

多车辆合乘问题的两阶段聚类启发式优化算法

邵增珍^{1,2} 王洪国¹ 刘弘^{1,2} 宋超超³ 孟春华³ 于洪玲³

¹(山东师范大学信息科学与工程学院 济南 250014)
²(山东省分布式计算机软件新技术重点实验室 济南 250014)
³(山东师范大学管理科学与工程学院 济南 250014)
(shaozengzhen@163.com)

Heuristic Optimization Algorithms of Multi-Carpooling Problem Based on Two-Stage Clustering

Shao Zengzhen^{1,2}, Wang Hongguo¹, Liu Hong^{1,2}, Song Chaochao³, Meng Chunhua³, and Yu Hongling³
¹(*Institute of Information Science and Engineering, Shandong Normal University, Jinan 250014*)
²(*Shandong Provincial Key Laboratory for Distributed Computer Software Novel Technology, Jinan 250014*)
³(*Institute of Management Science and Engineering, Shandong Normal University, Jinan 250014*)

Abstract In logistics engineering, transportation system and some other domains, the effect of carpooling is enormously significant. Excellent carpooling strategy may not only economize logistics costs, reduce the traffic jam, but be especially favorable for reducing noise and improving the environment. Thus, a two-stage clustering heuristic strategy is introduced to solve deterministic multi-carpooling problem. In the first stage of the strategy, the conception of matching degree is proposed to guide to assign service requirements to specific vehicle, hence the original problem can be split into several single-vehicle problems. And in the second stage of the strategy, based on priori clustering idea, the number of insertion operations in one single-vehicle matching progress is reduced greatly so as to improve the efficiency of the algorithm. To improve the success rate of matching and reduce total costs, the assignment schema of vehicles is not fixed but adjustable. Heuristic emigration and immigration operators are used in a new distribution when some requirements fail to be assigned or probably lead to long arcs in the last distribution. To verify the validity and practicability of the method, some actual samples are generated based on real map information. Experimental results show the algorithm may not only improve ride success rate greatly, reduce total vehicle costs, but also demonstrate strong practicality.

Key words multi-carpooling problem; two-stage clustering; matching degree; priori clustering; emigration and immigration operators; heuristic algorithms

摘要 车辆合乘问题研究在物流领域和交通领域意义重大. 良好的合成策略不仅可以节省物流成本, 降低交通拥塞, 在减少噪声及提高环境等方面也是很有利的. 针对确定性多车辆合乘匹配问题, 提出了两阶段聚类的启发式匹配策略: 第1阶段聚类过程提出匹配度的概念, 用于指导将服务需求分配到某一具体车辆, 从而将多车辆问题转化为单车辆问题; 第2阶段聚类过程基于“先验聚类”插入思想, 可降低单车辆匹配过程的插入试探次数, 从而提高算法效率. 为提高搭乘成功率并降低运营总成本, 通过迁移

对第 1 阶段聚类过程进行调整. 实际算例结果表明, 算法在可接受时间范围内不仅可提高搭乘成功率, 还明显降低车辆的运行成本, 表现出较强的实用性.

关键词 多车辆合乘匹配问题; 两阶段聚类; 匹配度; 先验聚类; 迁出/迁入算子; 启发式算法

中图法分类号 TP18

多车辆合乘匹配问题(multi-carpooling problem, MCP)^[1]也称为“拼车”问题, 是指相近路线的服务需求乘坐同一车辆出行, 合乘产生的费用分摊. 美国、新加坡等为应对能源危机, 早在 20 世纪就提出了合乘概念, 并对合乘过程进行了一些研究^[2-3]. 合乘问题在四方物流优化领域、城市交通领域都具有广泛应用, 是一个具有很大研究价值和应用领域的问题. MCP 问题属于一类特殊的装卸货^[4]问题, 其优化目标是在满足车辆出行时间约束基础上, 最大化搭乘成功率, 并尽量降低总体出行成本. 该问题同 PDP 问题又有不同之处, PDP 问题中的车辆在行驶路线、行驶时间等方面的约束较小, 而 MCP 问题中首先需要满足车辆本身要求, 然后再考虑其他服务需求的搭乘可能性. 有关 PDP 问题的研究成果较多, 如禁忌搜索算法^[5]以及分支定界法等^[6]; Cortes 等人^[7]提出利用分支定界法解决具有换乘特征的装卸货问题; Parragh^[8]提出了基于异质用户的 dial-a-ride 问题解决方案; Häme^[9]提出适应性插入算法解决窄时间窗口的单车 dial-a-ride 问题; 我国学者张瑾等人^[10]提出利用模拟退火算法解决出租车合乘问题; 翟泳等人^[11]提出基于交通路网的合乘路径匹配算法; 王万良等人^[12]提出了基于量子进化算法的车辆共享模型. 以上研究主要针对单车辆展开, 且大多停留在政策或理论研究阶段, 有关多车辆合乘的实用成果尚不多见.

本文针对确定性 MCP 问题展开研究. 基于提高搭乘成功率、降低总体成本的优化目标, 本文提出一种新的合乘匹配启发式算法——两阶段聚类启发式插入算法. 算例结果表明, 本算法能在较短时间内寻找到优化方案, 不仅能保证很高的搭乘成功率, 还可有效降低运行总成本.

1 多车辆合乘匹配问题数学模型

1.1 问题描述

某区域有 m 辆车组成车队 F , 车辆可在满足自身出行需要的基础上对外提供搭乘服务. 区域内有 n 个服务需求组成服务需求集合 P . 车辆的型号、空

余搭载量及行驶速度各不相同, 车辆出行信息及服务需求信息确定且保持不变. 不考虑服务需求的差异性, 假设: 1) 所有车辆及服务需求可在规定时间窗口内到达上车点; 2) 如车辆到达某一上车点的时刻早于服务时间窗口下界, 需等待至该时间窗口下界, 并在同一时刻离开; 3) 车队能满足服务需求中的最大容量, 且所有服务需求量的总和不大大于车队剩余容量. 问题目标是在尽可能提高搭乘成功率的前提下降低车队总体运行成本.

