第 24 卷第 5 期
2007 年 10 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 24 No. 5
Oct. 2007

文章编号: 1000−8152(2007)05−0807−04

基于混合算法的带时间窗车辆路径问题

陈宝文, 宋申民, 陈兴林

(哈尔滨工业大学 航天学院, 黑龙江 哈尔滨 150001)

摘要: 使用改进蚁群算法结合大规模邻域搜索算法解决带时窗限制的车辆路径问题. 首先对蚁群算法信息素及算法结构进行分析及改进, 并提出了新的解题策略, 由此得到可行解; 然后在区域改善部分用邻域搜索算法进一步提高解的性能. 给出混合算法计算Solomon100国际标准题库问题的结果, 并与同类方法的文献最优解进行比较.

关键词: 蚁群算法; 大规模邻域搜索算法; 带时间窗口车辆路径问题

中图分类号: TP301　　**文献标识码:** A

Hybrid metaheuristics for the vehicle routing problem with time windows

CHEN Bao-wen, SONG Shen-min, CHEN Xing-lin

(School of Astronautics, Harbin Institute of Technology, Harbin Heilongjiang 150001, China)

Abstract: A hybrid algorithm based on a modified ant colony optimization algorithm (ACO) and large neighborhood search (LNS) is proposed to solve a delivery vehicle routing problem with time window constraints (VRPTW). Firstly, an ant colony system is analyzed primarily. A new improved method of the traditional operation is then presented. This new ant colony optimization is used to get the initial solution of the vehicle routing problems with time window. In the local search improvement phase, large neighborhood search modules are also proposed to improve the result. Finally, Solomon's benchmark instances (VRPTW 100-customer) are tested for the algorithm and compared to the best solutions found in the literature.

Key words: ant colony optimization; large neighborhood search algorithm; vehicle routing problem with time window

1 引言(Introduction)

有时间窗车辆路径问题 (vehicle routing problem with time window, VRPTW)属NP-hard组合优化问题, 一般用启发式算法解决[1]. 蚁群算法 (ACO)作为新的仿生类算法, 对 TSP类问题有很大优势[2]. 1997年Bullnheimer等首次将蚁群算法应用到CVRP, 1999年Gambardella等首次成功将蚁群算法应用到VRPTW[3]. LNS算法作为另一种新出现的算法, 因其操作简易性和可靠性, 近年来发展很快. 2004年, Stefan Ropke提出自适应大规模邻域搜索 (adaptive large neighborhood search, ALNS)结构, 2005年David Pisinger等用此算法结构优化五类车辆路径问题, 包括VRPTW、CVRP、MDVRP、SDVRP、OVRP, 通过486个标准问题的测试, 说明算法的可靠性[4].

ACO还存在易停滞等现象, 同时单纯用LNS, 初始解差的问题非常突出, 需长时间优化. 本文算法主体结构综合ACO算法和LNS算法, 采用两阶段法优化: 第1阶段采用改进的ACO求初始解, 第2阶段用LNS算法改善解, 同时考虑时间窗的启发作用, 能快速找到高品质解.

2 启发式算法(Metaheuristic algorithm)

2.1 蚁群算法(Ant colony optimization)

基本蚁群算法模型[5]: 蚂蚁 k $(k = 1, 2, \cdots, m)$ 根据路径上的信息决定转移城市, $p_{ij}^k(t)$表示在 t时刻蚂蚁 k由城市 v_i转移到 v_j的概率.

$$p_{ij}^k(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum\limits_{r\in S_i^k}\tau_{ir}^{\alpha}(t)\eta_{ir}^{\beta}(t)}, j \in S_i^k, \\ 0, \qquad\qquad\qquad 其他, \end{cases} \tag{1}$$

其中: $\tau_{ij}(t)$表示 v_i到 v_j这条弧 E_{ij}的信息素强度, $\tau_{ij}(0) = C$. η_{ij}表示 E_{ij}上路径信息: 参数 α, β 分别表示信息素和路径信息的对转移概率

收稿日期: 2005−09−09; 收修改稿日期: 2006−11−10.
基金项目: 国家教委留学归国人员基金资助项目.

