

# 基于混合遗传模拟退火算法求解 TSP 问题

杜宗宗, 刘国栋

DU Zong-zong, LIU Guo-dong

江南大学 通信与控制工程学院, 江苏 无锡 214122

College of Communication and Control Engineering, Jiangnan University, Wuxi, Jiangsu 214122, China

E-mail: duzongzong@163.com

DU Zong-zong, LIU Guo-dong. Solution of TSP problem based on hybrid genetic simulated annealing algorithm. Computer Engineering and Applications, 2010, 46(29): 40-42.

**Abstract:** TSP is a classical NP-hard combinatorial optimization problem. Genetic algorithm is a method for solving this problem. But it is hard for genetic algorithm to find global optimization quickly and prevent premature convergence. This paper, in order to solve the problem, considers the characteristic of TSP, and puts forward a genetic simulated annealing algorithm, a hybrid of genetic and simulated annealing algorithm. In order to solve the inconsistency between diversity and convergent speed, this paper also adopts part greedy method to produce original population. The original population produced by this method is superior to the randomly produced original population. The simulation results demonstrate that, the proposed algorithm achieves considerable improvements, with respect to the basic genetic algorithm, in convergence speed, search quality and optimal solution output rate.

**Key words:** hybrid genetic algorithm; simulated annealing algorithm; traveling salesman problem

**摘 要:** TSP 问题是典型的 NP-hard 组合优化问题, 遗传算法是求解此类问题的一种方法, 但它存在如何较快地找到全局最优解, 并防止“早熟”收敛的问题。针对上述问题并结合 TSP 问题的特点, 提出将遗传算法与模拟退火算法相结合形成遗传模拟退火算法。为了解决群体的多样性和收敛速度的矛盾, 采用了部分近邻法来生成初始种群, 生成的初始种群优于随机产生初始种群。仿真实验结果证明, 该算法相对于基本遗传算法的收敛速度、搜索质量和最优解输出概率方面有了明显的提高。

**关键词:** 混合遗传算法; 模拟退火算法; 旅行商问题

DOI: 10.3778/j.issn.1002-8331.2010.29.011 文章编号: 1002-8331(2010)29-0040-03 文献标识码: A 中图分类号: TP301.6

## 1 引言

旅行商问题 (Traveling Salesman Problem, TSP) 可描述为: 已知  $N$  个城市之间的相互距离, 现有一推销员必须遍访这  $N$  个城市, 并且每个城市只能访问一次, 最后又必须返回出发城市。如何安排他对这些城市的访问次序, 使其旅行路线总长度最短。旅行商问题是一个典型的组合优化路径规划问题, 其可能的路径数目是与城市数目  $N$  呈指数型增长的, 一般很难精确地求出其最优解, 因而寻找其有效的近似求解算法具有重要的理论意义。另一方面, 很多实际应用问题, 经过简化处理后, 均可化为旅行商问题, 因而对旅行商问题求解方法的研究具有重要的应用价值。

近年来, 遗传算法由于它强大的优化能力已经被广泛应用于 TSP 问题的求解中<sup>[1-3]</sup>。遗传算法就其本质而言是处理复杂问题的一种鲁棒性强的启发式随机搜索方法。它是求解 TSP 问题的一种理想方法。遗传算法中一个较难解决的问题是如何较快地找到最优解、保证全局收敛性、提高局部寻优能力并防止“早熟”收敛问题。

将传统遗传算法<sup>[4]</sup>进行改进, 运用一种基于混合遗传模拟退火算法的路径规划方法可以有效解决上述问题。将基于启发式知识的遗传算子的设计方法、运用模拟退火算法对遗传算法进行优化的方法、采用了部分近邻法来生成初始种群应用于 TSP 问题的求解中。仿真结果表明使用该方法可以达到满意的规划效果和收敛速度, 说明混合遗传模拟退火算法的环境适应性较强。

## 2 混合遗传模拟退火算法

### 2.1 混合遗传模拟退火算法的基本思想

模拟退火算法是将退火思想引入组合优化领域, 提出的一种大规模组合优化问题的有效近似算法, 该算法模仿固体物质的退火过程。众所周知, 高温物体降温时其内能随之下降, 如果降温过程充分缓慢, 则在降温过程中物质体系始终处于平衡状态, 从而降到某一低温时其内能为最小。模仿退火过程的寻优方法称为模拟退火算法。

