

遗传算法, 空间平滑, TSP问题, 解

35-40

# 一种具有空间平滑技术的遗传算法研究

A Genetic Algorithm with Search Space Smoothing

陈建 李信满 柳玉辉

(东北大学软件中心 沈阳 110006)

0242-23

**Abstract** In this paper a genetic algorithm with search space smoothing is proposed for its "premature" problem. Based on TSP, a great amount experiment was made, and the results have proved that this method is very effective.

**Keywords** Premature problem, Genetic algorithm, TSP, Search space smoothing

## 1 引言

遗传算法从开始提出到现在,经过众多学者经过十几年的研究、发展,已广泛地应用到各个领域<sup>[1]</sup>。但在遗传算法应用过程中,容易发生“早熟”现象<sup>[2]</sup>。所谓“早熟”现象是指在找到最优解或近优解之前,遗传算法的迭代过程导致主导“基因”固定的一种染色体基因的确切现象。一般地认为,导致“早熟”现象的因素有这样几个方面:①选择算子选取不合理造成基因缺陷;②交叉算子构造不理想,引起模式缺陷;③遗传算法的参数选择不科学等。

为了避免“早熟”现象的产生,提高遗传算法的搜索能力,许多学者围绕以上几个方面进行了深入的探讨、研究,取得了许多有意义的成果和方法。但是这些方法都无法取得较为通用的结果,这就促使了混合遗传算法的兴起<sup>[3~5]</sup>。混合遗传算法的实质是通过不同算法的优点有机结合,改善单纯遗传算法的性能。一般地,混合遗传算法的性能在一定程度上优于单纯遗传算法的性能,但不能保证一定能获得问题的全局最优解,特别是对于多峰多谷的函数优化问题。

本文将从另一个角度来解决遗传算法应用中的“早熟”问题,即在遗传算法优化过程中引入搜索空间平滑技术<sup>[6,7]</sup>,我们称之为具有空间平滑的遗传算法。这种遗传算法只改变问题的解值空间结构,不涉及遗传算法的结构,可保持原有遗传算法的简单结构等特征。

## 2 搜索空间平滑技术的工作原理

局部极小点的存在,使得欲获得问题的全局最

优解变得非常困难,甚至无法获得问题的最优解,只能得到局部最优解。可以想象,如果局部极小点越少,那么获得问题最优解的机会就越多,搜索算法就越有效。通常,为了改善算法的性能,经常采用一些有效的启发式算法,如2-opt<sup>[8]</sup>、3-opt<sup>[9]</sup>等,试图通过领域中的某种变换,使问题的解跳出局部“陷阱”,获得问题的最优解。

搜索空间平滑技术可以限制问题搜索空间中局部极小点的数目是基于这样思想:设存在一个具有许多局部极小点的搜索空间,在该空间内很容易陷入一个局部极小点,如图1所示。如果应用某种平滑技术将原有的搜索空间进行平滑后,一些局部极小点被暂时“填满”,这样,对于一个搜索过程而言,陷入局部极小点的概率减少,获得全局最小点的机会就增加。这种平滑的搜索空间仅改变搜索空间的距离,而定性地保持了原有搜索空间的拓扑结构信息,如图2所示。由图2可以看出,平滑空间的全局最小点就非常接近于原搜索空间的全局最小点,如果采用已平滑的搜索空间中的全局最小点作为原搜索空间的初始点,那么,找到原有搜索空间中的全局最小点的概率就大大地增加了。

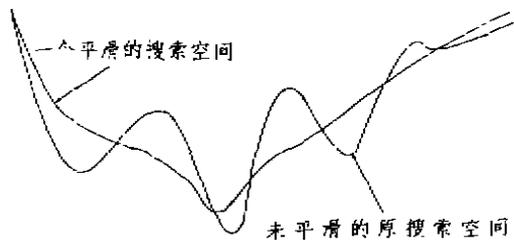


图1 平滑一个搜索空间的例子

在应用具体的平滑技术上存在许多不同的平滑方法,采用不同的平滑度量会产生不同的空间平滑状况.本文应用一个平滑系数 $\alpha$ 法,来衡量平滑的程度和相应的搜索空间的平滑度,通过改变平滑系数 $\alpha$ 的值,可以得到一系列的搜索空间的平滑程度,实现搜索空间从一个搜索空间到另一个搜索空间的变化,如图2所示.这里每个(上层)搜索空间是下层搜索空间的进一步平滑,其最优解是它紧下层的搜索空间中的一个初始解.

### 3 具有搜索空间平滑的遗传算法

#### 3.1 TSP问题的简介

旅行商问题(TSP)是一个广泛研究的组合优化问题之一.最简单的旅行商问题是指从某一个城市出发,访问 $n$ 个城市,最后返回到初始城市,寻找一条最短的路径,其中每一个城市只能访问一次,且城市间的距离是已知的.用数学方式表示如下:给定一个城市序列 $c_1, c_2, \dots, c_n$ 和相应城市间的距离 $d(c_i, c_j)$ ,寻找一条所有城市的一个访问排序,使得总距离之和最短.

