文章编号:1005-0523(2011)02-0102-05

旅行商问题推广及其混合智能算法

陈冬华

(华东交通大学基础科学学院,江西 南昌 330013)

摘要:旅行商问题(TSP)是典型的NP-hard问题,是组合优化研究领域中的热点问题之一。全体旅行商问题(CTSP)是TSP的变形推广,它是比TSP更复杂的一个问题,而且有着广泛的应用。遗传算法(GA)具有随机全局搜索能力,但对于系统反馈信息利用能力差,且收敛慢,求解效率低。蚁群系统(ACS)算法具有并行全局搜索能力,且在很大程度上避免了收敛到局部极小解从而陷入停止进化的可能性,但它也存在初期信息缺乏且收敛慢的缺点。用GA和ACS算法可组合成混合智能算法(CIA),用它来求解CSTP具有信息利用充分等较好性能,且收敛速度快。

关键词:TSP;CTSP;遗传算法;蚁群算法;混合智能算法。

中图分类号:TP183;TP301.6 **文献标识码**:A

1 TSP与CTSP

从上个世纪50年代起,学术界出版了大量关于旅行商问题[1]的文献。旅行商问题(traveling salesman problem,TSP)是指这样一个组合优化问题:某个商人欲到 n 座城市 $A=(a_1, a_2, \cdots, a_n)$ 去推销商品,希望选择一条路线使得商人走遍每座城市后(仅能访问一次)回到起点且所走路程最短。用图论术语来说[2],假设有一个图 $g=(v, e)$,其中 v 是顶点集(顶点对应于某个城市),e 是边集(城市之间的连接状态,每边赋予权重表示城市间距离),设 D 是由顶点 i 与顶点 j 之间的距离所组成的距离矩阵,如果任意两个城市的距离都是对称的,它所对应的是图论中的无向图,若两个城市间的距离是非对称的,它所对应于图论中的有向图。旅行商问题就是求出一条经过所有顶点且每个顶点只经过一次的具有最短路径的回路。

如果令城市 a_i 与 a_j 的距离为 d_{ij},用 x_{ij} 表示商人是否以顺序 i 访问城市 a_i 后接着访问城市 a_j(即,$x_{ij}=1$ 表示商人以顺序 i 访问城市 a_i 后接着访问城市 a_j,否则 $x_{ij}=0$),则TSP问题可以描述成如下优化问题

$$\min_{x_{ij}\in\{0,1\}} \sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} d_{ij}x_{ij}$$

$$\text{s.t.}\begin{cases}\sum_{j=1}^{n} x_{ij}=1, i=1,2,\cdots,n\\ \sum_{i=1}^{n} x_{ij}=1, j=1,2,\cdots,n\\ \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}=n\end{cases}$$

式中:s.t. 为约束条件。

TSP也是一个模拟进化计算(SEC)作为benchmark问题求解的典型问题。此问题中样本空间 $\Omega=\left\{\left(x_{ij}\right)_{n\times n}\in\{0,1\}^{n\times n}\middle|\sum_{j=1}^{n}x_{ij}=1,i=1,2,\cdots,n;\sum_{i=1}^{n}x_{ij}=1,j=1,2,\cdots,n;\sum_{i=1}^{n}\sum_{j=1}^{n}x_{ij}=n\right\}$;目标函数 f 是已知的,且可

收稿日期:2011-01-21

作者简介:陈冬华(1965—),男,副教授,硕士,研究方向为智能计算。

定义为

$$f\left(x_{ij}\right)=-\sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n}d_{ij}x_{ij}$$

TSP是典型的NP-hard问题,是组合优化领域研究中研究最多的问题之一,也是目前解决旅行优化领域里的研究热点。也有不少研究者对TSP进行了推广,如多旅行商问题[3-6],广义旅行商问题[7-14],中国旅行商问题[15],有向黑白旅行商问题[16]。

