

遗传算法和模拟退火算法求解 TSP 的性能分析

汪松泉¹,程家兴²

(1. 安徽大学 计算机科学与技术学院,安徽 合肥 230039;

2. 安徽大学 计算智能与信号处理教育部重点实验室,安徽 合肥 230039)

摘要:旅行商问题(Traveling Salesman Problem, TSP)是一个典型的组合优化问题,并且是一个 NP 难题,其可能的路径总数与城市数目是呈指数型增长的,所以一般很难精确地求出其最优解,因而寻找出有效的近似求解算法就具有重要的意义。目前求解 TSP 问题的主要方法有模拟退火算法(Simulated Annealing, SA)、遗传算法(Genetic Algorithm, GA)和神经网络算法等。GA 是模拟生物在自然环境中的遗传和进化过程而形成的一种自适应的全局优化概率搜索算法。SA 算法用于优化问题的出发点是基于物理中固体物质的退火过程与一般优化问题的相似性。文中将提出遗传算法和模拟退火算法求解 TSP 问题,通过试验比较两者求解 TSP 问题的性能,结果表明 GA 的性能要优于 SA 的性能。

关键词:遗传算法;模拟退火算法;TSP

中图分类号:TP301.6

文献标识码:A

文章编号:1673-629X(2009)11-0097-04

Performance Analysis on Solving Problem of TSP by Genetic Algorithm and Simulated Annealing

WANG Song-quan¹, CHENG Jia-xing²

(1. School of Computer Science and Technology in Anhui University, Hefei 230039, China;

2. Ministry of Edu. Key Lab. of Intelligent Computing & Signal Processing, Anhui Univ., Hefei 230039, China)

Abstract: TSP is a typical combination optimization problem, which is also an NP hard - problem. Its size is increased by exponential n . So, it is hard to find a precision result, and it is very important to search for the near result. Currently, the main method of solving TSP has GA, SA and the neural network algorithm. GA is a simulation of the natural environment in the biogenetic and evolutionary process of the formation of an adaptive search algorithm for global optimization probability. SA solves optimization problem, which the starting point is based on the physics of the annealing process of solids with the general similarity of optimization problems. Proposed two effective methods: genetic algorithm and simulated annealing, through the experiment, compare the two performance analysis, the results show that the GA's performance is superior to the performance of SA.

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

0 引言

TSP(Traveling Salesman Problem)问题是一个典型的易于描述却难以大规模求解的 NP-完全问题,对于这类问题很难用全局搜索法精确地求出其最优解,因此应用的有效算法寻找其最优或近似最优解具有重要的理论意义。另外,很多实际问题,如印刷电路板的钻孔线路方案,连锁店的货物配送路线等经简化处理后可以建模为 TSP,因此对 TSP 问题求解方法的研究也

具有重要的应用价值^[1]。

TSP 问题描述为:设有 n 个城市的集合 $city = \{C_1, C_2, \dots, C_n\}$,对于城市 $C_i, C_j \in city$,从 C_i 到 C_j 的距离记为 $d_{ij} \in \mathbf{R}^+$,这里假设 $d_{ij} = d_{ji}$,即考虑对称 TSP 问题。TSP 问题的解,就是在集合 $city$ 中找到一个不重复的全排列 $C_{k1}, C_{k2}, \dots, C_{kn}$,使其之间距离最短,也就是要确定一条经过每个城市的当且仅当一次并返回出发点的最短路线。其图论描述为:给定图 $G = (V, E)$,其中 V 为顶点集, E 为各顶点相互连接组成的边集。已知各顶点间的连接距离,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^[2]。

目前求解 TSP 问题的主要方法有模拟退火算法(Simulated Annealing, SA),遗传算法(Genetic Algo-

收稿日期:2009-03-02;修回日期:2009-06-06

基金项目:教育部博士点基金(200403057002)

作者简介:汪松泉(1984-),安徽怀宁人,硕士研究生,研究方向为机器学习、智能计算;程家兴,教授,博士生导师,研究方向为智能计算、算法分析及最优化方法。

rithm, GA) 和神经网络算法等。GA 是模拟生物在自然环境中的遗传和进化过程而形成的一种自适应的全局优化概率搜索算法, 具有较强的全局搜索性能。SA 算法用于优化问题的出发点是基于物理中固体物质的退火过程与一般优化问题的相似性, 它具有很好的局部搜索能力^[3]。文中将提出遗传算法和模拟退火算法求解 TSP 问题, 通过实验结果比较两者求解 TSP 问题的性能。

