文章编号:1006-2475(2012)03-0001-04

一种求旅行商问题的离散粒子群算法

邓伟林[1],胡桂武[2]

(1. 广东轻工职业技术学院计算机系,广东 广州 510300; 2. 广东商学院数学与计算科学系,广东 广州 510320)

摘要:针对旅行商问题提出一种离散粒子群算法。算法重新定义了速度及其与粒子位置的相关算子,设计了“距离排序矩阵”(保存距离城市由近到远的其他城市的矩阵),并根据它生成可动态变化的优秀基因库来指导粒子高效地进行全局搜索。本文用 TSPLIB 中的部分案例进行实验,实验结果表明,该算法在求解旅行商问题上有很好的性能,并且具有很好的鲁棒性。

关键词:群体智能算法; 离散粒子群算法; 优化算法; 旅行商问题; 进化计算

中图分类号:TP301.6　　**文献标识码**:A　　**doi**: 10.3969/j.issn.1006-2475.2012.03.001

Discrete Particle Swarm Optimization Algorithm for TSP

DENG Wei-lin[1], HU Gui-wu[2]

(1. Department of Computing, Guangdong Industry Technical College, Guangzhou 510300, China;
2. Department of Mathematics and Computer Science, Guangdong University of Business Studies, Guangzhou 510320, China)

Abstract: A discrete particle swarm optimization algorithm is proposed for solving traveling salesman problem (TSP). It defines a newly velocity. It also defines some new operators between particle and velocity. A matrix, which keeps the sorted ascending edges by distance for each city is created, and a variable size of gene pool is implemented to direct the particle to search around the space. In this paper, some cases from TSPLIB are tested; the experiments results show that the algorithm has good performance and good rudeness for solving TSP.

Key words: swarm intelligence algorithm; discrete particle swarm optimization; optimization algorithm; traveling salesman problem; evolutionary computation

0 引　言

旅行商问题(TSP)可以简单地描述为:图 G = (V,E),V 是结点的集合,E 是 V 中结点相互连接的边的集合,E 中的每条边都有一个正数的距离 w(e),在 G 中寻找一条 Hamilton 回路 path,使得这条回路 path 的距离: $W(c) = \sum_{e \in E(path)} w(e)$ 最小。TSP 描述很简单,但是求解却很困难。TSP 问题已经被验证了是一个 NP 难的问题。它具有很高的实用意义和理论意义。在工程应用中,很多问题其实本质上就是 TSP 问题,比如:物流配送车调度问题、网络路由优化问题等。而在理论上,TSP 问题通常被研究者们用来衡量优化算法的有效性和性能的标准。

目前求解 TSP 问题的算法大致可以分两类:一类是局部启发式算法,比如 2-opt、3-opt、LK 算法等。这类算法的优点是寻找局部最优解效率非常高,但是由于过分依赖于问题,容易陷入局部最优;另外一类是智能优化算法,例如,遗传算法[1-5]、模拟退火算法[6-7]、蚁群算法[8]、粒子群算法[9-13]等。这类方法独立于问题,目前已经取得了些阶段性的成果,而粒子群算法是近年的一个研究热点。

1 粒子群优化算法

粒子群优化算法(Particle Swarm Optimization, PSO)最早是由 Kennedy 提出,是一种基于迭代的优化算法。PSO 的基本思想来源于对鸟类的群体行为进行建模和仿真研究结果的启发。在 PSO 中,每个优化问题的解都是搜索空间中的一只鸟,称之为“粒

收稿日期:2011-10-26
基金项目:广东省自然科学基金资助项目(06301003);广东轻工职业技术学院科研启动基金资助项目(KY200817)
作者简介:邓伟林(1980-),男,广东韶关人,广东轻工职业技术学院计算机系讲师,硕士,CCF 会员,研究方向:遗传算法,群体智能;胡桂武(1970-),男,湖南冷水江人,广东商学院数学与计算科学系教授,博士,CCF 高级会员,研究方向:人工智能,生物信息学,数据挖掘。

子”。所有的粒子都有一个由被优化的函数决定的适应值,每个粒子还有一个速度决定它们飞翔的方向和距离。PSO 初始化为一群随机粒子(随机解),然后通过迭代找到最优解。在每一次迭代中,粒子通过跟踪两个“极值”来更新自己。第一个就是粒子本身个体极值 P_i,另一个极值是全局极值 P_g。在找到这两个最优值时,粒子根据公式(1)来更新自己的速度和位置:

$$V_i(t+1)=wV_i(t)+c1rand()(P_i-X(t))+c2rand()(P_g-X_i(t))$$

$$X_i(t+1)=X_i(t)+V_i(t+1) \quad (1)$$

其中,$V_i(t)$ 是粒子 i 在第 t 代的速度,$X_i(t)$ 是粒子 i 第 t 代的位置,rand() 是介于(0,1)之间的随机数,c1、c2 是学习因子,w 是惯性因子。

