一种求解多车辆合乘匹配问题的适应性算法

宋超超[1] 王洪国[2] 邵增珍[2] 杨福萍[2]

(山东师范大学管理科学与工程学院 济南 250014)[1] (山东师范大学信息科学与工程学院 济南 250014)[2]

摘 要 车辆合乘匹配问题是研究如何通过优化车辆路线及车辆-乘客匹配来搭乘尽量多的乘客的问题。目前国内外的研究多存在模型单一、脱离实际、算法效率不高等问题。针对该问题,提出一种基于吸引粒子群算法的问题求解方法。通过吸引粒子群算法进行多车辆问题向单车辆问题的转化,形成车辆同乘客之间的初次匹配。根据初次匹配结果利用先验聚类的思想将初次匹配结果进行排序,寻找较优需求序列排序方式。最后,通过相应的匹配再优化策略将需求序列进行再优化。对比实验表明,基于吸引粒子群算法的问题求解方式能以较高的搭乘成功率以及较低的花费完成车辆合乘匹配问题。

关键词 吸引粒子群,车辆合乘,先验聚类,需求序列

中图法分类号 TP18 **文献标识码** A

Adaptive Algorithm for MRMP

SONG Chao-chao[1] WANG Hong-guo[2] SHAO Zeng-zhen[2] YANG Fu-ping[2]

(Institute of Management Science and Engineering, Shandong Normal University, Shandong 250014, China)[1]

(Institute of Information Science and Engineering, Shandong Normal University, Shandong 250014, China)[2]

Abstract Multi-vehicle ride matching problem(MRMP) studies the problem of taking passengers as much as possible through optimizing vehicles' route and matching between vehicles and passengers. But at present, there are some problems in the researches such as models divorce from reality and low efficiency of algorithms. For this problem, this paper presented APSO(Attractive Particle Swarm Optimization) to solve this problem. The MRMP is transfered to RMP through APSO to form the first matching result between vehicles and passengers. And the best sort order is looked through sorting the result, making use of priori clustering based on the result from first matching. And at last, we optimized the solution one times more through optimized rules. Contrast experiment shows that the method based on APSO can solve the problem on a high rate of matching and a low cost.

Keywords APSO, MRMP, Prior clustering, Demand sequence

1 引言

随着城市车辆数量急剧增加,城市交通拥堵状况日益严重,车辆合乘匹配问题也日益引起各行业尤其是交通运输行业专家的关注。车辆合乘匹配问题是如何通过优化车辆行驶路径在一定约束条件下搭乘尽可能多乘客的问题。有效地解决该问题可以降低车辆空载率,提高车辆运行效率,进而可以降低运行成本,缓解环境污染状况,实现城市交通的协调顺畅运行。因此,国内外学者纷纷提出相应的解决方案。

国外的研究主要有,芬兰学者 Lauri Häme 等[1]提出一种自适应式插入算法来解决带窄时间窗口的单车辆合乘匹配问题,即通过先验聚类的思想对乘客的上下车顺序进行排序;Jean-François Cordeau · Gilbert Laporte 等[2]提出了车辆合乘问题的数学模型及相应的算法;国内的研究主要有:李相勇等[3]研究车辆路径问题的模型及算法,主要介绍了车辆路径问题数学模型的建立过程及相应的算法;段凤华等[4]在其博士毕业论文中针对带软时间窗约束的车辆路径问题进行了研究,通过相应的惩罚函数实现软时间窗的惩罚策略。但是,目前针对车辆合乘问题的研究主要是针对车辆本身的路径问题或者单车辆问题展开,多车辆合乘匹配问题研究成果较少。

本文主要对多车辆合乘匹配问题(multi-vehicle ride matching problem -MRMP)进行研究,提出了多车辆问题数学模型,克服了模型适用性不足的缺点。对该问题,本文分别从提高车辆-乘客搭乘率以及降低总体运行花费两个角度进行考虑。利用阶段化处理思想,首先,通过吸引粒子群算法进行车辆同需求之间的初次匹配,得到初次匹配结果。然后,通过先验聚类思想进行需求序列的优化排序,得到较优的排序结果。最后,通过相应的匹配再优化策略对得到的结果进行

到稿日期:2012-04-18 返修日期:2012-07-23 本文受山东省自然科学基金项目(ZR2011FQ029,ZR2011FL026),山东省科技发展计划项目(2011YD01099,2011YD01100),山东省高等学校科技计划项目(J11LG32)资助。

宋超超(1988—),男,硕士生,CCF 会员,主要研究方向为计算智能、物流优化等,E-mail:song881025@163.com;**王洪国**(1966—),男,教授,博士生导师,主要研究方向为人工智能、电子政务等;**邵增珍**(1976—),男,博士生,副教授,CCF 会员,主要研究方向为人工智能、人工社会等;**杨福萍**(1986—),女,硕士生,CCF 会员,主要研究方向为数据挖掘。