1.2 问题形式化

$\forall j \in F, d_j^+, d_j^-$ 为车辆起点和终点; $\forall i \in P, p_i^+, p_i^-$ 分别为服务需求上、下车点编号. 令集合 $F^+ = \{d_j^+ | j \in F\}, F^- = \{d_j^- | j \in F\}, P^+ = \{p_i^+ | i \in P\}, P^- = \{p_i^- | i \in P\}$. 将车辆本身视为特殊服务需求, 记所有上、下车点编号集合为 $N^+ = F^+ \cup P^+, N^- = F^- \cup P^-, N = N^+ \cup N^-$. 定义所有服务需求的上、下车点服务时间窗口为 $[e_x, l_x], x \in N$.

研究区域内点的集合为 $V = N \cup V'$, 其中 V' 表示车辆可经过的其他道路节点的集合, $|V'| = s$. V 中任意两点的路径距离已知. 定义图 $G = (V, E), E$ 为图 G 所有边的集合, $E = \{(x, y) | x, y \in V\}$. 定义 $A = \{a_{x,y} | x, y = 1, 2, \dots, 2(m+n) + s\}$ 为图 G 的距离矩阵. 不考虑路径方向性问题, 有 $a_{x,y} = a_{y,x}, a_{x,x} = 0$. 基于距离矩阵 A , 定义车辆 j 在任意两点间所用的行驶时间矩阵 $T^j = \{t_{x,y}^j | x, y = 1, 2, \dots, 2(m+n) + s\}$, 其中 $t_{x,y}^j = d_{x,y} / v_j, v_j$ 为车辆 j 的平均速度, $d_{x,y}$ 为顶点 x 到顶点 y 的最短路径长度. 设车辆固定运行成本为 fc_j , 单位里程内每增加一单位运载量增加成本为 ac_j, Q_j 为车辆 j 的最大容量, $j \in F$. $\forall i \in N^+, \Delta q_i^+$ 定义其上货量为 Δq_i^+ ; $\forall i \in N^-, \Delta q_i^-$ 定义卸货数量为 Δq_i^- . 为简单起见, $\forall i \in F \cup P$, 令 $\Delta q_{down(i)}^+ = -\Delta q_{up(i)}^+$.

1.2.1 变量定义

1) $X_{x,y}^j$: 二进制变量, $x, y \in V, x \neq y, j \in F$. 当车辆 j 到达节点 x , 且成功搭乘某服务需求, 或者车辆 j 行驶到搭乘服务需求的下车点 y , 并将该服务需求成功在该点下车时, $X_{x,y}^j = 1$, 否则 $X_{x,y}^j = 0$, 其中节点 y 是 x 的直接后继.

2) T_x^j :时刻变量, $x \in V, j \in F$ 为车辆 j 到达点 x 的实际时刻.

3) B_x^j :时刻变量, $x \in V, j \in F$ 为车辆 j 离开点 x 的时刻, $x \in N$ 时满足时间窗口约束, 有:

$$B_x^j = \max(T_x^j, e_x), x \in N. \quad (1)$$

4) q_x^j :货品数量, $x \in V, j \in F$ 为车辆 j 离开点 x 时的货品数量, 有 $q_x^j \leq Q^j$.

5) $Cost_j$:为车辆 j 在某次出车全程的总花费. 设车辆 j (出发点编号为 j) 离开出发点时的状态为 0 状态, 记为 $Status_{x_0}^j$. 每到达一个顶点后离开, 就转化为一种新的状态. 记为 $Status_{x_u}^j$ 是车辆 j 的第 u ($u > 0$) 个状态, $Status_{x_u}^j = (q_{x_{u-1}}^j, s_u^j)$, $q_{x_{u-1}}^j$ 是车辆 j 到达顶点 x_u 时的货品数量, 即车辆 j 在边 (x_{u-1}, x_u) 上运行时的货品数量; s_u^j 是顶点 x_u 到顶点 x_{u-1} 之间的路径长度, 即 $s_u^j = a_{x_{u-1}, x_u}$. 在这一段路程中, 车辆 j 的变动成本为 $ac_j \times s_u^j \times q_{x_{u-1}}^j$. 设车辆 j 到达其终点(顶点编号为 $m+n+j$) 时共经历了 U_j+1 (经过的弧的个数为 U_j) 个状态, 则该次搭乘行程中车辆 j 的总花费为

$$Cost_j = fc_j + \sum_{u=1}^{U_j} ac_j \times s_u^j \times q_{x_{u-1}}^j. \quad (2)$$

6) WT_x :时间变量, $x \in P$ 为 x 的等待时间, 即从最早上车时间到实际上车时间之间的时间, 有:

$$WT_x = T_x^j - e_x. \quad (3)$$

7) RT_x :时间变量, $x \in P$ 是服务需求 x 从上车点到下车点之间所用时间, 即乘车时间. 有:

$$RT_x = T_{m+n+x}^j - B_x^j, j \in F. \quad (4)$$

1.2.2 目标函数定义

考虑最大化搭乘成功率及总花费最少, 有:

$$\max \sum_{j \in F} \sum_{x \in P^+} \sum_{y \in V \setminus \{x\}} X_{x,y}^j; \quad (5)$$

$$\min \sum_{j \in F} Cost_j. \quad (6)$$

1.2.3 约束条件说明

约束 1. 时间窗口约束. 车辆 j 需在规定时间内窗口内到达上下车点, 否则无法满足服务需求. 有:

$$e_x \leq T_x^j \leq l_x, x \in N, j \in F. \quad (7)'$$

特殊情况下会出现车辆 j 早于时刻 e_x 到达顶点 x 的情况, 处理方法见前文所述, 则(7)'可修正为

$$T_x^j \leq l_x, x \in N, j \in F. \quad (7)$$

约束 2. 一次访问约束. 车辆 j 到某一个上、下车点, 每一个点只能被服务一次. 有:

$$\sum_{j \in F} \sum_{y \in V \setminus \{x\}} X_{x,y}^j \leq 1, x \in N; \quad (8)$$

$$\sum_{j \in F} \sum_{x \in V \setminus \{y\}} X_{x,y}^j \leq 1, y \in N. \quad (9)$$

式(8)表明从 x 点出发的车辆最多只有一辆; 式(9)表明从其他点进入 y 点的车辆最多只有一辆.