遗传算法虽然能从概率的意义上以随机的方式寻求到全

**作者简介:** 杜宗宗 (1983-), 男, 硕士研究生, 研究方向为机器人路径规划; 刘国栋 (1950-), 男, 教授, 博导, 研究方向为机器人控制、智能控制等。

**收稿日期:** 2009-03-25 **修回日期:** 2009-05-25

局最优解,但它在实际应用过程中也可能会产生一些问题。这些问题中最主要的是局部寻优能力差、收敛速度较慢、易陷入局部极值点等。而另一方面,模拟退火算法却具有较强的局部搜索和摆脱局部最优解的能力。所以使用遗传算法与模拟退火算法相结合的方法,是解决上述问题的有效途径。

由于遗传算法初始种群直接影响着收敛速度的快慢,所以在不影响种群多样性的前提下,提出部分近邻法生成初始种群。称这种将遗传算法与模拟退火算法相结合并采用部分近邻法生成初始种群的算法称为混合遗传模拟退火算法。

## 2.2 编码方式

求解TSP问题的各种编码中,多数以遍历城市的次序排列进行编码的,即整数编码<sup>[5-7]</sup>。这种方法简单、自然、容易理解。如城市编号为[3, 2, 5, 4, 7, 1, 6, 9, 8]表示从3号城市出发,经过2、5、4、7、1、6、9、8后,最终又回到了城市3。随机生成的个体都可能是问题的一个解。

## 2.3 适应度函数

TSP问题的一个个体  $X = [v_1, v_2, \dots, v_n]$  的适应度定义为个体所代表路径确切距离的倒数,也即

$$fit = \frac{1}{\sum_{i=1}^{n-1} d(v_i, v_{i+1}) + d(v_n, v_1)}$$

其中  $v_i$  为路径第  $i$  个顶点(城市),  $d(v_i, v_j)$  为两点间的距离,而且  $d(v_i, v_j) = d(v_j, v_i)$ , 在后面的运行过程中,算法试图使适应度函数最大化并认为使得该函数取得较大值的解为较优解。

## 2.4 初始种群生成方法

产生初始种群的方法有两种,一种是完全随机的方法产生的,它适用于对问题的解没有任何先验知识的情况;第二种是近邻算法,设有  $n$  个城市的TSP问题,城市集合  $X = [v_1, v_2, \dots, v_n]$ ,  $d(v_i, v_{i+1})$  表示城市  $v_1$  到城市  $v_{i+1}$  的距离,对所有的城市  $v_1$ , 将其他的  $n-1$  个城市按照与  $v_1$  的距离排序得到这些城市对  $v_1$  的近邻排序名次;把每一个城市的附近城市按邻近程度大小从大到小排列好放在一个二维数组中,命名为  $neighbor[n][n-1]$ 。从任意城市出发,每次选择尚未经过的城市中对当前城市的近邻排名最前者作为下一个城市,直至遍历所有  $n$  个城市,再返回出发点,这样就构成一个初始个体。以每一个城市分别作为出发点,用这样的方法可以生成  $n$  个初始个体。

用近邻算法生成的初始种群具有相当好的品质,但生成的初始种群多样性很小,从这样的初始种群开始进化,不一定能够收敛到全局最优解。

设计了混合算法生成初始群体即初始群体的一半个体是随机产生,另一半个体是由先验知识近邻算法生成。这样可结合具体问题,从问题中获取更多信息,从而使得算法更快更好找到最优解。即由近邻算法产生  $M/2$  个初始个体,然后再随机产生  $M/2$  个初始个体,得到一个种群规模为  $M$  的初始群体。

## 2.5 模拟退火算法的参数设置

### 2.5.1 初始温度的选取

通常初温  $T_0$  需要足够高,以确保算法从一开始就具有较强的遍历性,避免陷入局部最优解;然而初温过高,又会大大增加运行时间,降低运行效率<sup>[8]</sup>。初温的确定选择  $T_0 = k\delta$  的形

式,其中  $k$  为充分大的数,可以选  $k=10, 20, 100$  等试验值;  $\delta = fit_{\max} - fit_{\min}$ ,  $fit_{\max}$  为初始种群中最大的目标适应度值,  $fit_{\min}$  为初始种群中最小的目标适应度值。

### 2.5.2 温度更新函数的选取

温度更新函数,即温度的下降方法,用于在外循环中修改温度值。遗传算法的理论一般要求温度下降到零,整个系统以概率1收敛到全局最优解<sup>[9]</sup>。本文采取的温度更新函数为:  $T(t) = T_0 / (1 + \alpha t)$ 。这种温度下降的特点是开始时,温度下降是比较快的,越往后,降温速率较小,因此寻优重点在低温区。式中  $\alpha$  可以改善退火曲线形态。

### 2.5.3 算法的终止准则

采用零度法,即遗传模拟退火的最终温度为零,因而最为简单的原则是:给定一个比较小的正实数  $\varepsilon$ , 当温度  $T \leq \varepsilon$  时,算法停止。