下面将TSP变形推广成全体旅行商问题(caboodle traveling salesman problem,CTSP)。CTSP定义如下:某个商人拥有 s 种货物 $B=(b_1,b_2,\cdots,b_s)$,其中 $b_j(j=1,2,\cdots,s)$ 表示第 j 种货物的数量,欲到 n 座城市 $A=(A_1,A_2,\cdots,A_n)$ 去推销这些商品,$A_i=(a_1^i,a_2^i,\cdots,a_s^i)$,其中 $a_j^i(i=1,2,\cdots,n;\ j=1,2,\cdots,s)$ 表示第 i 座城市 A_i 对这 s 种货物各自的需求数量,商人事先不知道各座城市需求情况但希望找到一座城市一次性销售掉所有的商品(沿途不卖),销售完商品后随即返回不再访问其它剩余城市,需要选择一条路线使得商人推销完商品后(每座城市仅能访问一次,走过所有的城市后没有销售完商品即生意失败也返回起点城市)回到起点且所花时间最短(假设商人在途中速度始终不变)。

显然,CTSP具有现实实用价值与意义,也有些其它领域的问题可以转化为CTSP,比如信息检索等,但目前为止,CTSP少有人进行研究。

2　GA、ACS算法与CTSP描述

遗传算法[17](genetic algorithm,GA)是一类仿生优化算法,具有快速随机的大范围全局搜索能力,很容易与其它算法结合,但对于系统中的反馈信息利用不够,当求解到一定范围时往往做大量无为的冗余迭代,求精确解效率低。因为GA不受函数连续性、可导性等条件的约束,并以其固有的并行性,使GA广泛应用于组合优化、图像处理、人工智能、机器学习、专家系统等很多领域。

蚁群系统(ant colony system,ACS)算法[18]是意大利学者M. Dorigo等人于1991年首先提出的。1996年,M. Dorigo等又应用该算法求解非对称TSP以及车间作业调度问题(job-shop scheduling problem,JSP),且对蚁群算法中初始化参数对其性能的影响作了初步探讨,该算法可以使得蚂蚁在行进过程中不再局限于按照已积累的信息浓度信息选择路径,而是可以做到利用已有信息与搜索新路径并重,通过适当提高解的随机性和多样性,从而大大增强了基本蚁群算法的鲁棒性。ACS算法具有分布式并行全局搜索能力,能较大程度避免候选解陷入局部极小并导致系统收敛到伪最优解从而停止进化,但存在初期信息匮乏求解速度慢的缺点。我国在蚁群算法领域的研究起步较晚,从公开发表的论文来看,国内最早研究蚁群算法的是东北大学控制仿真研究中心的张纪会博士与徐心和教授,以投稿日期为标准,时间是1997年10月。

为了使用智能算法对CTSP进行路径规划,CTSP定义重新描述为:某地区共有 n 座城市,其编号分别为 $1,2,\cdots,n$,旅行商从城市1出发去执行销售任务,旅行商在城市 i 完成其销售任务的概率为 p_i,p_i 与 $p_j(i\neq j)$ 相互独立($i=1,2,\cdots,n;\ j=1,2,\cdots,n$)。不论旅行商在城市 i 是否能完成其销售任务,其在城市 i 因尝试执行任务而造成的时延均为 t_i。对城市1而言,$p_1=0$,$t_1=0$。旅行商从城市 i 旅行到城市 j 所需的时间为 t_{ij}(从城市 i 旅行到城市 j 距离 $d_{ij}=vt_{ij}$,假设 v 为常数。所以 t_{ij} 也可以用 d_{ij} 来代替)。若旅行商在某城市完成任务,则可由该城市直接返回出发城市1,而无需继续访问其余城市;若未能完成任务,则旅行商需要继续旅行到另一座城市直至其任务完成或遍历所有城市均不能完成其任务即任务失败为止(途中每座城市至多访问一次)。要求制定一个旅行商路径计划,使得旅行商完成任务所需时间的期望值最小。

