

求解 TSP 问题的拟人算法^①

吴 军, 李 建, 胡永泉

(西南石油大学 计算机科学学院, 成都 610500)

摘 要: 基于贪心算法提出了一种改进的求解旅行商问题(TSP)的拟人算法。该算法采用邻域定义, 主要思想是: 给定一个所有城市的全排列, 依此全排列的指挥用贪心算法生成一个回路。通过城市交换和城市序列平移, 在当前的邻域中搜索比它更好的解, 如能找到如此的解, 则使之成为新的当前解, 然后重复上述过程。在搜索的过程中, 采取跳坑策略以跳出局部最优解, 始终向目标最接近的方向搜索。算法结果与 Rego 提出的完全子路径搜索算法(F-SEC)做比较。

关键词: 旅行商问题; 拟人算法; 邻域搜索; 子序列平移; 跳坑策略

Personification Algorithm for Traveling Salesman Problem

WU Jun, LI Jian, HU Yong-Quan

(School of Computer Science, Southwest Petroleum University of China, Chengdu 610500, China)

Abstract: In this paper, a personification algorithm for solving the Traveling Salesman Problem (TSP) is proposed, which is based on original greedy algorithm. The algorithm adopts neighborhood definition, the main idea is that to give a full array of all cities, and so full array of command to generate a loop with a greedy algorithm. Through the city exchange and Sub-sequence migration, it generates a better solution in the current neighborhood search. If the solution is found, it becomes the new current solution. Then, repeat the process. In the search process, Off-trap strategies are used to jump from local optimal solution and guide the search in promising directions. The results of the algorithm are compared to the Full Sub-path Ejection Algorithm (F-SEC) proposed by Rego.

Keywords: TSP; personification algorithm; local search; sub-sequence migration; off-trap strategy

1 引言

旅行商问题(TSP 问题), 是目前国内外组合优化^[2]领域中的研究热点, 主要是研究如何寻找一种合适的办法以有效的方式来解决某个实际的离散问题。TSP 问题在科学领域和工程领域得到了广泛的应用^[3,4]。比如在构建通讯网络模型、设计公交车路线以及设计电路板布线等热门问题。因此, 旅行商问题具有很高的理论价值和实践价值。

2 问题的描述

简单的讲, TSP 问题即一个商人从自己的城市出发, 经过给定顾客的所在城市又能回到出发点, 遍历每个城市一次, 所经过的路径最短。

问题很容易陈述但难以实现。由于每次找到局部最优环路时, 形成下一环路总受环路之外这些城市的限制。所以如何寻找到最好的环路成为解决此问题的一个难题。TSP 问题形式化描述如下。记 $G=(V,E)$ 为赋权图, $V=(1,2,3,\dots,n)$ 为顶点集, E 为边集, 并且 $E=\{(V_i,V_j)|V_i,V_j\in V,i\neq j\}$, 各顶点间的距离 D_{ij} , 已知 $(D_{ij}>0,i,j\in V)$ 。如果对每个顶点 $V_i,V_j\in V$, 都有 $D_{ij}=D_{ji}$, 则称问题是对称的, 否则称问题是不对称的。对称的 TSP 问题是要在图中找出问题最短的哈密尔顿回路。通常, 求解 TSP 问题的算法可分为两类: 一类是完整算法; 另一类是近似算法。完整算法是任何给定的问题实例可产生最优解, 它通常基于分支定界。然而, 完整算法很耗时间, 在合理的计算时间内,

^① 收稿时间:2010-07-29;收到修改稿时间:2010-09-10

只能解决比较小型的问题实例。实现完整的算法,需要较高的编程技术。另外,近似算法可能在较短时间内生成问题的最优解或者接近最优解。

本文提出了一种拟人算法,采用的是近似算法。利用手动模型,在木板上钉钉子连线,启发出一些想法^[4]。然后将想法形式化,形成算法。根据问题实例计算的结果情况,反复多次改进,形成最终的算法。算法在设计过程中,基于一种邻域定义^[5]和一个新颖的解空间,采取模拟人跳坑的行为,跳出局部最优解,从而求得更好的解。在这种新的概念下,产生了一个具有代表性的有效的算法。用实验测试的方法来检验算法的效率。测试的 benchmark 实例来自国际公认的 TSPLIB。算法测试的结果跟完全子路径搜索算法 F-SEC 算法作比较。

