且距离最短的局部路径. 整个学习过程虽然运算复杂度高, 但是仅仅应用在精英个体上且有贪心法做了前期优化, 不会影响整个算法的时间效率. 本文采用常用的$(\mu + \lambda) - ES$ 选择策略. 描述如下: 在μ个父代个体与λ 个子代个体中选择μ个最好的个体进入下一代种群. 如果新一代父代个体(μ个)中存在最优个体优于精英个体中最差的个体, 则予以取代.

4 算法流程(Algorithm procedure)

流程如下：

1) 初始种群(μ个)和精英队列的初始化(current_gen $= 0$);

2) 满足终止条件则退出, 不满足转第3)步;

3) 由父代种群或精英队列交叉产生子代群

体(λ个)；

4) 对子群体(λ个)进行变异；

5) 将个体(即路径的整数代码)翻译成路径队列, 计算路径长度；

6) 精英群体按调整策略进行自学习；

7) 从$\mu+\lambda$个个体中选择个最优μ个体进入下一代种群, 如果新群体中最好的个体优于精英群体中最差的个体则予以取代；

8) current_gen = current_gen + 1 转第2)步.

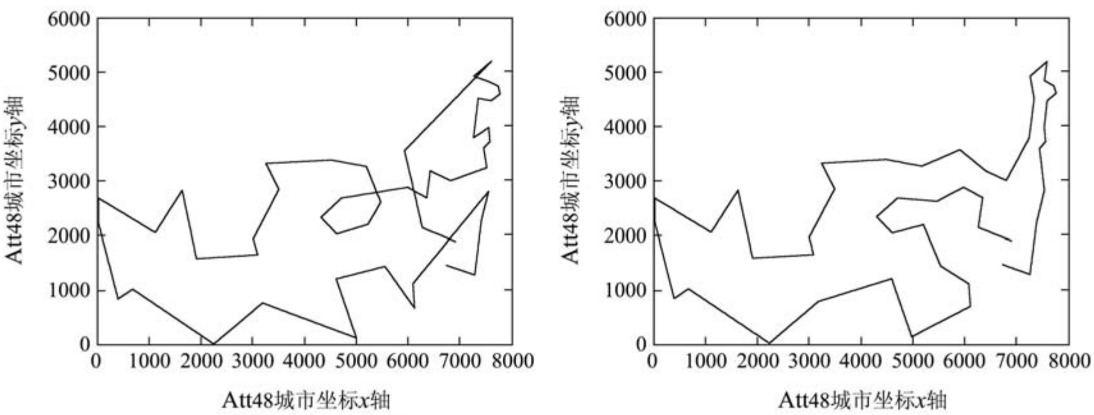
5　试验及结果(Experiments and results)

为了方便对比，本文选用了旅行商问题测试库(TSPLIB)[9]中的多个实例进行实验. 对于组合成员在700以下的实验, 新进化参数设置如下：$\mu=100, \lambda=300$, 进化代数total_gen = 1000. 新算法大部分程序均在Matlab 2006a平台上实现, 所有计算都是在Genuine Intel 2140@1.60G, 内存1G的PC机上完成, 其中大数高精度计算依托Matlab2006符号运算数学工具箱(Symbolic Math Toolbox Version 3.1.4)进行. 实验结果证明了新算法的有效性.