目前,PSO 算法在很多连续优化问题中得到成功应用,而在离散域和非数值化问题上的研究和应用则相对滞后。文献[9]提出了一种求解 TSP 的粒子群算法,引入了交换子和交换序的概念,其算法核心是把速度定义为点的交换集。位置加上速度的运算,则定义为依次交换交换集(速度)中所有的变换子。该算法在设置 100 个粒子,迭代 20000 次可以得到 TSPLIB 中 14 个城市的最优解。虽然该算法求解能力不佳,但是也为 PSO 求解 TSP 做了一个新的尝试;文献[10]提出的算法沿用了文献[9]对速度的定义,对速度和位置算式进行了适当的改造,使得其交换序更有针对性,具体改造如下:

$$X_i^1(t+1)=X_i(t)+(w*V_i(t)) \quad (2)$$

$$X_i^2(t+1)=X_i^1(t+1)+(\alpha*and()*(P_i-X_i^1(t+1))) \quad (3)$$

$$X_i^3(t+1)=X_i^2(t+1)+(\beta*rand()*(P_g-X_i^2(t+1))) \quad (4)$$

该算法求解 14 个城市的 TSP,设置 100 粒子迭代 778 次可以收敛到最优解,算法效率有了一定的提高,但是求解效率仍然不理想。

2 求 TSP 的离散粒子群算法

文献[9]和文献[10]求解 TSP 效率不佳的主要原因是:文中定义的 PSO 算法均把速度定义为结点的交换集。因为,根据 TSP 问题的特点,TSP 求的是最短哈密顿回路,直接影响其适应度的是边而不是点,因此,应该将边而不是将点作为基本操作元素来处理 TSP 问题。基于这个基本思路,提出一种适合求解 TSP 问题的离散粒子群算法(下称 TSP-DPSO 算法)。算法对 TSP 问题的解采用整数序列编码:假设 $X=\{X_1,X_2,\cdots,X_N\}$(N 为城市数),则粒子 X 代表访问路径为 $X_1\rightarrow X_2\rightarrow\cdots\rightarrow X_N\rightarrow X_1$。算法对粒子速度及其与粒子位置的运算给出了新的定义。

2.1 相关定义

定义 1 粒子速度:粒子的速度是若干条边组成的集合。

定义 2 离散减法算子 -:个体最优解或者全局最优解 P_{best} 与个体 P 的减法“-”定义为:在 P_{best} 中存在但是在个体 P 中不存在的边的集合,运算的结果相当于是粒子的速度。

例如:P_{best} = [4 1 6 5 2 3], P = [4 3 1 5 6 2]。P_{best} 的边有[(4,1),(1,6),(6,5),(5,2),(2,3),(3,4)];P 的边有[(4,3),(3,1),(1,5),(5,6),(6,2),(2,4)],因此 P_{best} - P = [(4,1),(1,6),(5,2),(2,3)]。

定义 3 离散乘法算子⊗:参数 c 与速度 V 的⊗运算是以 c 为概率随机选取速度中若干条边组成新的边集合,生成新的速度。

定义 4 离散加法算子⊕:个体 P 与速度 V 的⊕运算就是在个体 P 中依次加入速度 V 中的边的过程。边的加入通过基因片断的翻转来实现。

例如:个体 P = [4 3 1 5 6 2],速度 V = [(3,2),(4,5)]。那么 P⊕V 就是要在个体 P 中依次加入边(3,2)和(4,5):先加入边(3,2),那么需要翻转 P 中的“1562”基因,变为“2651”,于是 P 变为[4 3 2 6 5 1];然后加入边(4,5),需要翻转基因“3265”,翻转为“5623”,最后得到个体[4 5 6 2 3 1],也就是 P⊕V 的运算结果。

2.2 粒子更新公式

标准粒子群算法的粒子更新加入了粒子的速度项,代表了粒子的全局搜索功能。在 TSP-DPSO 中,也可以给粒子随机生成一个速度项,但是这样可能会使得粒子进化进入一个过于盲目的状态,影响全局搜索的效率。注意到在 TSP 的最优路径的边来自于距离较近的城市间的连接比率较大。表 1 的统计表明 TSPLIB 中最优路径的绝大部分边都来自城市与此城市最近的 4 个城市之一的连接。

