

# 基于近邻策略的旅行商问题求解

汪彤, 李云强

WANG Tong, LI Yun-qiang

解放军信息工程大学 电子技术学院 应用数学系, 郑州 450004

Department of Applied Mathematics, Electronic Technique Institute, PLA Information Engineering University, Zhengzhou 450004, China

E-mail: wangtong810601@126.com

WANG Tong, LI Yun-qiang. Solving travelling salesman problem based on nearest neighbor strategy. *Computer Engineering and Applications*, 2009, 45(28): 67-68.

**Abstract:** According to the specific properties of TSP and the spirit of neighborhood search, the paper presents a new genetic algorithm based on the nearest neighbor strategy to solve TSP. First, calculate the nearest neighbor schemas according to the TSP and use the schemas to generate the initial population. Then introduce one of the schemas randomly into every generation. Simulation tests show that the new algorithm increases the convergence speed heavily and has better effect on the process of GAs. As the city number increases, its superiority appears more obviously.

**Key words:** nearest neighbor strategy; Genetic Algorithm (GA); Travelling Salesman Problem (TSP)

**摘要:** 根据 TSP 问题的特征信息并借鉴邻域搜索算法的有关思想, 提出了一种基于近邻策略的 TSP 问题求解算法, 该算法首先依据 TSP 问题的特殊性求出相应的近邻模式, 再将近邻模式用于初始种群的生成, 然后在进化过程中随机引入这类模式。该算法可以大大缩短遗传进程, 提高进化效率。通过仿真实验, 验证了该算法的有效性, 并且随着城市数目的增加其优越性更为明显。

**关键词:** 近邻策略; 遗传算法; 旅行商问题

DOI: 10.3778/j.issn.1002-8331.2009.28.019 文章编号: 1002-8331(2009)28-0067-02 文献标识码: A 中图分类号: TP301.6

## 1 引言

遗传算法 (Genetic Algorithm, GA) 是一种借鉴生物界中自然选择和进化机制发展起来的随机、自适应全局搜索算法, 由 Holland 教授于 1975 年提出。Goldberg 总结了一种统一的最基本的遗传算法的形式 SGA (Simple Genetic Algorithms)。SGA 在选定了染色体编码方案后, 随机生成初始群体, 再通过选择、交叉、变异三种遗传算子的作用对种群不断实行优化, 直至找到问题的最优解或近似最优解。遗传算法具有很强的全局搜索能力, 但是对于大型旅行商问题 (TSP), 它存在收敛速度过慢的缺点。

TSP 是一个著名的 NP 难的组合优化问题, 对称 TSP 问题可描述为: 给定  $n$  个城市和每两个城市之间的距离, 求一条经过所有城市一次且仅一次的长度最短的回路。TSP 问题的数学表述: 寻找一条遍历  $n$  个城市的最短路径, 或者说搜索到自然数集  $X = \{1, 2, 3, \dots, n\}$  ( $X$  中元素表示对  $n$  个城市的编号) 的一个全排列  $\pi(X) = \{v_1, v_2, \dots, v_n\}$ , 使  $T_d = \sum_{i=1}^{n-1} d(v_i, v_{i+1}) + d(v_1, v_n)$  取最小值, 式中的  $d(v_i, v_{i+1})$  表示城市  $v_i$  到城市  $v_{i+1}$  的距离。

对于  $n$  个城市的 TSP, 其所有的旅程路线组合数目是  $(n-1)!/2$ 。当城市数目  $n$  增加时, 可能的路径数将呈指数级增长。这大大降低了遗传算法的求解速度、求解精度。一些研究表明, 根据 TSP 问题的特点, 通过把遗传算法与局部启发式

算法结合起来能有效提高求解大型 TSP 问题的效率。该文借助这一思想, 提出了一种近邻策略, 取得了较好的结果, 验证了算法的有效性。

## 2 近邻策略

从实际经验出发, 在某一时刻, 当某人从一城市前往下一城市时, 通常选择距离该城市最近的一个城市作为目标城市, 若此城市已遍历, 则选择除此城市外距离所处城市最近的另一城市作为目标城市, 这就是近邻策略的思想。按照该思想, 可得一遍历序列, 虽不能保证所得到的路径为最短路径, 但常常含有和最短路径相匹配的某些路段, 这些路段能够为求得最短路径提供信息。通过对 TSP 问题大量最优解的分析表明, 最优解在很大程度上包含这样的片断或相似片断。为叙述方便, 把这样的近邻片断称之为近邻模式, 并把这样的近邻模式与最优解相应片断的相似程度称之为匹配度, 即

$$\text{匹配度} = \frac{\text{近邻模式与最优解相应片断相同的路径数}}{\text{该片断总路径数}}$$

### 2.1 近邻模式

首先给出构造近邻模式的具体方法和步骤:

(1) 确定阈值  $\delta$

通常选取两两城市距离的平均值作为阈值, 若城市间距离悬殊太大, 则排除距离悬殊太大的城市, 对余下的城市, 求两两

城市距离的平均值作为阈值。

## (2) 构造近邻模式

从某一城市出发,寻找距离该城市最近的城市作为目标城市,若寻找到的这个城市已被遍历,则寻找除这个城市之外,距离所处城市最近的另一城市作为目标城市,当所处城市与目标城市间距离小于  $\delta$  时,把该目标城市列入所要到达的下一城市,以此类推,继续寻找,当所处城市与目标城市间距离大于  $\delta$  时,则中断这一次的搜索,得到一个近邻模式。再将下一城市作为起点开始新一回的搜索。当所有的城市搜索完毕时,得到了一系列的近邻模式,其数目为所有城市数目  $n$ 。

