基于改进蚁群算法的时间窗约束下物流配送车辆路径优化研究

●张玉春　余　炳　申风平

摘要：物流配送车辆路径优化作为一个涉及多影响因素、多目标需求的组合优化问题，其中带时间窗约束的物流配送车辆路径优化问题更是一个 NP 难题，较难得到最优解。文章分析带时间窗约束的车辆路径问题并建立相应数学模型，提出将变异和动态信息更新的改进蚁群算法应用于解决这类优化问题，同时仿真实验结果表明该算法能快速收敛于全局最优解，能有效地解决有时间窗约束下的物流配送车辆路径优化问题。

关键词：改进蚁群算法；时间窗约束车辆路径问题；物流配送

一、引言

物流配送过程中的车辆路径优化问题（Vehicle Routing Problem，VRP）作为物流配送优化的核心环节，一方面作为一项物流管理的重要内容，它对整个物流运输的速度、成本、效益起着至关重要的作用；另一方面随着现代物流配送的快速发展，物流配送越来越强调满足顾客种类、数量和时间等方面要求，提升顾客的满意度。配送车辆路径安排这一组合优化问题最初由 Dcmtzing & Ramser 于 1959 年提出，一直以来，作为交通运输和物流配送领域的一个核心问题，也成为一个运筹学、优化科学等学界研究的热点。

在实际应用中带时间窗的车辆路径问题（VRP With Time Windows，VRPTW）作为传统 VRP 问题的扩展和衍生，已被 Savelsbergh 证明是一个 NP 难题，对于大规模的 VRP 问题很难得到全局最优解。近年来在构造启发式算法和两阶段启发式算法的基础上发展起来的智能启发式算法如禁止搜索算法、模拟退火法、遗传算法、神经网络法、蚁群算法和粒子群算法等应用在有时间窗的车辆路径优化，取得了较好的效果。但这些算法存在着一些明显的缺陷，如：禁止搜索算法由于涉及复杂领域转换和求解策略，在现实中不易实现；模拟退火法也只能结合其它局部搜索算法构造混合算法应用；遗传算法不能保证最大的概率收敛于全局最优解；神经网络法、蚁群算法和粒子群算法易产生局部收敛和收敛速度较慢等。这篇文章研究一种高速收敛的改进蚁群算法，在该算法中，满足个点的时间窗约束的前提下采用一种新颖的动态信息新策略，以保证在每次搜索中，每只蚂蚁都对搜索做出贡献，同时还采取了一种独特的变异策略，以对每次搜索结果进行搜索，以对每次搜索的结果进行优化。

二、物流配送车辆路径优化问题的数学模型

时间窗约束下物流配送车辆路径优化问题可以描述为：从配送中心用多辆汽车向多个需求点送货，每个需求点的位置、需求量和时间窗约束一定，每辆汽车的载重量一定，要求合理安排汽车行驶路线，使总运输成本最小，并满足以下条件：

1. 每条配送路径上需求点的需求量之和不超过汽车载重量；

2. 每条配送路径的长度不超过汽车一次配送的最大行驶距离；

3. 每个需求点的需求量得到满足，且只能由一辆汽车送货；

4. 每个需求点的时间窗约束得到满足，且保证车辆工作总时间不超过其最长工作时间。

表 1　蚂蚁与配送车辆的关系

蚂蚁	车辆
蚂蚁走的两点之间路径	两配送之间的距离
路口	配送点

本文设配送中心有 M 辆汽车，第 k 辆汽车的载重量为 $Q_k(k=1,2,L,C)$ 其一次配送的最大行驶距离为 D_k，需要向 L 个需求点送货，每个需求点的需求量为 $q_i(i=1,2,L,L)$，时间窗为 $[e_i,u_i]$，其中 e_i 为任务 i 允许最早开始时间，如果车辆早于 e_i 到达，则需在 i 处等待；u_i 为任务 i 允许最迟开始时间，如果车辆晚于 u_i 到达，则任务 i 将被延迟进行。设 n_k 为第 k 辆汽车配送的需求点数（$n_k=0$ 表示未使用第 k 辆汽车），用集合 R_k 表示第 k 辆车的行驶路径，其中 r_{ki} 表示一个需求点，且这个需求点的路径 R_k 中的顺序为 i，$r_{k0}=0$ 表示配送中心。再设 t_{rki} 表示第 k 辆车在行驶路径 R_k 上到达 i 点的时刻，w_{rki} 表示第 k 辆车完成任务 i（如：验收、签单和卸货等）需要的时间。另外在目标函数中用 c_k 表示车辆 k 行驶的单位运输成本，p_e 表示在 e_i 之前到达需求点 i 单位时间的机会成本，p_u 表示在 u_i 之后到达需求点 i 单位时间的罚金成本。由此可建立如下物流配送车辆路径优化问题的数学模型。

