改进粒子群优化算法求解 TSP 问题

苏晋荣,王建珍
SU Jin-rong,WANG Jian-zhen

山西大学 商务学院 信息工程系,太原 030031
Department of Information Engineering,College of Business,Shanxi University,Taiyuan 030031,China
E-mail:sjr_su@tom.com

SU Jin-rong,WANG Jian-zhen.Improved particle swarm optimization for traveling salesman problem.Computer Engineering and Applications,2010,46(4):52-53.

Abstract: In allusion to particle swarm optimization being prone to get into local minimum,an improved particle swarm optimization algorithm is proposed.The algorithm draws on the thinking of the greedy algorithm to initialize the particle swarm. Two swarms are used to optimize synchronously,and crossover and mutation operators in genetic algorithm are introduced into the new algorithm to realize the sharing of information among swarms.This paper tests the algorithm with a Traveling Salesman Problem with 14 nodes.The result shows that the algorithm can break away from local minimum earlier and it has high convergence speed and convergence ratio.
Key words: Particle Swarm Optimization(PSO);Traveling Salesman Problem(TSP);greedy algorithm;crossover;mutation

摘 要:针对粒子群优化算法易陷入局部极值的缺点,提出一种改进粒子群算法,该算法借鉴贪婪算法的思想初始化种群,利用两个种群同时寻优,并将遗传算法中交叉和变异操作引入其中,实现种群间的信息共享。用 14 点 TSP 标准数据对算法性能进行了测试,结果表明该算法能够较早跳出局部最优,具有较高的收敛速度和收敛率。
关键词:粒子群优化算法;旅行商问题;贪婪算法;交叉;变异
DOI:10.3778/j.issn.1002-8331.2010.04.016 **文章编号:**1002-8331(2010)04-0052-02 **文献标识码:**A **中图分类号:**TP301.6

粒子群优化算法(Particle Swarm Optimization,PSO)是一种基于群体智能的进化算法，由 Eberhart 和 Kennedy 通过对鸟群、鱼群的模拟于 1995 年提出[1]。该算法利用群体优势为复杂问题的解决提供了新的方法。目前,PSO 算法在函数优化、神经网络训练、模糊系统控制、组合优化、入侵检测以及决策调度等多个领域得到广泛的应用[1-5]。PSO 算法有较强的全局搜索能力,但也有容易陷入局部极值导致地收敛精度低和不易收敛到全局最优的缺点。

旅行商问题(Traveling Salesman Problem,TSP)描述为:给定 n 个城市和两两城市之间的距离,求一条访问各城市一次且仅一次的最短路线。TSP 是著名的组合优化问题,是 NP-Hard 问题,常被用来验证智能启发式算法的有效性[6]。

文献[6]将交换子和交换序的概念引入 PSO 中,提出一种新的 PSO 算法,并将其成功用于一个 14 点 TSP 标准问题的求解中。基本 PSO 算法与目前解决 TSP 问题的经典算法相比,在解决问题的能力和速度上有一定的差距,但应用 PSO 解决 TSP 问题是一种崭新的尝试[7]。该文对基本 PSO 进行了改进,借鉴贪婪算法的思想产生初始种群,两个种群同时寻优,种群内部独立进行 PSO 运算,并以一定的变异概率进行变异,两个种群的个体最优之间进行交叉操作,提高种群质量,增加信息共享。仿真结果表明改进 PSO 算法具有较高的收敛速度和收敛率。

