

开放的车辆路线安排问题的模型与遗传算法

邓 猛 肖辉君 杨丰梅*

(北京化工大学理学院, 北京 100029)

摘 要: 针对开放的车辆路线安排问题,建立了以车流为基础的数学模型。在模型中利用罚函数法来化简约束条件,并设计了基于自然数编码的遗传算法。最后给出一个简单的算例来说明该模型及算法的应用。

关键词: 物流; 配送; 车辆路线安排问题; 罚函数; 遗传算法

中图分类号: O224

引 言

车辆路线安排问题 (Vehicle Routing Problem, VRP) 由 Danzig 和 Ramer 于 1959 年提出,之后有了很大的发展,在物流配送、交通运输和航空邮件等领域都得到了广泛的应用^[1]。VRP 是物流配送的核心问题,车辆配送路线的合理优化问题在整个物流运输的过程中扮演着至关重要角色。按照车辆对车场的所属关系划分,可以将 VRP 分为车辆路线封闭问题(车辆必须回到出发点)和车辆路线开放问题(车辆可以不返回出发点)两类^[2]。开放的车辆路线安排问题 (Open Vehicle Routing Problem, OVRP) 可以定义为:对一系列确定的顾客点,如何组织最适当的车辆行驶路线,使得车辆在满足顾客需求的条件下,达到费用最省,其中每一辆车行驶一条路线,每条路线开始于车场,经过若干顾客,最终结束于其中的一个顾客。目前研究较多的是车辆路线封闭问题,就是我们所熟知的 VRP 问题,但对于开放的车辆路线安排问题的研究结果还不多。

最早提到 OVRP 问题的文献是 Schrage 的文章^[3],但他只是描述了一些实际的路线问题,并没有给出具体的模型和算法。Bodin^[4]等描述了一个关于美国航空信件的邮递问题,并且运用 C—W 节约算法来求解。Sariklis 和 Powells^[5]给出了一种新的两阶段启发式算法,第一个阶段,先将顾客进行分

组,考虑车载量的限制得出最小组数和约束条件,再分派顾客到不同的组中以降低费用;第二阶段,在每一组中寻找它的一个最小生成树确定最终的车辆路线。Brandão^[6]提出了用禁忌搜索方法来求解 OVRP 问题,相比求得的解的质量优于上面的两阶段方法。本文借鉴 VRP 的以车流为基础的建模方式,建立了 OVRP 的数学模型,并通过罚函数法化简约束条件,设计了遗传算法进行求解,最后通过算例,对模型和算法的性能进行了验证。

1 OVRP 的数学模型

1.1 基本模型

考虑图 $G = (V, E)$, 其中 G 为一无向图, V 是顶点的集合, 定义为 $\{0, 1, 2, \dots, l\}$ (其中 0 对应为车场, 1、2、...、 l 对应为各个顾客点), E 为顶点之间关系的集合。

于是, OVRP 的一个基本数学模型可以表示为

$$\min z = \sum_{i,j,k} c_{ij} x_{ijk} \quad (1)$$

$$\text{s. t.} \quad \sum_i g_i y_{ki} = q, \quad \forall k \quad (2)$$

$$y_{ki} = 1, \quad (1, 2, \dots, l) \quad (3)$$

$$x_{ijk} = y_{kj}, \quad (j = 1, 2, \dots, l, \forall k) \quad (4)$$

$$x_{ijk}, y_{ki} \in \{0, 1\}, \quad (i, j = 0, 1, 2, \dots, l, \forall k) \quad (5)$$

其中 c_{ij} 记为从点 i 到点 j 的费用。每一个顾客有一已知的非负需求 g_i , (其中 $i = 1, 2, \dots, l$), 每辆车有相同的容量 q , 并假设 $g_i \leq q$, (其中 $i = 1, 2, \dots, l$)。决策变量记为

$$y_{ki} = \begin{cases} 1 & \text{点 } i \text{ 的任务由车辆 } k \text{ 完成} \\ 0 & \text{否则} \end{cases}$$

收稿日期: 2005-12-05

基金项目: 国家自然科学基金 (10171108)

第一作者: 男, 1979 年生, 硕士生

*通讯联系人

E-mail: yangfm@mail.buct.edu.cn

$$x_{ijk} = \begin{cases} 1 & \text{车辆 } k \text{ 由点 } i \text{ 行驶到点 } j \\ 0 & \text{否则} \end{cases}$$

