

文章编号:1001—9081(2004)06Z—0261—03

新型遗传模拟退火算法求解物流配送路径问题

阎 庆,鲍远律

(中国科学技术大学 自动化系,安徽 合肥 230027)

摘 要:文中提出了将遗传算法和模拟退火算法结合,并加入了记忆装置。根据这种想法设计了一种有记忆功能的遗传模拟退火算法,并进行了试验计算。结果表明:用这种有记忆功能的遗传模拟退火算法求解物流配送路径优化问题,可以在一定程度上解决一些问题,从而得到较高质量的解。

关键词:物流配送;遗传模拟退火算法;遗传算法;模拟退火算法;路径优化

中图分类号: TP301.6 文献标识码: A

1 引言

物流配送是现代化物流管理中的一个重要环节。它是指按用户的定货要求,在配送中心进行分货、配货,并将配好的货物及时送交收货人的活动。在物流配送业务中,存在许多优化决策的问题。本文只讨论物流配送路径优化问题。物流配送路径优化问题最早是在 1959 年由 Dantzig 和 Ramser 首先提出的,即所谓的车辆路径问题 VRP<sup>[1]</sup>。它也是目前在物流系统中较受关注的一个方面。它是指在客户需求位置已知的情况下,确定车辆在各个客户间的行程路线,使得运输路线最短或运输成本最低。VRP 问题已经被证明是一个 NP-hard 问题。也就是说当问题的规模较大时,很难得到全局最优解或满意解。而且随着问题规模的增大,算法的计算时间将以指数速度增加。因此研究的重点就转移到各种启发式算法上。求解物流配送路径优化问题的方法有很多,常用的有旅行商法、动态规划法、节约法、扫描法、分区配送法、方案评价法等。而遗传算法的出现为求解物流配送路径优化问题提供了新的工具。遗传算法<sup>[2]</sup>作为一种非数值并行算法,其思想起源于生物遗传学适者生存的自然规律。它对优化对象既不要求连续,也不要求可微,尤其适合求解 NP-hard 问题。到目前为止,已经有很多人都曾利用遗传算法求解物流配送路径优化问题并取得了一些研究成果。

2 物流配送路径优化问题的数学模型

物流配送路径优化问题一般可以这样描述:从某物流配送中心用多辆配送车辆向多个客户送货。每个客户的位置和货物需求量一定,每辆车的载重量一定,其一次配送的最大行驶距离一定。要求合理安排车辆配送路线,使目标函数得到最优。并满足以下条件:(1)每条配送路径上各客户需求量之和不超过配送车辆的载重量;(2)每条配送路径的长度不超过配送车辆一次配送的最大行驶距离;(3)每个客户的需求必须满足,且只能由一辆配送车送货。

设配送中心需要向  $k$  个客户送货,每个客户的货物需求量是  $g_i(i = 1, 2, \dots, k)$ ,每辆配送车的载重量是  $q$ ,且  $g_i < q$ 。首先为了安排路线需要对要使用的车辆数有一个估计。在实际情况中,货物装(卸)车越复杂,约束条件越多,一辆车的实际载货量就越小。在本文中使用了文献<sup>[5]</sup>的公式来确定需要

的车辆数:

$$m = [\sum_{i=1}^k g_i / aq] + 1$$

$m$  为所需车辆数, $[\ ]$  表示取整, $a$  为参数, $0 < a < 1$ 。约束条件越多,货物装(卸)越复杂, $a$  值越小。参考文献<sup>[2]</sup>,取  $a$  为 0.85。

下面建立此问题的数学模型: $c_{ij}$  表示点  $i$  到点  $j$  的运输成本,如时间,路程,花费等。配送中心编号为 0,各客户编号为  $i(i = 1, 2, \dots, k)$ ,定义变量如下:

$$x_{ijs} = \begin{cases} 1, & \text{车 } s \text{ 由 } i \text{ 驶向 } j \\ 0, & \text{否则} \end{cases}$$
$$y_{is} = \begin{cases} 1, & \text{点 } i \text{ 的货动任务由 } s \text{ 车完成} \\ 0, & \text{否则} \end{cases}$$