再优化，从而得到车辆合乘匹配问题的较优解。实验结果表明，本文提出的算法在一定程度上能大幅降低车辆初次匹配乘客时的时间，实现总体搭乘率高、车辆总体运行成本低的目标。

2 车辆合乘问题数学模型

2.1 问题描述

某区域内某车队有多辆车，车辆有固定时间窗口及起止点，容量、速度等指标各不相同。在有多个具有固定上下车站点及时间窗口的乘客的情况下，要求他们在其上下车时间窗口范围之内进行搭乘，最后通过车队总体搭载成功率最高以及总花费最少来作为目标。

2.2 问题形式化

假设车辆数量为 m，乘客的数量为 n。设 F 为车辆集合，$F=\{1,2,\cdots,m\}$，P 为乘客集合，$P=\{m+1,m+2,\cdots,m+n\}$。

2.2.1 符号定义

1) 对任意 $j\in F$，令 d_j^+ 表示车辆 j 的起点编号，d_j^- 表示车辆 j 的终点编号；对任意 $j\in P$，令 p_i^+ 表示乘客 i 的上车点编号，p_i^- 表示乘客 i 的下车点编号。定义集合 $F^+=\{d_j^+\mid j\in F\}$，$F^-=\{d_j^-\mid j\in F\}$，$P^+=\{p_i^+\mid i\in P\}$，$P^-=\{p_i^-\mid i\in P\}$ 分别表示车辆起点编号集合、车辆终点编号集合、乘客上车点编号集合和乘客下车点编号集合。为描述及计算方便，定义

$F^+=\{1,2,\cdots,m\}\quad P^+=\{m+1,m+2,\cdots,m+n\}$

$F^-=\{(m+n)+1,(m+n)+2,\cdots,(m+n)+m\}$

$P^-=\{(m+n)+(m+1),(m+n)+(m+2),\cdots,(m+n)+(m+n)\}$

可将司机看成特殊乘客，记所有上、下车点编号集合为

$N^+=F^+\cup P^+=\{1,2,\cdots,m,m+1,\cdots,m+n\}$

$N^-=F^-\cup P^-=\{(m+n)+1,(m+n)+2,\cdots,(m+n)+m,(m+n)+m+1,\cdots,(m+n)+(m+n)\}$

令 $N=N^+\cup N^-$ 表示所有上、下车点编号集合，则有

$N=\{1,2,\cdots,m,m+1,m+2,\cdots,m+n,(m+n)+1,\cdots,(m+n)+m,(m+n)+m+1,\cdots,(m+n)+(m+n)\}$

对任意 $x\in N^+$，定义上车点编号函数 $Up(x)=x$，下车点编号函数 $Down(x)=m+n+x$。

2)定义所有乘客的上、下车点的服务时间窗口为 $[e_x,l_x]$，$x\in N$。其中 e_x 为最早到达时间(时间约束下界)，l_x 为最迟到达时间(时间约束上界)。

3)设研究区域内点的集合为 V，$V=N\cup V'$，其中 V' 表示研究领域内车辆可经过的除上下车点之外的其它道路节点的集合。

4)对任意 $j\in F$，定义车辆固定运行成本为 fc_j，单位里程内每增加一乘客增加成本为 ac_j；定义 Q_j 为车辆 j 的最大载客量。

2.2.2 变量定义

1) $X_{x,y}^j$：二进制变量，$x,y\in V$，$x\neq y$，$j\in F$。$X_{x,y}^j=1$，说明车辆 j 从顶点 x 行驶到顶点 y(有方向性)。

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

3) B_x^j：时刻变量，$x\in V$，$j\in F$，为车辆 j 离开顶点 x 的时刻。当 $x\in N$ 时，需满足服务时间窗口约束，有

$$B_x^j=\max(T_x^j,e_x),x\in N \tag{1}$$

4) $Cost_j$：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 $Status_0^j$。每到达一个顶点后离开，就转化为一种新的状态。记 $Status_u^j$ 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-1} 时的乘客数量，即车辆 j 在边 (x_{u-1},x_u) 上运行时的乘客数量；s_u^j 是顶点 x_u 到顶点 x_{u-1} 之间的欧氏距离，即 $s_u^j=a_{x_{u-1},x_u}$，$x_u,x_{u-1}\in V$，其中 $a_{x,y}$ 表示站点 x 到站点 y 的距离。在这一段路程中，车辆 j 的变动成本为 $ac_j\cdot s_u^j\cdot q_{x_{u-1}}^j$。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(经过的边的个数为 U_j)个状态，则该次搭乘行程中车辆 j 的总花费为