实验 1　以TSPLIB95中Att48为例, 说明了新算法通过精英学习指导进化的过程. 图3(a)为贪心法求得的最优路径, 其整数编码为37278108504483160047173896925400992701832084244808202108386, 路径长度11865, 相应的排列为(1, 8, 38, 31, 22, 16, 3, 34, 41, 29, 2, 26, 4, 35, 45, 10, 24, 42, 5, 48, 39, 32, 21, 47, 11, 23, 14, 25, 13, 12, 15, 33, 46, 44, 18, 7, 28, 36, 30, 6, 37, 19, 27, 43, 17, 20, 40, 9). 通过精英学习、全局进化, 最后得全局最优解（如图3(b)). 全局最优的整数编码为37279319749735285213507811329519563713358680557416718960474, 路径长度为10481, (即$10628-147$, 城市9至城市1的距离为147), 相应的路径排列为(1, 8, 38, 31, 44, 18, 7, 28, 6, 37, 19, 27, 17, 43, 30, 36, 46, 33, 20, 47, 21, 32, 39, 48, 5, 42, 24, 10, 45, 35, 4, 26, 2, 29, 34, 41, 16, 22, 3, 23, 14, 25, 13, 11, 12, 15, 40, 9). 找到了已知的全局最优解.

实验 2　将新算法(LO-ES)的计算结果与改进的郭涛算法(IGT)[3]进行比较, 实验结果见表2, 表2中每个实例都是连续运行10次的统计结果.

实验 3　在TSPLIB95中挑选其他一些问题进行测试, 每个问题均独立运行20次结果如表3所示: p%表示所有实验中找到最优解的比率; gen表示算法找到最优解的代数近似值.



(a) 优化前的精英个体　　(b) 优化后的全局最优个体

图 3　贪心法最优解和新算法全局最优结果的比较

Fig. 3　Comparison of optimum solutions between greedy method and proposed algorithm

表 2　与改进的郭涛算法(IGT)[3]在TSPLIB95中3个问题的对比

Table 2　Comparison with IGT[3] algorithms on 3 benchmark problems in TSPLIB95

测试函数[9]	TSPLIB 提供最短路径[9]	IGT[3]		LO-ES	
		best	mean	best	mean
pr136(EUC_2D)	96772	96770.924122	96770.924122	96772	96772
pr144(EUC_2D)	58537	58535.221761	58542.129537	58537	58537
a280(EUC_2D)	2579	2586.769648	2588.568179	2579	2579

表2中EUC_2D[13]表示城市间的距离按2维空间欧式距离计算. 表3中ATT,GEO[13]类型的距离公式稍有不同. 所有计算均严格按照文献[17]的标准进行, 每一次的距离计算的结果都按要求取整, 故新算法的结果中没有小数部分. 表2实验数据显示, 新算法在表中3个实例问题上结果等于或优于IGT[3]算法. 从表2和3的实验结果表明：新算法在300个组合成员以下的实验中性能稳定, 均能可靠找到最优解. 表3中的U574和gr666[9]表明, 在高于500个成员的实验中, 也能较可靠地找到最优解, 证明空间转移机制的可靠性. 另外, 不同的大数计算工具对算法的运行时间影响很大, 在不计大数解码和编码时间情况下, U574(EUC_2D)的平均执行时间为6.9 s，gr666(GEO)的平均执行时间为8.4 s. 对于低于此规模的实验，新算明显优于文献[4, 5]提供的数据.

表 3 TSPLIB95[9]测试库中其它问题的实验结果

Table 3 Results of some others in TSPLIB95[9] benchmark library

测试函数[9]	TSPLIB提供最短路径[9]	best	mean	$p\%$	gen
att48(ATT)	10628	10628	10628	100	85
st70(EUC_2D)	675	675	675	100	156
berlin52(EUC_2D)	7542	7452	7452	100	105
lin105(EUC_2D)	14379	14379	14379	100	180
Line318(EUC_2D)	42029	42029	42029	100	364
U574(EUC_2D)	36905	36905	37028.2	95	392
gr666(GEO)	294358	294358	294436.8	90	399

6 结语(Conclusion)

基于空间映射思想结合进化算法来求解组合优化问题旨在利用连续域的方法来解决离散空间的问题. 实验显示了其解决组合优化问题的优势. 如果有高精度浮点数运算计算机做支持, 本算法很容易对多个移动Agent提供集中实时计算, 特别是在多机器人实时任务分配, 多机器人实时路径规划等应用领域. 在今后的研究中将继续提高算法的各项性能.

