第 29 卷第 8 期
2012 年 8 月
计 算 机 应 用 研 究
Application Research of Computers
Vol. 29 No. 8
Aug. 2012

融合可行基规则的粒子群优化算法及其应用

罗彩君
（陕西职业技术学院 计算机科学系，西安 710100）

摘　要：基本粒子群优化算法对于离散的优化问题处理不佳，容易陷入局部最优。针对基本粒子群优化算法处理离散型优化问题时的缺陷，提出了一种融合可行基规则的改进型粒子群优化算法，并用该算法求解车辆路径问题。实验结果表明，该算法的优化性能和求解精度均优于其他文献算法，在求解车辆路径问题中具有较高的应用价值。

关键词：粒子群优化算法；可行基规则；车辆路径问题
中图分类号：TP18　　**文献标志码：**A　　**文章编号：**1001-3695(2012)08-2909-03
doi:10.3969/j.issn.1001-3695.2012.08.028

Novel particle swarm optimization algorithm with feasibility-based rule and its application

LUO Cai-jun
(*Dept. of Computer Science, Shaanxi Vocational & Technical College, Xi'an* 710100, *China*)

Abstract: Basic particle swarm optimization algorithm falls into local optimum with little hindrance when it optimizes the discrete problem. To improve the performance of the standard PSO algorithm for solving discrete optimization problems, this paper proposed a novel particle swarm optimization algorithm with feasibility-based rule method. This paper used the proposed algorithm to solve the vehicle routing problem. Simulations show that proposed algorithm can obtain more powerful optimizing ability and higher optimizing precision for solving the vehicle routing problem than literature algorithms, and it has a good application value.

Key words: particle swarm optimization algorithm; feasibility-based rule; vehicle routing problem

0　引言

美国社会心理学家 Kennedy 博士和电气工程师 Eberhart 博士于 1995 年提出的粒子群优化算法（PSO）[1]，是通过模拟鸟群觅食行为而发展起来的一种基于群体协作的随机搜索算法，也是一种基于群体智能（swarm intelligence, SI）理论的优化算法。基本 PSO 算法由于具有参数少、收敛速度快等优点，非常适合于实值型数据处理，它在函数优化、神经网络训练、模式分类、调度优化、模糊系统控制等工程领域得到了广泛的应用。但是基本 PSO 算法对于离散的优化问题处理不佳，容易陷入局部最优。基于此，国内外研究者提出了多种改进型混合粒子群优化算法，文献[2]提出一种改进的变参数粒子群优化算法，该方法以进化状态因子计算策略和进化状态估计模型为基础，引入了算法参数控制和变异算子，提高了算法的收敛速度和全局优化能力；文献[3]提出一种基于排异竞争机制的粒子群优化算法，算法取消传统 PSO 算法中的全局最优值 gbest，通过设定竞争区域，使得当前种群中所有粒子和上一代种群中的精英粒子一同参与竞争，并通过对排异策略的动态调整，提高了算法后期的收敛速度和精度；文献[4]提出了一种基于进化状态估计的自适应粒子群算法，该算法综合考虑粒子演化以及粒子空间分布等信息，根据进化状态动态调整算法参数，具有很好的自适应性能；文献[5]提出一种新的云自适应粒子群优化算法（CPSO），此算法利用云滴具有随机性、稳定倾向性等特点，结合不同粒子与全局最优点的距离动态变化的性质，利用云自适应调整计算惯性权重。

不同于已有文献，本文把粒子群优化算法的全局寻优性和 Hooke-Jeeves 方法[6]的快速收敛性结合起来，提出了一种融合可行基规则的改进型粒子群优化算法，并用该算法求解车辆路径问题。