1 求解 TSP 的遗传算法

遗传算法(GA)是一个人工智能方法, 它的机理基于达尔文的生物进化论的适者生存原理, 模拟生物进化的步骤, 将繁殖、杂交、变异、竞争等概念引入到算法中。通过维持一组可行解, 并通过可行解的重新组合, 改进可行解在多维空间内的移动轨迹或趋向, 最终走向最优解。它克服了传统优化方法容易陷入局部极值的缺点, 是一种全局优化算法。遗传算法的基本原理^[4]如下:

(1) 定义一个目标函数;

(2) 将可行解群体在一定的约束条件下初始化, 每一个可行解用一个向量 x 来编码, 称为一条染色体, 向量的分量代表基因, 它对应可行解的某一决策变量;

(3) 计算群体中每条染色体 $x_i (i = 1, 2, 3, \dots, n)$ 所对应的目标函数值, 并以此计算适应值 F_i , 按 F_i 的大小来评估该可行解的好坏;

(4) 以优胜劣汰的机制, 将适应度差的染色体淘汰掉, 对幸存的染色体根据其适应值的好坏, 按概率随机选择, 进行繁殖, 形成新的群体;

(5) 通过杂交和变异的操作, 产生子代。杂交是随机选择两条染色体(双亲), 将某一点或多点的基因互换而产生两个新个体。变异是基因中的某一点或多点发生突变;

(6) 对子代群体重复步骤(3) ~ (5) 的操作, 进行新一轮遗传进化过程, 直到迭代收敛(适应值趋稳定)即找到了最优解或准最优解。

根据上面的基本原理把求解 TSP 问题分解成以下几个操作^[5]:

1) 初始化过程: 用 $v_1, v_2, v_3, \dots, v_n$ 代表所选 n 个城市。定义整数 $pop - size$ 作为染色体的个数, 并且随机产生 $pop - size$ 个初始染色体, 每个染色体为 1 到 18 的整数组成的随机序列。适应度 f 的计算: 对种群中的每个染色体 v_i , 计算其适应度, $f = od(t(i), t(i + 1))$ 。

评价函数 $eval(v_i)$: 用来对种群中的每个染色体 v_i 设定一个概率, 以使该染色体被选中的可能性与其

种群中其它染色体的适应性成比例, 即通过轮盘赌, 适应性强的染色体被选择产生后台的机会要大, 设 $\alpha \in (0, 1)$, 定义基于序的评价函数为 $eval(v_i) = \alpha * (1 - \alpha)^{(i - 1)}$ 。

2) 选择过程: 选择过程是以旋转赌轮 $pop - size$ 次为基础, 每次旋转都为新的种群选择一个染色体。赌轮是按每个染色体的适应度进行选择染色体的。

step1 对每个染色体 v_i , 计算累计概率 $q_i, q_0 = 0; q_i = \sigma eval(v_j) j = 1, \dots, i; i = 1, \dots, pop - size;$

step2 从区间 $(0, pop - size)$ 中产生一个随机数 r ;

step3 若 $q_{i-1} < r < q_i$, 则选择第 i 个染色体;

step4 重复 step2 和 step3 共 $pop - size$ 次, 这样可以得到 $pop - size$ 个复制的染色体。

3) grefenstette 编码: 由于常规的交叉运算和变异运算会使种群中产生一些无实际意义的染色体, 文中采用 grefenstette 编码可以避免这种情况的出现。所谓的 grefenstette 编码就是用所选队员在未选(不含淘汰)队员中的位置, 如:

8 15 2 16 10 7 4 3 11 14 6 12 9 5 18 13 17 1

对应:

8 14 2 13 8 6 3 2 5 7 3 4 3 2 4 2 2 1

4) 交叉过程: 文中采用常规单点交叉。为确定交叉操作的父代, 从 0 到 $pop - size$ 重复以下过程: 从 $[0, 1]$ 中产生一个随机数 r , 如果 $r < pc$, 则选择 v_i 作为一个父代。