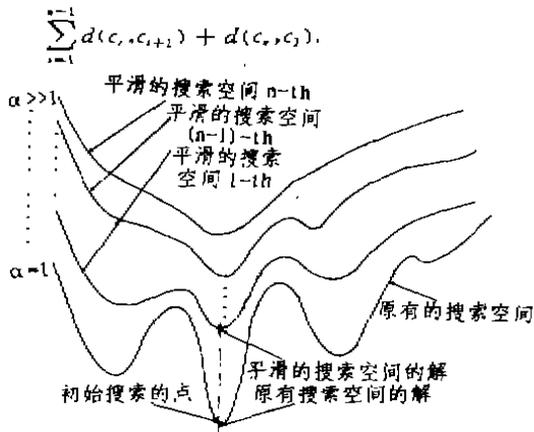


图2 空间平滑的一序列过程

TSP问题是一个NP难问题<sup>[8]</sup>,一个仅有12个城市的TSP问题具有39916800(=11!)条路径,每条可行的路径定义为解空间上的一个点,那么TSP问题的搜索空间就由所有可行路径组成.正是由于TSP问题的搜索空间具有许多局部极小点,导致了难以获得问题的精确解,尤其是城市规模较大的情况下.为此,本文提出一种具有搜索空间平滑的遗传算法,它能有效地获得TSP这个NP难问题的最优解或近优解.当然,具有搜索空间平滑的遗传算法可以推广应用到其他组合优化中的NP完全问题上.

#### 3.2 TSP问题的搜索空间平滑

设有一个给定的TSP问题, $n$ 是城市数目, $d_{ij}$ 是城市 $i$ 与城市 $j$ 间的距离( $i, j=1, 2, \dots, n$ ),不失一般性,假设所有的距离已经过归一化处理,使得 $0 \leq d_{ij} \leq 1$ ,而 $\bar{d}$ 是所有城市间的平均距离,即

$$\bar{d} = \frac{1}{n(n-1)} \sum_{i,j} d_{ij} \quad (1)$$

在平滑TSP问题的搜索空间时,采用一种平滑系数 $\alpha$ ,其具体方法如下方程所示:

$$d_{ij}(\alpha) = \begin{cases} \bar{d} - (d_{ij} - \bar{d})^\alpha, & d_{ij} \geq \bar{d} \\ \bar{d} - (\bar{d} - d_{ij})^\alpha, & d_{ij} < \bar{d} \end{cases} \quad (2)$$

其中, $\alpha \geq 1$ ,当 $\alpha$ 从一个太数逐步减小时,比如从10到1,那么就产生一系列简化的TSP问题, $\alpha$ 大时,产生的搜索空间就比较平滑地展开; $\alpha$ 小时,产生的搜索空间就有较多的凸凹不平的表面.在一系列TSP问题中,有两个极端情况:(1)如果 $\alpha \gg 1$ ,那么 $d_{ij}(\alpha) \rightarrow \bar{d}$ .这是任何一条可行路径都是全局最优解的TSP例子;(2)如果 $\alpha = 1$ ,那么 $d_{ij}(\alpha) = d_{ij}$ .这是原有的TSP问题.

按照方程(2)的平滑技术,通过 $\alpha$ 参数值的变化,可以得到一系列平滑的TSP问题,这样,就可以应用遗传算法进行求解.首先从一个近似典型的TSP问题(所有城市间的距离都一样)开始求解,即从一个相当“平滑”的搜索空间中找到全局最优解;然后,将该全局最优解作为下一个TSP问题的初始路径,即作为搜索空间较复杂( $\alpha$ 值减小的)的TSP问题的初始路径,再应用遗传算法进行求解;重复这个过程,直到参数 $\alpha = 1$ 为止,即最后求解原有的没有经过平滑技术处理的TSP问题.

#### 3.3 空间平滑的遗传算法的构成

本文以TSP问题为例,给出了具有空间平滑技术的遗传算法的算法结构,其具体流程如下:

算法结构:求解TSP的具有空间平滑的遗传算法()

```

begin
alpha = alpha_0
while (alpha >= 1) do
begin
for i := 1 to n do
for j := 1 to n do
d[i][j] := compute d_ij(alpha);
route_max := GA for TSP(d, route_max);
alpha := alpha - beta;
end;
end
    
```

现在,描述具有搜索空间平滑的遗传算法的核心部分—解TSP问题的遗传算法,其具体步骤如下:

步1 从 $M$ 个随机起点开始初始 $M$ 条路径, $M$ 为群体规模;