3 贪心算法的描述

贪心算法是很常见的算法之一,简单易行,容易实现。基本思路是:从问题的某一个初始状态出发,通过寻找局部最优解逐步逼近给定的目标,以尽可能快的速度产生一个全局最优解。一般来说,贪心算法的证明围绕着整个问题的最优解,一定是由在贪心策略中存在的子问题的最优解得来的。其具体步骤如下:

任意给定所有城市的一个序列 $Q = (Q_1 \cdots Q_n)$, 根据此序列的指挥,可产生一个回路。

步骤 1: 给定一个城市作为出发城市 $Q_i (1 \leq i \leq n)$ 。以出发城市作为当前城市,在当前路径中恰好有出发城市。转步骤 2。

步骤 2: 若所有城市均在环路中,则转步骤 3。在当前路径以外的所有城市中找到距离当前城市最近的城市。若这样的城市不止一个,则选取这些城市中序号最小的城市。将当前城市与此城市连接起来。此城市作为新的当前点。转步骤 2。

步骤 3: 将当前城市与出发城市连接起来。这样便给出了一个回路。算法停机。

4 解决策略

4.1 拟人算法分析

拟人算法是一类特殊的算法,其思想是建立在模拟人的行为特征的基础之上。它在理论上收敛于问题的最优解,可加速问题的求解过程。由于有些算法随着问题规模的增大,因而也就失去了作为近似算法所

应具有的控制精度。此算法在控制精度误差上有所改进,得到的结果尽可能的近似最优解。

4.2 邻域结构

任意给定所有城市的一个序列 Q , 在序列 Q 的指挥下,贪心算法可生成一条回路 P 。将序列 Q 视为 n 维空间中的一个点。序列 Q 的邻域结构定义如下:

两城市交换: 城市 $Q_i (1 \leq i \leq n-1)$ 和 Q_n 交换,从而生成一个新的序列 Q' , 这是 Q 的一个邻点。

单城市平移: 城市 $Q_i (1 \leq i \leq n-1)$ 逐步移动至 Q 的最后一个位置,每移动一位产生一个新的序列 Q' , 这是 Q 的一个邻接点。

城市子序列平移: 从一个城市到另一个城市,记为子序列 $Q_{ij} (1 \leq i < j \leq n)$, 序列 Q_{ij} 逐步移位至 Q 的最后一个位置。每移动一位产生一个新的序列 Q' , 这是 Q 的一个邻接点。

城市子序列反序并移位: 从一个城市到另一城市,记为子序列 $Q_{ij} (1 \leq i < j \leq n)$ 。序列反序 Q_{ji} 并逐步移位到 Q 的最后一个位置。每移动一位产生一个新的序列 Q' , 这是 Q 的一个邻接点。

4.3 跳坑策略

第一种跳坑策略是随机扰动。即均匀地在所有城市中随机选择一个城市 $Q_i (1 \leq i \leq n)$, 然后将这个城市均匀地随机移到一个新的位置上。上述过程重复 C 次,从而完成了跳坑。 C 是一个给定的常数。

第二种跳坑策略是随机地重新撒点。具体来说是均匀地随机生成一个所有城市的全排列作为当前点。

4.4 步骤

用 T 作为一个计数器来记录跳坑的次数, T_m 是绝对常数,令 $T=0$, $C=5$ (其中, C 在一般情况下取 5, 可根据城市的个数调整跳坑次数,目的是在最短时间内求得最优解)。

