基于蚁群算法求解带硬时间窗的 VRPSDP①

An Algorithm Based on Ant Colony Algorithm for VRPSDP with Hard Time Windows

殷佳林　蒋　泰　(桂林电子科技大学 计算机与控制学院 广西 桂林 541004)

摘　要： 建立了描述带硬时间窗的同时送取货的车辆路径问题(VRPSDPTW)的混合整数规划模型，给出了求解该模型的基于蚁群算法的改进的启发式算法。最后，通过实例计算，验证了算法的可行性和有效性，结果表明改进的蚁群算法在求解小规模问题（20 个客户点）时，其性能总体优于已有的同类问题算法。

关键词： 车辆路径 同时送取货问题 蚁群算法 带时间窗

1　引言

人类可持续发展的要求使得人们越来越重视环保问题。物流配送既是城市经济发展和消费生活多样化的支柱，同时它的发展也会对城市环境带来不利影响。为此新世纪对物流配送提出新的发展要求即绿色配送，这使得逆向物流的发展越来越受到重视。传统的车辆路径问题在逆向物流的环境下是基于先送货后取货的问题，即 VRPB(VRP with Backhaul)，但是这个模型仍然无法满足现实情况的需求，并且造成了车辆载重的浪费。在以后的发展中，配送和收取被同时考虑，而不是分开处理，这便是同时送货和取货的车辆路径问题即 VRPSDP(Vehicle Routing Problem with Simultaneous Delivery and Pick-up)。在现实生活中，VRPSDP 问题也较多，如啤酒、牛奶行业、铁路行包物流配送、邮件配送等行业。VRPSDP 问题是由 Min[1]通过解决中心图书馆和地方图书馆之间图书的发送与回库问题而提出。Gendreau et al.[2]是先研究了 TSPPD 问题并由此引出 VRPSDP 问题的模型。其他大多数文献是直接研究 VRPSDP 问题，大多使用先聚类后排序和插入的方式，在约束条件上没有刻意要求。该类问题已被证明是 NP-Hard 难题，由于问题的复杂性，该方面的研究较少。

2　带时间窗的VRPSDP的数学模型

2.1 问题描述

本文研究的单配送中心的带硬时间窗的 VRPSDP 问题可以描述为：用多台配送车辆将客户需要的货物从配送中心送到各个客户，并将各个客户供应的货物从客户处取货回到配送中心，在客户处送货和取货的时间有一定的约束，求如何合理安排车辆的路径，使目标函数得到最优化。可做如下假设：(1)配送中心和每个客户的位置固定；(2)客户的配送量和收取量已知；(3)车辆在任何行驶路径上的载重量不能超过车辆最大载重量；(4)每个客户只能由一辆车服务，必须且只能访问一次；(5)客户要求同时送货物和取货物；(6)配送车辆的类型相同，装载量一定；(7)每辆车从配送中心出发配送和取货之后要返回配送中心；(8)访问客户必须在客户指定的时间窗内；(9)每个客户时间窗已知。

2.2 VRPSDP 数学模型的建立

2.2.1 模型中参数的含义

R：客户点的集合，编号分别为 $1,2,...,N$ ；R_0：客户点与配送中心 $\{0\}$ 的集合，即 $R_0=\{0\}\cup R$ ；d_{ij}：节点和节点 j 的距离；C_{ijk}：第 k 辆车从客户 i 驶向客户 j 时单位距离的运输成本；d_i：配送车辆在第 i 个客户点处的送货量；p_i：配送车辆在第个客户点处的收取