的影响. S_i^k是蚂蚁 k在位置 i处的可行解域. 集合 $tabu_k(k=1,2,\cdots,m)$记录蚂蚁 k 已走城市, 随蚂蚁搜索进程作动态调整. 用 ρ 表示随着时间推移, 留在路径上信息素的挥发程度. 经过 n 时刻, 每个蚂蚁完成一次循环, 各路径上信息素按下式调整:

$$\tau_{ij}(t+n)=\rho\cdot\tau_{ij}(t)+\sum_{k=1}^{m}\Delta\tau_{ij}^k, \tag{2}$$

其中 $0<\rho<1$, $\sum\limits_{k=1}^{m}\Delta\tau_{ij}^k$表示本次循环中所有经历过路径 ij 的蚂蚁留在该路径上的信息素增量, 局部调整用 ant-cycle system调整方法.

$$\Delta\tau_{ij}^k=\begin{cases}Q/L_k, k\in(i,j),\\ 0, \quad \text{其他},\end{cases} \tag{3}$$

其中: Q是常数, L_k表示第 k只蚂蚁在本次循环中所走距离. 最大最小信息素策略(MMAS)[6]要求 $\tau_{ij}(t)\in[\tau_{\min}],\tau_{\max}$, 如果算法停滞, 需对信息素更新:

$$\tau_{ij}=(1-\rho)\cdot\tau_{ij}(t)+\rho\tau_0, \tag{4}$$

其中 $0<\rho<1$, $\tau_0=\tau_{\max}$.

2.2 大规模邻域搜索算法(Large neighborhood search algorithm)

LNS算法1998 年由Paul Shaw提出[7]. 它从初始解出发, 经过remove和re-insertion两个过程的不断优化, 最终得到优化解.

2.2.1 Remove 过程 (Remove process)

σ表示当前最好解, c 表示被移走客户, S表示被移走客户集, p表示移走客户数, σ'表示从 σ中移走 p 个客户后剩余的部分解, Paul Shaw 的策略: 首先, 从 σ中随机移走一个客户到 S 中, 作为集合 S中的第一个元素. 剩下的 $p-1$ 个元素按照如下的方法来选定: 每次从集合 S中随机选一客户 z, 将 σ中剩余的客户按照与 z的相关性由小到大排列. 从 σ中选出与 z的相关性最大的客户 c, 从 σ中移走 c中, 并把它加入到 S中. 重复该过程 $p-1$次, 直到剩下的 $p-1$个元素都选好. 相关性的定义:

$$\text{relateness}(i,j)=\frac{1}{t'_{ij}+v_{ij}}, \tag{5}$$

式中$t'_{ij}=\dfrac{t_{ij}}{\max t_{ij}}$, 取值属于 $(0,1)$. v_{ij}当 i 和 j 在同路径上时为0, 否则为1. 即地理位置相距近的客户的相关性比相距远的要大. 为避免仅根据相关性挑选客户, 相同客户有被反复选出的可能, Paul Shaw 在算法中加入随机因素: $\text{Random}([0,1])^B\times\sigma$ 中剩余的客户数目, 取出相关性大小序列中的某一客户. 可调节参数 B , 挑选相关性相对大小的客户. 移走的客户数 p 也可调节, 开始可设为1, 当解迭代iteration 次后仍不能得到改进, 则 p 增加1, p设置上界, 否则太大会增加运行时间.

2.2.2 Re-Insertion 过程 (Re-insertion process)

re-insertion 过程是将客户集 S 中的点重新插回部分解 σ', 以产生更好的解. 当 S 中的客户数目较多时, S 中的所有客户重新插回 σ' 中将会有很多种插入组合, 穷举不可行. 解决方式是首先计算 S 中每一客户的最佳插入位置. 将 S中的客户 c 插回部分解 σ' , 在多个插入位置中, 找出使目标值增加最少的那个位置, 并记下目标值. 比较 S 中各客户的最佳插入位置对应的目标值, 从中选取能使目标值增加最少的客户 c. 将 c插回部分解 σ' 中, 把 c 从 S 中移走, 同时搜索将 S 中剩余的客户都插回 σ'. 直到 S 为空集, 此时被移走的客户都已经插回 σ'. 比较解 σ_{new} 与最好解 σ, 如果 σ_{new}比 σ 要好, 则用 σ_{new} 替换 σ. 可以用分枝定界re-insertion 过程, 大多数情况能够确定在很短的时间内对 $p\leqslant 15$ 找到一个更好的解或证明没有更好的解. 然而对某些问题, 当 p 继续增大时, 该过程却显得耗时太长.