1　融合可行基规则的改进型粒子群优化算法

1.1　基本粒子群算法

PSO 模拟鸟群的捕食行为：一群鸟在随机搜索食物，在这个区域里只有一块食物；所有的鸟都不知道食物在哪里，但是它们知道当前的位置离食物还有多远；找到食物的最优策略就是搜寻目前离食物最近的鸟的周围区域。PSO 从这种模型中得到启示并用于解决优化问题。PSO 中，每个优化问题的解都是搜索空间中的一只鸟，称之为粒子。所有的粒子都有一个由被优化的函数决定的适应值（fitness value），每个粒子还有一个速度决定它们飞翔的方向和距离，然后粒子们就追随当前的最优粒子在解空间中搜索。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所找到的最优解，即个体极值 pbest；另一个极值是整个种群目前找到的最

收稿日期：2012-02-08；**修回日期：**2012-03-28
作者简介：罗彩君（1979-），女，湖南桂东人，讲师，硕士，主要研究方向为人工智能、粒子群算法等（lcjsiny@163.com）.

优解,即全局极值 gbest;另外也可以不用整个种群而只是用其中一部分最优粒子的邻居,那么在所有邻居中的极值就是局部极值。

PSO 算法可以表述为:随机初始化一个数量为 m 的粒子群,其中,迭代次数为 n,第 i 个粒子的位置为 $X_i=(x_{i1},x_{i2},\cdots,x_{in})$,速度 $V_i=(v_{i1},v_{i2},\cdots,v_{in})$。每一次迭代,粒子通过个体极值 $P_i=(P_{i1},P_{i2},\cdots,P_{in})$ 和全局极值 $G=(g_1,g_2,\cdots,g_n)$ 来不断更新自己的速度和位置,从而达到寻优。更新公式如下:

$$v_{k-1}=c_0v_k+c_1(\text{pbest}_k-x_k)+c_2(\text{gbest}_k-x_k) \tag{1}$$

$$x_{k+1}=x_k+v_{k-1} \tag{2}$$

其中:v_k 是粒子的速度;x_k 是当前粒子的位置;pbest_k 是粒子所找到的最优解的位置;gbest_k 是种群的最优解的位置;c_0、c_1、c_2 表示群体认知系数,c_0 一般取(0,1)间的随机数,c_1、c_2 一般取(0,2)间的随机数。

1.2 可行基规则

众所周知,惩罚函数法由于其简单且易实施一直是最流行的约束处理技术。约束优化问题的精确罚函数可构造为

$$F(x)=f(x)+\text{viol}(x) \tag{3}$$

$$\text{viol}(x)=M\sum_{j=1}^{N}[\max(g_j(x),0)] \tag{4}$$

这里称 viol(x)为约束偏差函数,其中 M 是一个适当大的正数。

由于在惩罚函数中同时考虑了目标函数和约束偏差函数,这种方法表现好坏与惩罚因子的选取有着直接的关系。然而选择适当的罚因子一般不是一件容易的事情,它与所讨论的问题有关。为了避免罚因子的选取,本文采用可行基规则[7]处理约束条件,具体叙述如下:

假设 $P_i(k)$ 代表第 i 粒子在第 k 代的历史最好位置 p_{best},$X_i(k+1)$ 代表第 i 粒子在 $k+1$ 代新产生的位置。按如下规则(可行基规则)对 $P_i(k)$ 进行更新,即在下列任一情形下取 $P_i(k+1)=X_i(k+1)$。

规则 1 $P_i(k)$是不可行的,但 $X_i(k+1)$是可行的。

规则 2 $P_i(k)$ 和 $X_i(k+1)$ 都是可行的,但 $f(X_i(k+1))<f(P_i(k))$。

规则 3 $P_i(k)$和 $X_i(k+1)$都是不可行的,但 $\text{viol}(X_i(k+1))<\text{viol}(P_i(k))$。

类似地,关于所有粒子的历史最好位置 g_{best},在每一代按照上述规则进行更新。

1.3 改进型粒子群优化算法

本文把粒子群优化算法的全局寻优性和 Hooke-Jeeves 方法的快速收敛性结合起来,利用可行基规则来避免惩罚函数法和乘子法的缺点,提出了一种改进型粒子群优化算法。该算法步骤如下:

a)随机初始化一群粒子的位置和速度。

b)以 $f(x)$评价每个粒子的适应度函数,即分别对每个粒子求问题的目标函数值。

c)以当前最好粒子位置 p_g 为初始点,使用 Hooke-Jeeves 搜索法进行优化计算求出最优解 x^*,并令 $p_g=x^*$。

d)根据式(3)和(4)更新每个粒子的速度和位置。

e)对每个粒子,将其经历过的最好位置 p_{best}按可行基规则更新。

f)对每个粒子,将其全局所经历的最好位置 p_g 按可行基规则更新。

g)如果没有达到结束条件(通常为足够好的函数值或达到一个预先给定的最大迭代次数或最优解停滞不再变化),则返回步骤 b)。

2 车辆路径问题的数学模型

车辆路径问题(vehicle routing problem,VRP)由 Dantzig 等人于 1959 年首次提出,它是指对一系列发货点(或收货点)组成适当的行车路径,使车辆有序地通过它们,在满足一定约束条件的情况下,达到一定的目标(诸如路程最短、费用最小,耗费时间尽量少等)[8]。约束条件一般有货物需求量、发送量、交发货时间、车辆容量限制,时间限制等;优化目标一般有路程最短、成本最低、时间最少等。

车辆路径问题描述如下:配送中心需要向 L 个客户(1,2,…,l)送货,第 i 个客户的货运量为 $g_i(i=1,2,\cdots,l)$,每辆车的载重量为 q,并且 $g_i<q$。求满足货运需求的最短行程路线。

本文建立的数学模型为:c_{ij}表示点 i 到点 j 的运输成本,如时间、路程、花费等;配送中心的编号为 0,客户的编号为 $i(i=1,2,\cdots,l)$,车辆编号为 k,最多可用 m 辆车,定义变量如下:

$$x_{ijk}=\begin{cases}1 & \text{车 } k \text{ 由 } i \text{ 驶向 } j\\ 0 & \text{else}\end{cases} \tag{5}$$

$$y_{ik}=\begin{cases}1 & \text{点 } i \text{ 的货运任务由 } k \text{ 完成}\\ 0 & \text{else}\end{cases} \tag{6}$$

车辆的路径问题是有约束的整数规划问题,即每个车辆路径上客户的任务之和不得超过该车辆的容量。完成服务时总的最小运输成本为

$$\min Z=\sum_{i=0}^{l}\sum_{j=0}^{l}\sum_{k=1}^{m}c_{ij}x_{ijk}+R\sum_{k=1}^{m}\max(\sum_{i=1}^{1}g_iy_{ik}-q,0) \tag{7}$$

设被调用的汽车都满足负载要求,表达式如下:

$$\sum_{i=1}^{1}g_iy_{ik}\leqslant q \quad k=1,2,\cdots,m \tag{8}$$

设每个客户仅由一辆车来负责配送并且仅被服务一次,表达式如下:

$$\sum_{k=1}^{m}y_{ik}=\begin{cases}1 & i=1,2,\cdots,l\\ m & i=0\end{cases} \tag{9}$$

模型中到达和离开每个客户的车辆相同,表达式如下:

$$\sum_{i=0}^{l}x_{ijk}=y_{ik} \quad j=1,2,\cdots,l;k=1,2,\cdots,m \tag{10}$$

$$\sum_{j=0}^{l}x_{ijk}=y_{ik} \quad i=1,2,\cdots,l;k=1,2,\cdots,m \tag{11}$$

设变量取值约束为 0 或 1,表达式如下:

$$x_{ijk}=0 \text{ 或 } 1 \quad i,j=0,1,\cdots,l;k=1,2,\cdots,m \tag{12}$$

$$y_{ik}=0 \text{ 或 } 1 \quad i=0,1,\cdots,l;k=1,2,\cdots,m \tag{13}$$

3 本文算法求解车辆路径问题的应用实例

为了验证本文算法求解车辆路径问题的性能,进行了两组实验。实验程序是用 VC++ 6.0 进行编程,在 Inter Core™2 Quad CPU 2.66 GHz、3 GB 内存、Windows XP SP3 的主机上运行的。实验过程如下:

a)本文算法与基本粒子群优化算法的对比实验。设配送中心的坐标为(14.5,13.0),表 1 为需求点的坐标数据以及客户的需求量数据,每辆车的最大装载量为 8.5 t,实验数据如表 1 所示。各个需求点之间以及需求点与配送中心的距离利用