① 基金项目:国家电子信息产业发展基金(信部运[2006]634 号)
收稿时间:2008-12-08

货量；V：配送车辆的集合，各个车辆的编号分别为 $1,2,...,K$ ；MD：配送车辆的最大行驶距离；Q：配送车辆的最大载货能力；D_{ij}^{k} ：第 k 辆车从客户 i 驶向客户 j 时，该车承载的未配送货物的总量；P_{ij}^{k} ：第 k 辆车从客户 i 驶向客户 j 时，该车承载的已收取货物的总量；F_i：表示完成第 i 个客户的配送和收取任务所需要的时间；T_{ij}：表示从客户行驶到客户 j 所需的时间；E_i：表示任务 i 的允许最早开始时间；L_i：表示任务 i 的允许最晚开始时间；W_i；表示车辆在客户 i 的等待时间；S_i：表示车辆到达客户 i 的时间；B_i：表示客户 i 任务的开始时间；X_{ij}^{k} ：为 0,1 变量。每个客户点 i 的配送和收取任务都必须在时间窗[E_i, L_i]内开始，如果车辆到达客户点的时间早于 E_i，则车辆需要在 i 处等待，直到时间窗开启才能进行工作，如果在时间窗内到达则直接开始工作；车辆到达客户 i 的时间不能晚于 L_i。

2.2.2 数学模型

目标函数：

$$Minimize\sum_{k=1}^{K}\sum_{i=0}^{N}\sum_{j=0}^{N}C_{ijk}d_{ij}X_{ij}^{k}$$

约束条件：

① $\sum_{i=0}^{N}\sum_{k=1}^{K}X_{ij}^{k}=1$　　$j\in R$

② $\sum_{i=0}^{N}\sum_{j=0}^{N}d_{ij}X_{ij}^{k}\le MD$　　$k\in V$

③ $\sum_{i=0}^{N}X_{ij}^{k}-\sum_{i=0}^{N}X_{ji}^{k}=0$　　$j\in R,k\in V$

④ $\sum_{j=1}^{N}X_{0j}^{k}\le 1$　　$k\in V$

⑤ $\sum_{i=0}^{N}\sum_{k=1}^{K}X_{ji}^{k}P_{ji}^{k}-\sum_{i=0}^{N}\sum_{k=1}^{K}X_{ij}^{k}P_{ij}^{k}=p_j$　　$j\in R$

⑥ $\sum_{i=0}^{N}\sum_{k=1}^{K}X_{ij}^{k}D_{ij}^{k}-\sum_{i=0}^{N}\sum_{k=1}^{K}X_{ji}^{k}D_{ji}^{k}=d_j$　　$j\in R$

⑦ $\sum_{k=1}^{K}X_{ij}^{k}\left(P_{ij}^{k}+D_{ij}^{k}\right)\le Q$　　$i\ne j,i,j\in R_0$

⑧ $\sum_{i=1}^{N}X_{0i}^{k}D_{0i}^{k}=\sum_{i=0}^{n}\sum_{j=1}^{n}X_{ij}^{k}d_j$　　$k\in V$

⑨ $\sum_{i=1}^{N}X_{i0}^{k}P_{i0}^{k}=\sum_{i=0}^{n}\sum_{j=1}^{n}X_{ij}^{k}p_j$　　$k\in V$

⑩ $d_{0j}\le\frac{MD}{2}$　　$j\in R$

⑪ $0\le\sum_{k=1}^{K}X_{ij}^{k}P_{ij}^{k}\le Q 0\le\sum_{k=1}^{K}X_{ij}^{k}D_{ij}^{k}\le Q$　　$j\in R$

⑫ $B_i=\max\left\{S_i,E_i\right\}$　　$i\in R$

⑬ $\sum_{k=0}^{K}\sum_{i=0}^{N}x_{ijk}(S_i+W_i+F_i+T_{ij})\le S_j$　　$j\in R$

⑭ $S_j\le L_i$　　$i\in R$

⑮ $X_{ij}^{k}\in\{0,1\}$　　$i,j\in R_0,k\in K$

⑯ $W_i\ge 0$　　$i,j\in R_0$

①到④确保配送车辆的约束条件；⑤到⑪表示要满足货物量的约束条件；⑫确定了车辆在客户点处的等待时间；⑬保证了车辆在服务过程中前一个客户点和后一个客户点在时间上的顺序关系；⑭保证满足客户的硬时间窗约束；⑮及⑯保证了变量约束和非负约束。