$$Cost_j=fc_j+\sum_{u=1}^{U_j}ac_j\cdot s_u^j\cdot q_{x_{u-1}}^j \tag{2}$$

2.2.3 目标函数定义

考虑搭乘成功率最高以及总花费最低。有

$$\max\sum_{j\in F}\sum_{x\in P^+}\sum_{y\in V\setminus\{x\}}X_{x,y}^j \tag{3}$$

$$\min\sum_{j\in F}Cost_j \tag{4}$$

2.2.4 约束条件说明

C1)时间窗口约束

车辆 j 必须在规定的时间窗口范围内到达上、下车点，否则乘客无法在规定时间内上、下车。有

$$e_x\leqslant T_x^j\leqslant l_x,x\in N,j\in F \tag{5}'$$

特殊情况下，可能会出现车辆 j 早于时刻 e_x 到达顶点 x，这也是合法的。此时司机可以在有限的时间范围内进行等待，则(5)′可以修正为

$$T_x^j\leqslant l_x,x\in N,j\in F \tag{5}$$

C2)一次访问约束

车辆 j 到某一个上、下车点，每一个乘客上下车点只能被服务一次。有

$$\sum_{j\in F}\sum_{y\in V\setminus\{x\}}X_{x,y}^j\leqslant 1,x\in N \tag{6}$$

$$\sum_{j\in F}\sum_{x\in V\setminus\{y\}}X_{x,y}^j\leqslant 1,y\in N \tag{7}$$

式(6)表明顶点 x 的出度最大为 1，即从该乘客点出发的车辆最多只有一辆；式(7)表明顶点 y 的最大入度为 1，即从任意其它点进入该乘客点的车辆最多只有一辆。

C3)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。

$$\sum_{y\in P^+}X_{d_j^+,y}^j=1,j\in F,d_j^+\in F^+ \tag{8}$$

$$\sum_{x\in P^-}X_{x,d_j^-}^j=1,j\in F,d_j^-\in F^- \tag{9}$$

式(8)保证车辆 j 一定从其出发点出发，式(9)保证车辆 j 一定能到达其终点。

C4) 成对约束

任何乘客的上车地点 x 和下车地点 $m+n+x$ 如被服务，则必须被同一辆汽车服务。有

$$\sum_{y_1\in V\setminus\{x\}}X_{x,y_1}^j-\sum_{y_2\in V\setminus\{m+n+x\}}X_{y_2,m+n+x}^j=0,x\in P^+,j\in F \tag{10}$$

C5) 访问顺序约束

任何乘客的上车时间加上乘车时间，应在该乘客的要求下车时间之前。有

$$T_x^j + RT_x \leqslant T_{n+n+x}^{j'}, x \in N^+, j, j' \in F \tag{11}$$

C6)车辆载客量约束

行驶过程中,任意时刻不能超过车辆最大载客量。有

$$q_x^j = initq_j, x \in F^+, j \in F \tag{12}$$

$$q_x^j = initq_j, x \in F^-, j \in F \tag{13}$$

$$q_x^j \leqslant Q_j, x \in P, j \in F \tag{14}$$

式(12)表示车辆 j 从起点出发时的原有载客量,式(13)表示车辆到达终点时所有搭乘全部下车完毕,只剩余原有载客量,$initq_x$ 为常量。

3 问题求解过程

对于车辆合乘匹配问题,本文的主要解决思路如下:首先进行初次匹配,通过吸引粒子群算法进行初次匹配结果的求解;然后通过先验聚类思想对初次匹配结果的需求序列进行优化排序;最后,通过匹配再优化策略进行匹配结果的优化处理,再次进行序列排序,得到最优解。解决过程如表 1 所列。

表 1 问题求解过程

```
问题求解过程框架
1. init( );//初始化参数;
2. s0* =GenerateInitSolution();//生成问题的初始解
3. 令 s** ⇐s0* ;// s** 是当前问题最优解
4. for i=1 to m do
5.     进行初次匹配过程;
6. end for
7. 需求序列优化排序;
8. repeat
9.    需求序列再优化;
10. 需求序列优化排序;
11. if f(s* )>f(s** ) then
12.    s** ⇐s* ;
13. end if
14. until 满足算法结束条件
15. return s**
```

其中,s^{**} 表示问题的最优解;m 表示车辆数目;s^* 表示某一次迭代过程得到的解;s_0^* 表示问题的初始解。

3.1 初次匹配

初次匹配过程是将多车辆合乘匹配问题向单车辆合乘匹配问题转化的中间桥梁,通过初次匹配过程将某一个范围内的乘客划分到某车辆上来,从而可以将研究角度转化成车辆内部乘客排序问题。

车辆合乘匹配问题为 NP-hard 问题[2],问题中的初次匹配过程也属于 NP-hard 问题,很难通过精确算法在多项式时间内来求解。本文通过吸引粒子群算法来解决合乘问题中的初次匹配过程。

3.1.1 吸引粒子群算法描述

粒子群算法(PSO)是由 Kennedy 和 Eberhart 等[5]于 1995 年通过对鸟群等智能群体的模拟实验得到的一种群智能算法。粒子群算法以其优越的性能在处理具体问题时有很好的表现。