参考文献(References):

[1] 邢文训, 谢金星．现代优化计算方法[M]．北京：清华大学出版社, 1999.
(XING Wenxun, XIE Jinxing. *Modern Methods of Optimal Computation*[M]. Beijing: Tsinghua University Press, 1999.)

[2] 陈志平, 徐宗本．计算机数学 —— 计算复杂性理论与NPC、NP难问题的求解[M]．北京：科学出版社, 2001.
(CHEN Zhingping, XU Zongben. *Computer Mathematics–Theory of Computational Complexity and Problem Solving for NPC, NP Puzzle*[M]. Beijing: Science Press, 2001.)

[3] TAO Guo, MICHALEWICZ Z. Evolutionary algorithms for the TSP[C] *//Proceedings of the 5th Parallel Problem Solving from Nature Conference, Lecture Notes in Computer Science 1498*. Berlin: Springer, 1998, 803 – 812.

[4] 蔡之华, 彭锦国, 高伟, 等. 一种改进的求解TSP问题的演化算法[J]. 计算机学报, 2005, 28(5): 823 – 828.
(CAI Zhihua, PENG Jinguo, GAO Wei, et al. An improved evolutionary algorithm for traveling salesman problem[J]. *Chinese Journal of Computers*, 2005, 28(5): 823 – 828.)

[5] 杨辉, 康立山, 陈毓屏. 一种基于构建基因库求解TSP问题的遗传算法[J]. 计算机学报, 2003, 26(12): 1753 – 1758.
(YANG Hui, KANG Lishan, CHEN Yuping. A gene-based gentic algorithm for TSP[J]. *Chinese Journal of Computers*, 2003, 26(12): 1753 – 1758.)

[6] 王宇平, 李华英. 求解TSP的量子遗传算法[J]. 计算机学报, 2007, 30(5): 748 – 755.
(WANG Yuping, LI Huaying. A novel quantum genetic algorithm for TSP[J]. *Chinese Journal of Computers*, 2007, 30(5): 748 – 755)

[7] MICHALEWICZ Z, SCHOENAUER M. Evolutionary algorithm for constrained parameter optimization problems[J]. *Evolutionary Computation*, 1996, 4(1): 1 – 32.

[8] 王勇, 蔡自兴, 周育人, 等. 约束优化进化算法[J]. 软件学报, 2009 20(1): 11 – 29.
(WANG Yong, CAI Zixing, ZHOU Yuren, et al. Constrained optimization evolutionary algorithms[J]. *Journal of Software*, 2009 20(1): 11 – 29.)

[9] 德国海德堡大学组合优化问题研究组. TSP问题标准测试数据库TSP95[DB/OL]. http://elib.zib.de/pub/mptestdata/tsp/tsplib/tsplib.html.

[10] 陈国良, 王煦法, 庄镇泉, 等. 遗传算法及其应用[M]. 北京: 人民邮电出版社, 1996.
(CHEN Guoliang, WANG Xufa，ZHUANG Zhenquan, et al. *Genetic Algorithms and Their Applications*[M]. Beijing: Posts & Telecom Press, 1996.)

[11] MHLENBEIN H, SCHLIERKAMP-VOOSEN D. Predictive models for the breeder genetic algorithm I: Continuous parameter optimization[J]. *Evolutionary Computation*, 1993, 1(1): 25 – 49.

[12] WANG Yong, CAI Zi-xing, GUO Guanqi et al. Multi-objective optimization and hybrid evolutionary algorithm to solve constrained optimization problems[J]. *IEEE Transactions on Systems, Man, And Cybernetics-Part B: Cybernetics*, 2007, 37(3): 560 – 575.