距离公式 $d_{ij}=\sqrt{(x_i-x_j)^2(y_i-y_j)^2}$ 计算。

表 1　实验数据

客户	坐标	需求	客户	坐标	需求
1	(12.8,8.5)	0.1	11	(6.7, 16.9)	0.9
2	(18.4,3.4)	0.4	12	(14.8,2.6)	1.3
3	(15.4,16.6)	1.2	13	(1.8, 8.7)	1.3
4	(18.9,15.2)	1.5	14	(17.1,11.0)	1.9
5	(15.5,11.6)	0.8	15	(7.4,1.0)	1.7
6	(3.9,10.6)	1.3	16	(0.2,2.8)	1.1
7	(10.6,7.6)	1.7	17	(11.9,19.8)	1.5
8	(8.6, 8.4)	0.6	18	(13.2,15.1)	1.6
9	(12.5,2.1)	1.2	19	(6.4,5.6)	1.7
10	(13.8,5.2)	0.4	20	(9.6,14.8)	1.5

两种算法的最优解平均值随进化代数的变化曲线如图 1 所示；两种算法的一次最优解随进化代数的变化曲线如图 2 所示。

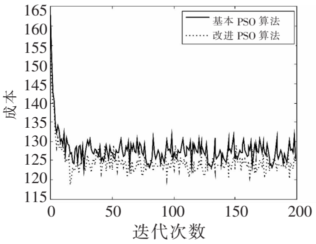

图 1　两种算法最优解平均值随进化代数的变化曲线

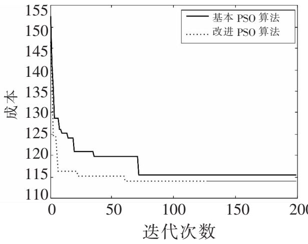

图 2　两种算法一次最优解随进化代数的变化曲线

从图 1 和 2 可以看出，本文算法的收敛速度明显高于基本 PSO 算法，物流配送的成本也低于基本 PSO 算法。这是由于本文算法把粒子群优化算法的全局寻优性和 Hooke-Jeeves 方法的快速收敛性结合起来，利用可行基规则来避免惩罚函数法和乘子法的缺点，促使粒子跳出局部极值区间，更容易在全局范围内搜索到最优值，提高了算法的收敛速度和精度。

b）本文算法与遗传算法的对比实验。测试采用 Solomon 的 r1 和 rc1 两类 benchmark 数据（数据来自 http://web.cba.neu.edu/~msolomon/problems.htm），r 类数据集节点的地理位置为随机生成，rc 类数据集的每个算例的一部分顾客需求是随机生成；另一部分顾客需求在地理位置或者服务时间窗上具有聚集特征。每个算例均独立运行 10 次，记录最好解。每个解包括使用的车辆数、总路径长度和平均运行时间。两种算法的求解结果如表 2 所示，加粗的字体显示了较好的解和较快的求解时间。