得到的数学模型如下所示:

$$\min Z = \sum_{i=0}^k \sum_{j=0}^k \sum_{s=1}^m c_{ij} x_{ijs} \tag{1}$$

$$\sum_{i=0}^k g_i y_{is} \leq q \quad s = 1, 2, \dots, m \tag{2}$$

$$\sum_{s=1}^m y_{is} = \begin{cases} 1 & i = 1, 2, \dots, k \\ m & i = 0 \end{cases} \tag{3}$$

$$\sum_{i=0}^k x_{ijs} = y_{js} \quad j = 1, \dots, k; s = 1, 2, \dots, m \tag{4}$$

$$\sum_{j=1}^k x_{ijs} = y_{is} \quad i = 0, 1, \dots, k; s = 1, 2, \dots, m \tag{5}$$

$$x_{ijs} = 0 \text{ 或 } 1 \quad i, j = 0, 1, \dots, k; s = 1, 2, \dots, m \tag{6}$$

上述模型中,式(2)为汽车容量约束;式(3)保证了每个客户的运输任务仅由一辆车完成,而所以运输任务则由  $m$  辆车协同完成;式(4)和式(5)限制了到达和离开某一客户的汽车有且仅有一辆。

3 路径优化问题的有记忆的遗传模拟退火算法

针对遗传算法的一些不足,作者将模拟退火算法与之结合,并加入了记忆装置,从而构造了物流配送路径优化问题的一种有记忆功能的遗传模拟退火算法。该算法的特点是扩大了原有遗传算法的搜索邻域,在一定概率控制下暂时接受一些恶化解。同时利用记忆装置保证了在一定终止条件下所得的最终解至少是搜索过程中曾得到所有解中的最优解。该算

法通过在常规的遗传算法基础上加入模拟退火算子和记忆装置而得到。

步骤1 给定群体规模  $maxpop, k=0$ ; 初始温度  $t_k=t_0$ , 产生初始群体  $pop(k)$ ; 对初始群体计算目标值  $f(i)$ , 找出使函数  $f_i(t_k)$  最小的染色体  $i$  和这个函数值  $f$ , 记  $i^*=i, f^*=f$ ;

其中,  $f_i(t_k)$  为状态  $i$  在温度为  $t_k$  时的目标值。 $i \in pop(k)$ , 即当代群体中的一个染色体。

步骤2 若满足结束条件则停止计算, 输出最优染色体  $i^*$  和最优解  $f^*$ ; 否则, 在群体  $pop(k)$  的每一个染色体  $i \in pop(k)$  的邻域中随机选取一个状态  $j \in N(i)$ , 按模拟退火中的接受概率:

$$A_{ij} = \min \left\{ 1, \exp \left( - \frac{f_j(t_k) - f_i(t_k)}{t_k} \right) \right\}$$

接受或拒绝  $j$ , 其中  $f_i(t_k), f_j(t_k)$  分别为状态  $i, j$  的目标值。这一阶段共需  $maxpop$  次迭代以选出新群体  $newpop1$ 。

步骤3 在  $newpop1(k+1)$  中计算适应度函数:

$$f_i(t_k) = \exp \left\{ - \frac{f_i(t_k) - f_{min}}{t_k} \right\}$$

其中,  $f_{min}$  是  $newpop1(k+1)$  中的最小值。由适应度函数决定的概率分布从  $newpop1$  中随机选  $maxpop$  个染色体形成种群  $newpop2$ 。

步骤4 按遗传算法的常规方法对  $newpop2$  进行交叉得到  $crosspop$ , 再变异得到  $mutpop$ 。

步骤5 令  $pop(k+1) = mutpop$ , 对  $pop(k+1)$  计算  $f_i(t_k)$ , 找出使函数  $f_i(t_k)$  最小的染色体  $i$  和这个函数值  $f$ , 如果  $f < f^*$ , 则令  $i^* = i, f^* = f, t_{k+1} = a \times t_k, k = k+1$ , 返回步骤2。

3.1 染色体结构

出于表示简单, 计算机处理方便的目的, 对于 VRP 问题的遗传算法编码通常都采用自然数编码。上节数学模型的解向量可编成一条长度为  $k+m+1$  的染色体  $(0, i_1, i_2, \dots, i_s, 0, i_j, \dots, i_k, 0, \dots, 0, i_p, \dots, i_q, 0)$ 。在整条染色体中, 自然数  $i_j$  表示第  $j$  个客户。0 的数目为  $m+1$  个, 代表配送中心, 并把自然数编码分为  $m$  段, 形成  $m$  个子路径, 表示由  $m$  辆车完成所有运输任务。这样的染色体编码可以解释为: 第一辆车从配送中心出发, 经过  $i_1, i_2, \dots, i_s$  客户后回到配送中心, 形成了子路径1; 第2辆车也从配送中心出发, 途经  $i_j, \dots, i_k$  客户后回到配送中心, 形成子路径2。 $m$  辆车依次出发, 完成所有运输任务, 构成  $m$  条子路径。如染色体 0123045067890 表示由三辆车完成9个客户的运输任务的路径安排:

子路径1: 配送中心 → 客户1 → 客户2 → 客户3 → 配送中心

子路径2: 配送中心 → 客户4 → 客户5 → 配送中心

子路径3: 配送中心 → 客户6 → 客户7 → 客户8 → 客户9 → 配送中心

3.2 遗传群体初始化

为了使算法收敛到全局最优, 遗传群体应具有一定的规模。但为了保证计算效率, 群体规模也不能太大。一般取群体规模取值在10到100之间。在初始化染色体时, 先生成  $k$  个客户的一个全排列, 再将  $m+1$  个0随机插入排列中。需要注意的是必须有两个0被安排在排列的头和尾, 并且在排列中不能有连续的2个0。这样构成一条满足问题需要的染色体。针

对此染色体, 随机选择两个位置上的元素进行交换, 并用算法对其调整, 使其成为新的满足要求的染色体。交换若干次, 直至生成满足群体规模数的染色体。

3.3 计算适应度函数

这里使用文献[3]的方法, 将容量约束式(2)转为运输成本的一部分, 运输成本变为:

$$Z = \sum_{i=0}^k \sum_{j=0}^k \sum_{s=0}^m c_{ij} x_{ijs} + M \sum_{s=1}^m \max \left( \sum_{i=0}^k g_i y_{is} - q, 0 \right)$$

其中  $M$  为一很大的正数, 表示当一辆车的货运量超过其最大载重量时的惩罚系数。 $M$  应趋向于无穷大。考虑到计算机处理的问题, 参考文献[6], 取  $M$  为1000000。将此运输成本函数作为目标函数。适应度函数采用一种加速适应度函数  $f_i(t_k) = \exp \left\{ - \frac{f_i(t_k) - f_{min}}{t_k} \right\}$ 。这种适应度函数加速性能比较好, 可以迅速改进适应度的值, 缩短算法运行时间。

3.4 交叉算子

将每代种群的染色体中适应度最大的染色体直接复制, 进入下一代。种群中其他染色体按其适应度的概率分布, 采用轮盘赌的方法, 产生子代。这样既保证了最优者可生存至下一代, 又保证了其余染色体可按生存竞争的方法生成子代, 使得算法可收敛到全局最优。选中的染色体按一定的概率——交叉率, 产生子代。文献[7]认为交叉率在0.6~0.8之间, 算法进化效果较好。

由于车辆路径问题的条件约束, 如果采用简单的交叉算子, 将可能产生大量的不可行解。因此这里采用文献[6]的交叉算子, 这种改进的交叉算子, 避免非可行解的产生, 具体做法如下:

(1) 随机产生交叉位, 如果在交叉位的外侧两端的父代基因都是0的话, 则对父代进行简单交叉; 如果交叉位的外侧两端的父代基因不为0的话, 则将左交叉位向左移, 直至移动到左交叉位的左端基因不为0时停止。以左交叉位为起点, 继续向右移动右交叉位, 直至右交叉位的右端基因不为0为止。这样就保证了父代染色体都用1条子路径来进行交叉。

(2) 经过第(1)步操作, 子代中仍会产生访问同一客户2次的情况, 需要对子代进行整理。在子代中选择那些具有重复情况的自然数, 如果该自然数并非在交叉位内的话, 则删除之。子代如存在未访问到某一客户的情况, 则在染色体的交叉位外补上该客户对应的自然数。

(3) 经过第(2)步的操作, 如果产生了某一子路径为空的情况, 即染色体中含有2个连续的0, 须继续进行第3步操作。将其中的1个0与染色体其他位上的客户自然码进行交换。对该客户自然码的要求是该码的前一位与后一位均不为0。本步操作可放在对子代的变异后进行。

如有父代1: 015604027680, 父代2: 071086402350。使用上述算法, 对父代1选定交叉位8, 11, 对父代2选定交叉位5, 7。经过第1步计算, 得子代1: 01560408640 和子代2: 0710273802350。使用算法的第2步, 整理子代1和子代2。得到子代1: 012537008640 和子代2: 014027380560。子代1中存在2个连续的0, 对其进行第3步操作。最后得到子代1: 012507308640。