设城市 i 所需求的货物需求向量 $\boldsymbol{A}_i=(a_1^i,a_2^i,\cdots,a_s^i)$,其中 $a_j^i(i=1,2,\cdots,n;j=1,2,\cdots,s)$ 均为非负值,表示城市 i 需求某种货物 j 的数量。完成销售任务必需满足旅行商具有货物资源(供应向量)$\boldsymbol{B}=(b_1,b_2,\cdots,b_s)$,其中 $b_j(j=1,2,\cdots,s)$ 表示该销售任务的第 j 种货物的数量。那些需求向量的每一维分量均大于供应量的城市完成销售任务的概率较大,即城市的需求模式和旅行商销售任务的供给模式越匹配,其完成任务的概率也就越大。可以按照如下公式对旅行商在 i 城市完成任务的概率 p_i 进行预测

$$p_i=\frac{\left|A_iB^T\right|}{\left|A_i\right\|B\right|}=\frac{\left|\sum_{j=1}^{i}a_j^ib_j\right|}{\left|\sum_{j=1}^{i}\left(a_j^i\right)^2\left|\sum_{j=1}^{i}\left(b_j\right)^2\right|\right|}$$

式中:T表示向量转置。

解决了CTSP,可以给出旅行商在不同城市的最优旅行路线,保证旅行商可以在最短时间内销售完商品,这在实际生活中具有指导意义,并且在其它领域应用中发挥作用,如在网络信息库中进行某种信息检索时,对检索系统规划出检索路径等等。

3 混合智能算法(combined intelligent algorithm,CIA)

ACS算法在求解TSP问题时,只需考虑城市之间的距离。CTSP除了需要考虑城市间距离上路径延时之外,还需要考虑在各城市完成任务概率和销售时间。旅行商蚂蚁除了会倾向于以较高的概率选择花费时间短、外激素浓度高的城市间路径之外,还会优先考虑那些完成任务概率高、销售时间短的城市,这是因为旅行商蚂蚁在访问过若干完成任务概率高、销售时间短的城市之后很可能就已经完成了预定任务,由此即可直接返回出发点城市而无须继续访问其余城市。按照这一基本思想,旅行商蚂蚁选择下一城市的概率为

$$p_{ij}^k=\begin{cases}\dfrac{\tau_{ij}\eta_{ij}^{\beta}}{\sum\limits_{i\in allowed_k}\tau_{ij}\eta_{is}^{\beta}}, j\in allowed_k\\ 0,\text{其它}\end{cases}\tag{1}$$

式中:p_{ij}^k为旅行商蚂蚁k由城市i旅行到城市j的概率;τ_{ij}为城市i和城市j之间路径上的外激素浓度(初始时刻可将所有路径上的外激素浓度设为一个常数τ_0);β是参数;$allowed_k=\{1,2,\cdots,n\}\backslash tabu_k$为旅行商蚂蚁$k$下一步允许选择的城市;而$tabu_k$为旅行商蚂蚁$k$到目前为止已访问过的城市;$\eta_{ij}=\dfrac{p_j}{d_{ij}t_j}$为启发式信息;$t_j$为在城市$j$的时延。

ACS算法允许当前找到最优路径的蚂蚁更新外激素浓度,称之为全局更新规则,算法式子为

$$\tau_{ij}\leftarrow(1-\rho)\tau_{ij}+\rho\Delta\tau_{ij}^m\tag{2}$$

式中:$\Delta\tau_{ij}^m=\begin{cases}\dfrac{1}{T_{\text{mb}}},\text{若蚂蚁经过}i\text{到达}j\\ 0,\text{其它}\end{cases}$;$\rho$为0至1之间的常数;$T_{\text{mb}}$为完成销售任务的最短时间。

在系统尚未收敛到全局最优解前,当前最优解很可能对应着局部极小值,这很有可能使得整个蚁群系统陷入局部极小而停止进化,为了避免此种情况发生,可对全局更新规则进行推广:在蚁群系统中只有当前循环中最优的那一只蚂蚁能够进行全局更新,令最优的k只蚂蚁同时更新所经路径上的信息素浓度,则将公式(2)修改为

$$\tau_{ij}\leftarrow(1-\rho)\tau_{ij}+\sum_{m=1}^{k}\rho\Delta\tau_{ij}^m\tag{3}$$

$$\Delta\tau_{ij}^m=\begin{cases}\dfrac{1}{mT_m},\text{若蚂蚁经过}i\text{到达}j\\ 0,\text{其它}\end{cases}\tag{4}$$