表 1 1/4 近距离城市间的连接占最优路径中边的比率

案例	Eli76	Eli101	Ch130	D198
比率(%)	96.0	97.0	92.3	90.9

因此,可以构造一个 N(N 为城市数)行(N-1)列的矩阵(命名为距离排序矩阵 sorted-edge)。矩阵第 i 行记录了与第 i 个城市距离由近到远的的 N-1 个城市。可从 sorted-edge 矩阵中选取其前 m(m≤N-1)列组成一个“短边库”,相当于是一个优秀基因库,随机从这个基因库中选取边组成粒子的速度,将会使得粒子在解空间的全局搜索更有效率和针对性。可以修

改参数 m 的值来调整粒子的全局搜索广度。在迭代前期,要求粒子有较强的全局搜索能力,避免过早收敛,这时需要一个较大的 m; 而在后期,则要求有较强的局部求精能力,以提高算法的效率,这时需要一个较小的 m。因此设计 m 的值在 N-1 和 4 之间的线性递减的策略:

$$m(t) = (N-1) - (N-5) * t/MaxGen \tag{5}$$

其中: t 为当前迭代的代数,MaxGen 为最大迭代代数。

基于以上新算子的定义和分析给出粒子更新公式:

$$\begin{cases} X_i(t+1) = X_i(t) \oplus (r1 \otimes (P_i - X_i(t))) & (\text{if}(r3 < \alpha)) \\ X_i(t+1) = X_i(t) \oplus (r2 \otimes (P_g - X_i(t))) & (\text{elseif}(r4 < \beta)) \\ X_i(t+1) = X_i(t) \oplus (r5 \otimes V_i(t)) & (\text{else}) \end{cases} \tag{6}$$

式(6) 中的变量说明:

(1) $i \in [1,2,...,N]$,其中 N 是粒子的个数,r1、r2、r3、r4、r5 是 0 到 1 的随机数。

(2) P_i 是第 i 个粒子的历史最优解,P_g 是全局最优解。

(3) $X_i(t)$ 是第 t 代,第 i 个粒子。

(4) $V_i(t)$ 的边从一个距离排序矩阵 sorted-edge 的前 m(t) 列随机选取。

从式(6) 可以看出,粒子更新由 3 个部分组成,第一部分是粒子以概率 α 进行⊕操作的"认知"部分,表示粒子本身思考,从自身吸取经验的过程。第二部分是粒子以概率(1 - α)β 进行⊕操作的"社会"部分,表示粒子从群体中学习。第三部分是以概率(1 - α)(1 - β)进行⊕操作的速度项部分,相当于是粒子的"变异"部分。粒子的"认知"部分和"社会"部分代表了粒子的局部求精能力,而粒子的"变异"部分代表了粒子的全局搜索能力。可以通过调整参数 α、β 均衡算法的局部求精能力和全局搜索能力。实验证明 3 个部分的叠加对算法性能并无明显改善,因此,为了简化算法,提高效率,在粒子的迭代更新中并没有叠加这 3 个部分的操作,而是采用可控概率的随机选取某一部分来更新粒子。

2.3　TSP-DPSO 算法流程

Step1　设置例子群大小 popsize,参数 α,β,最大迭代代数 Maxgen。随机生成粒子群 POP。

Step2　创建距离排序矩阵 sorted-edge。

Step3　求出粒子群子代个体。

Step3.1　利用式(5) 求出 m(t);

Step3.2　从 sorted-edge 的前 m(t) 列中随机生成若干条边,组成速度 $V_i(t)$;

Step3.3　利用式(6) 求出 $X_i(t+1)$;

Step3.4　如果 $X_i(t+1)$ 比 P_i 更优,则更新 P_i。

Step4　遍历粒子群中所有粒子的历史最佳 $P_i(1 \le i \le N)$,如果比全局最佳个体 P_g 距离小,则替代全局最佳。

Setp5　如果还没有满足结束条件或者还没到最大迭代代数,则转步骤 3,否则跳出循环,转步骤 6。

Step6　打印出收敛代数,收敛时间,全局最优个体和及其距离,画出收敛曲线、路径图。

TSP-DPSO 算法的在最坏的情况下,总循环次数和最大迭代次数 K,粒子群大小 M,城市数目 N 的平方的积成正比,所以该算法时间复杂度为 O(K* M* N* N)。

