

文章编号: 1000-6788(2004)04-0130-06

带时间窗车辆路径问题的粒子群算法

李 宁^{1,2}, 邹 彤¹, 孙德宝¹

1. 华中科技大学控制科学与工程系, 湖北 武汉 430074; 2 武汉理工大学计算机科学与技术学院, 湖北 武汉 430070)

摘要: 将粒子群算法(PSO)应用于带时间窗车辆路径优化问题(VRPTW), 构造车辆路径问题的粒子表达方法, 建立了此问题的粒子群算法, 并与遗传算法作了比较. 实验结果表明, 粒子群算法可以快速、有效求得带时间窗车辆路径问题的优化解, 是求解带时间窗车辆路径问题的一个较好方案.

关键词: 车辆路径问题; 粒子群算法; 优化

中图分类号: O 221. 1; U 116. 2

文献标识码: A

Particle Swarm Optimization for Vehicle Routing Problem with Time Windows

L IN ing^{1,2}, ZOU Tong¹, SUN De-bao¹

(1. Department of Automatic Control, Huazhong University of Science & Technology, Wuhan 430074, China; 2 School of Computer Science & Technology, Wuhan University of Technology, Wuhan 430070, China)

Abstract This paper introduces a proposal to extend the heuristic called "Particle Swarm Optimization"(PSO) to deal with the Vehicle Routing Problem with Time Windows (VRPTW), and proposes a novel Particle presentation for the vehicle routing problem. The PSO is compared with GA in the same VRPTW in experiments. Experimental results indicate that the PSO can effectively and quickly get optimal resolution of VRPTW, so it is proved to be an effective method for VRPTW.

Key words: vehicle routing problem; particle swarm optimization; optimization

1 引言

车辆路径问题(Vehicle Routing Problem, VRP)由Dantzig 和 Ramser 于1959年首次提出的, 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)^[1], 属于完全NP 问题, 在运筹、计算机、物流、管理等学科均有重要意义.

带时间窗的车辆路径问题(Vehicle Routing Problem With Time Windows, VRPTW)是在VRP 问题上加了客户要求访问的时间窗口. 由于在现实生活中许多问题都可以归结为VRPTW 问题来处理(如邮政投递、火车及公共汽车的调度等), 其处理的好坏将直接影响到企业的服务质量, 所以对它的研究越来越受到人们的重视. 先后出现了一般启发式算法和神经网络、遗传算法、禁忌搜索和模拟退火等智能化启发式算法, 也取得了一些较好的效果^[1].

粒子群算法(PSO, Particle Swarm Optimization)^[2]是最近出现的一种模拟鸟群飞行的仿生算法, 有着个体数目少、计算简单、鲁棒性好等优点, 在各类多维连续空间优化问题上均取得非常好的效果^[3]. 本文将 PSO 应用于车辆路径问题求解中, 取得了很好的效果.

收稿日期: 2003-04-30

资助项目: 航天技术创新基金项目

作者简介: 李宁(1972-), 男(汉), 湖北京山人, 博士研究生, 主要研究方向: 系统工程、人工智能、演化计算、人工生命; 孙德宝(1941-), 男(汉), 湖北云梦人, 教授, 博士生导师, 研究方向: 人工智能、信号处理等

2 有时间窗车辆路径问题的描述及数学模型

有时间窗车辆路径问题一般描述为: 有一个中心仓库, 拥有车 K 辆, 容量分别为 q_k ($k = 1, 2, \dots, K$); 现有 L 个发货点运输任务需要完成, 以 $1, 2, \dots, L$ 表示. 第 i 个发货点的货运量为 g_i ($i = 1, 2, \dots, L$), ($\max(g) \leq \max(q_i)$), 完成发货点 i 任务需要的时间 (装货或卸货) 表示为 T_i , 且任务 i 且必须在时间窗口 $[ET_i, LT_i]$ 完成, 其中 ET_i 为任务 i 的允许最早开始时间, LT_i 为任务 i 的允许最迟开始时间. 如果车辆到达发货点 i 的时间早于 ET_i , 则车辆需在 i 处等待; 如果车辆到达时间晚于 LT_i , 任务 i 将被延迟进行.