[13] 德国海德堡大学组合优化问题研究组. TSP问题标准测试数据库TSP95使用说明[EB/OL]. http://www.informatik.uni-heidelberg.de/groups/comopt/software/TSP LIB95/DOC.PS

[14] GUNTER Rudolph. Convergence analysis of canonical genetic algorithms[J]. *IEEE Transactions on Neural Networks*, 1994, 5(1): 96 – 101.

[15] 江瑞, 罗予频, 胡东成, 等.一种协调勘探和开采的遗传算法: 收敛性及性能分析[J]. 计算机学报, 2001, 24(12): 1233 – 1241.
(JIANG Rui, LUO Yuping, HU Dongcheng, et al. A genetic algorithm by coordinating exploration and exploitation-convergence properties and performance analyses[J]. *Chinese Journal of Computers*, 2001, 24(12): 1233 – 1241.)

[16] 刘静, 钟伟才, 刘芳, 等. 组织进化数值优化算法[J]. 计算机学报, 2004, 27(2): 157 – 167.
(LIU Jing, ZHONG Weicai, LIU Fang, et al. An organizational evolutionary algorithm for constrained and unconstrained optimization problems[J]. *Chinese Journal of Computers*, 2004, 27(2): 157 – 167.)

[17] SUZUKI J. A Markov chain analysis on simple genetic algorithm[J]. *IEEE Transactions on System, Man and Cybernetics*, 1995, 25(4): 655 – 659.

[18] IOSIFESCU M. *Finite Markov Processes and Their Applications*[M]. Chichester: Wiley, 1980

附录 收敛性[14~16](Appendix Convergence proof))

标准遗传算法中, 个体由二进制串表示, 而本文采用整数编码的形式. 若存在n个城市需要遍历, 那么搜索空间S是一个离散的搜索空间, 其大小为$n!$. 本文通过字典序映射, 将离散搜索空间进行编码映射到连续整数区间$[1, n!]$, 由2.1小节可知该整数区间对应的所有Hamilton回路均可在$[1,(n-1)!]$整数编码区间获得, 此区间用S'表示. 每个整数编码$x(x\in S')$所对应的适应值表示为$\mathrm{Fitness}(x)$, 令$F=\{\mathrm{Fitness}(x)|x\in S'\}$, 显然$|F|\leqslant|S'|$, 于是可以写成$F=\{F_1,F_2,F_3,\cdots,F_{|F|}\}$, 这里$F_1>F_2>F_3>\cdots>F_{|F|}$. 根据适应度的不同又可把整数编码区间$S'$内的整数集合划分为若干非空子集$\{S_i\}$, 这里$S_i=\{x|x\in S'$ and $\mathrm{Fitness}(x)=F_i, i=1,2,3,\cdots,|F|\}$, 于是有$\sum\limits_{i=1}^{|F|}|S_i|=|S|$; $|S_i|\neq\varnothing,\forall i=\{1,2,\cdots,|F|\}$; $S_i\cap S_j=\varnothing,\forall i\neq j$; $\bigcup\limits_{i=1}^{|F|}S_i=S$. 显然,$F_1$ 即为全局最优解F^*, 且集合S_1包含了所有适应度等于F_1的整数编码.

在本文提出的算法中, 整数编码构成种群, 群内的整数编码随着进化代数不断变化, 但是种群内个体的数目是固定不变的, 即种群$p=(x_1,x_2,\cdots,x_\mu)$. 令P表示所有种群的集合, 因为种群内个体允许相同, 所以种群可能的数目为