约束 3. 车辆出发点、到达点约束. 车辆必须从规定地点出发且到达指定终点. 有:

$$\sum_{y \in V} X_{d_j^+, y}^j = 1, j \in F, d_j^+ \in F^+; \quad (10)$$

$$\sum_{x \in V} X_{x, d_j^-}^j = 1, j \in F, d_j^- \in F^-; \quad (11)$$

其中式(10)保证车辆 j 一定从其出发点出发, 式(11)保证车辆 j 一定能到达其终点.

约束 4. 成对约束. 服务需求的上车点 x 和下车点 $m+n+x$ 如被服务, 则必须被同一辆车服务. 有:

$$\sum_{y_1 \in V \setminus \{x\}} X_{x, y_1}^j - \sum_{y_2 \in V \setminus \{m+n+x\}} X_{y_2, m+n+x}^j = 0, \quad x \in P^+, j \in F. \quad (12)$$

约束 5. 访问顺序约束. 服务需求的上车时刻加上乘车时间, 应在该服务需求的下车时刻之前. 有:

$$T_x^j + RT_x \leq T_{m+n+x}^j, x \in N^+, j \in F. \quad (13)$$

约束 6. 车辆搭载容限约束. 行驶过程中任意时刻不能超过车辆容限. 有:

$$q_x^j = initq_j, x \in F^+, j \in F; \quad (14)$$

$$q_x^j = initq_j, x \in F^-, j \in F; \quad (15)$$

$$q_x^j \leq Q_j, x \in P, j \in F. \quad (16)$$

式(14)表示车辆 j 从起点出发时的原有载货量, 式(15)表示车辆到达终点时所有搭乘全部完毕, 只剩余原有载货量, $initq_j$ 为常量; 式(16)表示车辆在搭乘过程中总货品数量不能超过车辆搭载容限.

2 两阶段聚类启发式算法

早期对路径优化问题的研究主要集中在精确算法, 如动态规划法、集分割法以及列生成法等. 这些方法追求精确解, 但计算量大, 仅适合小规模问题, 且其数学模型仅适合具体问题, 通用性较差. 对实际问题而言, 人们更关注的是能否在较短时间内找到满意解. 本文提出基于“两阶段聚类”的启发式算法 (two-stage clustering algorithms, TSCLA), 用于解决 MCP 问题. “两阶段聚类”的含义为: 第 1 阶段, 以 m 辆车辆为中心, 将服务需求集合聚类为 m 个子集, 这可将多车辆问题转化为单车辆问题; 第 2 阶段, 对每个服务需求子集采用先验聚类的方法再次聚类, 得到一个优先关系序列. 以上工作完成后, 在每个优先关系序列中利用启发式插入算法寻找单车辆问题最优解, 然后综合考虑所有车辆, 找到多车辆

问题的本次最优方案. 由于第 1 阶段聚类时可能会出现某服务需求归属多样性问题, 算法采取了迭代学习策略, 以提高某些特殊服务需求的归属问题, 从而提高搭乘效率, 减少算法陷于局部最优的机会.

2.1 几个定义

定义 1. 松弛时间 RTv . 服务需求 x 完成从上车点 x^{\uparrow} 到下车点 x^{\downarrow} 所用时间 $T_{x^{\uparrow}, x^{\downarrow}}^j$, 同 x^{\uparrow} 的时间窗口开始时刻到 x^{\downarrow} 时间窗口结束时刻之间的时间差值, 有 $RTv_x^j = (l_x^{\downarrow} - e_x^{\uparrow}) - T_{x^{\uparrow}, x^{\downarrow}}^j, j$ 为车辆编号.

定义 2. 松弛路程 RSv . 指服务需求 x 在松弛时间内可行驶的路程, 有 $RSv_x^j = RTv_x^j \times v_j$. 松弛时间或松弛路程越大搭载成功的可能性也越大.

定义 3. 线路拟合度 H . 指服务需求 x 的上车、下车点所确定的直线段同车辆 j 中同服务需求 x 邻近的初始行驶路径之间的线路拟合情况.

图 1 中有向线段 SD 由服务需求 x 的上车点 S 和下车点 D 确定, 有向曲线 L 为车辆 j 的初始行驶路径, 中间节点为车辆途经点. 从曲线 L 的各途经点集合中分别找出同点 S, D 距离最近的点 S', D' , 定义点 S', D' 确定的直线为 L' ; 设 SD 同 L' 的夹角为 α , 计算服务需求 x 同车辆 j 的拟合度:

$$H_x^j = \frac{1}{2} \left(e^{-\frac{1}{2C}(|S'-S|+|D'-D|)} + \frac{1}{2}(1 + \cos \alpha) \right),$$

(17)

其中 $|S'-S|, |D'-D|$ 为两点间距离; $C>0$ 为常量, 可取所研究凸平面空间的半径. $H_x^j \rightarrow 1$ 说明拟合度越大匹配成功的可能性也越大, 反之则越小.

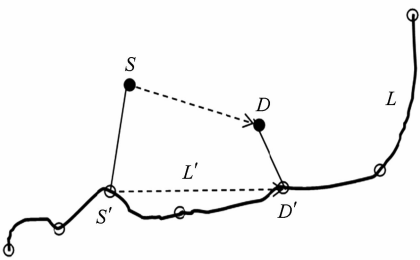


Fig. 1 Fitting between service requirement route and vehicle route.

图 1 服务需求路线同车辆行驶路径的拟合

定义 4. 匹配度 MP . 服务需求 x 同车辆 j 能成功匹配的可能性用 MP_x^j 表示. 匹配度是一个综合性指标, 同松弛路程、线路拟合情况、车辆初始剩余容量、服务需求容量等都有关系, 路程越松弛线路拟合度越高, 车辆剩余容量越大则匹配度越大; 而搭乘服务的需求容量越小匹配度越大. 有:

$$MP_s^j = \begin{cases} 0, (l_x^{\uparrow} < e_j^{\uparrow}) \text{ or } (e_x^{\downarrow} > l_j^{\downarrow}), \\ \beta_1 \times \frac{RSV_x^j}{\sum_{j \in F} RSV_x^j} + \beta_2 \times \frac{Q_j - initq_j}{\sum_{j \in F} (Q_j - initq_j)} + \\ \beta_3 \times \frac{H_x^j}{\sum_{x \in P} H_x^j} + \beta_4 \times \left[1 - \frac{\Delta q_x}{\sum_{x \in P} \Delta q_x} \right], \text{ else,} \end{cases}$$

(18)

其中, $\beta_1, \beta_2, \beta_3, \beta_4 \in [0, 1], \beta_1 + \beta_2 + \beta_3 + \beta_4 = 1$. 本文设 $\beta_k = 0.25, k = 1, 2, 3, 4$.