## 2.6 遗传算子

### 2.6.1 选择算子

选择算子采用确定式采样选择的基本思想是按照一种确定的方式来进行选择操作<sup>[10]</sup>,假设种群规模为  $M$ , 其具体的操作过程如下:

步骤1 计算种群中各个个体在下一代种群中的期望生存数目

$$N_i = \frac{Mfit_i}{\sum_{i=1}^M fit_i} \quad (i=1, 2, \dots, M)$$

其中  $fit_i$  为每个个体的适应度。

步骤2 用  $N_i$  的整数部分  $[N_i]$  确定各条对应个体在下一代种群中的生存数目。其中  $[x]$  表示取不大于  $x$  的最大整数,由该步共可确定出下一代中的  $\sum_{i=1}^M [N_i]$  个个体。

步骤3 按照  $N_i$  的小数部分对个体进行降序,顺序取前  $M - \sum_{i=1}^M [N_i]$  个个体加入到下一代群体中。至此可完全确定出下一代种群中的  $M$  个个体。

### 2.6.2 交叉算子

应用遗传算法求解TSP,通常把染色体表示成所有城市的一个排列在这种表示方法下,传统的杂交算子产生的向量很可能不是从头到尾的一个排列,即产生了无意义的路径。为了排除这一情况,必须要定义保持编码有效性的杂交算子。

采用贪婪交叉算子,这种交叉算子可以使交叉后的子代在保证可行性的同时,并很好地继承了父代的优秀基因,例如:交叉的两个个体为:

[3, 2, 5, 4, 7, 1, 6, 9, 8]

[4, 5, 2, 9, 3, 6, 8, 1, 7]

先任意选定两个个体中第一个城市作为起始城市。假如选择第一个个体中的城市3作为子代个体的起始城市。

根据9个城市之间  $9 \times 9$  的距离矩阵,比较两个个体中3到下一个城市的距离。即比较3与2和3与6的距离。不妨假设3与2的距离小于3与6的距离,那么子代染色体的第二个城市即为2。

同理,再根据距离矩阵比较2在两个个体中到下一个城市的距离,距离较短的下一个城市即为第三个城市。

如此反复直到个体结束,得出一个子代个体。再以另一个个体的第一个基因作为起始基因,按照同样的办法得出另外一个子代个体。

这种算子可以极大地加快算法的收敛速度,减小了交换操作的盲目性,提高了算法的执行效率。

### 2.6.3 变异算子

采用启发式变异算子。假设变异的个体为:[1,2,3,4,5,6,7,8,9]。

随机选择3个点,设2,6,8任意交换位置可以得到5个不同个体:

A1:[1,2,3,4,5,8,7,6,9]

A2:[1,6,3,4,5,2,7,8,9]

A3:[1,8,3,4,5,6,7,2,9]

A4:[1,6,3,4,5,8,7,2,9]

A5:[1,8,3,4,5,2,7,6,9]

从中选择最好的作为新的个体。

### 2.6.4 遗传算子的参数选择

因交叉操作是其主要操作,取  $0.7 < p_c < 0.9$ , 变异概率取  $0.01 < p_m < 0.2$ 。

### 2.6.5 遗传算法终止代数

大部分文献中仍需要指定最大终止进化代数,这使得算法的鲁棒性质受到限制。若最大终止进化代数设定过小,算法尚未收敛便已经终止,所得结果的品质就比较差;若最大终止进化代数设定过大,算法已经收敛后仍要无意义地计算很多代,浪费计算时间。为此,取消最大终止进化代数这一控制参数,代之以最大容许停滞代数  $Q$ ,当连续  $Q$  次迭代的最优解没有改进时,迭代过程中止。

## 2.7 模拟退火算法操作流程

操作流程采取内外双层循环,模拟退火部分采用 Metropolis 机制来接受和舍弃新解。

步骤1 设置种群规模大小  $M$ ; 最大容许停滞代数  $Q$ ; 输入城市的坐标数据。遗传代数计数器初始化:  $gen = 0$ ; 设置初始温度参数  $T = T_0$ ; 生成初始种群  $P(gen)$ , 种群中的每一个个体代表一条旅行商所走的路径; 初始化停滞代数计数器  $tolerance = 0$ 。

步骤2 用近邻算法生成  $M/2$  个初始个体,再用随机法生成另外  $M/2$  个初始个体。评价  $P(gen)$  中各条路径的适应值:  $fit_1, fit_2, \dots, fit_M$ , 并计算城市间的距离矩阵。

步骤3 对现有种群实施如下操作,直到产生出下一代新的种群:

(1)由2.6.1节中定义的选择算子从父代路径中进行子代路径选择操作:  $P_s(gen) \leftarrow selection[P(gen)]$ 。

(2)由2.6.2节中定义的交叉算子进行子代路径交叉操作: 由  $P_s(gen)$  中第  $i$  个个体  $P_{si}(gen)$  和第  $j$  个个体  $P_{sj}(gen)$  交叉得到新个体  $P_{ci}(gen)$  和  $P_{cj}(gen)$ , 并计算  $P_{ci}(gen)$  和  $P_{cj}(gen)$  的适应度函数值。

(3)根据下面两个公式的接受概率  $p_i$  和  $p_j$  来确定是接受  $P_{ci}(gen)$ 、 $P_{cj}(gen)$  还是拒绝  $P_{ci}(gen)$ 、 $P_{cj}(gen)$ , 最后得到交叉退火之后的新种群  $P_c(gen)$ 。

$$p_i = \begin{cases} 1, & fit(ci) \leq fit(si) \\ \exp\left(\frac{fit(si) - fit(ci)}{T}\right), & fit(ci) > fit(si) \end{cases}$$

$$p_j = \begin{cases} 1, & fit(cj) \leq fit(sj) \\ \exp\left(\frac{fit(sj) - fit(cj)}{T}\right), & fit(cj) > fit(sj) \end{cases}$$

(4)由2.6.3节中定义的变异算子进行子代路径变异操作,对第  $i$  个个体变异得到  $P_{mi}(gen)$  新个体,再以下面公式的概率接受  $P_{mi}(gen)$  个体。最后得到变异退火之后的新种群  $P_m(gen)$ 。

$$p(i \Rightarrow i') = \begin{cases} 1, & fit(i') \leq fit(i) \\ \exp\left(\frac{fit(i) - fit(i')}{T}\right), & fit(i') > fit(i) \end{cases}$$

其中,  $i$  为变异前状态,  $i'$  为变异后状态。

(5)比较相继两代的最优个体有无改善,若有改善,将停滞代数计数器  $tolerance$  重置为0,转到步骤1;若无改善则,  $tolerance$  置1,  $P(gen+1) = P_m(gen)$ ,  $gen = gen + 1$ , 转到步骤4。

步骤4 温度终止条件判断。若不满足温度终止条件,则:  $P(gen) = P_m(gen)$ ,  $gen = 0$ ; 按温度更新函数来更新温度参数  $T$ , 转向步骤3;若满足终止条件,则转向步骤5。

步骤5 操作流程结束。

## 3 仿真实验结果及分析

通过模拟30个城市地图情况下的计算机仿真来证明改进算法的性能。在仿真中,算法相关参数设定如下:种群规模  $M$  为100,染色体交叉概率  $p_c$  为0.7,变异概率  $p_m$  为0.02,最大容许停滞代数  $Q$  为50,起始温度参数  $k$  为100,温度更新参数  $\alpha$  为0.1,温度终止参数  $\varepsilon$  为0.1。

针对30个城市实现TSP问题,假设30个城市的坐标为: {87, 7; 95, 25; 83, 46; 71, 44; 64, 60; 68, 58; 83, 69; 87, 76; 74, 87; 71, 71; 58, 69; 54, 62; 51, 67; 31, 84; 41, 94; 5, 91; 7, 64; 22, 60; 25, 62; 18, 54; 4, 50; 13, 40; 18, 40; 24, 42; 25, 38; 41, 26; 45, 21; 44, 35; 58, 15; 62, 32}, 运用Matlab 7.0 仿真多次,种群进化到500代、1500代时的典型结果分别如图1与图2所示。

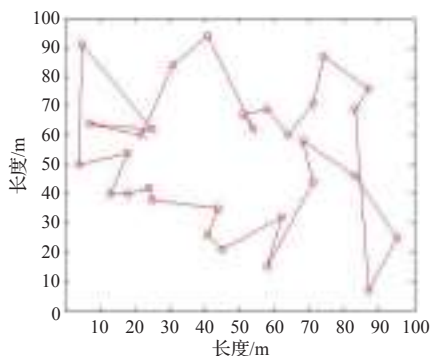


图1 进化代数为500时的路径

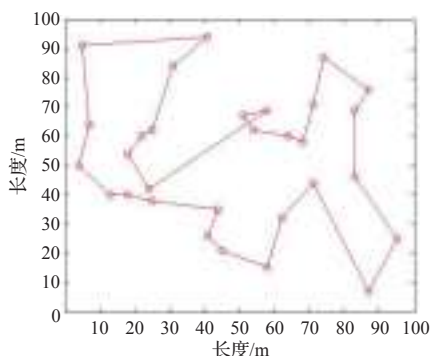


图2 进化代数为1500时的路径 (下转46页)