3 混合算法设计(Hybrid metaheuristics design)

3.1 算法设计思路(The method)

Gambardella和Taillard 1999年应用到VRPTW问题中的算法MACS-VRPTW[3]采用两个蚁群. (ACS-VEI)蚁群缩短车辆数, (ACS-TIME)蚁群缩短运行距离, 蚁群间用信息素交流.算法结构复杂, 并且对R和RC1类问题的解决效果不理想. 本文算法结构采用单蚁群加LNS算法, 算法结构简单, 能快速找到优化解. 以下为清晰起见, 举solomon100测试题中的C101标准问题为例子.

首先, 中心数目设计为25个(可选车辆数), 编号0～24的点具有相同的坐标、装载量及时间窗. 蚂蚁分置在不同中心, 每一只蚂蚁从各自的中心出发, 巡回城市, 算法时间步为125步. 中心间距离用距离矩阵的最大值代替, 避免能见度函数除零. 当蚂蚁重访中心时, 装载量和时间清零. 这样处理后采用MMAS算法框架. 同时改写公式(11), 加入时间窗口因素[8].

$$p_{ij}^k(t)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)wc(t)^{\gamma}}{\sum\limits_{r\in S_i^k}\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)wc(t)^{\gamma}}, j\in S_i^k,\\ 0, \qquad\qquad\qquad\qquad \text{其他},\end{cases} \tag{6}$$

其中$wc(t)^{\gamma}$为城市 j 的时间窗. 目标函数的设计体现了需要路径最短和车辆数少的特点.

$$f=F\cdot a+K\cdot b, \tag{7}$$

其中: $a+b=1,\ 0<a<1,\ 0<b<1$. F代表最短路径, K代表车辆数. 计算可得C101的解. 其中0∼24标号的城市为派送中心, 25∼124为原Solomon数据中的1∼100. 即如初始解的一条序列中6 81 79 78 77 80 82 84 83代表了一辆车的回路, 原序列标号为0 57 55 54 53 56 58 60 59. 接下来采用LNS算法, 抽取城市, 再插入可行解中改善最短路径. 如果目的对减小车辆数更重视, 那么操作时可先将其中一条城市点最少的路径中的城市全抽出来, 然后再插入别的路径中. 实际操作中采用随机抽取方式已取得较好结果, 最优解C101城市序列 (10辆车, 路径828.94).

车辆 1: 15 56 57 55 59 61 62 63 60 58

车辆 2: 17 67 66 65 64 68 70 69 72 75 74 76 73 71

车辆 3: 12 44 48 49 51 53 54 52 50 47 46 45

车辆 4: 14 81 79 78 77 80 82 84 83

车辆 5: 5 37 41 42 43 39 40 38 36

车辆 6: 20 91 89 87 86 98 96 85 88 92 90 93

车辆 7: 10 29 27 31 32 34 35 33 30 28 26 25 99

车辆 8: 7 105 102 100 95 94 97 101 103 104

车辆 9: 16 122 120 119 118 116 117 121 124 123

车辆10: 9 114 111 110 107 106 108 109 112 113 115

8 13 2 19 1 23 4 22 3 21 24 11 6 0为剩下的多余派送中心. 算法增加判断: 假如序列倒数第2位标号大于24, 则设定此解为不可行解, 舍弃.

3.2 算法步骤(Steps of algorithms)

Step 1 初始化. 载入城市及车辆信息. 初始化算法参数. 设 ψ^* 为最优解. ψ 为用ACS求得的初始解.

Step 2 复制中心数等同车辆数, 对于不同蚂蚁 k, 选择不同中心. $\mathrm{currenttime}_k \leftarrow 0,\ \mathrm{Load}_k \leftarrow 0$.

Step 3 候选点 j 为满足下列条件的可选点, 并设置禁忌表, $\mathrm{currenttime}_k + t_{ij} \leqslant e_j,\ \mathrm{Load}_k + q_j \leqslant \mathrm{Load}_{VEI},\ \mathrm{tabu}_k(j)=1$.

