

多旅行商问题遗传算法求解及其改进

熊翠, 吴慧萍, 李波

华中师范大学, 数学与统计学学院, 湖北, 武汉, 430079

摘要: 旅行商问题是一个著名的组合优化问题, 多旅行商回路是旅行商问题的扩展, 本文综合均衡度, 提出应用遗传算法求解多旅行商问题的算法设计, 并将其与模拟退火算法比较与结合, 有效提高了运算的速度和效率。

关键词: 多旅行商问题; 遗传算法; 均衡度; 模拟退火算法; 模拟退火遗传算法;

1. 引言

旅行商问题(Traveling Salesman Problem, 简称 TSP)^[1] 是一个著名的组合优化问题: 给定 n 个城市, 有一个旅行商从某一城市出发, 访问每个城市各一次后再回到原出发城市, 要求找出的巡回路径最短。如果用图论来描述, 那就是已知带权图 $G = (C, L)$, 寻出总权值最小的 Hamilton 圈。

多旅行商回路(MTSP) 是旅行商问题(TSP) 的扩展。MTSP 是指给出 N 个城市的集合, M 个推销商从目标城市出发, 分别走一条旅行路线, 使得每个城市有且仅有一个推销商走过, 最后回到原来的出发城市, 且总旅程最短。有关 MTSP 问题的研究在现实问题中有很大的使用价值^[2]。诸如: 交通运输、管道铺设、路线的选择、计算机网络的拓扑设计、邮递员送信等, 都可抽象成 TSP 或 MTSP 问题。

遗传算法 (Genetic Algorithm) 是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型, 是一种通过模拟自然进化过程搜索最优解的方法, 它最初由美国 Michigan 大学 J.Holland 教授于 1975 年首先提出来的, 是求解复杂的组合优化问题的有效方法, 遗传算法在求解 TSP 和 MTSP 问题中得到了广泛的应用。

而在求解多旅行商问题的时候, 常常只能保证总路程最短, 很少达到使每个推销商的路程均衡,

导致结果不合理, 推销商任务不平均, 不能合理利用资源。

2. 遗传算法模型

2.1 遗传算法简介

遗传算法是一种以自然选择和遗传理论为基础, 将生物进化过程中适者生存规则与同一群染色体的随机信息变换机制相结合的搜索算法。它通过给解向量编码、形成初始种群, 然后用变异、交叉重组、自然选择等算子, 进行并行迭代, 求得优化解。由于它采用随机运算, 对搜索空间无特殊要求, 无需求导, 具有运算简单、收敛速度快等优点。

其主要步骤为:

- 1) 先初始产生 l 个编码个体;
- 2) 计算每个个体的目标函数;
- 3) 利用轮盘法选出 N 个个体作为下一代变异对象;
- 4) 对选出的个体按概率循环变异, 交叉选择, 产生新一代群体;
- 5) 比较现有记录, 如果比现有记录更优, 记录下群体中最优的 l 个个体;

重复第二步, 这样遗传到足够多代后, 会收敛到一个近似最优解, 算法结束。

根据上述步骤, 我们可结合多旅行商问题的特点, 一一进行研究。

2.2 遗传个体的编码设计

2.2.1 旅行商问题的编码设计

在遗传算法中, 遗传个体的设计很重要, 需要体现个体的遗传特性, 在这儿设计编码如下:

以十个城市三个推销商为例, 设 A 为是个城市的最短距离矩阵, 用 A 矩阵对应的行列数 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 分别表示十个城市, 其中点 1 表示旅行商的出发城市。

在旅行商问题中, 我们将这十个城市, 保持起点在第一位, 按次序排列表示一个结果(即染色体): 如: 1, 2, 6, 3, 5, 4, 7, 8, 9, 10

路径表示:

1—2—6—3—5—4—7—8—9—10—1

2.2.2 多旅行商问题的编码设计

不妨假定此多旅行商问题存在着三条路线, 同样以十个城市为例, 我们需要插入两个虚点 11、12 用来表示路线的起点, 并形成新的编码, 可用来表示多旅行商问题的染色体: 如: 1, 6, 5, 2, 4, 11, 7, 8, 3, 12, 9, 10

三个路径表示为:

1—6—5—2—4—1;

1—7—8—3—1;

1—9—10—1。

设置节点 1, 11, 12 之间的距离为无穷大 (即永远不能达到), 到其它各点距离与 1 点一致, 得到新的最短距离矩阵 $D = (d_{ij})_{12 \times 12}$, 对应的行列数分别表示 12 个节点。并且这样遗传算法选择变异就不会得到 1—1—1 的路径。