$$\min z=\sum_{k=1}^{M} c_k[\sum_{i=1}^{n_k} d_{r_{k(i-1)}r_{ki}}+d_{r_{kn_k}r_{k0}}\mathrm{sign}(n_k)]+p_e\sum_{k=1}^{M}\sum_{i=1}^{n_k}\max(e_i-$$

$$t_{r_{ki}},0)+p_u\sum_{k=1}^{M}\sum_{i=1}^{n_k}(t_{r_{ki}}-u_i,0) \quad (1)$$

$$s.t.\sum_{i=1}^{n_k}q_{r_{ki}}\leq Q_k \quad (2)$$

$$\sum_{i=1}^{n_k}d_{r_{k(i-1)}r_{ki}}+d_{r_{knk}r_{k0}}sign(n_k)\leq D_k \quad (3)$$

$$\max(t_{r_{ki}}+w_{r_{ki}}+t_{r_{ki}r_{k0}})\leq u_0 \quad i=1,2,L,n_k\ \forall k \quad (4)$$

$$\min(t_{r_{ki}}-t_{r_{k0}r_{ki}})\geq e_0 \quad i=1,2,L,n_k\ \forall k \quad (5)$$

$$0\leq n_k\leq L \quad (6)$$

$$\sum_{k=1}^{c}n_k=L \quad (7)$$

$$R_k=\{r_{ki}|r_{ki}\in\{1,2,L,L\},i=1,2,L,n_k\} \quad (8)$$

$$R_{ki}IR_{kj}=\phi,\ \forall k_i\neq k_j \quad (9)$$

$$sign(n_k)=\begin{cases}1 & n_k\geq 1\\ 2 & \text{其它}\end{cases} \quad (10)$$

其中,式(1)为目标函数;式(2)保证每条路径上各需求点的需求量之和不超过汽车的重量;式(3)保证每条配送路径的长度不超过汽车一次配送的最大行驶距离;式(4)、(5)保证车辆的工作总时间不超过最长工作时间;(7)表明每条路径上的需求点都得到配送服务;式(8)为每条路径的需求点的组成;式(9)限制每个需求点只能由一辆汽车送货;式(10)第 k 辆汽车服务的客户数大于等于 1 时,说明该辆车参加了配送,则取 $sign(n_k)=1$,当第 k 辆汽车服务的客户小于 1 时,表示未使用该车,取 $sign(n_k)=0$。

三、算法的描述与实现

蚂蚁与配送之间的对应关系如表 1 所示。

在蚂蚁进行路径搜索时,如果找到一段很短的子路径(子解),它就释放出相应浓度的信息素,该信息素一方面直接影响位于子解的两个点上的蚂蚁;另一方面它会以该路径为中心向外扩散,影响其路径附近的其它蚂蚁的行为,使它们在寻找路径时会以更大的概率在下一步选择该路径。同时在时间窗约束下蚂蚁开辟新的路径遵循以下规则:从当前路径的最后一个路口出发,到所有未访问的路口中开始服务时间最早的的那个路口。只有当开始服务时间超过路口的时间窗时,才主动开辟一条新路径,并从未访问的路口中重新限制出发点,将所有未访问过的路口中开始服务最早的那个路口作为最新路径的第一个路口。通过这种基于时间窗约束下信息素扩散的协作方式,一方面指导蚂蚁在满足新的路口时间窗内开辟下一路径,另一方面其他蚂蚁在选择下一个路口时选择到最优路径的干扰性会降低。从而在满足各路口的时间窗要求的同时使算法的收敛程度大大提高,提升算法搜索的效率和成功率。

表 2 各客户点基本信息

序号	1	2	3	4	5	6	7	8
q_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
$[e_i,u_i]$	[1.5,4]	[4,6]	[1,2]	[4,7]	[3,5.5]	[2,5]	[5,8]	[1,4]

表 3 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 4 实验的对比结果

算法分类	搜索成功率	平均行驶成本	平均成功搜索时间
遗传算法	24%	993.6	17.6
基本蚁群算法	41%	949.6	10.28
改进蚁群算法	47.6%	937.1	9.39

