

蚁群算法在求解 TSP 问题上与遗传算法的对比研究^①

A Comparison Study of GA and ACA on TSP

(江西理工大学 信息工程学院 江西 赣州 341000)

康岚兰 李康顺 (江西理工大学 应用科学学院 江西 赣州 341000)

(中国科学院自动化研究所 北京 100080)

摘要: 蚁群算法(ACA)与遗传算法(GA)都属于仿生型优化算法,是解决组合优化问题的强有力工具,并都分别成功应用于旅行商问题(TSP)问题中。本文通过实验验证了两种算法在解决 TSP 问题上各自的优缺点,并给出了未来的进一步研究方向。

关键词: 蚁群算法 遗传算法 旅行商问题

1 引言

遗传算法和蚁群算法都是受生物进化论的启发而提出来的一种仿生算法,不同在于遗传算法是模拟基因进化的过程,而蚁群算法模拟的是由简单个体组成的群体行为,即:社会系统。两种算法都应用于求解组合优化问题上,并取得了一定的成果。本文将两种算法应用于 TSP^[1]问题中,通过实验对比,分析两种算法在求解组合优化上的优劣性,同时提出改进的建议。

2 TSP 问题

TSP^[1]问题(即:旅行商问题),就是指给定 n 个城市和两两城市之间的距离,要求确定一条经过各城市当且仅当一次的最短路线。其图论描述为:给定图 $G=(V, A)$,其中 V 为顶点集, A 为各顶点相互连接组成的边集,已知各顶点间的边接距离,要求确定一条长度最短的 Hamilton 回路,即遍历所有顶点当且仅当一次的最短回路。

选择旅行商问题作为测试问题的原因主要有:

- ① 它是一个最短路径问题,蚁群算法和遗传算法很容易适应这类问题;
- ② 很容易理解,不会因为太多的术语而使得算

法行为的解释难以理解;

- ③ TSP 是典型的组合优化难题,常常用来验证某一算法的有效性,便于与其他算法比较。

3 蚁群算法基本原理和 TSP 问题求解

蚁群算法(Ant Colony Algorithm, ACA)由意大利学者 Colomi A, Dorigo M 和 Maniezzo V 于 1991 年首先提出,用蚁群在搜索食物源的搜索的过程中所体现出来的寻优能力来解决一些离散系统优化中的困难问题。蚂蚁在觅食的过程中通过一种称之为信息素(Pheromone)的物质相互传递信息[2]。更具体地,蚂蚁在运动过程中能够在其所经过的路径上留下信息素,而且在运动过程中能够感受到这种信息素的存在及其强度,并以此指导自己的运动方向。蚂蚁倾向于朝着信息素浓度高的方向前进,因此,由大量蚂蚁组成的蚁群的行为便表现出一种信息的正反馈现象:某一路径上走过的蚂蚁越多,则后来者选择该路径的概率就越大。蚁群就是通过个体之间这种信息交换机制来彼此协作达到搜索食物的目的。

为讨论问题方便,现作如下符号说明:

^① 基金项目:国家重大基础研究(973)项目(2004CB318103);江西省教育厅科学技术研究项目,演化自适应智能天线的研究与开发(第一),江西省教育厅 2006-2007,1 万元,项目编号:赣教技字[2007]205 号

设 m 表示蚁群中蚂蚁的数量, n 表示城市数量, d_{ij} ($i, j=1, 2, \dots, n$) 表示城市 i 和城市 j 之间的距离, $\tau_{ij}(t)$ 表示 t 时刻在边 $e(i, j)$ 上的信息量。

$$\Delta\tau_{ij}^{(k)}(t) = \begin{cases} \frac{Q}{d_{ij}}, & \text{若蚂蚁 } k \text{ 在循环中经过边 } e(i, j) \\ 0, & \text{否则} \end{cases} \quad (6)$$

① 初始时刻, 各条路径上的信息量相等, 设 $\tau_{ij}(0) = C$ (C 为常数), 蚂蚁 k ($k=1, 2, \dots, m$) 在运动过程中, 根据各条路径上以信息量为变量的概率决定转移方向:

其中 Q 为常数, 表示每只蚂蚁周游一遍留下的信息总量, L_k 为蚂蚁 k 在本次循环中所走路程的长, d_{ij} 为城市 i 与城市 j 之间的距离。

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta}{\sum_{s \in allowed_k} \tau_{is}^\alpha \cdot \eta_{is}^\beta}, & \text{若 } j \in allowed_k \\ 0, & \text{否则} \end{cases} \quad (1)$$