3 求解VRPSDP的蚁群优化算法

蚁群算法是由意大利学者 M.Dorigo 等人首先提出。蚂蚁在觅食过程中会在其经过的路径上释放出一种特殊的分泌物—信息素，使得一定范围内的其他蚂蚁能够感知信息素的存在及其强度，并以此指导自己的行为；蚂蚁倾向于沿着信息素强度高的方向移动，当某些路径上走过的蚂蚁越多，后来蚂蚁选择该路径的概率就越高[3]。蚁群算法是一种并行的算法，全局搜索能力强，蚂蚁之间通过信息交流加速进化过程，利用正反馈原理，具有较强的发现较好解的能力。

4 求解带硬时间窗VRPSDP的关键实现技术

针对 VRPSDPTW 问题的复杂性，对 ACS 的初始可行解、初始化信息，路径转移规则，轨迹更新规则的具体设计如下：

4.1 初始可行解的生成进行改进

好的初始解可以决定蚂蚁的搜索空间同时又可以加快解的收敛速度。因此本文以 Clark 与 Wright (1964)提出的节约法[4]的概念为基础，加入时窗、总距离最小化和车容量限制因素修改节约值计算方式，以此来生成 VRPSDPTW 的初始解。带硬时间窗的 VRPSDP 要求车辆必须在时间窗内对客户服务,所以给等待时间一个惩罚值，来避免不必要的浪费。此外在同时送取货的条件下，取货使得装载量很容易超出车辆的最大装载能力，对于节点的送货量和取货量也给予一定的惩罚值。定义节约值的计算公式为：

$$S(i,j)=\begin{cases}s(i,j)-X+Y, Q_{now}>\lambda*Q_{start}\\ s(i,j)-X-Y, Q_{now}<\lambda*Q_{start}\end{cases}\quad(1)$$

其中，$X=\omega\left[MAX(E_j-B_j),0\right]$ 表示时间惩罚函数；

送货取货的差函数 $Y=\varphi(d_i-p_i)$；修改前的节约值 $s_{ij}=d_{i0}+d_{0j}-d_{ij}$，$Q_{now}$ 表示车辆的当前配送载货量，Q_{start} 表示车辆出仓时的配送载货量，ω、φ、λ 均为可调参数，变化范围在(0,1)之间。

初始解构造的算法流程如图 1 所示：

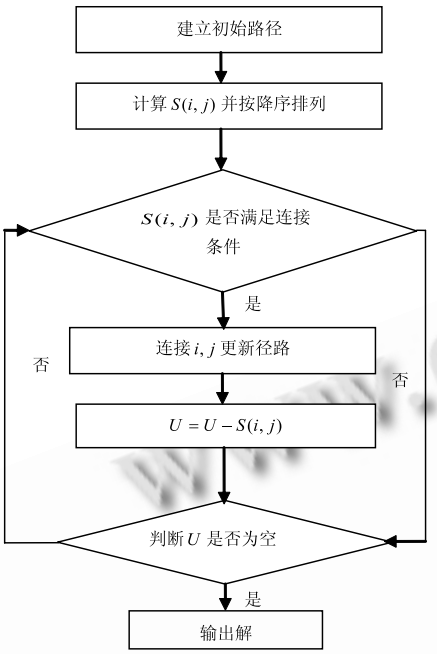

图 1　初始解构造流程图

4.2 初始化信息素

采用节约法构造初始解 s_0，然后初始化信息素 τ_0。

$$\tau_0=1/n*f(s_0),\quad \text{n 为客户数量} \tag{2}$$

4.3 路径转移规则

借鉴 Dorigo 等的 Ant-Q 算法思想[5]，将确定性与随机选择结合起来选择策略。

$$j=\begin{cases}\arg\max_{h\in\pi}\left\{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}[S_{ij}]^{\gamma}\right\} & q\le r_0\\ J & \text{否则}\end{cases} \tag{3}$$

$$J=P_{ij}^{k}=\begin{cases}\dfrac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}[S_{ij}]^{\gamma}}{\sum_{h\in\pi}[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}[S_{ij}]^{\gamma}} & \text{如果 j}\in\pi\\ 0 & \text{否则}\end{cases} \tag{4}$$