将所选的父代两两组队, 随机产生一个位置进行交叉, 如:

8 14 2 13 8 6 3 2 5 7 3 4 3 2 4 2 2 1

6 12 3 5 6 8 5 6 3 1 8 5 6 3 3 2 1 1

交叉后为:

8 14 2 13 8 6 3 2 5 1 8 5 6 3 3 2 1 1

6 12 3 5 6 8 5 6 3 7 3 4 3 2 4 2 2 1

5) 变异过程: 文中采用均匀多点变异。类似交叉操作中选择父代的过程, 在 $r < pm$ 的标准下选择多个染色体 v_i 作为父代。对每一个选择的父代, 随机选择多个位置, 使其在每位置按均匀变异(该变异点 xk 的取值范围为 $[ukmin, ukmax]$, 产生一个 $[0, 1]$ 中随机数 r , 该点变异为 $xk = ukmin + r(ukmax - ukmin)$ 操作。如:

8 14 2 13 8 6 3 2 5 7 3 4 3 2 4 2 2 1

变异后:

8 14 2 13 10 6 3 2 2 7 3 4 5 2 4 1 2 1

6) 反 grefenstette 编码:交叉和变异都是在 grefenstette 编码之后进行的,为了循环操作和返回最终结果,必须逆 grefenstette 编码过程,将编码恢复到自然编码。

7) 循环操作:判断是否满足设定的带数 $xzome$, 否,则跳出适应度 f 的计算;是,结束遗传操作,跳出。

2 求解 TSP 的模拟退火算法

模拟退火法(Simulated Annealing, SA)是模拟热力学中经典粒子系统的降温过程来求解规划问题的极值。当孤立粒子系统的温度以足够慢的速度下降时,系统近似处于热力学平衡状态,最后系统将达到本身的最低能量状态,即基态,这相当于能量函数的全局极小点。由于模拟退火法能够有效地解决大规模的组优化问题,且对规划问题的要求极小,该方法已成为一种具有发展前景的一种优化方法^[6]。

模拟退火法的基本原理^[7]如下:

(1) 给定初始温度 T_0 ,及初始点 x ,计算该点的函数值 $f(x)$;

(2) 随机产生扰动 Δx ,得到新点 $x' = x + \Delta x$,计算新点函数值 $f(x')$,及函数值差 $\Delta f = f(x') - f(x)$;

(3) 若 $\Delta f \leq 0$,则接受新点,作为下一次模拟的初始点;

(4) 若 $\Delta f > 0$,则计算新点接受概率: $p(\Delta f) = \exp(-\Delta f K \cdot T)$,产生 $[0,1]$ 区间上均匀分布的伪随机数 $r, r \in [0,1]$,如果 $p(\Delta f) \geq r$,则接受新点作为下一次模拟的初始点;否则放弃新点,仍取原来的点作为下一次模拟的初始点。

以上步骤称为 Metropolis 过程。按照一定的退火方案逐渐降低温度,重复 Metropolis 过程,就构成了模拟退火算法。当系统温度足够低时,就认为达到了全局最优状态。按照热力学分子运动理论,粒子作无规则运动时,它具有的能量带有随机性,温度较高时,系统的内能较大,但是对某个粒子而言,它所具有的能量较小。因此算法要记录整个退火过程中出现的能量较小的点。

根据上面的基本原理把求解 TSP 问题分解成以下几个操作^[8]:

(1) 解空间:解空间 S 是遍访每个城市恰好一次的所有路径,解可以

表示为 $\{w_1, w_2, \dots, w_n\}$, w_1, \dots, w_n 是 $1, 2, \dots, n$ 的一个排列,表明 w_1 城市出发,依次经过 w_2, \dots, w_n 城市,再返回 w_1 城市。初始解可选为 $(1, \dots, n)$;