Step 4 计算选择概率. 若可选点集有中心, 则通往该点路径信息素设置 $\tau_{\min}$. 若可选点全为派送中心, 则将等概率选某一中心. 使用公式 (8), 采用选择机制选择 j点 $\mathrm{currenttime}_k \leftarrow \max(\mathrm{currenttime}_k + t_{ij}, b_j),\ \mathrm{Load}_k \leftarrow \mathrm{Load}_k + q_j \leqslant \mathrm{Load}_{VEI}$. 若 j点为派送中心, 则: $\mathrm{Load}_k \leftarrow 0,\ \mathrm{currenttime}_k \leftarrow 0$ 直到候选集为空. 2-opt操作后得到 ψ.

Step 5 解序列调整. 初始解 ψ 循环执行: 按相关性抽取城市点 z, $\mathrm{vector}(S) \leftarrow z$. 可以插入位置 j 满足 $\Sigma_i \in \mathrm{loop}, \mathrm{Load}_k + q_z \leqslant \mathrm{Load}_{\mathrm{VEI}}$, $\mathrm{currenttime}_{j-1} + t_{(j-1),z} \leqslant e_z,\ \mathrm{currenttime}_{zj} + t_z \leqslant e_j$ 按城市插回序列增加最少距离修补序列. 得到新解 $\sigma_{\mathrm{new}} = \min(\sigma_{\mathrm{new}}, \psi)$

Step 6 更新信息素: $\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \rho\, J_\psi$

Step 7 使用MMAS思想$\tau < \tau_{\min} \Rightarrow \tau \leftarrow \tau_{\min}$, $\tau > \tau_{\max} \Rightarrow \tau \leftarrow \tau_{\max}$, 如果固定次数 ψ^* 不更新, $\tau_{ij} = (1-\rho)\cdot\tau_{ij}(t) + \rho\tau_0$.

4 仿真实验及结果分析 (Simulation and analysis)

本文程序采用C++编写, 调用MATLAB7.0绘图. 执行机器PentiumⅣ2.4 GHz, 512 M内存. 数据采用Solomon100标准问题. 蚁群算法主要参数:

$$Q=1,\ \alpha=1,\ \beta=2.5,\ \gamma=1,\ \rho=0.1,$$
$$\tau_{\min}=0.00024,\ \tau_{\max}=1.$$

选择机制用轮盘赌. LNS参数iteration过程的迭代次数30, 若找不到更好的解, 可增加上限到100; 上选择机制用轮盘赌. LNS参数iteration过程的迭代次数30, 若找不到更好的解, 可增加上限到100; p上界为10, B 取13. 目标函数参数 $a=0.6,\ b=0.4$. 混合算法总运行次数3000次. 以下是MATLAB7.0绘出的其中一个最优路径.

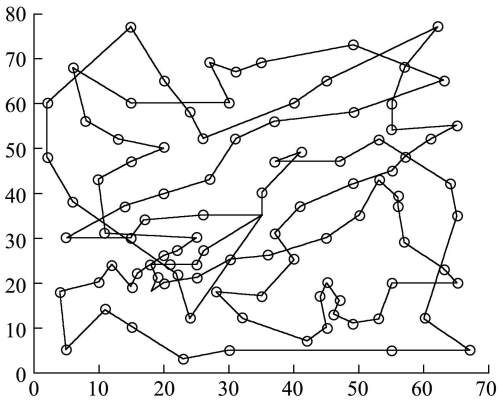

图 1 R207最优路径

Fig. 1 R207 best routing

混合算法对所有问题都能很快找到合适的次优解, 并且在某些题型中可找到目前知道的最优解. 所比较文献来自如下网址: http://www.idsia.ch/ luca/macs-vrptw/solutions/welcome.htm.

表1给出与同类文献最优结果比较, 表2给出混合算法对Solomon100的所有计算结果.