$$|P|=\begin{pmatrix}(n-1)!+\mu-1\\ \mu\end{pmatrix}.$$

为了衡量种群的优劣, 需要定义种群的适应度. 对于种群p其适应度定义为: $\mathrm{Fitness}(p)=\max\{\mathrm{Fitness}(x_i)|i=1,2,\cdots,\mu\}$. 这样就有$F_{|F|}\leqslant\mathrm{Fitness}(p)\leqslant F_1,\forall p\in P$. 进而可以把集合$P$划分为非空子集$\{p_i\}$, $p_i=\{p|p\in P$ and $\mathrm{Fitness}(p)=F_i, i=1,2,\cdots,|F|\}$. 显然, $\sum\limits_{i=1}^{|F|}|p_i|=|P||p_i|\neq\varnothing,\forall i=\{1,2,\cdots,|F|\}p_i\cap p_j=\varnothing,\forall i\neq j\bigcup\limits_{i=1}^{|F|}p_i=P$. 其中集合$p_1$包含所有适应度为$F^*$的种群. 设$p_{ij}$是$p_i$中第$j$个种群$i=1,2,\cdots,|F|$, $j=1,2,\cdots,|P_i|$. 在各进化算子的作用下, 从p_{ij}到p_{kl}的种群状态转移可以表示为$p_{ij}\to p_{kl}$, 其状态转移概率可以表示为$\Pr(p_{ij}\to p_{kl})$或者$\mathrm{pr}_{ij,kl}$. 从种群p_{ij}直接转变到种群集合p_k中某一种群用$p_{ij}\to p_k$表示, 其状态转移概率表示为$\Pr(p_{ij}\to p_k)$ 或$\mathrm{pr}_{ij,k}$, 相应地, 从种群集合p_i直接转变到种群集合p_k用$p_i\to p_k$表示, 其状态转移概率表示为$\Pr(p_i\to p_k)$ 或$\mathrm{pr}_{i,k}$. 综上所述, 可以得到$\mathrm{pr}_{ij,k}=\sum\limits_{l=1}^{|p_k|}\mathrm{pr}_{ij,kl}$, $\sum\limits_{i=1}^{|F|}\mathrm{pr}_{ij,k}=1$, $\mathrm{pr}_{i,k}\geqslant\mathrm{pr}_{ij,k}$. 明确了以上概念的, 在文献[15~17]的基础上, 将给出新算法全局收敛性的一般性定义并对其全局收敛性做出证明.

定义 1 一个进化算法收敛于全局最优解, 当且仅当$\lim\limits_{t\to\infty}\mathrm{pr}\{\mathrm{Fitness}(p_t)=F^*\}=1$, 这里pr表示概率, p_t表示第t代种群.

定理 1[18] 令p'是一个n阶可归约随机矩阵, 即通过相同的行变换和列变换后可以得到$p'=\begin{pmatrix}C&0\\R&T\end{pmatrix}$, 其中$C$是一个$m$阶本原随机矩阵且$R,T\neq0$, 则

$$p'^{\infty}=\lim_{k\to\infty}p'^k=\lim_{k\to\infty}\begin{pmatrix}C^k&0\\ \sum\limits_{i=0}^{k-1}T^iRC^{k-i}&T^k\end{pmatrix}=\begin{pmatrix}C^{\infty}&0\\R^{\infty}&0\end{pmatrix}$$

是一个稳定的随机矩阵, 且$p'^{\infty}=e'\cdot p'^{\infty}$, 这里$e=(1,1,\cdots,1)$, $p'^{\infty}=p'^0\cdot\lim\limits_{k\to\infty}p'^k=p'^0\cdot p'^{\infty}$ 唯一确定且与初始分布p'^0无关, 并且当$1\leqslant i\leqslant m$ 时, $p'^{\infty}_i>0$; $m\leqslant i\leqslant n$ 时, $p'^{\infty}_i=0$.