(2) 目标函数:目标函数为访问所有城市的路径总长度。要求的最优路径为目标函数为最小值时对应的路径。

新路径的产生:随机产生 1 和 n 之间的两相异数 k 和 m ,不妨假设 $k < m$,则将原路径

$$(w_1, w_2, \dots, w_k, w_{k+1}, \dots, w_m, w_{m+1}, \dots, w_n)$$

变为新路径:

$$(w_1, w_2, \dots, w_m, w_{k+1}, \dots, w_k, w_{m+1}, \dots, w_n)。$$

上述变换方法就是将 k 和 m 对应的两个城市在路径序列中交换位置,称为 2-opt 映射。

3 实验仿真结果

在 Visual C++ 编译环境下,分别对城市数目为 10, 20, 30, 42, 51, 利用模拟退火算法和遗传算法来求解 TSP 问题,其求得的 TSP 路径的平均距离如表 1 所示。

表 1 模拟退火算法和遗传算法求解 TSP 问题的性能比较

城市数目(TSP路径的平均距离)					
算法类型	$n = 10$	$n = 20$	$n = 30$	$n = 42$	$n = 51$
模拟退火算法	170.5985	273.0956	317.0335	437.4417	463.5962
遗传算法	160.6494	245.3362	305.3587	392.1063	424.3753