2.2 一次聚类: 基于匹配度的启发式聚类算法

基于式(18)可得到所有服务需求对所有车辆的 m 行 n 列匹配度矩阵 MP . 定义 MP 的行编号为车辆编号 vid , 列编号为服务需求编号 pid . 匹配度矩阵刻画了特定车辆对特定服务需求的匹配能力, 但不能反映车辆或服务需求的总体匹配能力. 如能从统计意义上得到车辆及服务需求的平均匹配能力, 则显然有利于提高搭乘成功率. 对矩阵 MP 进行如下变换:

- 1) 求各行计算数学期望 $vExp$ 及标准差 $vStd$, 并按式(19)计算各车辆的匹配能力 vMA ;
- 2) 对各行按如下策略排列: 先以 vMA 为第 1 关键字降序排列; 如值相同则再以 $vExp$ 为第 2 关键字降序排列; 如前两个关键字都相同, 则再以 $vstd$ 为第 3 关键字升序排列;
- 3) 求各列数学期望 $pExp$ 及标准差 $pStd$, 并按式(20)计算各服务需求的匹配能力 pMA ;
- 4) 对列各进行重排, 策略同步骤 2), 由此得到矩阵 MP' .

相比 MP , 矩阵 MP' 右侧增加了 3 列: 数学期望 $vExp$ 、标准差 $vStd$ 及车辆匹配能力 vMA , 其中 vMA 按式(19)计算:

$$vMA^j = \eta vExp^j - (1 - \eta)vStd^j, j \in F, (19)$$

$vExp^j$ 为第 j 行向量 MP' 的数学期望, $vStd^j$ 为第 j 行向量的标准差. 同理可基于 $pExp, pStd$ 计算服务需求的匹配能力 pMA , 则矩阵 MP' 下侧又增加 3 行, 有:

$$pMA_x = \eta pExp_x - (1 - \eta)pStd_x, x \in P. (20)$$

$vExp$ 反映了车辆对所有搭乘需求的平均能力, 该值越大说明车辆对搭乘的可能贡献越大; $vStd$ 反映了车辆对搭乘需求能力的分布情况, 该值越小说明车辆提供给各服务需求的搭乘能力越相似; 而 vMA 综合了数学期望和标准差的影响. $\eta \in [0,$

Table 3 Changed Reachable Relationship

表 3 变换之后的可达关系示例

Position	1 [↑]	1 [↓]	2 [↑]	3 [↑]	2 [↓]	3 [↓]	4 [↑]	4 [↓]	Sum of Row
1 [↑]	1	1	1	1	1	1	1	1	8
1 [↓]	0	1	1	1	1	1	1	1	7
2 [↑]	0	1	1	1	1	1	1	1	7
3 [↑]	0	1	1	1	1	1	1	1	7
2 [↓]	0	0	0	0	1	1	1	1	4
3 [↓]	0	0	0	0	0	1	1	1	3
4 [↑]	0	0	0	0	0	1	1	1	3
4 [↓]	0	0	0	0	0	0	0	1	1

2.4 启发式聚类迁移规则

一次聚类完成后,服务需求被限制到一辆车上.为解决该问题,本文提出聚类迁移规则,用以扩展寻优范围.借鉴文献[13]中“长弧在高质量解中出现的概率较小”的结论,算法对搭乘失败或导致长弧的服务需求执行选择性“迁出-迁入”操作.

2.4.1 迁出启发式规则

对集合 P^j 中匹配失败的服务需求 x ,迁出概率为 1.对那些虽成功匹配但却导致长弧的服务需求做如下操作:定义弧长阈值 $\delta = \theta \times s^* / (m + n - 1)$,其中 $\theta \in [0.5, 1.5]$ 为调节因子, s^* 为本次迭代中生成的最优路径的路径总长度,即

$$s^* = \sum_{j \in F} \sum_{x \in V} \sum_{y \in V} X_{x,y}^j \times a_{x,y}, \forall x \in P^j,$$

设 $Lx_l^\uparrow, Lx_r^\uparrow$ 分别表示点 x^\uparrow 到其前续节点、后续节点的弧长度, $Lx_l^\downarrow, Lx_r^\downarrow$ 分别表示 x^\downarrow 到其前续节点、后续节点的弧长度.定义 $L_x = \max(Lx_l^\uparrow, Lx_r^\uparrow, Lx_l^\downarrow, Lx_r^\downarrow)$ 为 x 所连接的弧的最大长度.计算 P^j 中每个服务需求的弧最大长度的最大值,记该服务需求的编号为 mx^j ,不妨假设 $mx^j = x$.定义 $emP_x^{P^j}$ 为从集合 P^j 中将 x 迁出的概率,有:

$$emP_x^{P^j} = \begin{cases} 0, & L_x - \delta < 0, \\ \min\left(\frac{L_x}{\delta} - 1, 1\right), & L_x - \delta \geq 0. \end{cases} \quad (22)$$

2.4.2 迁入启发式规则

确定 x 的迁入目标子集需考虑以下因素:1) 服务需求同车辆的匹配度;2) 车辆搭乘后的时间松弛程度.定义 $imP_x^{P^k}$ 为将元素 x 迁入子集 P^k 的概率,有:

$$imP_x^{P^k} = \gamma \times \frac{MP_x^k}{m \times pExp_x - MP_x^x} + (1 - \gamma) \times \frac{f_{ts}(P^k)}{\sum_{l \in F \setminus \{j\}} f_{ts}(P^l)}, k \in F \setminus \{j\}, \quad (23)$$

其中:

$$f_{ts}(P^k) = \sum_{i=1}^{2r} (l_{w_i} - B_{w_i}^k). \quad (24)$$

对式(24),为方便描述,定义车辆 k 对应子集 $P^k = \{k, x_{k,1}, x_{k,2}, \dots, x_{k,r-1}\}$, $|P^k| = r$,子集 P^k 中各元素构成的上、下车点集合为 $VP^k = \{vx_1, vx_x, \dots, vx_{2r}\}$,则式(24)所描述函数 f_{ts} 即为车辆 k 在上一轮迭代完成后所表现出来的总的松弛时间,该值越大,说明子集 P^k 越容易接受新的服务需求.按照式(23)计算得到除去迁出子集 P^j 之外的所有其他子集的迁入概率,按照轮赌策略,即可确定唯一迁入子集. $\gamma \in (0, 1)$ 为迁入概率影响因子,其取值可根据情况适当调整.