它们的区别在于蚁周系统得用的是整体信息, 而蚁密和蚁量系统利用的是局部信息, 在求解 TSP 问题中, 显然蚁周系统的性能较好。本文采用蚁周系统。

其中 η_{ij} 表示边 $e(i, j)$ 的能见度, 用某种启发式算法算出, 一般取 $\eta_{ij} = 1/d_{ij}$, $P_{ij}^k(t)$ 表示在 t 时刻蚂蚁 k 由位置 i 转移到位置 j 的概率, $allowed_k = \{0, 1, \dots, m-1\} - tabu_k$ 表示蚂蚁 k 下一步允许选择的城市的集合, $tabu_k$ ($k=1, 2, \dots, m$) 表示蚂蚁 k 已经走过城市的集合, 开始时 $tabu_k$ 中只有一个元素, 即蚂蚁 k 的出发城市, 随着进化的进行, $tabu_k$ 中的元素不断增加, α 表示轨迹的相对重要性, β 表示能见度的相对重要性。

③ 循环以上步骤, 直到周游次数达到指定次数或在一定时间内没有新的更优解出现。见图 1

② 当蚂蚁完成一次循环后, 各路径上信息量根据以下公式进行更新:

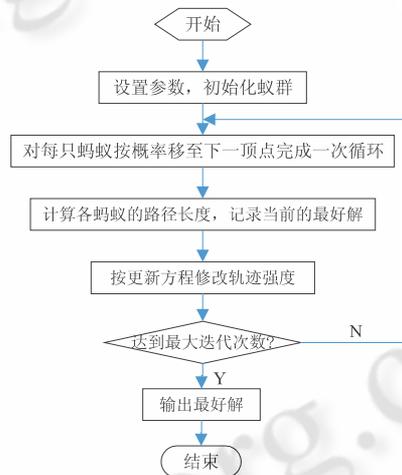


图 1 蚁群算法流程图

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (2)$$

其中 $\rho \in (0, 1)$ 表示信息素含量 $\tau_{ij}(t)$ 随时间的推移而衰减的程度, $\Delta\tau_{ij}$ 表示本次循环中路径 $e(i, j)$ 上的信息量增量:

4 遗传算法基本原理和 TSP 问题求解

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^{(k)}(t) \quad (3)$$

$\Delta\tau_{ij}^{(k)}(t)$ 表示蚂蚁 k 在本次循环中城市 i 和城市 j 之间留下的信息量, $\Delta\tau_{ij}^{(k)}(t)$ 根据表达形式的不同, Dorigo M 曾给出不同的模型, 分别称为蚁周系统 (ant-cycle system)、蚁量系统 (ant-quantity system)、蚁密系统 (ant-density system)。它们的计算公式见公式 (4)~(6):

遗传算法^[3] (Genetic Algorithm, GA) 是模拟达尔文的遗

在蚁周系统 (ant-cycle system) 中:

传选择和自然淘汰的生物进化过程的计算模型。它的思想源

$$\Delta\tau_{ij}^{(k)}(t) = \begin{cases} \frac{Q}{L_k}, & \text{若蚂蚁 } k \text{ 在循环只经过边 } e(i, j) \\ 0, & \text{否则} \end{cases} \quad (4)$$

于生物遗传学和适者生存的自然规律, 是具有"生存+检测"迭代过程的随机化概率搜索算法。遗传算法以一种群体中的所有个体为对象, 并利用随机化技术指导一个被编码的参数空间进行高效搜索。其中, 选择、交叉和变异构成了遗传算法的遗传操作, 参数编码、初始群体的设定、适应度函数的设计、遗传操作的设计、控制参数的设定, 这 5 个要素组成了遗传算法的核心内容。

在蚁量系统 (ant-quantity system) 中:

通常遗传算法的设计主要有以下步骤:

$$\Delta\tau_{ij}^{(k)}(t) = \begin{cases} Q, & \text{若蚂蚁 } k \text{ 在循环中经过边} \\ 0, & \text{否则} \end{cases} \quad (5)$$

① 确定数据编码方案, 随机产生一组初始个体构成初始种群。

在蚁密系统 (ant-density system) 中:

本文采用一种对旅行最自然的编码方案:路径表示法。例如 (7 4 9 5 6 1 2 8 10)代表从城市 7 出发,经由城市 4—9—5—6—1—2—8—10,最后又回到城市 4 的一条路径。

② 给出评价个体优劣的适配值函数,并评价每一个体的适配值(适应度)。

个体的评价是对每一个体计算其路径的长度,并把该长度作为个体的适应度函数,则规定适应度函数为 $f(x)$:

$$f(x) = \sum_{i=1}^{n-1} d(i, j+1) + d(1, n) \quad (7)$$

其中, $d(i, j)$ 表示城市 i 与城市 j 之间的距离。适应度越小的个体,该个体的路径越短,该个体则越好。

③ 判断算法是否满足收敛准则,若满足则输出搜索结果,否则执行以下步骤。

④ 根据适配值大小以一定方式执行复制操作。

产生初始群体后,根据计算的个体适应度,对整个群体进行适应度从小到大排序,设定选择百分比 p ,为保持群体的多样性,在进行选择时,规定选择 $p/2$ 的好的个体,再选择 $p/2$ 的个体放入到交配池中进行交叉、变异等遗传操作。

⑤ 按交叉概率 p_c 执行交叉操作。

采用 Davis 提出的 OX 算子。即通过从一个亲体中挑选一个子序列旅行并保存为一个亲体的城市相对次序来构造后代。例如,两个亲体(切割点以“|”标记):

$$p_1 = (1\ 2\ 3\ | \ 4\ 5\ 6\ 7\ | \ 8\ 9)$$

$$p_2 = (4\ 5\ 2\ | \ 1\ 8\ 7\ 6\ | \ 9\ 3)$$

为得到后代,首先将割点之间的片段拷贝到后代中,得:

$$o_1 = (x\ x\ x\ | \ 4\ 5\ 6\ 7\ | \ x\ x)$$

$$o_2 = (x\ x\ x\ | \ 1\ 8\ 7\ 6\ | \ x\ x)$$

为得到后代 o_1 ,再将 p_2 中已在 o_1 中的城市 4、5、6、和 7 后,得到 2-1-8-9-3,接着将该序列顺次放在 o_1 中,得:

$$o_1 = (2\ 1\ 8\ | \ 4\ 5\ 6\ 7\ | \ 9\ 3)$$

类似可得另一个后代:

$$o_2 = (2\ 3\ 4\ | \ 1\ 8\ 7\ 6\ | \ 5\ 9)$$

⑥ 按变异概率 p_m 执行交叉操作。

采用倒置变异。即在染色体上随机地选择两点,

将两点间的子串反转。例如:

原个体 (1 2 3 4 5 6 7 8 9)

随机选择两点 (1 2 3 | 4 5 6 7 | 8 9)

倒置后的个体 (1 2 3 | 7 6 5 4 | 8 9)

⑦ 返回步骤③。

见图 2

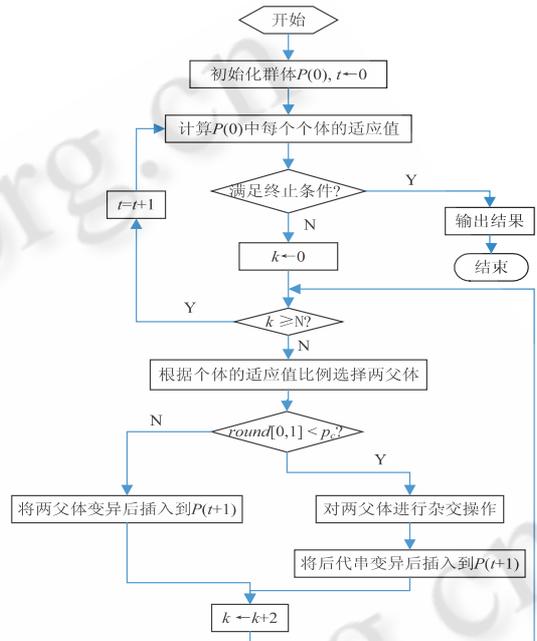


图 2 遗传算法流程图

5 算法的对比研究

本文中用蚁群系统作为蚁群算法的代表,与基本遗传算法进行比较。表 1 给出了两种算法在城市数分别为 30、48、51 的 TSP 问题上的仿真结果,并在相同条件下分别运行 20 次和 50 次,同时给出了不同问题在 TSPLIB^[7]中的结果。其中算法参数设定如下:

$$\rho = 0.1, \alpha = 1, \beta = 5, m = 60, C = 10, Q = 1000, p_c = 0.75, P_m = 0.05$$