求满足货运要求的运行费用最少的车辆行驶线路. 此问题称之为带时间窗的车辆优化调度问题. 它又可分为两类:

1) 硬时间窗VRP

硬时间窗VRP 指每项任务必须在要求的时间范围内完成, 出这个时间范围, 则得到的解为不可行解.

2) 软时间窗VRP

软时间窗VRP 指如果某项任务不能在要求的时间范围内完成, 则给予一定的惩罚:

若车辆在 ET_i 之前到达点 i , 则车辆在此等待, 增加了时间成本;

若车辆在 LT_i 之后到达点 i , 则服务被延迟, 须支付一定的罚金成本.

本文采用文献[1]提出的数学模型, 将中心仓库编号为0, 发货点编号为 $1, 2, \dots, L$, 任务及中心仓库均以点 i ($i = 0, 1, \dots, L$) 来表示. 定义变量如下:

$$y_{ki} = \begin{cases} 1 & \text{发货点 } i \text{ 的任务由在 } k \text{ 完成} \\ 0 & \text{否则} \end{cases} \quad x_{ijk} = \begin{cases} 1 & \text{车 } k \text{ 从点 } i \text{ 行驶到点 } j \\ 0 & \text{否则} \end{cases}$$

c_{ij} 表示为从点 i 到 j 的运输成本, 它的含义可以是距离、费用、时间等. s_i 表示车辆到达任务点 i 的时间, p_E 表示在 ET_i 之前到达任务点 i 等待的单位时间成本, p_L 表示在 LT_i 之后到达任务点 i 的单位时间所得到的罚金成本; 若车辆在 ET_i 之前到达点 i , 则增加机会成本 $p_E \times (s_i - ET_i)$, 若车辆在 LT_i 之后到达点 i , 则增加罚金成本 $p_L \times (LT_i - s_i)$.

则可得到车辆优化调度数学模型如下:

$$\min z = \sum_i \sum_j \sum_k c_{ij} x_{ijk} + p_E \sum_{i=1}^L \max(ET_i - s_i, 0) + p_L \sum_{i=1}^L \max(s_i - LT_i, 0) \quad (1)$$

$$\text{s.t.} \quad \sum_i g_i y_{ki} \leq q_k \quad \forall k \quad (2)$$

$$\sum_k y_{ki} = 1 \quad i = 1, 2, \dots, L \quad (3)$$

$$\sum_i x_{ijk} = y_{kj} \quad j = 0, 1, \dots, L; \quad \forall k \quad (4)$$

$$\sum_j x_{ijk} = y_{ki} \quad i = 0, 1, \dots, L; \quad \forall k \quad (5)$$

$$X = (x_{ijk}) \quad S \quad (6)$$

$$x_{ijk} = 0 \text{ 或 } 1 \quad i, j = 0, 1, \dots, L; \quad \forall k \quad (7)$$

$$y_{ki} = 0 \text{ 或 } 1 \quad i = 0, 1, \dots, L; \quad \forall k \quad (8)$$

当(1)中 $p_E = p_L$ 时, 以上模型为硬时间窗VRP问题.

该模型要求每个发货点都得到车辆的配送服务, 并限制每个发货点的需求仅能由某一车辆来完成; 同时保证每条路径上的各发货点的总需求量不超过此条路径配送车的容量. 优化问题也就是在满足以上约束条件下, 使总成本 z 最小.

这个模型通用性很强, 经过参数的不同设定, 可以将其转换为其他组合优化问题的数学模型. 若式(1)中 $ET_i = 0, LT_i$, 则VRPTW模型就变成了普通的VRP模型; 若仅有一个车辆被利用, 则该问题就变成了TSP问题; 若去掉约束(2), 则变成了 m TSPTW问题.