粒子群算法是根据群体运行规律抽象出来的一种智能算法[6]。本文通过粒子群算法的基本思想,引入吸引的概念来改进基本粒子群算法,进而使得该算法更适合解决本文所研究问题中的初次匹配过程。速度更新公式如下:

$$v_{ik} = \omega v_{ik} + step \cdot [c_1 \cdot (p_{ik} - x_{ik}) + c_2 \cdot s(x_{ik})] \tag{15}$$

$$s(x) = A_1 \cdot (p_{jk} - x_{ik}) + A_2 \cdot (p_{ck} - x_{ik}) + A_3 + A_4 \cdot (p_{ek} - x_{ik}) \tag{16}$$

$$x_{ik}' = x_{ik} + v_{ik} \tag{17}$$

式中,ω 表示惯性因子;c_1,c_2 分别表示自身经验影响因子以及社会经验影响因子;p_{ik} 表示该粒子当前最优位置;p_{jk} 表示全局最优位置;p_{ck} 表示当前群体的中心位置;p_{ek} 表示被觅食个体的位置信息。

其中 $step$ 的定义如下:

$$step_i = \begin{cases} \dfrac{n \cdot attration_i}{\sum_{j=1}^{n} attration_j}, & d_{i0} = 0 \\ \dfrac{n \cdot attration_i}{\sum_{j=1}^{n} attration_j} \cdot \dfrac{d_i}{d_{i0}}, & d_{i0} \neq 0 \end{cases} \tag{18}$$

式中,$step_i$ 表示朝向吸引力方向的动态移动步长;d_i 表示个体向吸引力方向移动时动态变化的距离;d_{i0} 表示在最初感受到吸引力时的距离。

$$attraction = \alpha \cdot \frac{fq}{sum(fq)} + \beta \cdot \frac{sum(cd) - cd}{sum(cd)} \tag{19}$$

式中,$\alpha+\beta=1$,α、β 分别表示适应度值以及密度对吸引力的影响因子,fq 为当前个体的适应度值,cd 为该个体区域范围内的个体密度。

环境变量用来描述当前时期整体的适应度值情况,其更新公式如下:

$$Envir_k = \frac{(\sum_{i=1}^{|N|} fq_i(k))/|N|}{(\sum_{j=1}^{k}(\sum_{i=1}^{|N|} fq_i(j)/|N|))/k} \tag{20}$$

式中,k 表示经过迭代的次数;N 是粒子的集合;$fq_i(j)$ 表示粒子 i 在第 j 代的适应度。

基于以上内容,抽象 A_1、A_2、A_3、A_4 的取值,见表 2。

表 2 状态变量取值表

变量	状态			
	(1)	(2)	(3)	(4)
A_1	1	0	0	0
A_2	0	1	0	0
A_3	0	0	1	0
A_4	0	0	0	1

根据前述,吸引粒子群算法基本流程见表 3。

表 3 吸引粒子群算法基本流程

```
吸引粒子群算法-APSO 框架
1. init( );//初始化参数;
2. for i=1 to Num do
3.    GenerateInitSolution();//将初始位置作为个体当前最优位置
4.    Fitness();//计算适应度值
5. end for
6. NewGPosition();//更新全局最优位置
7. repeat Sn
8.    CheckEnvir();//计算环境
9.    for i=1 to Num do
10.       CheckAttraction();//计算吸引力以及步长
11.       UpdateVelocity();//更新速度
12.       NewSPosition();//更新最优位置
13.   end for
14.   NewGPosition();//更新全局最优位置
15. until 满足停止条件
16. return s**
```

其中,s^{**} 表示最优解,Num 表示群体规模,Sn 表示迭代次数。

3.1.2 初次匹配及求解过程描述

1)初次匹配描述

某范围内有多辆车,在这些不同质车辆初始路线周围分布着不同搭乘要求的乘客,如何在多辆车路径微可调的初始条件下以及乘客运输需求的制约下,将乘客与车辆进行匹配来保证总搭乘率最高或者总花费最小。

本文通过吸引粒子群算法根据车辆初始路径中各站点的多种因素[11]得到某车辆经过该站点时允许的调整范围,车辆可以在该范围内微调车辆路线,从而实现搭载乘客的目的。

初次匹配过程模型构建过程如下:

$$\text{目标函数:}f(x_i)=M_i/\sum_{j}^{num_i}Pnum_{i,j} \tag{21}$$

匹配数 M_i:表示车辆 i 匹配到的乘客数量;站点数 num_i:表示车辆 i 初始路径中的主要站点数量;上下车点数量 $Pnum_{i,j}$:表示车辆 i 的第 j 个站点在其半径范围内包含的乘客上下车点数量;匹配率 $f(x_i)$:表示车辆 i 的匹配率。

$$R_i=\sum_{j}^{num_i}R_{i,j} \tag{22}$$

式中,R_i 表示车辆 i 的总路线调整半径。

站点半径:根据车辆初始路径中的各主要站点,考虑交通状况、道路本身状况、周围环境状况、车辆在该点的调整能力来定义某车辆在该站点可调整的半径范围。

$$R_{i,j}=(\lambda_1\cdot\frac{Traff_j}{\sum_{j}^{k}Traff_j}+\lambda_2\cdot\frac{Road_j}{\sum_{j}^{k}Road_j}+\lambda_3\cdot\frac{Envir_j}{\sum_{j}^{k}Envir_j}+\lambda_4\cdot\frac{Vehicle_j}{\sum_{j}^{k}Vehicle_j})\cdot Ajust_i \tag{23}$$