定理 2 在新算法中, $\forall i,k\in\{1,2,\cdots,|F|\}$, 有

$$\mathrm{pr}_{i,k}\begin{cases}>0,& k\leqslant i,\\ =0,& k>i.\end{cases}$$

证 在$\mu+\lambda$策略中, 交叉和变异算子产生的新个体, 只影响子代种群, 在选择机制的作用下, 只有适应度的优于当前父代种群的个体才能进入下一父代种群. 父代群体可以看作是子代群体的优秀个体保留群体. 问题的最优解必然最终出现在父代群体中. 所以这里仅以父代种群为研究对象. $\forall p_{ij}\in p_i\Rightarrow\exists x\in p_{ij}$, $\mathrm{Fitness}(x)=F_i, j=1,2,\cdots,|P_i|$. 设在进化算子的作用下, 种群$p_{ij}$转变为种群$p_{kl}$. $x\in p_{ij}$且适应值一定不小于其它个体, 用文献[14]类似的方法可知. 如果$x\in p_{kl}$, 则

$$\mathrm{Fitness}(p_{kl})\geqslant\mathrm{Fitness}(p_{ij})\Rightarrow k\leqslant i\forall k>i,$$

$$\mathrm{pr}_{ij,kl}=0 \Rightarrow \mathrm{pr}_{ij,k}=\sum_{i=1}^{|p_k|}\mathrm{pr}_{ij,kl}=0 \Rightarrow \mathrm{pr}_{i,k}=0.$$

给定另一个个体$x' \in p_{kl}$,

$$\mathrm{Fitness}(x')=F_k, k \leqslant i$$

且$x \neq x'$, 由x产生x'的概率大于0. 由此可知在进化算子作用后, 从p_{ij}到p_k中任一种群的概率必然也大于0, 即$\mathrm{pr}_{ij,k} \geqslant 0$, 又$\mathrm{pr}_{i,k} \geqslant \mathrm{pr}_{ij,k}$, 所以$\mathrm{pr}_{i,k}>0, \forall k \leqslant i$.

证毕.

该定理说明, 适应度低的种群可以转移到适应度相同或适应度更高的种群, 而不能从适应度高的种群转移到适应度低的种群. 因此一旦新算法进入到集合p_1中就不可能逃逸出来.

定理 3 新算法具有全局收敛性.

证 每一个$p_i, i=1,2,\cdots,|F|$ 可以看作是可以看作是有限Markov链上的一个状态, 根据定理2, 该Markov链的转移矩阵可表示为

$$p'=\begin{pmatrix} \mathrm{pr}_{1,1} & 0 & \cdots & 0 \\ \mathrm{pr}_{2,1} & \mathrm{pr}_{2,2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{pr}_{|F|,1} & \mathrm{pr}_{|F|,1} & \cdots & \mathrm{pr}_{|F|,1} \end{pmatrix}=\begin{pmatrix} C & 0 \\ R & T \end{pmatrix},$$

$T \neq 0, C=(\mathrm{pr}_{1,1})=1 \neq 0,$

$$R=(\mathrm{pr}_{2,1},\mathrm{pr}_{3,1},\cdots,\mathrm{pr}_{|F|,1})^{\mathrm{T}}>0.$$

根据定理1有

$$p'^{\infty}=\lim_{k\to\infty}p'^{k}=\lim_{k\to\infty}\begin{pmatrix} C^k & 0 \\ \sum_{i=0}^{k-1}T^iRC^{k-i} & T^k \end{pmatrix}=\begin{pmatrix} C^{\infty} & 0 \\ R^{\infty} & 0 \end{pmatrix},$$

其中$C^{\infty}=1$, $R^{\infty}=(1,1,\cdots,1)^{\mathrm{T}}$. 这样$p'^{\infty}$就是一个稳定的随机矩阵且

$$p'^{\infty}=\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & \cdots & 0 \end{pmatrix}.$$

因此, 根据定义1 可知新算法具有全局收敛性. 证毕.

作者简介:

肖赤心 (1973—), 男, 讲师, 博士研究生, 主要研究领域为进化计算、神经网络、自然语言处理, E-mail: chixinxiao@hotmail.com;

蔡自兴 (1938—), 男, 教授, 博士生导师, 主要研究领域为人工智能、计算智能、智能控制等;

王 勇 (1980—), 男, 讲师, 博士研究生, 主要研究领域为进化计算、约束优化、多目标优化.