1 基本粒子群优化算法

设粒子的群体规模为 N,粒子当前的位置表示为 $X_i^k=(x_1^k,x_2^k,\cdots,x_n^k,\cdots,x_N^k)$,$x_n^k\in[l_n,u_n]$,$1\leqslant n\leqslant N$,$l_n$ 和 u_n 分别表示第 n 维空间的上下边界;当前速度表示为 $V_i^k=(v_1^k,v_2^k,\cdots,v_n^k,\cdots,v_N^k)$,$V_i^k$ 在最大值 $V_{\max}^k=(v_{\max,1}^k,v_{\max,2}^k,\cdots,v_{\max,n}^k,\cdots,v_{\max,N}^k)$ 和最小值 $V_{\min}^k=(v_{\min,1}^k,v_{\min,2}^k,\cdots,v_{\min,n}^k,\cdots,v_{\min,N}^k)$ 之间,粒子的速度和位置的更新方程如式(1)和式(2)所示:

$$V_i^{k+1}=\omega V_i^k+c_1r_1(P_i^k-X_i^k)+c_2r_2(P_g^k-X_i^k) \tag{1}$$

$$X_i^{k+1}=X_i^k+V_i^{k+1} \tag{2}$$

其中,c_1、c_2 为常数,称为学习因子,用来调节个体极值和全局极值的相对重要性;r_1、r_2 是(0,1)上均匀分布的随机数;P_i^k、P_g^k 分别表示粒子第 k 次迭代的个体极值点位置和全局极值点位置。

文献[6]提出交换子和交换序的概念,重新定义了基本 PSO

基金项目:山西省自然科学基金(the Natural Science Foundation of Shanxi Province of China under Grant No.20101135)。
作者简介:苏晋荣(1981-),女,助教,主要研究方向为智能优化算法;王建珍(1965-),女,教授,主要研究方向为计算机网络通信与信息安全。
收稿日期:2009-09-07 **修回日期:**2009-11-18

的速度更新和位置更新公式：

$$V_{id}'=V_{id}\oplus\alpha(P_{id}-X_{id})\oplus\beta(P_{gd}-X_{id}) \tag{3}$$

$$X_{id}'=X_{id}+V_{id}' \tag{4}$$

公式（3）中，α、β（$\alpha,\beta\in[0,1]$）为随机数，$(P_{id}-X_{id})$为粒子与个体极值的交换序，以概率α保留，$(P_{gd}-X_{id})$为粒子与全局极值的交换序，以概率β保留。公式（4）中，粒子的位置按照交换序V_{id}'进行更新。

2 改进粒子群优化算法

基本PSO算法求解TSP问题时，容易陷入局部最优，影响了算法的收敛率和收敛速度。针对PSO算法的这些缺点，进行了适当改进。

（1）初始种群的产生借鉴贪婪算法（Greedy Algorithm）的思想，每一步都取局部最优。这样产生的初始种群全局最优值已经比较接近问题的解，因此可以节省搜索时间，提高算法收敛速度。

（2）粒子以一定概率进行变异。贪婪算法产生的初始种群粒子质量较高，但是，与随机产生的种群相比，贪婪算法产生的种群多样化性能较差，这将直接影响算法的全局搜索性能。因此，在粒子进化过程中，引入变异操作来增加种群的多样化，从而增强算法的全局搜索能力。粒子是否进行变异，取决于变异阈值m，m为大于最小误差的值，如果两次进化适应度之差小于m且大于最小误差，说明此次搜索粒子群没有明显进化且不满足结束条件，此时对粒子进行变异来增加种群的多样化。

（3）两个种群同时寻优，种群内部独立进行PSO进化，种群个体最优之间以一定概率进行交叉。两个种群同时寻优可以减小算法陷入局部最优的概率，种群间个体最优的交叉能够增强两种群间以及粒子间的信息共享，及时传递最优值信息，提高粒子向更好解进化的速度。选择种群的个体最优进行交叉相当于生物界少数优良个体进行交叉，有利于产生优良的后代，并保证了父代优良品性的繁衍，符合生物界优胜劣汰的进化机制，因此，交叉产生的个体最优有利于种群的进化。

另外，采用带有惯性权重ω的速度更新公式，并令其在进化过程中线性递减。

$$V_{id}'=\omega V_{id}\oplus\alpha(P_{id}-X_{id})\oplus\beta(P_{gd}-X_{id}) \tag{5}$$