目标函数(1)表示合理的安排车辆的行驶路线使得费用最省,式(2)表示每条路线上各个顾客的需求量之和不超过车辆的容量限制,式(3)表示每个顾客只能由一辆车来满足其需求,式(4)表示一辆车开始于车场并结束于其中的某个顾客点的一条路线,式(5)是0-1约束。

若每项任务要求在一定的时间范围内完成,则问题就成为有时间窗的车辆路线安排问题。设顾客点 j 的时间窗为 $[a_j, b_j]$, 其中 a_j 为顾客 j 允许最早到达时间, b_j 为顾客允许的最迟到达时间。考虑有时间窗的车辆路线安排问题有以下的附加约束: 如果 $x_{ijk} = 1$, (即车辆 k 从点 i 行驶到点 j , 有 $s_i + t_{ij} = s_j$, 其中 $s_0 = 0, i \neq j$), 那么, $a_j - s_j - b_j$, (其中 $j = 1, 2, \dots, l; s_j$ 为车辆到达点 j 的时间; t_{ij} 为车辆从点 i 行驶到点 j 所用时间)。

1.2 OVRP 模型中约束条件的简化

记 $g(V) = \sum_{i \in V} g_i$ 为顾客的需求总和, 由 $m = \lceil g(V)/q \rceil + 1$ 来确定完成任务所需要的车辆数。本文采用罚函数的方法来处理约束, 即将约束条件变为目标函数的一部分。因此, 对于基本模型中最复杂的约束就是顾客需求约束(2), 通过罚函数将其简化掉, 从而目标函数简化为:

$$\min z = \sum_{i,j,k} c_{ij} x_{ijk} + M \max_{k=1}^m \left(\sum_{i=1}^l g_i y_{ki} - q, 0 \right)$$

其中 M 取为较大的正数。如果对应的一个解违反了约束, 则给予一定的惩罚, 使其具有相应较小的适应度, 在搜索中允许维持一定的不可行解, 从而使搜索从可行区域和不可行区域两个方向搜索最优解。

对于时间窗约束也可以将约束转化为惩罚项来处理。时间窗约束可以分为软时间窗约束和硬时间窗约束, 所谓软时间窗约束就是当不能按要求时间完成任务时, 则处以罚值。设表示车辆在任务点处等待单位时间的机会成本, 表示车辆在要求时间之后到达时单位时间所处罚的罚值。所以, 若车辆在 a_j 之前到达点 j , 产生成本 $c_2(a_j - s_j)$; 若车辆在 b_j 之后到达点 j , 处以罚款 $c_3(s_j - b_j)$ 。

以罚函数法来处理软时间窗约束, 其目标函数变为

$$\min z = \sum_{i,j,k} c_{ij} x_{ijk} + M \max_{k=1}^m \left(\sum_{i=1}^l g_i y_{ki} - \right.$$

$$\left. q, 0 \right) + c_2 \sum_{j=1}^l \max(a_j - s_j, 0) + c_3 \sum_{j=1}^l \max(s_j - b_j, 0)$$

硬时间窗约束是指每项任务必须在要求的时间窗内完成, 如果超出了相应的时间范围, 则得到的解就是不可行解。

令 $c_2 = c_3 = M$, 即时间窗必须满足, 目标函数变为

$$\min z = \sum_{i,j,k} c_{ij} x_{ijk} + M \max_{k=1}^m \left(\sum_{i=1}^l g_i y_{ki} - q, 0 \right) + M \sum_{j=1}^l \max(a_j - s_j, 0) + M \sum_{j=1}^l \max(s_j - b_j, 0)$$

经过上述约束的简化, 虽然目标函数变得复杂起来, 但约束条件就只剩下(3)、(4)、(5)这样的简单约束, 这对于我们设计的启发式算法是非常有意义的。

2 算法设计

遗传算法是上世纪70年代由美国 Holland 教授和他的学生建立发展起来的, 是一种基于生物遗传学适者生存的自然规律的全局性概率搜索算法。它具有极强的鲁棒性和全局性, 适合于非凸空间中复杂的多极值优化和组合优化问题, 在机器学习、自动控制、人工智能和计算机科学等领域有广泛的应用。近年来已有不少学者成功应用遗传算法来解决 VRP 问题。本文设计了适合 OVRP 问题求解的遗传算法。下面给出该算法的主要参数和求解步骤。