步骤 1: 生成初始解。均匀地随机生成所有城市的一个序列 Q_0 。用贪心算法可生成一条回路 P_0 。序列 Q_0 为当前点。转步骤 2。

步骤 2: 邻域搜索。依次检查当前点的邻域。如果当前点的邻域中,有比当前点更好的点。也就是说,依此邻点的指挥生成的回路,比依当前点指挥生成的回路短。则找到第一个这样的邻点,将此邻点记为当前点。转步骤 2。如果当前点的邻域中没有比当前点更好的点,当前点是局部最优,则转步骤 3。

步骤 3: 跳坑策略。 $T=T+1$ 。检查停机条件: 如果

$T=T_m$ 或找到了最优回路,则算法停机。如果不满足停机条件。则执行跳坑策略,然后转到步骤2。具体来说,执行跳坑策略的过程如下:如果T能被100整除。则用第二种跳坑策略(随机地重新撒点)跳坑。如果T不能被100整除。则用第一种跳坑策略(随机扰动)跳坑。

5 结果分析

对每个 benchmark 问题实例,拟人算法独立计算10次,统计结果表1所示。对每次计算,统计相对误差(RE)。计算公式为: $RE = 100\% * (LS - opt) / opt$,

其中 opt 是指最优回路的长度。LS 表示本文提出的拟人算法所生成的回路的长度。F-SEC 和拟人算法一样,独立计算10次。

在27个 benchmark 问题实例中,拟人算法算出了25个问题实例的最优解。而F-SEC算出了其中17个问题实例的最优解。可见拟人算法优于F-SEC算法。BRE表示10次计算生成的最短回路的相对误差。WRE表示10次计算生成的最长回路的相对误差。ARE表示10次计算生成回路的算术平均值的相对误差。ART表示10次计算的计算时间的算术平均值。

表1 27个 TSPLIB 实例的拟人算法和完全子路径搜索(F-SEC)算法的比较

Problem	Opt	拟人算法				F-SEC			
		BRE	ARE	WRE	ART	BRE	ARE	WRE	ART
ber127	118282	0.00	0.12	0.17	196	0.12	0.28	0.56	16
d198	15780	0.01	0.09	0.12	392	0.30	0.63	1.14	26
eil101	629	0.00	0.01	0.03	47	0.00	0.02	0.16	12
eil15	426	0.00	0.00	0.00	4	0.00	0.02	0.24	3
eil76	538	0.00	0.01	0.02	7	0.00	0.02	0.19	4
kroA100	21282	0.00	0.00	0.00	7	0.00	0.05	0.11	8
kroA150	26524	0.00	0.01	0.02	189	0.00	0.19	0.31	27
kroA200	29368	0.00	0.18	0.49	429	0.27	0.52	0.73	28
kroB100	22141	0.00	0.03	0.12	41	0.00	0.45	1.02	14
kroB150	26130	0.00	0.01	0.03	139	0.02	0.18	0.25	21
kroB200	29437	0.01	0.05	0.18	391	0.11	0.34	0.47	28
kroC100	20749	0.00	0.00	0.00	4	0.00	0.12	0.34	14
kroD100	21294	0.00	0.00	0.00	5	0.00	0.47	1.06	18
kroE100	22068	0.00	0.00	0.00	7	0.00	0.18	0.38	15
pr107	44303	0.00	0.00	0.00	3	0.05	0.64	2.74	16
pr124	59030	0.00	0.00	0.00	4	0.10	0.48	0.99	17
pr136	96772	0.00	0.00	0.00	5	0.15	0.49	0.79	21
pr144	58537	0.00	0.00	0.00	8	0.00	0.77	1.41	21
pr152	73682	0.00	0.00	0.00	11	0.90	2.87	4.57	21
pr226	80369	0.00	0.04	0.16	96	0.22	0.91	1.56	37
Pr76	108159	0.00	0.00	0.00	3	0.00	0.66	1.12	11
Rat99	1211	0.00	0.02	0.06	7	0.00	0.06	0.17	13
Rd100	7910	0.00	0.01	0.13	17	0.00	0.03	0.09	13
St70	675	0.00	0.00	0.00	3	0.00	0.18	0.74	9
U159	42080	0.00	0.00	0.00	13	0.00	0.08	0.21	24
Ts225	126643	0.00	0.03	0.19	192	0.00	0.18	0.25	23
Lin105	14379	0.00	0.01	0.02	7	0.00	0.10	0.41	8