表 1 两种算法在求解不同 TSP 问题上的实验结果

问题名称	运行次数	ACS			GA			TSPLIB 结果
		最优解	平均值	最差解	最优解	平均值	最差解	
oliver30	20	423.74	429.7	432.46	423.74	456.68	502.57	423.74
	50	423.74	428.76	432.45	423.74	425.7	439.84	
att48	20	33780	35595	36534	37880	38833	38894	33522
	50	33780	35533	36534	35633	38541	42458	
Eil51	20	426	428.4	431.42	495	512	536	426
	50	426	428	432.01	494	514	554	

从以上算法分析和实验结果可见:蚁群优化算法和遗传算法都是一种结合了分布式计算,易于并行实现,同时具有较强鲁棒性的概率搜索仿生算法。蚁群算法是模拟由简单个体组成的群落与环境以及个体之间的互动行为,即社会系统^[6],从而可能产生不可预测的群体行为,而遗传算法是模拟基因进化的过程。它们在实现 TSP 问题中的共同点有:

① 都容易出现过早收敛,陷入局部最优值。

② 算法本身在参数选择与取值上无法在理论上得不到严格的证明,只能通过实验的方法来确定其优化组合。

③ 对搜索空间两者都没有特殊的要求,如可导性、连续性、凹凸性等辅助信息,实用范围更加广泛。

区别在于:蚁群算法是通过信息素的累积和更新而收敛于最优路径,但初期信息素匮乏,导致算法速度慢,而遗传算法虽具有快速全局搜索能力,但对于系统中的反馈信息却没有利用,往往导致无为的冗余迭代,求解效率低。因此,在城市规模大于 30 时,遗传算法的搜索能力将逐渐下降,大到一定程度时,如想在迭代次数内获得最优解,但时间太长,无法忍受。此时蚁群算法表现好于遗传算法,可在很少的迭代次数内达到最优解。当城市规模过大时,蚁群算法可能出现停滞现象,从而也无法得到最优解。

6 结语和展望

为了克服两种算法各自的缺陷,形成优势互补,利用遗传算法的随机搜索、快速性^[1]和蚁群算法的正反馈机制、求解效率高等特性,将两种算法融合,核心思想如下:

首先由遗传算法产生较优解,较优的路径留下信息素,其他不改变,然后让蚂蚁按照蚁群算法完成一次遍历后,再让蚂蚁作遗传算法的交叉操作和变异操作,

有可能经过交叉操作和变异操作的解不一定得到改善,只有改善的蚂蚁路径,才代替原来的路径。

将上述遗传蚁群混合算法应用于 TSP 问题中进行检验,从而有望得到一种更优的新的启发式算法,这是未来进一步的研究方向,在改进上述算法时,可进一步引入一些改进遗传算法或蚁群算法,如在混合算法中的蚁群算法部分可引入最大最小蚂蚁系统^[8](Max - MIN Ant System, MMAS)以及最优 - 最差蚂蚁系统^[11](Best - Worst Ant System, BWAS)等。

参考文献

- 1 李士勇,陈永强,李研. 蚁群算法及其应用. 哈尔滨: 哈尔滨工业出版社, 2004. 9.
- 2 Colomi A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies. In: The First European conference on Artificial Life, France: Elsevier, 1991: 134 - 142.
- 3 潘正君,康立山,陈毓屏. 演化计算. 北京: 清华大学出版社, 1998. 5.
- 4 高尚,杨静宇. 群智能算法及其应用. 北京: 中国水利水电出版社, 2006. 9.
- 5 彭丹平等. 求解 TSP 的一种改进遗传算法. 计算机工程与应用, 2006. 13
- 6 冯静,舒宁. 群智能理论及其应用研究. 计算机工程与应用, 2006. 17.
- 7 <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp>.
- 8 Stutzle T, Hoos H. Improvements on the Ant System: Introducing MAX - MIN ant System. In Proceedings of the International Conference in Artificial Neural Networks and Genetic Algorithms, Springer Verlag, Wien. 1997. 245 - 249.