2.1 编码

本文考虑基于自然数编码的遗传算法。编码 $(0, i_1, i_2, \dots, 0, i_3, \dots, 0, i_{l-1}, i_l)$ 对应的 $0 i_1 i_2 \dots 0 i_3 \dots 0 i_{l-1} i_l$ 表示一个染色体, 一个染色体对应为一条行车路线。如染色体 04503201 表示的行车路线为子路线 1: 车场 0—顾客 4—顾客 5; 子路线 2: 车场 0—顾客 3—顾客 2; 子路线 3: 车场 0—顾客 1。

2.2 适应函数

适应函数可以设置为

$$f(x) = \begin{cases} C_{\max} - z, & \text{若 } z < C_{\max} \\ 0, & \text{否则} \end{cases}$$

其中 z 为加上罚值以后的目标函数, C_{\max} 为一个较大的正数。

2.3 初始化群体

随机产生 $1, 2, \dots, l$ 的全排列 $i_1 i_2 \dots i_3 \dots i_{l-1} i_l$, 在第一位前面添上 0, 形成 $0 i_1 i_2 \dots i_3 \dots i_{l-1} i_l$ 。

如果 $\sum_{j=1}^{s-1} g_j < q$, 且 $\sum_{j=1}^s g_j > q$, 则在 i_{s-1}, i_s 之间插入 0, 如此继续, 直到剩下的 $(m-1)$ 个 0 全部插入为止, 这样就构成了一个染色体。反复进行操作, 形成 n 个染色体。

2.4 遗传算子

遗传算子包括选择算子、交叉算子和变异算子。

选择算子: 设初始群体为 $P = \{a_1, a_2, \dots, a_n\}$, a_i 为群体中的染色体, 各染色体所对应的适应函数为 $f(a_i)$, 则选择概率为 $p(a_i) = \frac{f(a_i)}{\sum_{j=1}^n f(a_j)}$ 。在选

择染色体进行操作时, 就根据选择概率进行选择。

交叉算子: 为了保持遗传操作中双亲的优良特性, 构造最大保留交叉算子。

最大保留交叉算子步骤:

(1) 如果染色体交叉点处的两个基因都为 0, 则直接进行顺序交叉运算;

(2) 如果染色体交叉点处的基因不全为 0, 则将交叉点左移或右移, 直到两个交叉点处的基因都为 0, 再进行顺序交叉运算。

例如, 父代 1: 0120 | 345 | 067809, 父代 2: 0137026504809, 产生一个子代 0170345026809。其中顺序交叉运算步骤如下:

(1) 从第一个双亲中随机选择一个子串;

(2) 将子串复制到一个空子串的相应位置, 产生一个原始后代;

(3) 删去第二父代中已有的顾客, 得到原始后代需要的其他顾客的顺序;

(4) 按照这个顺序, 从左到右将这些顾客定位到空缺的位置上。

例如, 父代 1: 57 | 4913 | 628, 父代 2: 123456789, 其中 4913 为匹配段。从而得到一个子代 25 | 4913 | 678。

变异算子: 对于变异算子选择插入变异, 随机选择一个顾客点 i , $i \neq 0$, 将它插入到一个随机的位置上。

2.5 终止条件

遗传算法的终止条件有多种方式, 本文采用设定最大迭代次数的方法和群体的收敛程度来作为终止条件。最大迭代次数的方法比较简单但不很准确, 根据群体的收敛程度来进行判断就是通过测定群体中的基因多样性来控制。即迭代 500 次, 然后

终止算法; 或者, 如果某一代群体中染色体的平均适应度与最佳染色体适应度的商大于 90%, 则终止算法。

2.6 算法步骤

(1) 使用自然数编码方法, 以自然数数列来表示行车路线;

(2) 设置控制参数 (即群体规模 n , 交叉概率 p_c , 变异概率 p_m 和迭代次数 m);

(3) 根据随机化的方法初始化群体 $P(0)$, 群体中包括 n 个染色体;

(4) 评价群体, 计算适应值函数;

(5) 若满足算法终止条件, 则停止, 否则继续;

(6) 根据适应值比例法选择群体;

(7) 进行交叉、变异操作, 得到下一代群体 $P(t)$;