(下转第244页)

表2 烟草配送路径优化中遗传算法与节约算法的对比

运算序号	1	2	3	4	5	6	7	8
遗传算法求得总距离 Z (千米)	81.4	82.1	81.5	80.8	81.3	80.5	81.2	81.5
节约算法求得总距离 Z (千米)	96.8	91.5	89	88.6	86.9	92.9	94	89.1

从表中数据可以看出,程序运行8次得出的结果都优于节约法所求得的结果。且第6次还得到了最优解80.5千米,其对应的配送路径方案为:0—3—6—8—4—1—0—5—7—2—0(其中0表示配送中心)。

可见,利用遗传算法可以快速有效地求得优化烟草配送路径的最优解或近似最优解。

4 结束语

实验表明,先建立优化烟草配送路径的数学模型,然后用遗传算法优化求解,这是一种性能优良的启发式搜索方法。可以快速有效地求得优化烟草配送路径的最优解。文中所使用的编码方法、适应值的求法以及选择、交叉和变异算子,对求解类似的组合优化问题具有一定的参考价值。

参考文献

1 蔡希贤,夏士智.物流合理化的数量方法.武汉:华中工学院出版社,1985.

2 张翠军,张有华,秦彭,等.基于有时间窗车辆路径问题的混合蚁群算法.计算机工程与设计,2008,29(4):920-922.

3 Chen CH, Ting CJ, An improved ant colony system algorithm for the vehicle routing problem. Journal of the Chinese Institute of Industrial Engineers, 2006,23:115-126.

4 张颖,刘艳秋.软计算方法.北京:科学出版社,2002.

5 李军,郭耀煌.物流配送车辆优化调度理论与方法.北京:中国物资出版社,2001.

6 姜大立,杨西龙,杜文,等.车辆路径问题的遗传算法研究.系统工程理论与实践,1999,19(6):40-44.

7 陈国良.遗传算法及其应用.北京:人民邮电出版社,1996.

8 Dorigo M, Boeabeau E, Theraola G. Ant algorithms and stigmergy. Future Generation Computer System, 2000,16: 851-871.

9 Holland J.遗传算法的基本理论与应用.李敏强译.北京:科学出版社,2003.

(上接第250页)

6 结束语

本文提出了一种新颖的求解TSP问题的算法。通过邻域结构调整和两种策略的调用,更加完善了在重组过程中丢失的序列,增加了算法组合优化。

27个TSPLIB问题实例中,计算其它二个问题实例时,相对于最优解的误差可为十万分之一左右。并且,通过结果比较,最好、最坏以及平均误差减小了很多,效率上得到了很大的改进。此算法是一种效率较高的求解TSP问题的算法。

参考文献

1 Rego C. Relaxed tours and path ejections for the traveling salesman problem. European Journal of Operational Research,

1998,106(2-3):522-528.

2 Chatterjee S, et al. Genetic algorithms and travelling salesman problems. European J. of Opnl. Res. 1996,93(3):490-510.

3 Reinelt G. TSPLIB - a traveling salesman problem library. ORSA Journal on Computing, 1991,3(4):376-385.

4 Wang RL, Tang Z, Cao QP. A learning method in Hopfield neural network for combinatorial optimization problem. Neurocomputing, 2002,48(1-4):1021-1024.

5 Wang L, Huang QW. Novel. Local search method for the traveling salesman problem. Journal of Southwest Jiaotong University, 2005,13(1):1005-2429.