2.5 计算复杂性分析

采用两阶段聚类算法解决 MCP 问题,可有效降低问题的计算复杂度,适合解决大规模问题.算法主要包括匹配度矩阵计算、一次聚类、聚类迁移、二次聚类等过程.在匹配度计算过程中,松弛时间及松弛路程的计算复杂度为 $O(m + n)$,搭乘距离的计算复杂度为 $O(m \times n)$,生成匹配度矩阵的计算复杂度为 $O(m \times n)$,则匹配度矩阵的计算复杂度为 $2 \times O(m \times n) + O(m + n)$.在一次聚类过程中,计算各行/列的混合指标的复杂度为 $2 \times O(m \times n) + O(m)$,对各行/列进行排序的复杂度为 $2 \times O(n \log n)$,一次聚类时的计算复杂度为 $O(m \times n)$,则一次聚类过程的计算复杂度为 $3 \times O(m \times n) + 2 \times O(n \log n) + O(m)$.在聚类迁移过程中,迁出概率的计算复杂度为 $O(n) + 2 \times O(m)$,迁入概率的计算复杂度为 $O(m^2) + O(n)$,确定迁入子集的计算复杂度为 $O(m^2)$,则总体聚类迁移过程的计算复杂度为 $2 \times (O(n) + O(m) + O(m^2))$.综合以上分析,整个 TSCA 算法的计算复杂度约为 $m \times O((m + n)^4) + 7 \times O(n^2) + 2 \times O(n \log n) + 9 \times O(n) \leq O(n^5)$,说明算法为多项式算法,具有可接受的计算复杂度.

3 数值实验及结果分析

3.1 实验数据

为验证 TSCA 算法的有效性 & 性能,本文以中国济南市环路以内的方形路网为研究对象进行实验,其中 $m = 10, n = 30, \eta, \gamma$ 取值分别为 0.3, 0.6 和 0.9.为验证聚类迁移效果,我们对是否增加迁移过程进行了对比分析.实验内容包括:1) 不同参数设置对同一问题的影响比较;2) 迁移过程对解的影响.实验环境: Intel® Celeron® CPU E3300, 2.50 GHz, 4 GB 内存, 操作系统为 Windows XP Professional. 车辆基本信息及服务需求基本信息如表 4、表 5 所示:

Table 4 Basic Information of Vehicles

表 4 车辆基本信息表

vid	Start/End Position & Time Window		Mean Velocity	Fixed Cost & Variable Cost	Maximum Capacity & Initial Loading
1	(59,81)	(1 139,676)	50	50	500
	(1,10)	(30,40)		5	100
2	(221,50)	(1 278,59)	50	60	600
	(5,14)	(30,40)		5	150
3	(446,639)	(1 531,441)	50	60	600
	(2,11)	(35,45)		5	150
4	(442,185)	(1 648,58)	60	70	700
	(15,25)	(40,50)		7	200
5	(481,271)	(1 456,204)	60	70	700
	(10,20)	(35,45)		7	200
6	(164,436)	(1 650,376)	60	80	800
	(8,18)	(40,50)		8	250
7	(452,363)	(1 630,220)	60	80	800
	(7,17)	(40,50)		8	250
8	(601,531)	(1 226,504)	70	90	900
	(2,12)	(15,25)		9	300
9	(448,41)	(835,720)	70	90	900
	(10,20)	(20,30)		9	300
10	(588,71)	(1 411,459)	70	100	1 000
	(14,24)	(35,40)		10	400

Table 5 Basic Information of Service Requirements

表 5 服务需求基本信息

pid	Start/End Position & Time Window		Capacity Required	pid	Start/End Position & Time Window		Capacity Required
1	(126,256)	(601,531)	150	16	(601,443)	(1 151,443)	120
	(5,10)	(20,25)			(15,20)	(30,35)	
2	(310,425)	(712,629)	120	17	(813,441)	(1 413,397)	140
	(10,15)	(25,30)			(20,25)	(35,40)	
3	(601,443)	(902,717)	80	18	(618,324)	(971,381)	100
	(15,29)	(30,35)			(10,15)	(20,25)	
4	(448,41)	(797,46)	90	19	(746,406)	(1 156,352)	902
	(10,15)	(18,25)			(15,20)	(25,30)	
5	(697,45)	(1 071,17)	100	20	(873,390)	(1 354,298)	80
	(10,20)	(25,30)			(20,25)	(30,35)	
6	(577,639)	(1 411,459)	160	21	(736,533)	(961,499)	110
	(5,10)	(30,35)			(1,5)	(10,15)	
7	(712,629)	(1 307,496)	130	22	(822,539)	(1 102,507)	150
	(5,10)	(25,30)			(5,10)	(10,25)	
8	(835,720)	(1 226,504)	70	23	(588,71)	(736,533)	110
	(10,15)	(25,30)			(10,15)	(25,30)	
9	(585,207)	(1 438,70)	90	24	(585,257)	(712,629)	140
	(15,20)	(35,40)			(15,20)	(25,30)	
10	(795,247)	(1 278,59)	80	25	(662,374)	(835,720)	120
	(20,25)	(30,35)			(15,20)	(25,30)	
11	(967,247)	(1 291,111)	110	26	(697,45)	(1 084,225)	110
	(25,30)	(35,40)			(15,20)	(25,30)	
12	(452,363)	(904,298)	130	27	(952,146)	(1 152,409)	90
	(10,15)	(25,30)			(20,25)	(30,35)	
13	(678,397)	(1 250,248)	90	28	(1 152,316)	(1 413,397)	100
	(15,20)	(30,35)			(30,35)	(35,40)	
14	(809,330)	(1 105,272)	100	29	(749,406)	(1 053,380)	150
	(20,25)	(30,35)			(45,50)	(60,66)	