## 2.2 复杂度分析

通过以上给出的方法和步骤来生成近邻模式,其时间复杂度仅为  $O(n^2)$ ,因此在算法运行过程中是切实可行的。并且在  $n$  较大的情况下,构造近邻模式的时间开销要远小于后续遗传进程的时间开销。

## 3 基于近邻策略的 TSP 问题求解

该文提出的这种策略首先将近邻模式用于初始种群的生成,然后在进化过程中随机引入这类模式。该策略可以有效地缩短进化过程,提高求解效率。

### 3.1 利用近邻模式生成初始种群

利用近邻模式生成初始种群的方法:

首先用 2.1 节中所述的近邻模式的生成方法来生成  $n$  个近邻模式,保持  $n$  个近邻模式的确定位不变,对非确定位,采用随机生成的方法,即可得到  $n$  个初始个体。为避免遗传算法陷入早熟收敛,用随机生成的方法再生成另外的  $N-n$  个初始个体(其中  $N$  为设定的初始种群规模),让这  $N$  个个体共同作为初始种群。

### 3.2 基于近邻策略的遗传算法及 TSP 求解步骤

下面给出了基于近邻策略遗传算法求解 TSP 问题的具体步骤,如下:

(1) 种群初始化:用 3.1 节方法生成初始种群。

(2) 遗传策略:①选择,采用轮盘赌及最优解保留策略,并在每一代随机选择一个近邻模式生成一个个体替换当前种群中的最差个体。②交叉,用较为常用的顺序交叉算子,该算子能较为完整地保留部分结点的相对访问顺序。③变异,采用双点逆转变异算子。

(3) 对算法终止条件进行判断:如果条件不满足,则转(2),否则,结束。

由于近邻模式与最优解相应的路径片断具有极高的匹配度,它能够在整个算法的搜索过程中具有良好的导向作用,避免了随机搜索的盲目性,大大提高了算法的速率及效率。

## 4 仿真实验及分析

为验证该算法的有效性,对算法进行实验。算法的求解效率主要涉及三个方面:(1)所求近似解精度。(2)算法的时间效率。(3)算法对问题规模的适应度。

TSPLIB 是一个研究 TSP 问题的常用数据集,选取 TSPLIB 中 16 个城市的数据集 *ulysess16*, 22 个城市的数据集 *ulysess22* 分别进行测试,已知最好解分别为 6 859 km, 7 013 km, 用 VC6.0 对算法进行实现,将文中的基于近邻策略的遗传算法(NSGA)与 SGA 进行对比,为公平比较,初始种群均为  $N$ ,交叉

概率为 0.6,变异概率为 0.1。

表 1 对 *ulysess16* 求解情况

	所求最优解/km	求解偏差/(%)	得到最优解的进化代数
SGA 求解情况	6 902	0.6	230
NSGA 求解情况	6 859	0	127

由表 1 可以看出,SGA 在第 230 代才得到近似最优解,而 NSGA 在第 127 代就得到了全局最优解,无论在求解精度还是在求解速度上都明显优于 SGA。

对该实验进行分析:

在 *ulysess16* 中,最优解路径为:

0-13-12-11-6-5-14-4-10-8-9-15-2-1-3-7-0

取阈值  $\delta=800$ ,得到的近邻模式分别为:

0-7-15-12-11-13-6-5-14-4;

1-2-0-7-15-12-11-13-6-5-14-4;

2-1-3-7-15-0-12-11-13-6-5-14-4;

3-7-0-15-12-11-13-6-5-14-4;

4-14-13-12-11-6-5;

5-6-11-12-13-14-4;

6-5-13-12-11-14;

7-0-15-12-11-13-6-5-9-8-14-4;

8-9-6-5-13-12-11-14-4;

9-6-5-13-12-11-14-4;

11-12-13-6-5-9-8-14-4;

12-11-13-6-5-14-4;

13-12-11-6-5-14-4;

14-13-12-11-6-5-9-8;

15-0-7-3-1-2

因为城市 10 与其他 15 个城市的距离都大于阈值  $\delta$ ,所以城市 10 不生成近邻模式,各近邻模式与最优解片断的匹配度分别为:5/9, 6/11, 7/12, 6/10, 5/6, 5/6, 3/5, 5/11, 5/8, 4/7, 5/8, 4/6, 6/6, 5/7, 4/5, 匹配度的平均值为 66.72%,并且在最优解的 16 条路径中有 11 条路径已被遍历。这就充分说明了用 NSGA 能快速有效求解的原因。

表 2 对 *ulysess22* 求解情况

	所求最优解/km	求解偏差/(%)	得到最优解的进化代数
SGA 求解情况	7 260	3.5	397
NSGA 求解情况	7 013	0	236

由表 2 可以看出,随着城市数目的增加,NSGA 相对 SGA 的求解优越性就表现地更为明显,SGA 在第 397 代也只得到近似最优解,而 NSGA 在第 236 代就获得了全局最优解。

对该实验进行分析:

在 *ulysess22* 中,最优解路径为:

0-13-12-11-6-5-14-4-10-8-9-18-19-20-15-2-1-16-21-3-17-7-0

取阈值  $\delta=800$ ,得到的近邻模式分别为:

0-7-21-3-17-16-1-2-15-12-11-13-6-5-14-4;

1-2-16-21-3-17-7-0-15-12-11-13-6-5-14-4;

2-1-16-21-3-17-7-0-15-12-11-13-6-5-14-4;

3-17-21-16-1-2-0-7-15-12-11-13-6-5-14-4;

4-14-13-12-11-6-5-18-19-20-9-8;

(下转 71 页)