3.5 变异算子

交叉产生的子代依一定概率——变异率发生变异。变异的策略是随机交换2个基因的位置。在变异后, 需要用交叉算子

的第3步操作整理子代,以保证可行性。

3.6 结束条件

当算法的当前进化代数大于预先设定的  $N$  时,算法结束。

3.7 邻域结构

每个染色体的邻域包括任意交换其两个基因所产生的所有染色体。

表1 试验1初始数据

门店	配送中心	门店1	门店2	门店3	门店4	门店5	门店6	门店7	门店8
位置	(31,9)	(76,38)	(77,16)	(90,82)	(60,74)	(76,86)	(11,31)	(29,90)	(10,60)
需求量	0	2.46	0.41	2.16	2.27	1.83	3.76	2.54	2.39

根据各仓库的需求量,计算出需要的汽车数: $m=\lceil 17.82/(0.85\times 8)\rceil+1=3$ ,文献[6]采用传统的遗传算法的各算子,并对其中的交叉算子进行了改造,取群体规模为20,进化代数为50,应用此程序他费时3s得到的结果为:

子路径1:0→8→7→4→0

子路径2:0→6→0

子路径3:0→5→1→2→3→0

运输距离成本:476.29

而我们的算法在上面的算法中加入了一个模拟退火算

3.8 衰减函数

$$t_{k+1}=a\times t_k$$

4 实验分析

实验初始数据取自参考文献[6]。

实验1,随机生成1个有8个门店的VRP问题,初始数据如下:

子,取初始退火温度为10,衰减系数取0.85使用第三节所述算法步骤,在奔腾四的计算机上计算,耗时2s,得结果如下:

子路径1:0→6→0

子路径2:0→2→1→3→5→0

子路径3:0→8→7→4→0

运输距离成本:476.290507

进化代数为:20

实验2,随机生成1个有20个门店的VRP问题,初始数据如下:

表2 试验2初始数据

门店	配送中心	门店1	门店2	门店3	门店4	门店5	门店6	门店7	门店8	门店9	门店10
位置	(52,4)	(15,49)	(0,61)	(51,15)	(25,71)	(38,62)	(35,45)	(100,41)	(10,52)	(26,79)	(87,7)
需求量	0	1.64	1.31	1.43	3.38	1.13	3.77	3.48	0.39	0.24	1.03

续表2

门店	门店11	门店12	门店13	门店14	门店15	门店16	门店17	门店18	门店19	门店20
位置	(24,89)	(19,25)	(20,99)	(73,91)	(100,95)	(7,73)	(69,86)	(24,3)	(66,14)	(9,30)
需求量	0.24	2.60	1.00	0.65	0.58	2.56	2.69	3.26	2.97	

计算得:需6辆车。用文献[6]的算法取群体规模100,进化代数分别设为20,50,100,得到的结果不同:

表3 原来算法的结果

实验	1	2	3
进化代数	20	50	100
运输成本	1153.12	964.48	908.44

而采用本文的算法,初始退火温度取10,衰减系数取0.85,在奔腾四的计算机上计算,则结果如下:

表4 本算法的结果

实验	1	2	3
进化代数	20	50	100
运输成本	1153.96	963.67	906.29

从以上两个实验可以看出:采用本文中所述的算法,要得到相同的结果可以缩短进化代数,从而节约运算时间。而采用相同的进化代数,虽然在进化代数较少时对结果的改进并不明显,但随着进化代数的增加,必然得到更好的结果。

5 结束语

本文使用文献[6]中比较合理的交叉算子和变异算子,采

用自然数编码将模拟退火算法与传统的遗传算法相结合,用来求解物流系统中车辆路径问题。这种算法所需的进化代数明显减少,获得了较好的结果。实验结果表明此算法在解决诸如车辆路径问题这类NP-hard问题确实可行,并有较好的性能。而且随着问题规模的增大,这种缩短进化代数的效果将更加明显。

参考文献

[1] 郭耀煌,李军著.车辆优化调度[M].成都:成都科技大学,1994.

[2] 邢文训,谢金星.现代优化计算方法[M].北京:清华大学出版社,1999.140.

[3] 郎茂祥.物流配送车辆调度问题的模型和算法研究[D].北京:北方交通大学,2002.

[4] 郎茂祥,胡思继.用混和遗传算法求解物流配送路径优化问题的研究[J].中国管理科学,2002,10(5):51—56.

[5] 李军,谢秉磊,郭耀煌.非满载车辆调度问题的遗传算法[J].系统工程理论与实践,2000,20(3):235—239.

[6] 唐坤.车辆路径问题中的遗传算法设计[J].东北大学学报(自然科学版),2002,28(1):66—70.

[7] 姜大立,杨西龙,杜文,等.车辆路径问题的遗传算法研究[J].系统工程理论与实践,1999 19(6):40—45.