3 粒子群算法(Particle Swarm Optimization)

粒子群算法由 Kennedy 和 Eberhart 在1995年提出,该算法模拟鸟集群飞行觅食的行为,通过鸟之间的集体协作使群体达到最优目的。

在 PSO 系统中,每个备选解被称为一个“粒子”(Particle),多个粒子共存,合作寻优(近似鸟群寻找食物),每个粒子根据它自身的“经验”和相邻粒子群的最佳“经验”在问题空间中向更好的位置“飞行”,搜索最优解。

PSO 算法数学表示如下(Shi and Eberhart)^[3]:

设搜索空间为 D 维,总粒子数为 n 第 i 个粒子位置表示为向量 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$; 第 i 个粒子“飞行”历史中的过去最优位置(即该位置对应解最优)为 $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$, 其中第 g 个粒子的过去最优位置 P_g 为所有 $P_i (i=1, 2, \dots, n)$ 中的最优; 第 i 个粒子的位置变化率(速度)为向量 $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ 。每个粒子的位置按如下公式进行变化(“飞行”):

$$\begin{aligned} v_{id}(t+1) &= w \times v_{id}(t) + c_1 \times \text{rand}() \times [p_{id}(t) - x_{id}(t)] + c_2 \times \text{rand}() \times [p_{gd}(t) - x_{id}(t)] \\ x_{id}(t+1) &= x_{id}(t) + v_{id}(t+1) \quad 1 \leq i \leq n; \quad 1 \leq d \leq D \end{aligned} \quad (10)$$

其中, c_1, c_2 为正常数,称为加速因子; $\text{rand}()$ 为 $[0, 1]$ 之间的随机数; w 称惯性因子, w 较大适于对解空间进行大范围探查(exploration), w 较小适于进行小范围开挖(exploitation)。第 d ($1 \leq d \leq D$) 维的位置变化范围为 $[-XMAX_d, XMAX_d]$, 速度变化范围为 $[-VMAX_d, VMAX_d]$, 迭代中若位置和速度超过边界范围则取边界值 Maurice Clerc 对上述参数进行了分析,给出了 PSO 算法收敛的参数条件^[4]。

粒子群初始位置和速度随机产生,然后按公式(9)、(10)进行迭代,直至找到满意的解。目前,常用的粒子群算法将全体粒子群(Global)分成若干个有部分粒子重叠的相邻子群,每个粒子根据子群(Local)内历史最优 P_i 调整位置,即公式(9)中 P_{gd} 换为 P_{id} 。PSO 算法可用伪代码表示如下:

随机初始化粒子群中的每个粒子(Particle);

Do

For 每个粒子(each particle)

按式(1)计算适应度;

If 粒子当前适应度优于该粒子历史最优适应度

Then 记历史最优适应度值和位置(P_i)为该粒子当前适应度和位置(X_i);

End

选择当前粒子群(或相邻子群)中适应度最优的粒子;

If 当前粒子群(或相邻子群)中最优适应度优于群内历史最优适应度

Then 记粒子群(或相邻子群)历史最优适应度和最优位置 P_g (or P_l) 为当前群内最优适应度和最优粒子的位置;

For 每个粒子

按式(9)计算粒子飞行速度 V_i ;

按式(10)更新粒子位置 X_i ;

End

While 适应度最小误差标准或最大迭代次数均未达到。

近几年的研究和实践表明,PSO 在多维空间多峰问题寻优、动态目标寻优方面有着速度快、解质量高、鲁棒性好等优点^[3]。

4 车辆路径问题的粒子群算法

4.1 构造粒子表达方式

如何找到一个合适的表达方法,使粒子与解对应,是实现算法的关键问题之一。借鉴文献[5]的思路,本文中构造一个 $2L$ 维的空间对应 L 个发货点任务的 VRP 问题,每个发货点任务对应两维:完成该任务车辆

的编号 k , 该任务在 k 车行驶路径中的次序 r . 为表达和计算方便, 将每个粒子对应的 $2L$ 维向量 X 分成两个 L 维向量: X_v (表示各任务对应的车辆编号) 和 X_r (表示各任务在对应的车辆路径中的执行次序).