3.2 实验过程及分析

我们称不考虑迁移策略的算法为 iTSCA 算法. 设定一次聚类的次数不超过 30 次, 且当总成本不再有明显降低, 或连续 3 次相邻总成本之差小于某一阈值时算法结束. 实验结果如表 6 所示. 可以发现, 当 $\eta=0.6, \gamma=0.6$ 且增加迁移策略时, 实验效果较好, 这说明:

- 1) 在一次聚类过程中, 计算车辆或服务需求的搭乘能力时, 数学期望的影响作用强于其标准差;
- 2) 在迁移过程中, 车辆当前的时间松弛度比其匹配度更重要, 因为匹配度反应的是其原有的匹配能力, 而当前的时间松弛度反应的是当前的匹配能力;
- 3) 迁移过程能提高算法的效能, 但从计算时间上略有增加.

Table 6 Experimental Results ($m=10, n=30$)
表 6 实验结果 ($m=10, n=30$)

Parameters		iTSCA				TSCA			
η	γ	Carpooling success rate in optimal solution/%	Total cost in optimal solution	Run time/s	Run times of the 1st clustering when optimal solution appears firstly/ total run times	Carpooling success rate in optimal solution/%	Total cost in optimal solution	Run time/s	Run times of the 1st clustering when optimal solution appears firstly/ total run times
0.3	0.3	86.21	32 014	1 205	21/24	93.10	31 613	1 273	18/21
0.3	0.6	89.66	31 972	1 147	19/23	93.10	32 104	1 189	17/20
0.3	0.9	86.21	31 470	1 106	19/22	89.66	31 912	1 159	18/19
0.6	0.3	93.10	30 309	1 011	18/20	96.55	30 061	1 180	17/20
0.6	0.6	93.10	30 319	1 008	18/20	96.55	29 773	1 028	15/17
0.6	0.9	89.66	31 024	941	17/19	93.10	31 019	1 113	17/19
0.9	0.3	86.21	32 235	1 156	20/23	89.66	32 009	1 209	17/20
0.9	0.6	86.21	31 547	1 140	19/23	93.10	31 600	1 199	18/20
0.9	0.9	82.76	29 976	1 112	20/22	89.66	32 004	1 260	18/21

当 $\eta=0.6, \gamma=0.6$ 时, 对 TSCA 算法的计算过程分别进行跟踪, 得到匹配度矩阵, 如表 7 所示. 可以发现, 29 号服务需求对所有车辆的匹配度均为 0, 说明没有任何车辆可以满足其服务要求. 经分析发现, 这是因为 29 号的时间窗口同所有车辆的时间约束冲突, 无法匹配成功. 这种情况下, 系统需将该服务需求排除, 进入匹配过程的服务需求变成 29 个. 基于匹配矩阵, 按照 TSCA 算法中的一次聚类规则, 得到车辆同服务需求的匹配关系:

1(1,2,3), 2(4,5), 3(6,7,8),
4(9,10,11), 5(12,13,14), 6(15,16,17),
7(18,19,20), 8(21,22), 9(23,24,25,30),
10(26,27,28)

以“9(23,24,25,30)”来说明以上匹配关系的具体含义: 9 号车匹配了 23, 24, 25, 30 号 4 个服务需求. 需要说明, 本例中出现的连续的服务需求聚集到一辆车上是进行了特殊处理, 以方便问题描述, 实际应用中的编号是随机的. 一次聚类完成后, TSCA 将进行单车体的先验聚类插入过程. 分析 9 号车辆发现, 由于 30 号需求的上车点和下车点分布范围同车辆的匹配度平均较小, 虽然在一次聚类中匹配到 9

号车, 但由于效果欠佳导致总体成本大幅上升, 因此在迁移过程中选择迁出; 而在迁入 8 号车后, 由于难以找到合适插入位置, 从而导致 30 号需求搭乘失败. 图 2 描述了 5, 6 号车初始路线及搭乘后的路线对比, 以及 8, 9 号两车的初始路线同 30 号服务需求之间的位置对比关系. 可以发现, 如果 30 号服务需求先搭乘 9 号车, 然后在合适位置换乘 8 号车, 则有可能在约束范围内成功搭乘, 但由于本算法尚不支持换乘, 导致 30 号搭乘失败.

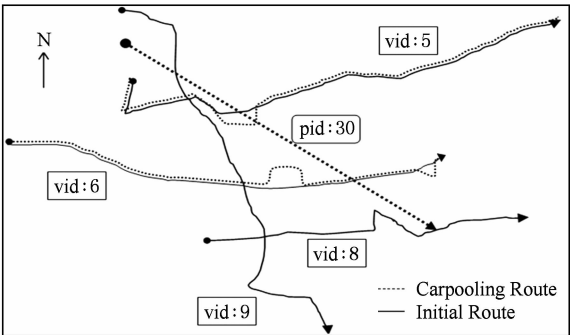


Fig. 2 Comparison between original routes and after carpooling.
图 2 车辆初始路径及搭乘后路线对比图

Table 7 Matching Matrix ($m=10,n=30$)