步2 选择交配对使在平均性能之上的个体得到更多的子代;

步3 交叉和变异,产生新子代;

步4 如果不满足终止准则,则回到步3;否则,停止进行。

这里,具体的遗传算法的各种算子设置为:

1)编码方式:采用排列编码方式。

2)交叉算子:采用修改的分类匹配算子,其工作原理:它从捐送串上随机选取一个子串,接着把包含在被选子串上的元从接受串上删除,然后再把这个子串拷贝到接受串上。图3是这个交叉算子的一个实例。交叉概率设定为0.80。

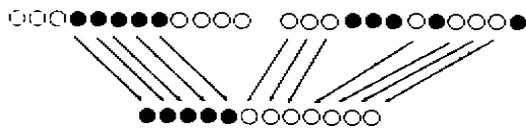


图3 修改的分类匹配的实例

3)变异算子:应用交换排序变异算子,在染色体串上随机选定两个位置,将这两个位置间的顺序逆排。变异概率设定为0.01。

4)选择机制:适应值函数  $f$  定义为:

$$f = f_{max} + (f_{max} - f_{min}) / M \quad (3)$$

其中,  $f_{max}$  和  $f_{min}$  分别表示当前代中染色体的最大距离和最小距离,  $M$  是群体规模,选择概率定义为:

$$P(i) = (f - f(i)) / (M \times f - \sum_{j=1}^M f(j)) \quad (4)$$

其中,  $f(i)$  是染色体  $i$  的路径距离。

5)群体规模定为  $M=30$ 。

6)结束准则:不同规模的 TSP 问题,采用不同的最大迭代代数。

## 4 仿真结果与分析

本节针对不同规模的 TSP 问题,通过大量的比较实验研究,来考察具有搜索空间平滑技术的遗传算法的搜索能力,以下给出几个典型的仿真结果。图4(略)是两种遗传算法在相同的参数设置下,针对 Oliver TSP 问题<sup>[10]</sup>的性能比较,一种是本文所提出的具有空间平滑技术的遗传算法(GAS),另一种是文[11]所提出的2-opt与遗传算法所构成的混合遗传算法(HGA)。图5(略)是 GAS 所求得的 Oliver TSP 问题的最优解的城市遍历图。图6(略)和图7(略)分别是48城市的 TSP<sup>[11]</sup>的寻优结果和最优解的城市遍历图,而图8(略)和图9(略)分别是 Grot442城市 TSP<sup>[11]</sup>的寻优结果和最优解的城市遍历图。从仿真结果不难看出,GAS 具有较强的搜索能力,它能够有效地跳出“局部”最优解的陷阱,获得问题的近优解。

## 参考文献

- 1 Chen Xiong, Xu Xinhe. New Heuristics with Search Space Smoothing for Rolling Planning. In: Proceeding of CWCICIA, Xian, 1997
- 2 Gu J. Local Search for Satisfiability Problem IEEE Trans. on System, Man and Cybernetics, 1993, 23(4): 1108~1129
- 3 Lin S. Computer Solutions of the Traveling Salesman Problem. Bell System Tech. J., 1965, 44: 2245~2269
- 4 Lin S, Kernughan, B W. An Effective Heuristic Algorithm for the Traveling Salesman Problem. Operations Research, 1973, 21: 498~516
- 5 Oliver I M D, et al. A study of permutation Crossover Operators on the TSP. In: Proc. 2nd Int. Conf. on Genetic algorithm and Their Applications. Cambridge, MA, 1987. 224~230
- 6 陈雄:[博士论文].东北大学,1998

(上接第18页)

ND-Polya 语言中的这种机制具有如下一些优点:(1)它提供了协变多态计算模型,语言的表达能力较强,符合人们的思维习惯,使用自然、灵活。(2)它保证了类型安全。(3)方法在类型格上的搜索由系统自动完成,程序员无额外负担。(4)保证了修改子类时,不须改动父类。(5)保证了语言的模块性。

## 参考文献

- 1 Boyland J, Castagna G. Type-Safe Compilation of Covariant Specialization: A Practical Case. Nov. 1995
- 2 Castagna G. Covariance and Contravariance: Conflict

without a Cause. ACM Transactions on Programming Languages and Systems, 17(3)

- 3 Bruce B, et al. Subtyping is not a good "Match" for object-oriented languages. Nov. 1996
- 4 Bruce K, et al. On Binary Methods. Theory and Practice of Object Systems 1(Number:3)
- 5 Boyland J, Castagna G. Parasitic Methods. An Implementation of Multi-Methods for Java.
- 6 谢高严,徐永森,詹志远,单成. TransFrame 语言中类型依赖关系及其实现. 微型计算机, 1997, 17(增刊)
- 7 McDermud J A. Software Engineer's Reference Book. Butterworth-Heinemann Ltd, 1991