$\lambda_1+\lambda_2+\lambda_3+\lambda_4=1$;$\lambda_{1-4}$ 分别表示某个影响因素的作用能力因子;$R_{i,j}$ 表示车辆 i 在 j 点的聚类半径;$Traff_j$ 表示 j 点的交通状况;$Road_j$ 表示 j 点本身的路网分布情况;$Envir_j$ 表示 j 点周围的环境状况;$Vehicle_j$ 表示 j 点的车辆路线调整能力。

$$Ajust_i=(loff_i-eup_i)\cdot V_i-d_i \tag{24}$$

式中,$Ajust_i$ 表示车辆 i 的可调路程,即在车辆可调时间范围内的可调整路径长度;$loff_i$ 表示车辆时间窗口中最晚到达时间;eup_i 表示车辆时间窗口中最早出发时间;V_i 表示车辆的速度;d_i 表示车辆 i 的初始路径中的总距离长度。

乘客状态信息表:[0,1,1,…,0],表示乘客当前的匹配状态,如果已匹配则置 1,否则置 0。

2)初次匹配过程粒子群算法

应用粒子群算法解决该问题的关键在于找到一种合适的表现方式也即粒子的编码方式[7],使之与问题对应起来。本文粒子群的编码方式如下:

车辆路径内容:[1,3,6,9,13,15,7],则此时的粒子长度即为 7,每一位表示该站点的可调整半径值,该值取 0~R_i 之间的值,如粒子编码为[100,235,150,200,340,200,300],其中该辆车的总可调整半径为 R_i=100+235+150+200+340+200+300=1525。速度编码为[30,−40,45,35,−60,20,−30],其中 $\sum_{j=0}^{num_j}V_j=0$,保证在优化过程中粒子所有位之和恒等于 R_i。

具体的求解过程如下:

步骤 1 初始化粒子群体

(1)将 $I_i\cdot Num$ 个粒子位置向量中的每一位根据半径计算式(23)来计算,$Num-I_i\cdot Num$ 个粒子在总半径之和约束下随机生成;速度向量在总和为 0 的约束下随机生成。

(2)根据式(21)来计算目标函数值,更新个体自身最优位置以及全局最优位置。

步骤 2 循环迭代,直到满足结束条件或者达到最大迭代次数限制

(3)对于每一个粒子,通过式(15)来计算该粒子速度向量的值,根据式(17)计算该粒子的位置向量的值。

(4)根据位置向量中的每一位值,判断在该半径范围内的乘客上下车点信息,计算成对率,更新乘客状态表。

(5)根据成对率比较更新每一粒子的自身历史最优值 p_{ik} 以及全局历史最优值 p_{jk}。

步骤 3 满足结束条件输出最优位置向量

其中 I_i 表示车辆 i 的路线调整惯性,取值范围[0,1],该值越大说明惯性越大,车辆的调整变化范围越小,越难调整线路;反之,调整范围越大。

结束以上过程就得到了车辆同乘客之间的初次匹配结果,见表 4。

表 4 初次匹配结果表

车辆编号	乘客集合
1	{1,2,3}
2	{4,5,30}
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}
10	{26,27,28}

此时就将多车辆问题转化成了单车辆问题,下一步通过需求序列优化排序进行单车辆内部乘客需求序列的排序,得到问题的较优解。

3.2 需求序列优化排序

需求序列优化排序实际上是寻找一个有序服务序列,使得聚集到该车辆上的乘客尽可能实现搭乘。如($x^{\uparrow}$,$y^{\uparrow}$,$z^{\uparrow}$,$z^{\downarrow}$,$x^{\downarrow}$,$y^{\downarrow}$)是一个包含 3 名乘客 x,y,z 上、下车点的服务序列。不考虑时间约束、车辆容量约束时,显然这种服务序列的数量巨大。传统插入算法的基本原则[8-10]是将新乘客的上、下车点插入已有服务序列中并保持序列可行,后期采用路径交换等原则生成多种新解,但效果并不理想。芬兰学者 Lauir Hame[1]提出了基于"先验聚类"(a prior clustering)的单车辆插入算法,其在解决 DARP 问题时作用明显。