改进算法步骤如下：

步骤1 用贪婪算法产生两个初始种群A群和B群，随机产生两个基本交换序作为两个种群的初始速度，设定惯性权重ω、变异阈值m以及最小误差；

步骤2 如果满足结束条件（一个种群两次进化适应度之差小于最小误差），转步骤6；

步骤3 如果满足变异条件，则进行变异；

步骤4 以一定概率将A群粒子的个体最优与B群粒子的个体最优进行交叉，产生新的个体最优；

步骤5 两个种群按公式（5）和公式（4）更新粒子的速度和位置，产生两个新的全局最优，转步骤2；

步骤6 停止搜索，显示结果。

3 算法测试

这里采用14点的TSP标准问题对算法有效性进行了验证（问题来源：http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/tsp/），表1为14点TSP问题的描述，目前已知14点TSP问题的最好解为30.878 5，巡回路径为1–10–9–11–8–13–7–12–6–5–4–3–14–2–1。

表1　14点TSP问题

序号	1	2	3	4	5	6	7
X	16.47	16.47	20.09	22.39	25.23	22.00	20.47
Y	96.10	94.44	92.54	93.37	97.24	96.05	97.02
序号	8	9	10	11	12	13	14
X	17.20	16.30	14.05	16.53	21.52	19.41	20.09
Y	96.29	97.38	98.12	97.38	95.59	97.13	94.55

采用Matlab6.1为编程工具，实验环境为Pentium 4，2.93 GHz CPU，512 MB RAM，Windows XP系统。为便于比较，将该文算法与基本PSO算法、遗传算法分别连续进行30次实验。几种算法中，种群的粒子数均为50，遗传算法采用顺序交叉，交叉概率为0.9，变异概率为0.1。速度交换序的最大交换子个数为7，α、β为（0，1）的随机数，惯性权重ω从1.4线性递减到0.5，最大迭代次数2 000，变异阈值m为1E–4，最小误差为1E–10，交叉概率为0.9。实验结果如表2所示，得到的最优结果巡回路径如图1所示，巡回路径与目前已知最好解一致。

表2　实验结果比较

方法	最优值	最差值	收敛率/（%）	平均迭代次数
基本PSO	30.878 5	31.819 4	20	231.6
遗传算法	30.878 5	–	100	52.4
该文PSO	30.878 5	–	100	33.5

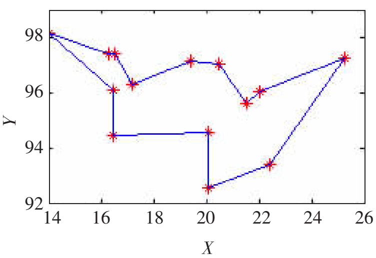

图1　算法获得最优结果（路径长度：30.878 5）

表2中，收敛率是30次实验中，算法收敛到最优值30.878 5的次数与实验次数30之比；由表2数据得知，该文PSO算法与基本PSO相比较，收敛率和收敛速度均有明显提高；与目前较成熟的遗传算法相比，二者收敛率相同，但是该文算法的收敛速度比遗传算法略有提高。

30次实验中，该文PSO算法有几次在早期陷入局部极值31.208 8，但均在100代左右跳出局部极值，找到目前最优值30.878 5，可见，改进算法在一定程度上克服了陷入局部最优的缺陷。另外，由贪婪算法产生的初始种群对提高算法的收敛速度起了重要作用，算法曾在第2代就收敛到最优值30.878 5。表3为贪婪算法和随机产生两种方法产生初始种群全局最优值平均值的比较，随着种群规模的增加，全局最优值逐渐减小。种群数为50时，贪婪算法产生的全局最优与目前的最好解30.878 5已经非常接近，仅差1.004，这在很大程度上减少了算法的迭代次数，提高了算法的收敛速度。

表3　两种方法产生初始种群全局最优值平均值比较

种群数	随机产生	贪婪算法
20	49.432 0	32.417 9
30	49.165 2	32.012 2
50	46.795 4	31.882 5

（下转75页）