表 2　本文算法与遗传算法求解车辆路径问题结果的比较

测试数据	遗传算法			本文算法		
	车辆数目/辆	路径长度/km	耗时/s	车辆数目/辆	路径长度/km	耗时/s
r101	19	1 652.52	**53.45**	**19**	**1 650.80**	1 797.42
r102	17	1 524.98	**54.67**	**17**	**1 490.76**	2 052.45
r103	14	1 398.05	**56.98**	**14**	**1 224.41**	2 602.71
r104	11	1 234.24	**59.15**	**10**	**1 021.03**	2 834.58
r105	14	1 426.87	**57.10**	**14**	**1 385.28**	2 334.16
r106	13	1 356.95	**56.65**	**13**	**1 244.50**	2 553.87
r107	12	1 343.72	**57.21**	**11**	**1 091.68**	2 902.44
r108	11	1 191.11	**59.29**	**10**	**969.47**	2 974.27
r109	13	1 356.91	**56.05**	**12**	**1 153.89**	2 601.46
r110	13	1 348.06	**56.57**	**11**	**1 112.46**	2 759.56
r111	12	1 365.91	**56.72**	**11**	**1 098.59**	2 844.26
r112	11	1 239.85	**56.09**	**10**	**986.35**	3 130.37
rc101	15	1 692.95	**57.32**	**15**	**1 637.28**	2 073.48
rc102	14	1 574.51	**58.24**	**14**	**1 470.93**	2 396.90
rc103	13	1 452.79	**58.15**	**12**	**1 293.66**	2 464.38
rc104	12	1 328.30	**56.93**	**10**	**1 159.12**	2 708.69
rc105	**14**	**1 727.40**	**55.30**	15	1 537.45	2 451.61
rc106	14	1 581.05	**54.33**	**13**	**1 382.84**	2 459.04
rc107	13	1 527.35	**54.48**	**11**	**1 249.40**	2617.51
rc108	12	1 495.46	**54.17**	**11**	**1 145.42**	2 858.49

从表 2 的对比结果可以看出，本文算法耗时较长，这是因为 Hooke-Jeeves 方法搜索需要在个体的基础上搜索大量的邻域结构，从而增加了程序的运行时间。但是，从求解质量上看，除算例 rc105 之外，本文算法均明显优于遗传算法，说明本文算法的局部搜索策略可以显著提高算法的寻优能力。

4　结束语

本文针对基本粒子群优化算法处理离散型优化问题时易陷入局部极值、收敛精度不高的缺点，把粒子群优化算法的全局寻优性和 Hooke-Jeeves 方法的快速收敛性结合起来，提出了一种融合可行基规则的改进型粒子群优化算法，并用该算法求解车辆路径问题。实验表明，新算法能有效避免一般粒子群优化算法出现的早熟收敛问题，具有较强的全局搜索能力和较好的收敛速度，能有效地解决离散型优化问题。对车辆路径问题的实验结果表明，本文算法求解的速度和精度均优于改进前的算法，在求解车辆路径问题中具有较高的应用价值。下一步将重点优化算子，以进一步提高算法的效率。

参考文献：

[1] KENNEDY J，EBERHERT R. Particle swarm optimization[C]//Proc of IEEE International Conference on Neural Networks. Piscataway：IEEE Press，1995：1942-1948.

[2] 赵成业，闫正兵，刘兴高. 改进的变参数粒子群优化算法[J]. 浙江大学学报：工学版，2011，45(2)：2099-2102.

[3] 谭阳，唐德权，全惠云. 一种排异竞争的粒子群优化算法[J]. 系统仿真学报，2011，23(12)：2635-2641.

[4] ZHAN Zhi-hui，ZHANG Jun，LI Yun，*et al*. Adaptive particle swarm optimization [J]. IEEE Trans on Systems，Man and Cybernetic，2009，39(6)：1362-1391.

[5] 张艳琼. 改进的云自适应粒子群优化算法[J]. 计算机应用研究，2010，27(9)：3250-3252.

[6] LIU Ye-qing，LIU San-yang，GU Ming-tao. Hooke-Jeeves algorithm for linear support vector machine [J]. Journal of Systems Engineering and Electronics，2010，21(1)：138-141.

[7] HE Qie，WANG Ling. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization [J]. Applied Mathematics and Computation，2007，186(2)：1407-1422.

[8] EKSIOGLU B，VURAL A V，REISMAN A. The vehicle routing problem：a taxonomic review [J]. Computers & Industrial Engineering，2009，57(4)：1472-1483.