假设 1−4 号乘客的上下车点集合为{$1^{\uparrow}$,$1^{\downarrow}$,$2^{\uparrow}$,$2^{\downarrow}$,$3^{\uparrow}$,$3^{\downarrow}$,$4^{\uparrow}$,$4^{\downarrow}$},根据各点优先关系,得{$1^{\uparrow}$}<{$1^{\uparrow}$,$2^{\uparrow}$,$3^{\uparrow}$}<{$2^{\downarrow}$}<{$3^{\downarrow}$,$4^{\uparrow}$}<{$4^{\downarrow}$}。显然,{$1^{\uparrow}$}、{$2^{\downarrow}$}、{$4^{\downarrow}$}集合中的元素位置已经确定,后面的插入操作中,只需对集合{$1^{\uparrow}$,$2^{\uparrow}$,$3^{\uparrow}$}和{$3^{\downarrow}$,$4^{\uparrow}$}中的元素进行内部排序即可。

3.3 匹配再优化策略

初次匹配完成后,乘客被限制到某一特定车辆上。从全

局考虑，将其匹配到另一车辆也许会有更好的效果。基于此，本文提出匹配再优化策略，以扩展寻优范围，提高解的质量。

3.3.1 迁出策略

1)对集合 P^j 中匹配失败的乘客 x，是一定要从 P^j 中迁出的，即其迁出概率为 1。

2)对匹配成功但可能导致长弧的乘客的操作，定义弧长阈值 $\delta=\theta\cdot S(s^*)/(m+n-1)$，其中 $\theta\in[0.5,1.5]$为调节因子，$S(s^*)$为本次迭代中生成的最优路径的路径总长度，即 $S(s^*)=\sum_{j\in F}\sum_{x\in V}\sum_{y\in V}X^j_{x,y}\cdot a_{x,y}$。$\forall x\in P^j$，设 $Lx_l^{\uparrow}$，$Lx_r^{\uparrow}$ 分别表示乘客 x 的上车点 $x^{\uparrow}$ 到其前续结点、后续结点的弧长度，$Lx_l^{\downarrow}$，$Lx_r^{\downarrow}$ 分别表示乘客 x 的下车点 $x^{\downarrow}$ 到其前续结点、后续结点的弧长度。定义 $L_x=\max(Lx_l^{\uparrow},Lx_r^{\uparrow},Lx_l^{\downarrow},Lx_r^{\downarrow})$为乘客 x 所连接的弧的最大长度。计算 P^j 中每个乘客的弧最大长度的最大值，记该乘客的编号为 mx^j，不妨假设 $mx^j=x$。定义 emP_x^{Pj} 为从集合 P^j 中将乘客 x 迁出的概率，有：

$$emP_x^{Pj}=\begin{cases}0, & L_x-\delta<0\\ \min(\dfrac{L_x}{\delta}-1,1) & L_x-\delta\geqslant 0\end{cases}\tag{25}$$

如果 emP_x^{Pj} 较大，说明 P^j 的元素 x 形成了长弧，将 x 迁出的可能也越大。实际应用中，可设置当 emP_x^{Pj} 超出某阈值 η 时，需将 x 迁出，如图 1 所示。

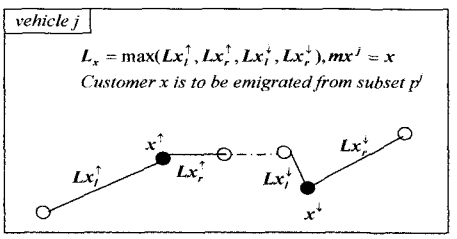

图 1 乘客 x 迁出子集 P^j 示例

3.3.2 迁入策略

确定 x 的迁入目标子集需考虑以下因素：车辆搭乘后的时间松弛程度。定义 imP_x^{Pk} 为将元素 x 迁入子集 P^k 的概率，有

$$imP_x^{Pk}=\frac{f_{ts}(P^k)}{\sum_{i\in F\backslash\{j\}}f_{ts}(P^i)},k\in F\backslash\{j\}\tag{26}$$

其中，

$$f_{ts}(P^k)=\sum_{i=1}^{2r}(l_{ux_i}-B^k_{ux_i})\tag{27}$$

对式(27)，为方便描述，定义车辆 k 对应子集 $P^k=\{k,x_{k,1},x_{k,2},\cdots,x_{k,r-1}\}$，$|P^k|=r$，子集 P^k 中各元素构成的上、下车点集合为 $VP^k=\{vx_1,vx_x,\cdots,vx_{2r}\}$，则式(27)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 越容易接受新的乘客。按照式(26)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。

3.4 算法复杂度分析

采用启发式策略及智能算法解决 MRMP 问题可以有效降低问题求解的计算复杂度，适合解决大规模问题。根据本文流程，求解方法的主要部分包括通过吸引粒子群算法求解初次匹配过程、需求序列优化排序过程、匹配再优化过程。其中，吸引粒子群算法复杂度为 $m*Num*Sn*Pdim\approx O(n^3)$；按照文献[1]中描述，需求序列优化排序过程的复杂度在 $O(n^4)\sim O(n^5)$之间；匹配再优化过程中迁出策略的复杂度为 $O(n)+2*O(m)$，迁入策略的计算复杂度为 $O(n)$。综合以上分析，整个求解过程的算法复杂度为 $O(n^3)+O(n^5)+O(n)+2*O(m)+O(n)\leqslant O(n^5)$，说明该求解过程为多项式过程，总体运算复杂度在可接受范围之内。