(8) 若满足终止条件, 则停止, 否则转 (4)。

3 数据模拟分析

利用 matlab 程序来实现上述算法, 算例中只考虑了车辆有容量限制的 OVRP 模型 (时间窗约束与容量限制约束处理方式相同)。假设有 9 个顾客点, 由车场 0 派出车辆来完成各个顾客的需求。随机产生车场和各个顾客的位置坐标 (X_i, Y_i) , 各个顾客的需求量 g_i 在 $(0, 1)$ 内随机的产生, 如表 1 所示。假设车辆的容量相同 $q = 1$, $c_{ij} = c_{ji}$, $i \neq j$, c_{ij} 只与行驶距离 d_{ij} 有关, $c_{ii} = M$, M 为一趋于无穷大的正数。 $m = \lceil g(V)/q \rceil + 1$ 由得到所需车辆数。

根据给出的算例, 应用上述遗传算法进行求解, 设群体规模为 $n = 50$, 交叉概率 $p_c = 1.00$, 变异概率 $p_m = 0.01$, 迭代次数 $m = 500$ 。10 次求解的计算结果如表 2 所示。10 次费用的平均值为 240.8208。其中第 9 次的计算结果最好, 其满意解为 08125 07394 06, 费用为 229.2354, 得到了较好的计算结果。从计算效率上来看, 遗传算法的计算速度是相当快的, 利用奔腾 2 系列配置的电脑, 计算时间不超过 1 秒。通过观察计算中迭代群体的平均费用, 可知在迭代过程中, 群体的整体趋势是不断优化的。从整体上看每次计算结果都要优于初始群体。

由于 OVRP 问题是 NP 难的组合优化问题, 求出最优解往往是比较困难的, 通过遗传算法可以求得满意解; 而且从所需时间上来看, 求解的速度很快。对于 OVRP 这种优化问题, 用遗传算法来求解是有很大的优越性的。

表 1 初始数据
Table 1 Initial data

节点	坐标	需求量
0	(50 ,69)	0
1	(70 ,37)	0. 10
2	(42 ,86)	0. 50
3	(30 ,85)	0. 45
4	(18 ,59)	0. 14
5	(19 ,49)	0. 07
6	(68 ,89)	0. 29
7	(30 ,82)	0. 12
8	(54 ,64)	0. 30
9	(15 ,81)	0. 18

表 2 计算结果
Table 2 Calculated results

计算次序	迭代次数	最优费用
1	81	241. 3223
2	168	254. 7384
3	117	246. 2326
4	107	241. 6642
5	101	255. 0255
6	140	234. 0712
7	102	246. 0509
8	167	230. 0584
9	122	229. 2354
10	138	229. 8091

4 结束语

区别于传统的车辆路线安排问题 ,针对开放的车辆路线安排问题 ,建立了以车流为基础的数学模型 ,根据所研究问题的特点 ,应用遗传算法进行求解。遗传算法是一种强有力的且应用范围十分广泛的随机搜索优化技术 ,它对许多传统方法难以解决的问题是非常有效的。算例中的求解结果表明应用遗传算法求解 OVRP 问题 ,可以取得较好的计算结果 ,并且有较高的计算效率。

参 考 文 献

[1] 赵秋红. 几类物流优化模型的研究[D]. 北京:北京航空航天大学 ,2002.

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

[3] Schrage L. Formulation and structure of more complex/ realistic routing and scheduling problems[J]. Networks , 1981 , 11 : 229 - 232.

[4] Bodin L , Golden B , Assad A , *et al.* Routing and scheduling of vehicles and crews: the state of art [J] . Computers & Operations Research , 1983 , 10 : 63 - 211.

[5] Sariklis D , Powell S. A heuristic method for the open vehicle routing problem[J]. Journal of the Operational Research Society , 2000 , 51 :564 - 573.

[6] Brandão J. A tabu search algorithm for the open vehicle routing problem[J]. European Journal of Operational Research , 2004 , 157 : 552 - 564.

A genetic algorithm for the open vehicle routing problem

DENG Meng XIAO Hui-jun YANG Feng-mei

(School of Science , Beijing University of Chemical Technology , Beijing 100029 , China)

Abstract : In this paper , the open vehicle routing problem (OVRP) is studied and a model based on the vehicle flow is set up. In the model , a penalty function is used to treat the constraint conditions. A genetic algorithm based on decimal coding has been designed in order to obtain solutions for the model. Finally , a simple example is used to illustrate the use of the algorithm.

Key words : logistics ; distribution ; vehicle routing problem ; penalty function ; genetic algorithm