用 MATLAB 对 42 个城市的 TSP 问题分别用模拟退火算法和遗传算法进行计算,求解路径和最优路径如图 1 和图 2 所示。

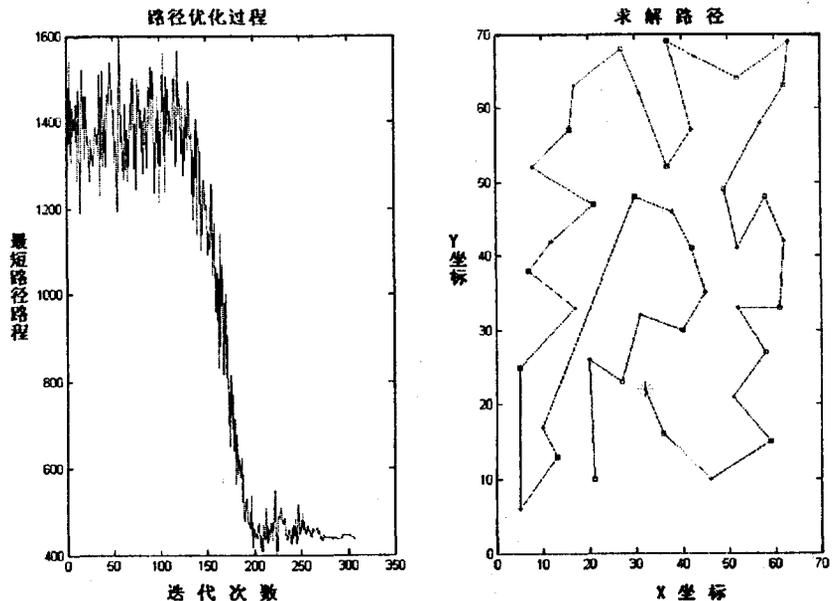


图 1 模拟退火算法计算的结果

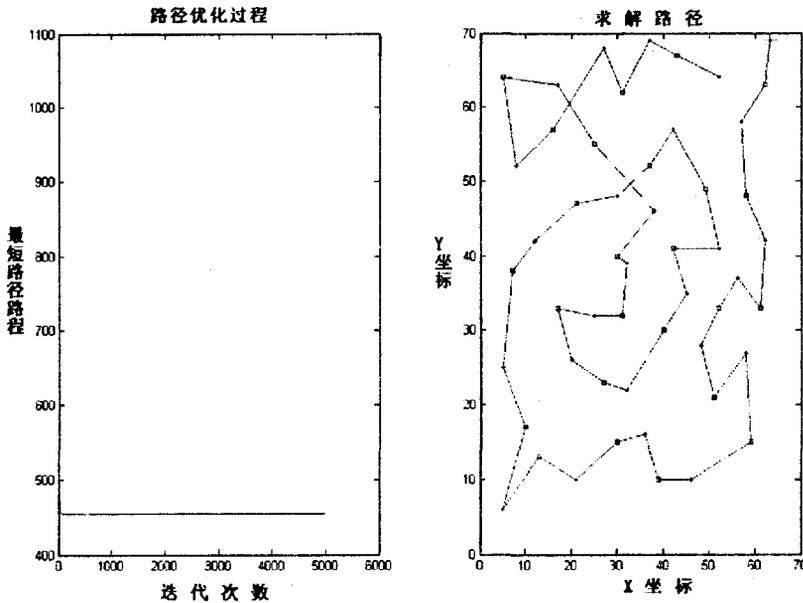


图 2 遗传算法计算的结果

4 结束语

遗传算法的主要特点是简单、通用、鲁棒性强,适用于并行分布处理,应用范围广。文中分析了遗传算法和模拟退火算法的基本原理,并将它应用于解决 TSP 问题,取得了较好的优化结果。通过实验得出遗传算法在求解 TSP 问题时,能够获得比模拟退火更好的答案,而且也比模拟退火算法的搜索速度要快,遗传算法的搜索过程是从一个母体到另一个母体的过程,使它能够有效放弃潜在的局部解,而用更加有效的算法

获得最优解。

参考文献:

[1] Grefenstette J. Genetic algorithms for the traveling salesman problem [C]//in: proc. of 1 int. conf. on genetic algorithms and their applications. [s. l.]: Lawrence Erlbaum Association, 1985:166 - 168.

[2] 邢桂华. 用 MATLAB 实现中国旅行商问题的求解[J]. 微计算机应用, 2004 (2):218 - 222.

[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[J]. 重庆邮电大学学报, 2008(10):624 - 625.

[4] 高经纬, 张煦, 李峰, 等. 求解 TSP 问题的遗传算法实现[J]. 计算机时代, 2004(2):19 - 21.

[5] 宋丹, 傅明, 朱亨荣, 等. 一种改进的遗传算法及其在 TSP 中的实现[J]. 微机发展(现更名: 计算机技术与发展), 2004, 14(6):21 - 23.

[6] 魏平, 徐成贤. 基于模拟退火算法优化分析与研究[J]. 装备制造技术, 2008(7):1 - 3.

[7] Yu Shiaw - Shian, Tsai Wen - Hsiang. A new thinning algorithm for gray - scales by the relaxation technique[J]. Pattern Recognition, 1990, 23(10):1067 - 1077.

[8] 许智宏, 宋勃, 董建波. 用蚂蚁算法和模拟退火算法解大规模 TSP 问题的研究[J]. 计算机工程与科学, 2008(10): 43 - 44.

(上接第 96 页)

4 结束语

本体在知识表示与知识推理方面发挥着重要作用,本体表示语言与描述逻辑相结合,通过对基于本体 DL 的语义推理机制进行阐述分析,较详细地探讨了在本体构建过程中,利用本体进行逻辑判定,实现自动推理语义,描述逻辑的作用十分重要。描述逻辑提供了完备高效的推理服务^[5]。其推理机制以归约判定和一致性判定为核心,能够实现本体知识库上的多种推理,在本体的构建阶段发挥了重要的作用。

表 4 一致性检测实例

```
//数据一致性检测
If ( ValidityTest(in Price)) System. Out .println ("ok");
检测结果:
冲突
-Error("conflict"): "Two individuals both same and different,
May be due to disjoint classes or price properties"
Culprit = eg: 012
Implicated node:eg: Name023SpeciaMIB
```

参考文献:

[1] 刘琼, 李保敏. 一种果品领域本体库的构建方法[J]. 计算机技术与发展, 2009, 19(1):197 - 203.

[2] Berners - Lee T. The Semantic Web[J]. Scientific American, 2001(6):1 - 6.

[3] 李文斌, 刘椿年. 对 OWL 及其逻辑基础的研究[J]. 计算机应用, 2004, 24:146 - 148.

[4] Baader F. Handbook of Description Logic[M]. Cambridge: Cambridge University, 2003.

[5] 聂卉, 龙朝晖. 描述逻辑语义推理机制的应用研究[J]. 信息检索技术, 2006(11):61 - 64.

[6] Noy N F, McGuinness D L. Ontology development: a guide to increasing your first ontology[R]. USA: Standford University, 2001.

[7] 张娜, 李宝敏. 语义检索及其关键技术研究[J]. 计算机技术与发展, 2006, 16(11):22 - 25.

[8] 浓国海, 穆斌, 胡学钢. 语义 Web 本体及本体库系统设计技术[J]. 微机发展(现更名: 计算机技术与发展), 2004, 14(7):78 - 81.