4 实验及结果分析

为了验证本文求解方法的有效性，进行了相应的实验，形成了一套完整的原型系统。另外，由于车辆同乘客的搭乘数据较难获得，因此我们通过一定的数据生成方法，形成同现实社会符合度较高的实验数据。

4.1 数据生成方法

二维平面内数据生成过程为：

1) 车辆速度生成：对 $\forall j\in F$，随机生成[40,70]范围内的车辆平均速度 v_j，同时可得所有车辆的平均速度 va。

2) 车辆、乘客上下车点信息生成：对 $\forall j\in F$，随机生成两个点 d_j^+ 和 d_j^-，满足 $|d_j^+-d_j^-|\geqslant 100\sqrt{2}$；对 $\forall i\in P$，随机生成两个点 p_i^+ 和 p_i^-，满足 $|P_i^+-P_i^-|\geqslant 10$。

3) 车辆、乘客上下车点时间窗口生成：对 $\forall x^{\uparrow}\in F^+\cup P^+$，根据 Poisson 分布原则生成上车点 $x^{\uparrow}$ 的最早到达时间 $e_{x^{\uparrow}}$；按照下式计算乘客 x 的上车点 $x^{\uparrow}$ 的最迟到达时间 $l_{x^{\uparrow}}=e_{x^{\uparrow}}+rand(5,15)$；按照下式计算乘客 x 的下车点 $x^{\downarrow}$ 的最早到达时间 $e_{x^{\downarrow}}=e_{x^{\uparrow}}+|x^{\uparrow}-x^{\downarrow}|/va$；按照下式计算乘客 x 的下车点 $x_{\downarrow}$ 的最迟到达时间 $l_{x^{\downarrow}}=e_{x^{\downarrow}}+rand(5,15)$。

方便起见，定义所研究的时间范围为(0,200)，以分钟计。以上数据一旦确定，在后面各实验中将不再改变，这样可对各种策略及参数选择进行有效对比。另外，本研究基于如下计算机平台进行试验：Intel(R) Celeron(R) CPU E3300，2.50GHz，双核，2G 内存，操作系统为 Windows XP Professional。

路网及站点数据信息：我们以济南市二环路以内作为研究对象，以该范围内的实际地图作为实验路网，通过真实的路径生成相应的站点及线路，根据以上数据生成方法生成车辆以及乘客的相应数据信息。车辆及乘客信息见表 5、表 6。

表 5 车辆信息

车辆编号	起点及时间窗	终点及时间窗	初始路径
1	1,[1,10]	63,[30,40]	1,2,3,10,8,13,12,36,37,38,39,40,63
2	4,[5,14]	74,[30,40]	4,5,18,19,28,29,49,72,73,74
3	9,[2,11]	84,[35,45]	9,11,37,38,39,40,63,64,80,82,84
4	6,[15,25]	93,[40,50]	6,17,16,21,26,31,51,71,75,92,93
5	7,[10,20]	77,[35,45]	7,14,15,33,22,32,47,48,52,76,70,96,77
6	3,[8,18]	86,[40,50]	3,10,8,13,35,43,44,55,89,62,68,65,79,81,83,85,86
7	14,[7,17]	95,[40,50]	14,15,33,23,34,46,45,54,53,66,67,69,97,78,94,88,95
8	12,[2,12]	64,[15,25]	12,36,42,56,57,59,60,61,64
9	5,[10.20]	39,[20,30]	5,18,17,16,15,33,23,35,36,37,38,39
10	18,[14,24]	82,[35,40]	18,19,28,29,30,50,51,52,102,91,67,90,79,81,82

表 6　乘客信息表

乘客编号	起点及时间窗	终点及时间窗
1	2,[5,10]	12,[20,25]
2	10,[10,15]	37,[25,30]
3	13,[15,29]	40,[30,35]
4	5,[10,15]	28,[18,25]
5	19,[15,20]	72,[25,30]
6	11,[5,10]	82,[30,35]
7	37,[5,10]	80,[25,30]
8	39,[10,15]	64,[25,30]
9	17,[15.20]	92,[35,40]
10	26,[20,25]	74,[30,35]
11	31,[25,30]	75,[35,40]
12	14,[10,15]	47,[25,30]
13	23,[15,20]	70,[30,35]
14	32,[20,25]	52,[30,35]
15	10,[10,15]	105,[20,25]
16	13,[15,20]	68,[30,35]
17	43,[20,25]	81,[35,40]
18	15,[10,15]	54,[20,25]
19	34,[15,20]	67,[25,30]
20	45,[20,25]	97,[30,35]
21	36,[1,5]	59,[10,15]
22	42,[5,10]	61,[10,25]
23	18,[10,15]	36,[20,25]
24	16,[15,20]	37,[25,30]
25	33,[15,20]	39,[25,30]
26	19,[15,20]	51,[25,30]
27	30,[20,25]	90,[30,35]
28	91,[30,35]	81,[35,40]
29	34,[45.50]	53,[60,66]
30	6,[20,26]	61,[39,55]