具体的算法如下:

Step1:初始化,设置待定参数和最大进化代数;

Step2:随机选择每只蚂蚁的位置;

Step3:计算每只蚂蚁 k 将要转移的位置,假设为 j,上一个位置假设为 i。按所需等待时间较短和时间窗较窄的优先原则计算下一路口 j 的时间窗宽度和所在路口 i 到达下一路口 j 的时间; 然后按往下一路口 j 的路径长度以及路径上的信息量计算转移概率,计算公式为:

$$pP^{k}_{ij}=\begin{cases}\tau^{\alpha}_{ij}(t)\cdot\eta^{\beta}_{ij}(t)/\sum\limits_{s\in allowed_k}\tau^{\alpha}_{is}(t)\cdot\eta^{\beta}_{is}(t) & j\in tabu_k\\ 0 & otherwise\end{cases} \quad (9)$$

其中,$\eta_{ij}(t)=\frac{1}{d_{ij}}$

式中,设有路口 m 个,蚂蚁 n 个,$d_{ij}=(i,j=1,2,\wedge,m)$表示路口 i 和路口 j 之间的距离,$\tau_{ij}(t)$表示 t 时刻在路口 i 与路口 j 连线上信息素的浓度(初始时刻,各条路径上信息素的浓度相同)。

蚂蚁 $k(k=1,2,\wedge,n)$在运动过程中,根据各条路径上的信息素浓度决定转移方向,$tabu_k(k=1,2,\wedge,n)$为蚂蚁 k 已走过路口的集合,开始时 $tabu_k$ 中只有一个元素,即蚂蚁 k 的出发路口,随着进化的进行,$tabu_k$ 中的元素不断增加,随着时间的推移,以前留在各条路径上的信息素逐渐消逝,用参数 $1-\rho$ 表示信息素的挥发程度,经过 m 个时刻(所要指出的是:m 个时刻远大于任意一只蚂蚁 k 完成任意路径蚂蚁 $r_{k(i-1)ki}$ 所需时间),蚂蚁完成一次循环,各路径上的信息素逐渐消逝,用参数 $1-\rho$ 表示信息素的挥发程度。经过 m 个时刻,蚂蚁完成一次循环,各路径上的信息素的浓度根据(10)式调整,

$$\tau_{ij}(t+m)=\rho*\tau_{ij}(t)+\Delta\tau_{ij}$$

$$\Delta\tau_{ij}=\sum_{k=1}^{n}\Delta\tau^{k}_{ij} \quad \rho\in[0,1] \quad (10)$$

$\Delta\tau^k_{ij}$ 表示第 k 只蚂蚁在本次循环中留在路边 ij 上的信息素的浓度，$\Delta\tau_{ij}$ 表示本次循环所有蚂蚁在路径 ij 上所释放的信息素浓度之和。

Step4：按下式计算路径 ij 上新产生的信息素浓度：

$$\Delta\tau^k_{ij}=\frac{Q}{d_{ij}} \tag{11}$$

Q 为信息素强度，也就是蚂蚁经过所留轨迹数量的一个常数，它影响算法的收敛程度。

Step5：按下式计算从 i 扩散到其它路径 il 上的信息素的浓度：

$$\Delta\tau^k_{ij}=D^k_{il} \tag{12}$$

$$D^k_{il}=\begin{cases} r*Q/d_{ij}(1-d_{il}*(d_{il})^w\tan\theta/d^{w+1}), & \text{if } d_{il}<d^{-w+1}/(d_{ij})^w*\tan\theta \\ 0, & \text{otherwise} \end{cases}$$

(13)

w 为大于 1 的可调常数，d 为各城市的平均距离，r 为小于 1 的可调常数，θ 参数为一锐角。

Step6：按下式计算从 j 扩散到其它路径 jl 上的信息浓度。

$$\Delta\tau^k_{il}=D^k_{jl} \tag{14}$$

$$D^k_{jl}=\begin{cases} r*Q/d_{ij}(1-d_{jl}*(d_{jl})^w\tan\theta/d^{w+1}), & \text{if } d_{jl}<d^{-w+1}/(d_{ij})^w*\tan\theta \\ 0, & \text{otherwise} \end{cases}$$

(15)