2.3 遗传算法的目标函数

遗传算法涉及到个体选择, 淘汰适应性差的个体, 因此, 需要建立目标函数来评价个体的好坏。

对于目标函数的确定, 可以从四个方面进行考虑。

1) 总路程最短

即为三条路线距离的总和。即为 $S = \sum_{i=1}^3 l_i$,

其中 l_i 表示第 i 条线路的长度。

2) 均衡度

在保证了总路程最短的情况下, 很有可能会出现其中某一条路线过长、负载过重, 而某一条路线过短, 甚至完全没有经过任何点就直接回到原点, 这样的多旅行商问题就没有任何意义了。所以我们需要引用均衡度的概念。

均衡度的意义在于让每一条路线尽量保持均衡, 故只要能满足这一条件的约束函数都能定义为均衡度。

一般地, 传统的均衡度定义为

$$J_c = \frac{\max_{i,j \in \{1,2,3\}, i \neq j} \{l_i - l_j\}}{\max_{i \in \{1,2,3\}} \{l_i\}}$$

此时, J_c 为定义在 0 到 1 之间的数, 其值越接近 0 时表示每条线路越接近。

此种定义有以下几点好处: i) 已经归一化处理; ii) 定义明确清晰。但此定义有一个较大的问题在于

运算量大, 表示复杂。

为了简化计算, 可以定义均衡度为

$$J = \max(l_1, l_2, l_3)$$

即让三条线中最长的那条尽可能短。此定义方式较传统的定义方式有以下几个特点:

i) 量纲问题易处理

对于这种定义的均衡度, 它较传统的定义方式在量纲的问题上更接近总路程, 处理起来也比较简单。

ii) 符合所有定义要求

实际上, 只要能够使得最长的路线都能保持最短, 那么三条线路的长度就相差就很小了。因此它也是符合均衡度的定义的。

iii) 定义非常简单, 运算量大大降低。

3) 综合考虑总路程和均衡度下的目标函数

要想综合考虑总路程和均衡度, 首先我们需要统一量纲。我们发现, S 与 $3J$ 的量纲是相同的。

我们令 $J' = 3J$

于是我们可以得到总目标函数为:

$$\begin{aligned} Z &= aS + bJ', \\ a + b &= 1, \end{aligned}$$

其中, a, b 分别为 S 和 J' 的权重,

当 $a > b$ 时, 即总路程的权重大, 此时总路程占主要因素。

当 $a < b$ 时, 即均衡度的权重大, 此时均衡度为主要因素。

为了是总路程及均衡度对总目标的影响程度相同, 我们可将其权重设置为 1: 1, 于是得到目标函数:

$$Z = \frac{1}{2}S + \frac{1}{2}J', (J' = 3J)$$

4) l_i 的确定

通过遗传算法的编码可得到矩阵组

$$\{X^m = (x_{ij}^m)_{12 \times 12}, m = 1, 2, 3\}$$

其中,

$$x_{ij}^m = \begin{cases} 1 & \text{第 } m \text{ 个旅行商会从 } i \text{ 出发到达 } j \\ 0 & \text{否则} \end{cases}$$

$$\text{显然, } l_1 = \sum_{i,j=1}^{12} x_{ij}^1 * d_{ij}, \quad l_2 = \sum_{i,j=1}^{12} x_{ij}^2 * d_{ij},$$

$$l_3 = \sum_{i,j=1}^{12} x_{ij}^3 * d_{ij}$$

2.4 轮盘选择

对于下代种群的选择,先计算每个个体被选择的概率,假设每代种群有 N 个,设第 i 个个体的目标函数值为 Z_i 。要求目标函数值越小,被选中的概率越大。

常出现的处理办法是,引入适应度函数的定义,

$S_i = 1/Z_i$, 即为目标函数值的倒数。

当然可以由此定义第 i 个个体被选中概率为:

$$P_i^1 = \frac{S_i}{\sum_{i=1}^N S_i} = \frac{1/Z_i}{\sum_{i=1}^N 1/Z_i}$$

此时, P_i^1 显然满足目标值越小,被选概率越大的条件。

被选概率的定义并不是唯一的。例如:

$$P_i^2 = \frac{\sum_{j=1}^N Z_j - Z_i}{\sum_{i=1}^N (\sum_{j=1}^N Z_j - Z_i)}$$

也同样满足要求,那么究竟要如何判别一个定义的好坏呢?