式中:m为所对应的蚂蚁;T_m为第m最优解对应的蚂蚁完成任务的时间。

这样全局更新规则的推广使得解的多样性得到了适当提高,在很大程度上避免了候选解陷入局部极小并导致系统收敛到伪最优解而停止进化。并在此ACS算法基础上引入GA,基本思想是汲取两种算法的

优点,克服各自缺点,起到优势互补作用,本文针对CTSP提出一种新的混合智能算法(combined intelligent algorithm,CIA)。CIA在求解速度效率上优于ACS算法,在求解精确度上效率优于GA,是收敛速度快与求解效率高的比较好的一种启发式算法。

4　CIA描述与实例

CIA基本思路是首先采用GA,充分利用GA的快速性、随机性、全局收敛性,并把GA产生的结果作为有关问题的初始信息素分布,然后再采用ACS算法,在有一定初始信息素分布的情况下,利用ACS算法的并行性、正反馈性、求解精度效率高等特点。CIA运算如下:

步骤一:定义目标函数和适应度函数;初始化GA中相关参数;随机生成一组实数编码;置GC:=0,GC为GA算法运行次数。

步骤二:设置控制遗传代数t:=0。

步骤三:根据适应度函数选择X,Y;对X,Y进行OX交叉;对X,Y进行逆转变异。

步骤四:设置控制遗传代数标准T;若$t<T$,则置t:=t+1,转入步骤三,否则继续。

步骤五:设置GA控制运行次数GC_{max};若$GC<GC_{max}$(其中GC_{max}为GA最大运行次数),则转入步骤二,否则继续。

步骤六:生成若干组GA优化解;初始化ACS算法中相关参数;根据GA的优化解生成信息素初始分布,将m只旅行商蚂蚁置于n座城市;置NC:=0,NC为ACS算法运行次数。

步骤七:每只蚂蚁根据公式(1)状态转换规则旅行到下一座城市,并按公式(3)进行信息素局部更新;m只蚂蚁遍历n座城市后,按公式(4)进行信息素强度范围进行控制;将m只蚂蚁随机置于n座城市,对禁忌表和路径序列进行初始化。

步骤八:设置ACS算法控制运行次数NC_{max};若$NC<NC_{max}$(其中NC_{max}为ACS最大运行次数),则转入步骤七,否则继续。

步骤九:输出CIA的最优解。

应用CIA求解CTSP时,在求解初始阶段采用GA,充分利用GA的全局优化机制在较短时间内取得一组优化解,然后将这组优化解转化为对应路径上信息素的初始浓度,从而引导蚂蚁的寻找路径过程,减少了单独使用ACS算法时初始计算过程的盲目性与随机性。在构造解的过程中按照ACS算法的局部更新规则在所经过的路径上分泌信息素,并且每一代的前k个优胜解都会按照ACS算法的全局更新规则对所经过的路径进行信息浓度更新。经过若干代之后,那些延时短且沿途城市销售完成任务概率高、总时间费时短的路径上信息浓度会逐渐高于其它路径,后继蚂蚁在信息素指引下,会以更高的概率选择那些比较优化的路径,如此反复就构成了一个正反馈过程,这样会使得绝大多数蚂蚁首选优化路径,从而快速给出最优解。

为了验证CIA与GA及ACS算法的寻优效率,下面研究一个实例,如:求旅行商去16座城市销售的最佳路线。16座城市(城市编号为1至16)地理位置横坐标与纵坐标(x,y)分别为:1(38.24,20.42),2(39.57,26.15),3(40.56,25.32),4(36.26,23.12),5(33.48,10.54),6(37.56,12.19),7(38.42,13.11),8(37.52,20.44),9(41.23,9.10),10(41.17,13.05)11(36.08,−5.21),12(38.47,15.13),13(38.15,15.35),14(37.51,15.17),15(35.49,14.32),16(39.36,19.56)。采用Matlab2008a仿真平台,在Pentium4,2.8 GHz,512 MB内存的微机上进行仿真实验。求出的最优解(用城市编号顺序表示路线)为:1,14,13,12,7,6,15,5,11,9,10,16,3,2,4,8。3种算法的效果见表1。