3　实验结果

为了验证算法的有效性,本文选取了 TSPLIB[14] 中部分案例,使用 Matlab R2008a 编程,在 P4 3G 的 CPU,1G 的 RAM 的 PC 上测试。

实验 1　burma14 的求解。

参数 α = β = 0.4,粒子数为 30,最大迭代次数为 100,测试 50 次。实验表明,算法都能稳定收敛,平均 34 代可以收敛到最优解。

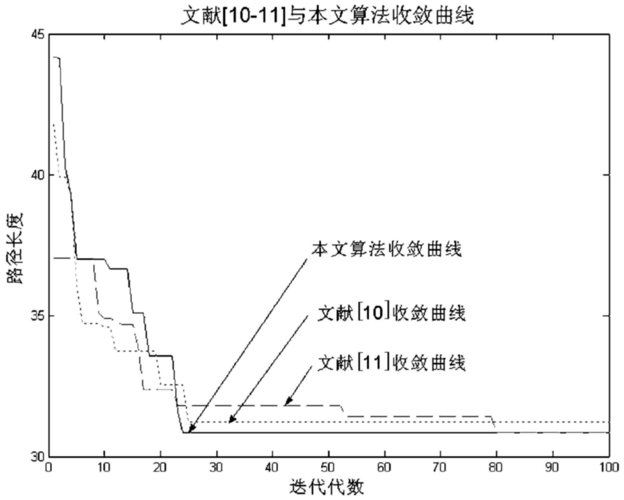

图 1　文献[10]和文献[11]算法与本文算法的收敛曲线

图 1 给出了其中一次实验的文献[10]和文献[11]算法与本文算法的收敛曲线。本文算法在 24 代收敛于最优值: 30.8785,文献[11]算法在第 80 代收敛于最优值,而文献[10]在 100 代内未收敛于最优解。表 2 是 50 次试验的平均求解性能。

表 2　文献[9-11]算法和本文算法的性能比较

性能比较	文献[9]	文献[10]	文献[11]	本文
粒子数	100	100	100	30
平均搜索代数	20000	778	82	34
运行时间(s)	62.5	27.4	1.8	0.8

从表 2 可以看出 TSP-DPSO 算法求解 burma14,在搜索搜索空间和运行时间上比文献[9-11]均有明

显优势。

其它案例的实验结果如表 3 所示: 最大迭代 5000 次,30 个粒子,计算 30 次,其他参数同实验 1。TSPLIB 最优解均由 TSPLIB 提供的路径求得。

表 3 本文算法对其他实例实验结果

实例	本算法平均解	本算法标准差	本算法最优解	TSPLIB 最优解	求得最佳解次数
eli51	429.1	3.2	428.9	430.0	24
st70	683.9	10.3	677.8	678.6	26
eli76	548.6	9.9	545.4	545.4	22
eli101	651.9	11.0	642.3	642.3	12
ch130	6165.4	28.6	6112.7	6100.9	0
a280	2598.0	30.5	2586.8	2586.8	8

从表 3 可以看出,本文算法对 eli51、st70 求解的结果均优于 TSPLIB 中提供的最佳路径,所求路径如图 2、图 3 所示。对 eli76、eli101、a280 可以求出 TSPLIB 最优解,对 ch130 求出的解也非常接近最优解,并且本算法所求解具有较小的标准差,具有很好的稳定性。

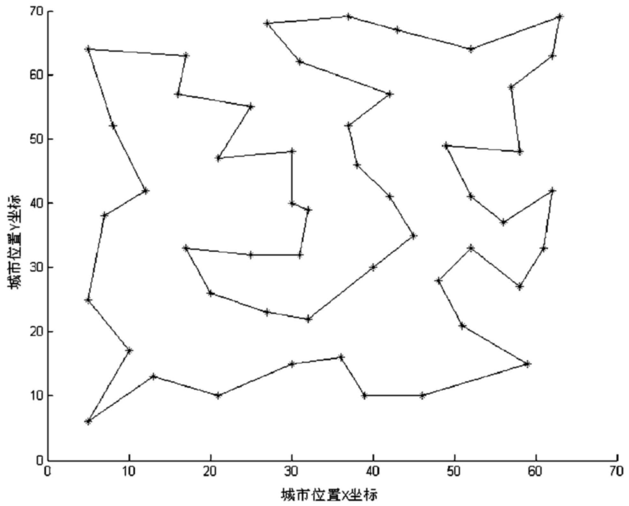

图 2 本文算法对 eli51 求得最佳路径图(路径长度:428.8718)