在所定义的概率中,若 N 个个体被选概率相差不大,则循环的次数将会很多,而被选概率相差很大,循环次数则会下降,也会迅速收敛。那么我们所想要得到的是 $P_{\max} - P_{\min}$ 尽可能大,于是可得比较如下:

$$P_{\max}^1 - P_{\min}^1 = \frac{Z_{\min} - Z_{\max}}{Z_{\min} Z_{\max} \sum_{i=1}^N 1/Z_i} \quad (I)$$

$$P_{\max}^2 - P_{\min}^2 = \frac{Z_{\min} - Z_{\max}}{\sum_{i=1}^N \left(\sum_{j=1}^N Z_j - Z_i \right)} \quad (II)$$

显然, (I) 式要比 (II) 式值更大也更优一些,于是可取 $P_i = P_i^1$, 而对于其他的定义方式,也可以通过进行这样的对比,进行逐一分析。

然后对各个概率求累积概率:

$$Q_i = \sum_{j=1}^i P_j$$

最后用轮盘选择法进行选择,每一轮产生一个 0-1 的随机数,将随机数作为指针选择个体,选出

下一代的 N 个个体。

2.5 部分匹配交叉与交换变异

遗传算法每代都要进行交叉及变异,以便产生新的更优的个体,在遗传算法中起着核心地位。这里针对 MTSP 采用部分匹配交叉与交换变异。

2.5.1 部分匹配交叉算子

部分匹配交叉操作要求随机选取两个交叉点,以便确定一个匹配段,根据两个父个体中两个交叉点之间的中间段给出的映射关系生成两个子个体。例如:

对于下面两个父个体的表示,随机地选择两个交叉点"|"。

$$P_1: (1\ 2\ 3 | 4\ 5\ 6\ 7 | 8\ 9)$$

$$P_2: (4\ 5\ 2 | 1\ 8\ 7\ 6 | 9\ 3)$$

首先,两个交叉点之间的中间段交换,得到

$$Q_1: (x\ x\ x | 1\ 8\ 7\ 6 | x\ x)$$

$$Q_2: (x\ x\ x | 4\ 5\ 6\ 7 | x\ x)$$

其中, x 表示暂未定义码,得到中间段映射关系为:

$$1 \leftrightarrow 4, 8 \leftrightarrow 5, 7 \leftrightarrow 6, 6 \leftrightarrow 7$$

然后将未选定的城市码 2, 3, 9 保留,于是得到

$$Q_1: (x\ 2\ 3 | 1\ 8\ 7\ 6 | x\ 9)$$

$$Q_2: (x\ x\ 2 | 4\ 5\ 6\ 7 | 9\ 3)$$

而对于, Q_1 中第一个 x , 可以由 $1 \leftrightarrow 4$, 可得为 4。其他也依次可得,结果如下:

$$Q_1: (4\ 2\ 3 | 1\ 8\ 7\ 6 | 5\ 9)$$

$$Q_2: (1\ 8\ 2 | 4\ 5\ 6\ 7 | 9\ 3)$$

2.5.2 交换变异

交换变异,即交换两个随机位置上的基因。常用的是反转变异,即为在染色体上随机的选择两个点,将两个点间的子串完全反转。

例如: 1, 6, 5, 2, 4, 11, 7, 8, 3, 12, 9, 10

将第 2 到第 7 节点反序得到新个体:

1, 7, 11, 4, 2, 5, 6, 8, 3, 12, 9, 10

3. 实例分析

上一节介绍了,遗传算法求解 MTSP 问题的方

法, 现在来实例求解一个问题来验证算法的有效性。

我们在网络上查到中国 31 个主要城市的坐标^[3], 城市与城市之间仅考虑其直线距离。数据如下:

表 1 31 个城市坐标值

城市	x 轴	y 轴	城市	x 轴	y 轴
武汉	3507	1633	呼和浩特	3238	2771
长沙	3394	1357	沈阳	4196	2996
郑州	3429	2092	长春	4312	3210
合肥	3780	1788	哈尔滨	4386	3430
西安	3007	2030	石家庄	3488	2465
成都	2545	1643	北京	3639	2685
太原	3326	2444	南昌	3676	1422
西宁	2381	2324	天津	3712	2601
兰州	2562	2244	济南	3715	2322
银川	2788	2509	南京	3918	1821
南宁	2935	760	上海	4177	1756
拉萨	1304	1688	杭州	4061	1630
乌鲁					
木齐	1332	3305	福州	4029	1162
贵阳	2778	1174	广州	3439	799
昆明	2370	1025	台北	4263	1069
海口	3140	450			