τ_{ij} 是节点 i 和节点 j 之间的信息素量。$\eta_{ij}=1/d_{ij}$ 是节点 i 和节点 j 之间的距离的倒数。S_{ij} 中 $s_{ij}=d_{i0}+d_{0j}-d_{ij}$ 反映两点直接连接与将两点分别与配送中心连接所能获得的路径长度的节约量。S_{ij} 的取值由 Q_{now} 和 $\lambda*Q_{start}$ 的大小决定。当 $Q_{now}>\lambda*Q_{start}$，此时选择送货量大而取货量小时间窗开启早的客户点；反之则选择取货量大而送货量小时间窗开启早的客户点。公式(3)、(4)表示当 $q\le r_0$ 时，按照(3)式选择最好的弧，否则按照(4)式选择。

4.4 轨迹更新规则：局部更新和全局更新

局部更新是指蚂蚁在构造解的过程中，每移动一步都要对相应的弧段上的信息素进行局部更新，这样既可以在一个迭代周期内加强各蚂蚁之间的协作；又能在本次迭代周期内所有的蚂蚁将在前一次全局更新过的最好路径的有限相邻区域内搜索。

$$\begin{aligned}&\tau(i,j)=(1-\rho)\cdot\tau(i,j)+\rho\cdot\tau_0\\&\rho\left(0\le\rho\le1\right)\end{aligned} \tag{5}$$

ρ 是信息挥发参数。全局信息素更新的目的是在最短路径上加入更多的信息素，应用全局信息素更新规则 $\tau(i,j)=(1-\alpha)*\tau(i,j)+\alpha*\Delta\tau(i,j)$ 来改变最短路径上的所有的弧 (i,j) 相关联的信息素值，其中 $\alpha\left(0\le\alpha\le1\right)$ 是信息素挥发参数。式中 L_{gb} 是目前得到全局最优解的路径长度。

$$\Delta\tau\left(i,j\right)=\begin{cases}\dfrac{1}{L_{gb}} & \text{如果弧 (i, j)}\in\text{全局最短路线}\\ 0 & \text{否则}\end{cases} \tag{6}$$

4.5 求解 VRPSDP 的 ACO-TS 算法执行流程

步骤 1. 初始化：设置 ACO 算法最大迭代次数 $I^{\max}$、当前迭代次数 $I=0$ 和蚂蚁数 m，参数信息素挥发系数 ρ、阀值 r_0、随机数 q 以及相对影响因子 $\alpha,\beta,\gamma,\lambda$。

步骤 2. 初始化信息素 τ_0，按公式(2)设置每条弧的信息素 $\tau(i,j)=\tau_0$，设置终止条件。

步骤 3. 构造路径：对于蚂蚁访问的每个客户节点首先进行可行性判断，判断其是否满足约束条件：①满足未被访问过；②满足车辆最大行驶距离限制；③满足车辆载重量限制；④满足时间窗条件。如果存在可行性节点，按照公式(3)或公式(4)选择移动到下一个节点 j，并按照公式(5)进行信息素的局部更新，以本次迭代的最优解为初始解，执行 2-交换法邻域搜索，重复步骤 3，否则转到步骤 4。

步骤 4. 所有客户节点都已在当前解集中，计算目标函数值,进行全局信息素更新；转到步骤 6，否则转到步骤 5。

步骤 5. 重新开始一条新的路径并重复上述构造过程，如在寻找过程中无法找到可行的下一个客户时，返回配送中心，然后再从配送中心出发直到所有的客户点都被访问。

步骤 6. 在可行解中选出本次迭代最优解,算法结束。

5 实验及结果分析

本文设计的算法对文献[6]中的实例(表 1)进行计算，其中站点的坐标为(3.2，14.1)，车辆的最大载重量为 8，车辆一次配送的最大距离是 50，车辆的行驶速度为 20。计算结果见表 2。表中 N 表示客户编号、X 表示横坐标、Y 表示纵坐标、D 为需求量、P 为供应量、Tds 表示送货开始时刻、Tde 送货结束时刻表示、Pds 表示取货开始时刻、Pde 表示取货结束时刻。