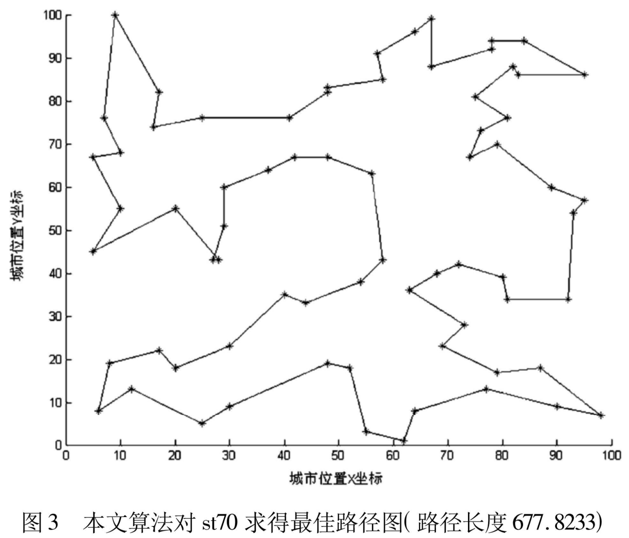

图 3 本文算法对 st70 求得最佳路径图(路径长度 677.8233)

4 结束语

本文根据 TSP 问题的特点,提出了一种适合求解 TSP 问题的离散粒子群算法。定义速度为边集合,重新定义了对粒子和速度的“-”,“$\otimes$”,“$\oplus$”算子,导出距离排序矩阵的前 m(t) 列,构造优秀基因库指导粒子进行高效的全局搜索,其中 m(t) 在迭代中递减可以控制粒子在迭代前期具有较强的全局搜索能力,在迭代后期具有较强的局部求精能力,使得算法既不会过早收敛到局部最优,同时又具有较高的收敛速度。实验证明该算法在求解 TSP 问题中比文献[9-11]中的算法更有效。本文算法可以高效率地求出 TSPLIB 的最佳解或满意解,并且算法有很小的标准差,具有很好的鲁棒性。

参考文献:

[1] Goldberg D E, Robert Lingle Jr. Alleles loci and the traveling sales-man problem [C]//Proc of the 1st Int. Conf. on Genetic Algorithms and Their Applications. Pittsburgh, 1985: 154-159.

[2] Grefenstette J J, Gopal R, Rosmaita B, et al. Genetic algorithm for the traveling salesman problem [C]//Proc of the 1st Int Conf on Genetic Algorithms and Their Applications. Pittsburgh, 1985: 160-168.

[3] Tao Guo, Michalewicz Z. Inver-over operator for the TSP [C]//Proceedings of the 5th Parallel Problem Solving from Nature. 1998: 803-812.

[4] 蔡之华,彭锦华,高伟,等,一种改进的求解 TSP 问题的演化算法[J]. 计算机学报,2005,28(5):823-828.

[5] 王海龙,周辉仁,魏颖辉. 基于遗传算法的一类多旅行商问题研究[J]. 计算机应用,2009,29(1):119-122.

[6] 杨理云. 用模拟退火算法求解旅行商问题[J]. 微电子学与计算机,2007,24(5):193-196.

[7] 江新姿,高尚,陈建忠. 求解旅行商问题的模拟退火蚁群算法[J]. 计算机工程与设计,2008,29(6):1491-1493.

[8] Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies [C]//Proc. 1st European Conf Artificial Life Plans. Elsevier, France, 1991: 134-142.

[9] Wang Kangping, Huang Lan, Zhou Chunguang, et al. Particle swarm optimization for traveling salesman problem [C]//Proceedings of the 2nd International Conference on Machine Learning and Cybernetics. Xi'an, 2003: 1583-1587.

[10] 肖健梅,李军军,王锡淮. 改进微粒群优化算法求解旅行商问题[J]. 计算机工程与应用,2004,40(35):50-52.

[11] 王东,吴湘滨,毛先成,等. 一种改进的求解 TSP 混合粒子群优化算法[J]. 计算机工程,2008,34(6):185-187.

[12] 叶安新. 基于改进粒子群优化算法的 TSP 问题研究[J]. 计算机与现代化,2011(4):1-3.

[13] 詹仕华,王长缨,钟一文. 求解 TSP 问题的伪贪婪离散粒子群优化算法[J]. 小型微型计算机系统,2011,32(1):181-184.