表 7 匹配矩阵($m=10,n=30$)

vid	pid														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.93	0.95	0.96	0.36	0.72	0.55	0.76	0.75	0.11	0.47	0.3	0.66	0.02	0.86	0.21
2	0.46	0.82	0.66	0.97	0.95	0.02	0.37	0.21	0.57	0.78	0.77	0.12	0.81	0.32	0.68
3	0.78	0.28	0.12	0.48	0.84	0.97	0.95	0.96	0.67	0.03	0.87	0.23	0.59	0.42	0.78
4	0.03	0.39	0.75	0.58	0.79	0.78	0.14	0.5	0.96	0.94	0.97	0.33	0.69	0.05	0.89
5	0.66	0.49	0.85	0.69	0.05	0.41	0.24	0.6	0.81	0.8	0.16	0.93	0.95	0.92	0.84
6	0.76	0.6	0.81	0.32	0.15	0.51	0.87	0.71	0.07	0.75	0.26	0.62	0.46	0.81	0.92
7	0.87	0.23	0.06	0.42	0.78	0.62	0.82	0.81	0.17	0.53	0.36	0.72	0.08	0.77	0.28
8	0.82	0.33	0.69	0.53	0.88	0.72	0.08	0.44	0.27	0.63	0.84	0.83	0.19	0.02	0.38
9	0.6	0.43	0.79	0.63	0.84	0.35	0.18	0.54	0.75	0.74	0.1	0.78	0.29	0.65	0.49
10	0.34	0.7	0.54	0.9	0.26	0.09	0.45	0.81	0.65	0.01	0.84	0.2	0.56	0.4	0.76

vid	pid														
	16	17	18	19	20	21	22	23	24	25	26	27	28	29*	30*
1	0.05	0.41	0.77	0.6	0.81	0.32	0.16	0.52	0.35	0.71	0.07	0.76	0.27	0	0.63
2	0.51	0.87	0.23	0.07	0.43	0.26	0.62	0.83	0.82	0.18	0.54	0.37	0.73	0	0.57
3	0.14	0.83	0.34	0.17	0.53	0.89	0.73	0.09	0.44	0.28	0.64	0.48	0.83	0	0.19
4	0.25	0.08	0.44	0.8	0.64	0.85	0.35	0.19	0.55	0.39	0.74	0.1	0.79	0	0.3
5	0.35	0.71	0.55	0.75	0.26	0.1	0.46	0.3	0.65	0.01	0.85	0.21	0.57	0	0.4
6	0.96	0.91	0.17	0.01	0.37	0.2	0.56	0.77	0.76	0.12	0.48	0.31	0.67	0	0.51
7	0.11	0.47	0.95	0.96	0.93	0.83	0.67	0.03	0.39	0.22	0.58	0.42	0.78	0	0.13
8	0.74	0.58	0.79	0.3	0.13	0.95	0.96	0.49	0.33	0.69	0.04	0.88	0.24	0	0.86
9	0.85	0.2	0.04	0.4	0.24	0.6	0.8	0.95	0.96	0.97	0.79	0.15	0.51	0	0.85
10	0.11	0.8	0.31	0.15	0.5	0.86	0.7	0.06	0.42	0.25	0.95	0.94	0.92	0	0.61

二次聚类及迁移规则可有效提高搭乘的匹配成功率,同时可降低总成本,如图 3 所示. 在本次匹配过

程中,除 30 号服务需求最终未能搭乘成功、29 号服务需求不满足搭乘约束外,其余所有需求全部搭乘

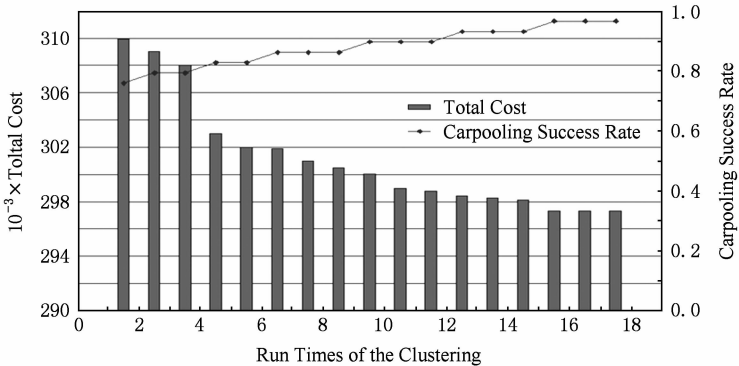


Fig. 3 Illustration of total cost and riding success rate.

图 3 总成本/搭乘成功率示意图

成功. 因此, 算法执行到最后, 系统已经到达当前最优, 总成本为 29 773, 搭乘成功率为 96. 55% (29 个人参与搭乘匹配, 最终 28 个人搭乘成功).

4 结 论

本文构建了多车辆合乘匹配问题的数学模型, 并提出解决此问题的两阶段聚类算法 TSCA. 通过对中国济南市的 10 辆车及 30 个服务需求的案例结果分析, TSCA 算法能够在较快时间给出优秀的搭乘方案, 整体搭乘成本及搭乘成功率较高. 本实验还进行了 30 辆车、100 个服务需求的实验, 实验结果类似, 但是运算时间较长. 本文研究的是确定性多车辆合乘匹配问题, 尚没有涉及车辆换乘问题, 也没有考虑路况变化, 即交通流的实时变化对搭乘匹配的影响, 这也是后期将要开展的研究方向.

参 考 文 献

[1] Ferguson E. The rise and fall of the american carpool: 1970-1990 [J]. *Transportation*, 1997, 24(4): 349-376

[2] Dailey D J, Loseff D, Meyers D. Seattle smart traveler: Dynamic ridematching on the world wide Web [J]. *Transportation Research Part C*, 1999, 7(1): 17-32

[3] Buliung R N, Soltys K, Bui R, et al. Catching a ride on the information super-highway: Toward an understanding of internet-based carpool formation and use [J]. *Transportation*, 2010, 37(6): 849-873

[4] Karaoglan I, Altiparmak F, Kara I, et al. The location-routing problem with simultaneous pickup and delivery: Formulations and a heuristic approach [J]. *Omega-International Journal of Management Science*, 2012, 40(4): 465-477

[5] Bianchessi N, Righini G. Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery [J]. *Computers & Operations Research*, 2007, 34(2): 578-594

[6] Cordeau J F. A branch-and-cut algorithm for the dial-a-ride problem [J]. *Operations Research*, 2006, 54(3): 573-586

[7] Cortes C E, Matamala M, Contardo C. The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method [J]. *European Journal of Operational Research*, 2010, 200(2): 711-724

[8] Parragh S N. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem [J]. *Transportation Research Part C-emerging Technologies*, 2011, 19(5): 912-930

[9] Håme L. An adaptive insertion algorithm for the single-vehicle dial-a-ride problem with narrow time windows [J]. *European Journal of Operational Research*, 2011, 209(1): 11-22