例如, 设 VRP 问题中发货点任务数为 7, 车辆数为 3, 若某粒子的位置向量 X 为:

发货点任务号: 1 2 3 4 5 6 7

X_v : 1 2 2 2 2 3 3

X_r : 1 4 3 1 2 2 1

则该粒子对应解路径为:

车1: 0 1 0

车2: 0 4 5 3 2 0

车3: 0 7 6 0

粒子速度向量 V 与之对应表示为 V_v 和 V_r .

该表示方法的最大优点是使每个发货点都得到车辆的配送服务, 并限制每个发货点的需求仅能由某一车辆来完成, 使解的可行化过程计算大大减少. 虽然该表示方法的维数较高, 但由于 PSO 算法在多维寻优问题有着非常好的特性^[6], 维数的增加并未增加计算的复杂性, 这一点在实验结果中可以看到.

4.2 算法实现过程

前面所述 PSO 算法为连续空间算法, 而 VRP 问题为整数规划问题, 因此在算法实现过程中要作相应修改. 具体实现步骤如下:

步骤1 初始化粒子群.

粒子群划分成若干个两两相互重叠的相邻子群;

每个粒子位置向量 X_v 的每一维随机取 $1 \sim K$ (车辆数) 之间的整数, X_r 的每一维随机取 $1 \sim L$ (发货点任务数) 之间的实数;

每个速度向量 V_v 的每一维随机取 $-(K-1) \sim (K-1)$ (车辆数) 之间的整数, V_r 的每一维随机取 $-(L-1) \sim (L-1)$ 之间的实数;

用评价函数 $Eval$ 评价所有粒子;

将初始评价价值作为个体历史最优解 P_i , 并寻找各子群内的最优解 P_l 和总群体内最优解 P_g .

步骤2 重复执行以下步骤, 直到满足终止条件或达到最大迭代次数.

对每一个粒子, 按式 (9) 计算 V_v 、 V_r ; 按照式 (9) 计算 X_v 、 X_r , 其中 X_v 向上取整; 当 V 、 X 超过其范围时按边界取值.

用评价函数 $Eval$ 评价所有粒子;

若某个粒子的当前评价价值优于其历史最优评价价值, 则记当前评价价值为该历史最优评价价值, 同时将当前位置向量记为该粒子历史最优位置 P_i ;

寻找当前各相邻子群内最优和总群体内最优解, 若优于历史最优解则更新 P_l 、 P_g .

对于子群内有多个个体同为最优值的情况, 则随机取其中一个为子群内当前最优解. 其中, 评价函数 $E-val$ 完成以下任务:

1) 根据公式 (1) 计算该粒子所代表路径方案的行驶成本 Z , 在计算中发货点任务的执行次序要根据对应 X_r 值的大小顺序, 由小到大执行.

2) 将 X_r 按执行顺序进行重新整数序规范. 例如, 某粒子初始化和迭代一次后结果如下:

X_v : 1 2 2 2 2 3 3

X_r : 5 3 2 6 2 1 2 2 5 0 5 4 2

则评价后重新规范的 X_r 是: 1 3 4 1 2 1 2

5 实验结果及分析

为便于比较结果, 用 Matlab 6.1 编写了 VRPTW 问题的粒子群算法和遗传算法 (GA) 程序. 并在同一 P41.7G、256M RAM、Win2000 操作系统的计算机上运行. 其中, GA 采用了文 [1] 中的染色体编码方式和

相关参数; PSO 采用满足文献[4]中约束条件的参数 .