同时只研究三个旅行商的情况, 令种群个体数为 50, 遗传 1000 代, 交叉概率为 0.8, 而变异概率为 0.15。

用 C 语言编程, 运行 10 次, 取最优值, 同样也计算只考虑均衡度和只考虑总路程, 得到对比表如下:

表 2 遗传算法结果

	l_1	l_2	l_3	均衡度	总路程
只考虑均衡度	8736	8812	8583	8812	26131
只考虑总路程	633	14669	2376	14669	17688
都考虑	6106	6477	7643	7643	20225

这样我们可以看到, 两者都考虑, 比只考虑均衡度的总路程有所减少, 且保证了各个旅行商路程

的均衡度, 而只考虑总路程的结果, 均衡度很不好, 结果极不合理。由此, 算法得到了理想的结果。

4. 与其他算法的比较

4.1 模拟退火算法简介

模拟退火算法(Simulated Annealing, 简称 SA)是基于 Monte Carlo 迭代求解法的一种启发式随机搜索算法, SA 以一定的概率选择邻域中目标值较小的状态。它模拟固体物质退火过程的热平衡问题与随机搜索寻优问题的相似性来达到寻找全局最优或近似全局最优的目的。在搜索最优解的过程中, 模拟退火法除了可以接受优化解外, 还有一个随机接受准则(Metropolis 准则)有限度地接受恶化解, 并且接受恶化解的概率慢慢趋向于 0, 这使得算法有可能从局部极值区域中跳出, 即可能找到全局最优解, 并保证了算法的收敛性。

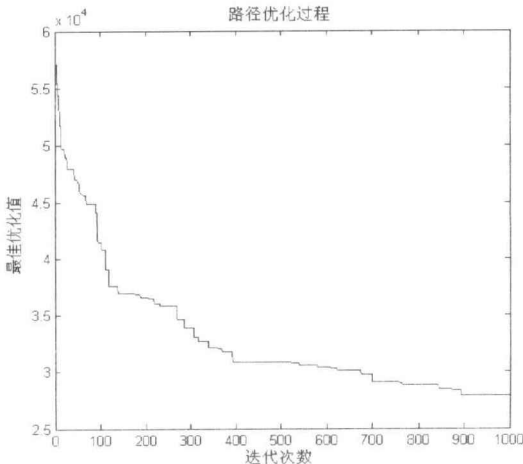
4.2 实例比较

通过使用相同的例子, 我们可以得到以下结论:

表 3 两算法仿真结果优劣性的比较

算法	最短的总路线长	各个圈长(km)	最优评价函数值	程序运行时间(s)
遗传算法	20225km	6106 6477 7643	27868	12
模拟退火算法	17731km	5527 5616 6578	24318	113

两算法的收敛情况和具体的求解路径如下图所示:



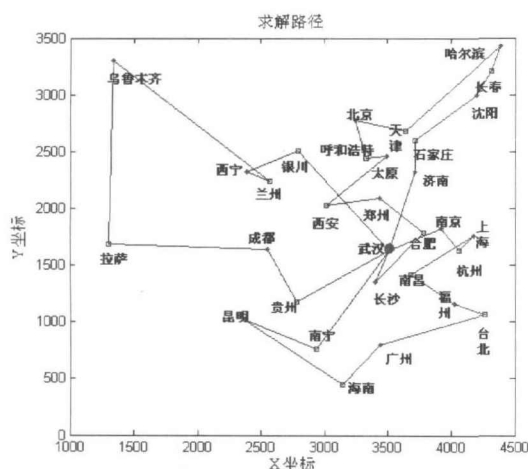


图1 遗传算法的收敛情况和具体的求解路径

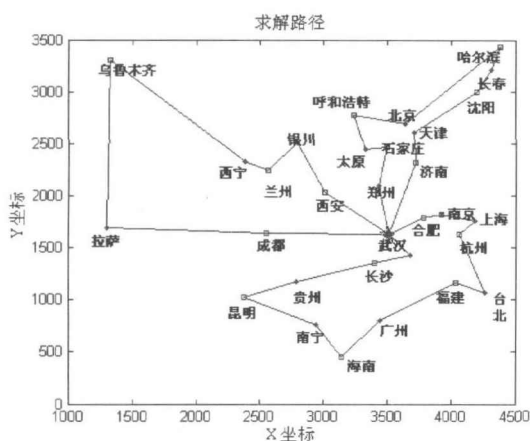
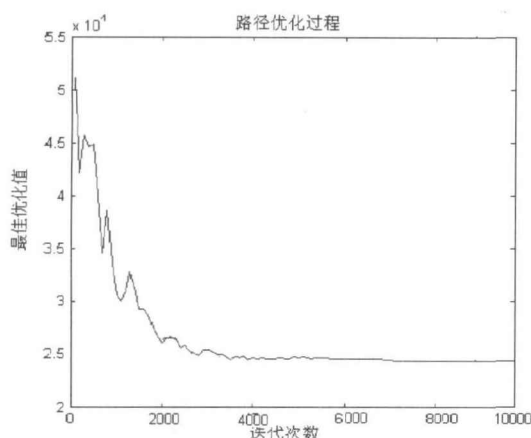


图2 模拟退火算法的收敛情况和具体的求解路径

由以上图标,我们可以总结出以下几点规律:

1) 遗传算法收敛速度很快,但结果相对较差。其主要原因在于遗传算法很容易陷入局部最优解,即为“早熟”。在很多情况下,遗传算法对于解决

大规模计算量问题在结果上可能导致不够精确。

2) 模拟退火算法收敛速度较慢,但结果相对较好。其主要原因在于模拟退火算法较遗传算法最大的不同在于其采用的是单个个体进行进化,一般新解优于当前解才接受新解,并且还需要通过温度参数 t 进行选择,并通过变异操作产生新个体。因为也不容易陷入局部解,但搜索范围的扩大同时也使得收敛速度的下降

3) 遗传算法容易出现过早收敛,从而陷入局部最优解,而模拟退火算法等一些优化算法具有很强的局部搜索能力,因而常常将局部搜索算法插入到遗传算法主循环中,形成混合算法,即退火遗传算法。

4.3 模拟退火遗传算法^[5]

步骤如下:

1) 给定种群规模 l , $k=0$; 初始温度 $t_k=t_0$, 产生满足载重约束的初始种群 $Pop(l)$, 对初始种群计算目标函数 $Z(t_k)$, 找出目标函数值最小(最大)的染色体 x 和这个函数值 h , 记 $\min p=x$, $s=h$ 。

2) 若满足结束条件停止计算, 输出最优染色体 $\min p$ 和 s ; 否则, 对初始种群, 在每一个染色体 i 的领域 $N(i)$ 中随机选取一个状态, 按模拟退火中的接受概率

$$p_r = \min \left[1, \exp \left(\frac{Z_i(t_k) - Z_j(t_k)}{t_k} \right) \right]$$

接受 j 。对每个染色体 i 迭代 n 次, 其中 $Z_i(t_k)$, $Z_j(t_k)$ 分别为状态 i 和 j 的函数值, t_k 为当前退火温度。共迭代 l 次选出新种群 $Pop1(l)$ 。

3) 在 $Pop1(l)$ 中计算适应度函数 $S(t_k) = 1/Z(t_k)$, 然后采用改进的比例选择策略进行染色体的选择。将在 $Pop1(N)$ 中的染色体, 按 $S(t_k)$ 值排序, 将值最大的染色体复制一个直接进入下一代, 下一代种群中剩下的 $N-1$ 个染色体用轮盘选择法产生。这样首先可以保证最优个体可以生存到下一代, 既给了适应度较大的个体以较大的机会进入下一代, 又避免了个体间因适应值不同而被选入下一代的机会悬殊。最后形成种群 $Pop2(l)$ 。

4) 分别以概率 P_c 和 P_m 对 $Pop2(l)$ 进行交叉和变异操作。

5) 计算种群 $Pop2(I)$ 中个体的目标函数值, 找出函数值最小的染色体 y 和这个函数值 g 。如果 $g < s(g > s)$, 则令 $\min p = y(\max p = y)$, $s = g$ 。 a 为退火率, $k = k + 1$, $t_k = a \times t_k$, 返回步骤 2。

4.4 与混合算法的对比

通过采用的相同的数据, 可以得到

表 4 与混合算法仿真结果优劣性的比较

算法	最短的总路线长	各个圈长(km)	最优评价函数值	程序运行时间(s)
遗传算法	20225km	6106 6477 7643	27868	12
模拟退火算法	17731km	5527 5616 6578	24318	113
模拟退火遗传算法	17731km	5527 5616 6578	24318	31