表1　三种算法效果表

Tab.1　Effect of three algorithms

算法名称	最小迭代次数	平均运行时间/s	平均值	最优值
GA	959	7.82	193.75	193.12
ACS	944	7.63	193.64	193.12
CIA	827	6.89	193.20	193.12

实验表明，CIA比GA及ACS算法寻优效率更高、更好。

参考文献：

[1] 王有为.基于禁忌表的捕鱼搜索算法及其在旅行商问题中的实验研究[J]. 系统工程理论与实践，2008，2(2)：230-631.
[2] 刘新，刘任任，侯经川. 求解旅行商问题的整体优先算法[J]. 计算机应用，2007，5(5)：1204-1207.
[3] 李天龙，吕勇哉. 基于自组织优化算法的一类多旅行商问题[J]. 计算机应用，2010，2(2)：458-460.
[4] 党建武，靳蕃. 神经网络求解MTSP的应用研究[J]. 铁道学报，1997，19(2)：63-69.
[5] WA C E，HAN J，MANN R C. A neural network algorithm for the multiple traveling salesmen problem[J]. Biological Cybernetics，1989，61(1)：11-19.
[6] 代坤，鲁士文，蒋祥刚. 基于遗传算法的多人旅行商问题求解[J]. 计算机工程，2004，30(16)：139-141.
[7] 黄可为，汪定伟. 热轧计划中的多旅行商问题及其计算方法[J]. 计算机应用研究，2007，7(24)：43-57.
[8] HAYKIN S，LABORDERE A L. The record balancing problem：A dynamic programming solution of a generalized traveling salesman problem [J]. Real World Applications，1969(2)：43-49.
[9] SAKSENA J P. Mathematical mode of scheduling clients through welfare agencies[J]. Computers & Structures，1970(8)：185-200.
[10] SRIVASTAVA S S，KUMAR S，GARG R C，et al. Generalized traveling salesman problem through n sets of nodes[J]. Computers & Structures，1969(7)：97-101.
[11] 赵曦，叶和平. 广义旅行商问题及其求解[J]. 东莞理工学院学报，2007，5(14)：75-80.
[12] LIEN Y N，MA E. Transformation of the generalized traveling salesman problem into the standard traveling salesman problem [J]. Information Sciences，1993(74)：177-189.
[13] VLADIMIR D，ZORAN S. An efficient transformation of the generalized traveling salesman problem into the traveling salesman on digraphs[J]. Informatics and Computer Science，1997(192)：105-110.
[14] 余国兴，丁玉成，李涤尘. 平面多轮廓加工路径优化模型及其近似算法[J]. 西安交通大学学报，2004，1(38)：39-42.
[15] 王怡雯，丛爽. 用随机神经网络优化求解C-TSP[J]. 吉林大学学报：信息科学版，2004，4(22)：359-363.
[16] 江贺，张宪超，陈国良. 有向黑白旅行商问题[J]. 计算机学报，2007，3(3)：431-439.
[17] 吴微，周春光，梁艳春. 智能计算[M]. 北京：高等教育出版社. 2009：119-146.
[18] 徐宗本. 计算智能[M]. 北京：高等教育出版社. 2006：111-114.

CTSP and Combined Intelligent Algorithm

Chen Donghua

（School of Basic Sciences，East China Jiaotong University，Nanchang 330013，China）

Abstract：Traveling salesman problem (TSP)，a typical NP-hard problem，is one of the hot researching topics in the combined optimization area. It has already attracted many researchers from all areas. CTSP，a promotion of TSP deformation，is more complex than TSP，and has a wide range of applications. Genetic algorithm (GA) has the ability of conducting a stochastic global searching. However，using ability of feedback information is poor，convergence is slow，and its solving efficiency is low. Ant colony system (ACS) algorithm not only has the ability of parallel global searching，but also can avoid converging to a local minimal solution to possibility of stopping evolution. It lacks initial information and slow convergence. GA and ACS can be combined into CIA，which can be used to solve CTSP，having good properties of fully using information and fast convergence.

Key words：TSP；CTSP；genetic algorithm；ACS algorithm；combined intelligent algorithm