实验1 采用了文[1]中无时间窗VRP 的例子, 问题为一个有7个发货点任务的车辆路径问题, 各任务点的坐标及货运量见表1 .

表1 各发货点坐标及货运量

序 号	0	1	2	3	4	5	6	7
坐 标	(18, 54)	(22, 60)	(58, 69)	(71, 71)	(83, 46)	(91, 38)	(24, 42)	(18, 40)
货运量 (g_i)		0 89	0 14	0 28	0 33	0 21	0 41	0 57

0表示中心仓库, 设车辆容量皆为 $q= 1. 0$, 由3辆车完成所有任务 .(最优路径距离为217. 81)
GA 参数: 群体规模 $n= 40$; 交叉概率 $P_c= 0. 6$; 变异概率 $P_m= 0. 2$; 轮盘赌法选择子代, 最大代数200 .
PSO 参数: 粒子数 $n= 40$; 分为2个子群, 子群规模为22, 子群间重叠的粒子数为2个(子群规模的1/10); $w= 0. 729$; $c_1= c_2= 1. 49445$; 最大代数200 .

两种方法各运行50次, 结果如表2所示 .表3则给出 PSO 每次达到最优路径的代数 .

表2 实验1 GA、PSO 方法结果对比

方法	达到最优 路径次数	未达最优 路径次数	达到最优路径 平均代数	达到最优路径 的平均时间(s)
GA	32	18	53. 9	32. 3
PSO	50	0	28. 36	3. 04

表3 实验1PSO 方法达到最优路径的代数

7	23	2	17	7	17	13	7	41	19
28	11	33	14	21	23	11	71	82	24
13	58	36	20	10	3	8	5	65	35
9	2	15	76	25	67	30	55	9	29
21	6	38	9	43	148	1	29	3	79

实验结果表明, PSO 方法对该问题的搜索成功率为100%, 远远高于 GA 方法的64%, 而且达到最优路径的速度比 GA 方法提高近10倍左右 .说明在该问题上使用 PSO 方法的效果远远优于 GA 方法 .这一结果与文献[5]中 GA 和 PSO 在并行处理器任务调度分配问题中的比较结果非常近似 .

实验2 采用文[1]中VRPTW 的例子, 该问题有8项货物运输任务(编号为1, 2, ..., 8), 各任务货运量 g_i 、卸货时间 t_i 及每项任务时间窗 $[ET_i, LT_i]$ 由表4给出 .这些任务由中心仓库0发出的3辆容量为8吨的车辆完成, 中心仓库与各任务点间及各任务点之间的距离由表5给出, 车速50, 单位运输成本为1, 超出时间窗的单位惩罚为: $p_E= 50, p_L= 50$.

表4 各发货点坐标及货运量

任务序号	1	2	3	4	5	6	7	8
货运量(g_i)	2	1. 5	4. 5	3	1. 5	4	2. 5	3
T_i	1	2	1	3	2	2. 5	3	0. 8
$[ET_i, LT_i]$	[1, 4]	[4, 6]	[1, 2]	[4, 7]	[3, 5. 5]	[2, 5]	[5, 8]	[1. 5, 4]

已知最少行驶成本路径为:

车1: 0 6 4 0
车2: 0 3 1 2 0
车3: 0 8 5 7 0
总行驶成本 $Z= 910$.

对 GA 和 PSO 各运行50次, 两种算法参数与例1相同, 结果对比如表6所示 .

实验结果表明, PSO 对该问题的搜索成功率为46%, 远高于 GA 的24% .同时, 在平均行驶成本和平均搜索成功时间方面, PSO 也明显优于 GA .说明在该问题上 PSO 方法也优于 GA 方法 .