表 1 混合算法结果与文献中最优解比较

Table 1 Hybrid metaheuristics result compared to the best solutions found in the literature

Instance	best result		Our best results		Time/s
R202	3	1202.529	3	1197.665	185.33
R204	2	856.364	2	849.619	209.48
R207	2	894.889	2	890.608	195.41
R209	3	921.659	3	916.474	215.83
RC207	3	1068.855	3	1062.048	182.38

表 2 混合算法结果

Table 2 Hybrid metaheuristics results

Instance	Our	best result	Instance	Our	best result
C101	10	828.94	C201	3	591.56
C102	10	868.69	C202	3	591.56
C103	10	828.14	C203	3	591.17
C104	10	824.78	C204	3	590.60
C105	10	828.94	C205	3	588.88
C106	10	828.94	C206	3	588.49
C107	10	828.94	C207	3	588.29
C108	10	828.94	C208	3	589.86
C109	10	828.94			
R101	19	1650.80	R201	4	1253.26
R102	14	1526.18	R202	3	1197.67
R103	13	1306.73	R203	3	945.55
R104	12	1227.65	R204	2	849.62
R105	13	1444.52	R205	3	1008.52
R106	12	1252.03	R206	3	913.18
R107	12	1256.61	R207	2	890.61
R108	9	960.88	R208	2	734.85
R109	11	1214.54	R209	3	916.47
R110	10	1119.00	R210	3	954.12
R111	11	1097.38	R211	2	933.75
R112	9	1009.73			
RC101	16	1684.10	RC201	4	1406.94
RC102	14	1526.18	RC202	3	1407.52
RC103	13	1362.72	RC203	3	1073.39
RC104	12	1227.65	RC204	3	806.12
RC105	13	1633.72	RC205	4	1326.83
RC106	11	1424.73	RC206	3	1160.91
RC107	12	1256.61	RC207	3	1062.05
RC108	11	1221.35	RC208	3	832.36

5 结束语 (Conclusion)

本文提出一种由改进蚁群算法和LNS结合的算法, 并成功应用于VRPTW问题. 算法结构的安排, 使ACO和LNS互为补充, 提高寻优的精确性, 能有效解决VRPTW类问题. 对于大部分题型, 算法能很快找到近似最优解, 所得最优解与同类文献的比较证明算法的优良性. 不过如果LNS算法有太多的交换动作, 会导致算法运行时间过长.所以如何进一步提高算法时效性, 应成为今后研究重点.

参考文献(References):

[1] JEAN-FRANCOIS CORDEAU, GILBERT LAPORTE, ANNE MERCIER. A unified tabu search heuristic for vehicle routing problems with time windows[J]. *J of the Operational Research Society*, 2000, 52(4): 928 – 936.

[2] DORIGO M, STUTZLE T. *Ant Colony Optimization*[M]. Cambridge, MA: MIT Press, 2004.

[3] GAMBARDELLA L M, TAILLARD E, AGAZZI G. MACS VRPTW: Vehicle routing problem with time windows[C] // *New Ideas in Optimization*. London: McGraw Hill, 1999.

[4] ROPKE S, PISINGER D. A unified heuristic for vehicle routing problems with backhauls[J]. *European J of Operational Research*, 2006, 171(3): 750 – 775.

[5] DORIGO M, GAMBARDELLA L M. Ant colony system: A cooperative learning approach to the traveling salesman problem[J]. *IEEE Trans on Evolutionary Computation*, 1997, 1(1): 53 – 66.

[6] TOMAS S, HOLGER H H. Max-min ant system[J]. *Future Generation Computer Systems*, 2000, 16(8): 889 – 914

[7] SHAW P. Using constraint programming and local search methods to solve vehicle routing problems[C] //*Principles of the Fourth Int Conf on Principles and Practice of Constraint Programming*. Berlin, Germany: Springer-Verlag, 1998: 417 – 431.

[8] TAN K C, LEE L H, ZHU Q L, et al. Heuristic methods for vehicle routing problem with time windows[J]. *Artificial Intelligence in Engineering*, 2001, 15(3): 281 – 295.

作者简介:

陈宝文 (1979—), 男, 博士研究生, 目前研究方向为智能控制、智能计算等领域, E-mail: chenbaowen@hit.edu.cn;

宋申民 (1968—), 男, 博士, 副教授, 目前研究方向智能优化与智能控制在机器人中应用, E-mail: songshenmin@163.com;

陈兴林 (1963—), 男, 博士, 教授, 博士生导师, 目前研究方向为仿人步行智能机器人、计算计视觉, E-mail:chenxl@hit.com.