其结果图为:

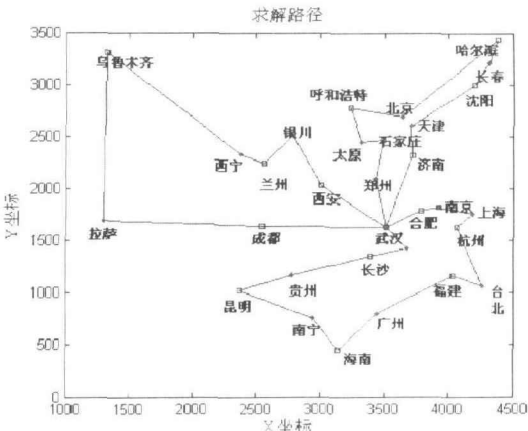
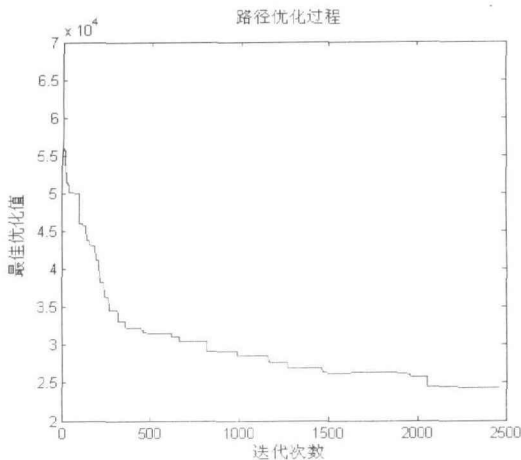


图 3 模拟退火遗传算法的收敛情况和具体的求解路径

通过结果我们发现, 改进后的模拟退火遗传算法, 即能够保证遗传算法较快的收敛速度, 也能够保证模拟退火算法结果的精确性, 从而实现了我们的目的。

5. 结语

目前, 针对旅行商问题的研究与算法比较是比较多的, 但对于多旅行商问题, 无论是在编码的设计上, 约束条件的增加直接导致了算法复杂度的增加。在这类问题的比较就变得很少了。

本文针对综合均衡度的多旅行商问题的遗传算法求解, 具有以下几个特点:

- 1) 结合多旅行商问题的特点, 专门为其设计了一种编码, 有效解决了多个旅行商的问题。
- 2) 通过综合考虑均衡度的复杂性、量纲问题及直观性, 重新提出了均衡度的一种简单清晰的定义方式。
- 3) 对了达到较快收敛的目的, 给出了一套评价被选概率优劣的方法, 也确定了较优的被选概率的定义方案。
- 4) 通过将遗传算法与模拟退火算法的结果进行实例比较, 找到了遗传算法较模拟退火算法的优势和劣势, 并提出了改进算法——模拟退火遗传算法, 最终有效地提高了运算的效率和准确度。

基金项目: 国家自然科学基金(批准号: 70871050) 和全国大学生创新试验计划项目(081051109)。

参考文献:

- [1] 王立志, 汪定伟, 闫杨. 一类多旅行商问题的计算及仿真. 系统仿真学报, 2009年第20期
- [2] 卢厚清, 王辉东, 黄杰, 李波. 任务均分的多旅行商问题[J]. 系统工程, 第23卷第2期
- [3] 李敏, 吴浪, 张开碧. 求解旅行商问题的几种算法的比较研究. 重庆邮电大学学报(自然科学版) 第20卷第5期
- [4] Nirwan Ansari, Edwin Hou著, 李军, 边肇祺译. 用于优化的计算智能. 清华大学出版社 1999年第1版
- [5] 张建光. 基于退火遗传算法的战时非满载车辆调度. 国防科学技术大学工程硕士学位论文. 2009年 P39-47

作者简介:

熊翠(1988-), 女, 湖南长沙人, 华中师范大学数学与统计学学院

Email: haoyoulibo@163.com

The Solution and Improvement of Genetic Algorithm in Multi-traveling Salesman problem

Abstract: Traveling salesman problem is the a well-known combinatorial optimization problems, and the multi-traveling salesman problem is the expansion of it. Combining with balancing, this article proposed the design of genetic algorithm for multi-traveling salesman problem, and, compared and combination with the simulated annealing algorithm, which improved the speed and efficiency of operation effectively.

Keywords: multi-traveling salesman problem; genetic algorithm; balancing; simulate annealing algorithm; simulated annealing and genetic algorithm