为验证实验结果是否具备一般性, 随机产生多个VRPTW 问题, 使用不同群体数和不同参数的 GA

和 PSO 算法进行实验, 所取得结果近似 根据多次试验得到以下经验规律

1) 当发货点任务数的增加, 必须适当增加粒子数来避免收敛于局部最优 对于发货点任务数的相同的不同 VRPTW 问题, 使之达到较高搜索成功率(搜索到最优路径的几率)的最少粒子数也有所不同 一般而言, 对无时间窗 VRP, 当粒子数为任务数的6~ 8倍左右时, 搜索成功率一般能达到90% 以上; 而对有时间窗 VRP, 粒子数则要为粒子数的10倍左右;

表5 任务点与中心仓库及各任务点间距离

距离	0	1	2	3	4	5	6	7	8
0	0	40	60	75	90	200	100	160	80
1	40	0	65	40	100	50	75	110	100
2	60	65	0	75	100	100	75	75	75
3	75	40	75	0	100	50	90	90	150
4	90	100	100	100	0	100	75	75	100
5	200	50	100	50	100	0	70	90	75
6	100	75	75	90	75	70	0	70	100
7	160	110	75	90	75	90	70	0	100
8	80	100	75	150	100	75	100	100	0

表6 实验2 GA、PSO 方法结果对比

	搜索成功率	平均行驶成本	平均成功搜索时间
GA	24%	993 6	18 41s
PSO	46%	940 5	8 53

2) 使用不同的相邻子群数对搜索成功率和达到最优行驶成本路径的平均代数有影响, 采用2~ 4个子群数的效果最佳;

3) 子群之间重叠粒子个数对搜索成功率和达到最小行驶成本路径的平均代数也有一定影响, 采用子群规模的1/10~ 1/20左右效果最好;

4) 当粒子数大于任务数的5~ 6倍之后, 粒子数的增加对于达到最优解的搜索时间、50次试验结果平均行驶成本并无很大影响, 即此时粒子数与寻优结果的相关性不大 .

文献[6]在对典型连续非线性多维函数使用 PSO 方法寻优的经验研究中, 发现粒子数与寻优结果的相关性并不大; 通过自适应修改惯量 w 的方法, 可以克服在非常复杂空间寻优时收敛于局部最优解的问题 但 VRPTW 属于整数规划问题, 实验中采用自适应修改惯量 w 的方法, 并未能解决收敛于局部最优解的问题 .

6 结论

分析 PSO 方法, 可以看出它与 GA 等其他演化算法的最大不同在于 .

- 1) 迭代运算中只涉及到初等运算, 且运算量非常少;
- 2) 每个粒子能直接获取群体历史经验和个体历史经验, 比在其他方法中使用精英集(elitism)的方法更有效;
- 3) 整个粒子群被划分为几个的子群, 且子群之间有一定重叠, 从而使收敛于局部最优解的几率大大减少 .

正因为如此, 本文将 PSO 应用于带时间窗车辆路径问题求解中, 取得了很好的效果, 有着运算速度快、鲁棒性好、解的质量与个体数目相关性小、所获得的解质量高等诸多优点 .

参考文献:

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

[2] Kennedy J, Eberhart R C Particle swam optimization[A] Proc IEEE International Conference on Neural Networks, IV [C] Piscataway, NJ: IEEE Service Center, 1995 1942- 1948

[3] Eberhart R C, Shi Y Particle swam optimization: developments, applications and resources[A] Proc Congress on Evolutionary Computation 2001[C] Piscataway, NJ: IEEE Press, 2001 81- 86

[4] Maurice Clerc, James Kennedy The particle swam-explosion, stability, and convergence in a multidimensional complex space[J] IEEE Transaction on Evolutionary Computation, 2002, 6(1): 58- 73

[5] A yed Salmen, In tiaz A hmad, Sabah A l- M adani Particle swam optimization for task assignment problem [J] M i-croprocessors and M icrosystem s, 2002, 26: 363- 371

[6] Shi Y, Eberhart R C Empirical study of particle swam optimization [A] Proceedings of the 1999 Congress on Evo-lutionary Computation [C] Piscataway, NJ: IEEE Service Center, 1999 1945- 1950