表 1　客户的数据

N	1	2	3	4	5	6	7	8	9	10
X	3.8	15.2	18.6	11.9	10.2	5.3	0.6	6.1	7.6	16.0
Y	5.5	10.9	12.9	8.2	9.5	9.6	9.9	15.0	19.2	15.7
D	0.8	0.6	0.4	1.6	0.8	0.6	1.9	1.3	1.8	1.8
P	0.5	1.6	0.7	0.3	0.6	1.4	1.7	1.0	0.9	1.9
Tds	8.8	2.4	0.1	9.8	9.6	0.3	5.9	1.7	0.6	9.8
Tde	14.6	8.1	4.9	14.1	13.8	6.0	11.6	7.5	5.2	13.9
Pds	6.0	8.0	6.8	0.6	1.7	5.6	8.0	4.5	4.9	6.1
Pde	14.6	13.8	12.3	5.0	6.3	10.4	12.6	8.5	10.3	12.0
N	11	12	13	14	15	16	17	18	19	20
X	15.3	1.6	9.0	5.4	7.8	18.6	14.5	15.0	9.8	1.4
Y	15.2	14.7	9.2	13.3	10.0	7.8	5.3	18.7	5.0	6.9
D	0.4	1.6	1.1	1.6	1.0	0.8	1.4	1.2	0.4	1.4
P	1.0	1.8	1.4	0.4	1.8	1.8	0.8	0.2	0.5	1.0
Tds	4.7	7.0	0.1	0.1	9.0	7.4	0.5	8.3	5.7	3.0
Tde	10.3	12.4	5.0	4.7	14.7	12.5	5.6	13.0	10.9	7.5
Pds	7.1	4.0	2.9	5.1	8.1	6.7	3.6	0.0	1.1	2.9
Pde	11.8	9.8	7.1	9.8	13.9	11.6	9.4	5.8	6.1	7.0

表 2　文献[6]的结果和本文的结果比较

算法	文献[6]中遗传算法	蚁群算法
使用车辆	5	5
行驶距离	153.8	144.2

为了验证改进的蚁群算法的性能，以表 1 中的数据为例，进行算法对比实验。表 2 是实验的最好结果。可见，本文设计的算法所求解的质量比文献[6]中的算法提高了 6.24%，由于配送路径在现实中是要不断重复行驶的，因此这在实际中的经济效益还是很明显的。

6 结语

本文根据物流配送领域中的一些实际情况研究了 VRPSDP 问题，以所有车辆总的行驶费用为目标建立相应的整数规划模型。结合蚁群算法，将车辆行驶时间、客户时间窗、服务时间等因素考虑进来，设计了求解 VRPSDPTW 的改进蚁群算法。最后，为检验算法的性能，与其他算法进行了比较与分析。结果表明，该算法收敛速度快，求解质量高，与同类文献相比可以减少行驶距离，从而节省总时间，因此本文的模型和算法是有效的。如果能将多站点、多车型的条件下的 VRPSDP 将更具有适用价值。

参考文献

1 Min H. The multiple vehicle routing problem with simultaneous delivery and pick-up points.Transportation Research A. 1989,23A:377－386.

2 Gendreau M, Laporte G, Vigo D. Heuristics for the traveling saleman problem with pickup and delivery. Computers and Operations Research, 1999,26:699－714.

3 段海滨.蚁群算法原理及应用.北京:科学出版社, 2005:26－29.

4 Clarke JG, Wright W. Seheduling of Vehicle from a Central Depot to a number of Delivery Points. Operation Researeh, 1964,12:568－581.

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

6 郎茂祥.物流配送车辆调度问题的模型和算法研究[博士学位论文].北京:北方交通大学, 2002.