Step7：若本次循环中每只蚂蚁都执行了 Step3~Step6，则转至 Step8，否则转向 Step3。

Step8：按式(10)更新各条路径上的信息素浓度，此时式(10)中取 m=1。

Step9：如果每只蚂蚁都完成了一个完整的路径，则转向 Step10，否则转向 Step3。

Step10：是否达到指定的进化代数或者所求得的解在最近若干代中无明显改进，如果出现这样情况，则转向 Step11，否则转向 Step3。

Step11：输出优化结果。

四、仿真实例

在这篇文章中，应用文献“7”和文献“10”中提出的算例进行仿真。由中心仓库 0 向 8 个需求点运输货物(编号为 1，2，…，8)，各任务货运量 $q_i(i=1,2,L,8)$、每客户所需工作时间及每项客户任务的时间窗由表 1 给出。这些任务发出的 3 辆容量为 8 吨的车辆完成，中心仓库与各客户点间及各客户点间的距离由表 2 给出，车速 50，单位运输成本为 1，超出时间窗的单位惩罚为：$p_e=50$，$p_u=50$。

用 1，2，L，L 表示各需求点，因为共有 M 辆汽车，最多存在 M 条配送路线，每条配送路径都始于配送中心，也终于配送中心。为了在编码中反映配送路线，采用了增加 M-1 虚拟配送中心的方法，分别用 L+1，L+2，L，L+M-1 表示。这样 1，2，L，L+M-1 这路径 L+M-1 个互不重复的自然数的通知数的随机排列就构成了一个个体，并对应一种配送路径方案。并将描述的算法用 MATLAB 编程，生成 NDMACO.m 文件运行程序。

在同一台计算机上将本文提出的改进蚁群算法运算结果与遗传算法、基本蚁群算法的求解结果进行比较。

五、结论

在这篇文章中，在有时间窗约束下的物流车辆配送路径问题应用变异和动态信息素更新的改进蚁群算法，即将 m 只蚂蚁均匀放置于 n 个边部配送点，采用最近邻居节点选择原则，在此基础上满足各路口时间窗约束，同时进行动态局部信息素更新并用变异算法加速局部寻优使收敛速度提高较多，在同样一台计算机上运算结果与遗传算法和基本蚁群算法等其他算法求解结果相比具有明显的优越性。因此，应用文中描述的算法，可以进行有时间窗约束的物流配送车辆路径优化，且满足各点时间窗需求的情况下较快地得到近似的最优解，为今后解决有时间窗约束下的物流配送车辆路径优化问题提供一定的参考。

参考文献：

1. 盛丽俊，周溪召．带有时间窗的车辆路径问题优化．上海海事大学学报，2007,(4):64-67.

2. 姜凌，沈桂兰．基于蚁群算法的物流配送车辆路径优化问题研究.首都经济贸易大学学报，2010,(1):71-74.

3. 即茂祥，胡思继.用混合遗传算法求解物流配送路径优化问题的研究.中国管理科学，2002,(10):51-56.

4. 亓霞，陈森发，黄鹍等.基于免疫算法的物流配送车辆路径优化问题研究.土木工程学报，2003,(7):43-46.

5. 张毅，申彦杰.时间窗约束下的车辆路径问题多目标优化算法.数学的实践与认识，2009,(7):124-130.

6. 刘北林，高爽.配送车辆路径优化问题算法研究.商业经济，2008,(9):31-33.

7. 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.

8. 朱庆保，杨志军.基于变异和动态信息素更新的蚁群优化算法．软件学报，2004,(2):185-92.

9. 肖国玺，彭玉青，林涛等．基于蚂蚁算法的物流配送计划的生成.河北工业大学学报，2005,(4):163-165.

10. 肖力.基于改进蚁群算法的物流配送问题研究.计算机仿真，2008,(4):182-185.

11. Dorigo M，Grambardalla L M. Ant colony system: a cooperative learning approach to the traveling salesmen problem. IEEE Transaction Son Evolutionary Computer，1997,(1):53-56.

12. 李志威，张旭梅．基于动态扫描和蚂蚁算法的物流配送网络优化研究．管理工程学报，2006,(4):9-12.

基金项目：国家自然科学基金资助项目“机动多目标跟踪融合策略研究”(项目号：61005026)。

作者简介：张玉春，南京航空航天大学管理科学与工程博士，兰州理工大学国际经济管理学院副教授、硕士生导师；余炳，兰州理工大学国际经济管理学院硕士生；申风平，兰州理工大学经济管理学院教授、硕士生导师。

收稿日期：2010-11-09。