[10] Zhang Jin, He Ruichun. Solving dynamic taxipooling problem with simulated annealing algorithm [J]. *Journal of Lanzhou Jiaotong University*, 2008, 27(3): 85-88 (in Chinese)
(张瑾, 何瑞春. 解决动态出租车拼车问题的模拟退火算法 [J]. *兰州交通大学学报*, 2008, 27(3): 85-88)

[11] Zhai Yong, Yang Jinliang, Lian Jian, et al. Path matching algorithm for carpool information retrieval [J]. *Computer and Communications*, 2007, 25(134): 27-30 (in Chinese)
(翟泳, 杨金梁, 连剑, 等. 合乘出行信息检索的路径匹配算法 [J]. *交通与计算机*, 2007, 25(134): 27-30)

[12] Wang Wanliang, Huang Haipeng, Zhao Yanwei, et al. Dynamic customer demand VRP with soft time windows based on vehicle sharing [J]. *Computer Integrated Manufacturing Systems*, 2011, 17(5): 1056-1063 (in Chinese)
(王万良, 黄海鹏, 赵燕伟, 等. 基于车辆共享的软时间窗动态需求车辆路径问题 [J]. *计算机集成制造系统*, 2011, 17(5): 1056-1063)

[13] Toth P, Vigo D. The granular tabu search and its application to the vehicle routing problem [J]. *INFORMS Journal on Computing*, 2003, 15(4): 333-346



Shao Zengzhen, born in 1976. Associated professor. Member of China Computer Federation. His current research interests include logistics optimization and intelligent algorithms.



Wang Hongguo, born in 1966. Professor of Shandong Normal University. His main research interests include E-government and optimization theory (wang666666@126.com).



Liu Hong, born in 1955. Received her PhD degree in computer science from the Chinese Academy of Sciences, Beijing, China in 1998. Professor of Shandong Normal University. Senior member of China Computer Federation. Her main research interests include artificial intelligence, CSCW, and innovative design (hongliu@sdsu.edu.cn).



Song Chaochao, born in 1988. Master candidate. Student member of China Computer Federation. His main research interests include logistics optimization and intelligent algorithms(415846946@qq.com).



Meng Chunhua, born in 1989. Master candidate. Student member of China Computer Federation. Her main research interests include logistics optimization and intelligent algorithms(492041950@qq.com).



Yu Hongling, born in 1988. Master candidate. Her main research interests include logistics optimization(649753825@qq.com).

研发动态

美制造出世界首台碳纳米管计算机

美国斯坦福大学的工程师在新一代电子设备领域取得突破性进展,首次采用碳纳米管建造出计算机原型,比现在基于硅芯片模式的计算机更小、更快且更节能.晶体管不断缩小,以使一个芯片上可以排列更多.但随着晶体管变得越来越细小,在狭窄的空间里就会浪费更多电力、产生更多热量.而研究表明,碳纳米管能非常有效地指挥和控制电力.碳纳米管一直被认为是硅晶体管潜在的继任者.这项研究将引导研究人员去探索如何在未来10年开发出更小、更高效节能的处理器.他们利用这种技术组装了具有178个晶体管的基础计算机.该碳纳米管计算机可执行一些如计数和数字排序等任务,运行一个基本的操作系统.虽然碳纳米管计算机可能还需要数年时间才趋于成熟,但这一突破已经凸显未来碳纳米管半导体以产业规模生产的可能性(摘自:<http://www.stdaily.com/>,2013-09-27,科技日报).

英科学家开发出超触觉技术系统

最近,英国布里斯托尔大学“互动与绘图”(BIG)项目组研究人员开发出一种系统,能让用户在一个交互界面上体验到多部位的触觉感受(多点触觉反馈),而无需碰触或拿着任何设备.研究人员称,这种多点触控表面能让人们在公共场合轻松互动,人们不仅能感觉到显示屏上的内容,而且能在触摸前接受到看不见的信息.利用超声辐射力向用户发射触觉感受还是一门新技术.该研究提出了一种超触觉理论,即一种能在空中产生触觉反馈的系统设计.在空中传播的超声波会产生不同的压力,如果许多超声波同时到达同一位置,就会在该点产生明显的压力变化.超声转换器阵列能对空中目标施加压力,通过屏幕投射出触觉感受,直接作用于用户手上.一系列超声转换器发出频率很高的超声波,当所有声波时间在同一位置会合,就在人体皮肤上造成了感觉.研究人员还探索了可能让超触觉大展身手的3个新领域:空中手势指令、触觉信息层和可视化限制显示器,并为每个领域编写了应用程序(摘自:<http://www.stdaily.com/>,2013-10-10,科技日报).

深度学习新算法完成字里行间的情绪识别

近日,斯坦福大学的研究生 Richard Socher 和 Andrew Ng(Google 深度学习项目工程师之一),以及一位语言学及人工智能领域的专家 Chris Manning,共同研究开发了一种深度学习的新算法,这种算法被称为 Neural Analysis of Sentiment,缩写为 NaSent. NaSent 算法从人脑中得到灵感,旨在改善当前书面语言的分析方法.深度学习领域是计算机科学一个新兴领域,通俗说来就是构建像人脑一样处理数据的计算机程序.深度学习首先发起于学术界,目前各大互联网巨头也纷纷投入研究,如 Google 的猫脸识别以及 Facebook 的深度学习团队(摘自:<http://www.csdn.net/>,2013-10-12,CSDN).

“电喷”打印打出三维超精细结构

美国伊利诺斯大学厄本纳-香槟分校、芝加哥大学、韩国汉阳大学等的科研小组开发出一种制作纳米结构的新方法,将“从上到下”的喷墨打印和“从下到上”的自组装技术结合在一起,以一种“电喷”打印方式自动形成三维的超精细结构,所造出的纳米材料可用在半导体和磁存储工业中.用聚合物、DNA、蛋白质及其他“软”材料制造纳米结构,在新型电子工业、诊断设备和化学传感器中有很大学术潜力.但这些材料大部分都与光刻平印类的技术不相容,平印是集成电路的传统工艺.研究小组利用一种自组材料——嵌段共聚物改进了传统光照平印工艺,使其也能用于“软”材料领域.这种局部沉积嵌段共聚物的喷墨打印技术称为电流动态打印或电喷打印,操作起来更像办公室用的喷墨打印机(摘自:<http://www.stdaily.com/>,2013-09-30,科技日报).