4.2　参数设置

求解站点半径范围 $R_{i,j}$ 时，λ_{1-4} 分别用来调节各种因素在半径范围计算中的影响，取值范围为[0,1]，本文将 4 种因素视为是等影响力的，因此设置参数均为 0.25。

在描述迁移时，根据迁出概率同迁出阈值的对比来确定是否迁出。如果该阈值设置过大，会使得迁移惯性较大，一些本该迁出的乘客没能迁出；如果该阈值设置过小，会使得原先的初次匹配结果有较大范围的变动，从而大范围地影响整体的匹配结果，效果欠佳。因此，本文取迁出阈值 η 为 0.6，既能满足迁移需要又不至于对初次匹配结果造成重大影响。

在计算弧长阈值时定义弧长影响因子 $\theta\in[0.5,1.5]$，用来影响弧长阈值的计算。如果 θ 取值过大则会使得弧长阈值在一定范围内较大，降低迁出概率；如果 θ 取值过小则会使得弧长阈值在一定范围内较小，增加了迁出概率。因此，本文取值为 1，使得对迁出概率计算的影响较为适中。

吸引粒子群算法中用到的惯性权重因子 $\omega\in[0.6,1]$，在粒子群算法中，如果该值取值较大，就会缩小粒子的发掘空间，但搜索过程会比较细密；反之，会扩大粒子的发掘空间，但搜索过程不太细密。基于此，我们取值 0.8。另外，对于学习因子 $c_1,c_2\in[0,4]$，其主要用来加强或减弱自身经验以及社会经验在粒子进化中的影响，本文均取值 2。给定 $m=10$ 辆车以及 $n=30$ 个乘客，根据问题规模设定粒子群规模[12] 为 $Num=50$，迭代次数上限为 200。

4.3　实验结果分析

最终车辆运行路线表见表 7，其中包含车辆编号、搭载乘客集合以及最终运行路线，运行路线结构含义为："站点编号，时刻，(乘客集合)"，站点之间通过"/"分开。如车辆 1，其搭载车辆集合为{1,2,3}，最终运行线路为 1,1,()→2,5,(1)→10,12,(1,2)→13,19,(1,2,3)→12,21,(2,3)→37,26,(3)→40,33,()→63,38,()。

表 7　车辆最终运行路线表

车辆编号	搭载乘客集合	最终运行路线(只包含车辆、乘客上下车点)
1	{1,2,3}	1,1,()/2,5,(1)/ 10,12,(1,2)/ 13,19,(1,2,3)/12,21,(2,3)/ 37,26,(3)/40,33,()/63,38,()
2	{4,5}	4,5,()/5,10,(4)/ 19,16,(4,5)/28,18,(5)/72,26,()/ 74,32,()
3	{6,7,8}	9,2,()/11,5,(6)/37,8,(6,7)/ 39,13,(6,7,8)/ 80,26,(6,8)/64,28,(6)/ 82,33,()/84,36,()
4	{9,10,11}	6,15,()/17,18,(9)/ 26,23,(9,10)/31,26,(9,10,11)/74,34,(9,11)/75,35,(9)/92,38,()/93,42,()
5	{12,13,14}	7,10,()/14,12,(12)/ 23,18,(12,13)/ 32,24,(12,13,14)/47,26,(13,14)/ 52,31,(13)/ 70,34,()/ 77,38,()
6	{15,16,17}	3,8,()/10,11,(15)/8,14,(15)/13,17,(15,16)/ 43,21,(15,16,17)/ 105,24,(16,17)/ 68,31,(17)/ 81,37,()/ 86,45,()
7	{18,19,20}	14,7,()/15,10,(18)/ 34,15,(18,19)/ 45,22,(18,19,20)/54,24,(19,20)/ 67,30,(20)/ 97,34,()/ 95,40,()
8	{21,22}	12,2,()/36,4,(21)/42,6,(21,22)/ 59,10,(22)/ 61,14,()/64,16,()
9	{23,24,25}	5,10,()/18,13,(23)/ 16,16,(23,24)/ 33,19,(23,24,25) /36,23,(24,25)/37,25,(25)/ 39,29,()/39,29,()
10	{26,27,28}	18,14,()/19,16,(26)/ 30,23,(26,27)/ 51,27,(27)/91,30,(27,28)/90,32,(28)/ 81,37,()/82,38,()

不失一般性，我们通过两辆车来进行对比分析。

由图 2、图 3 可知，由于搭载乘客，车辆的路线进行了微调，这符合本文伊始的假设。

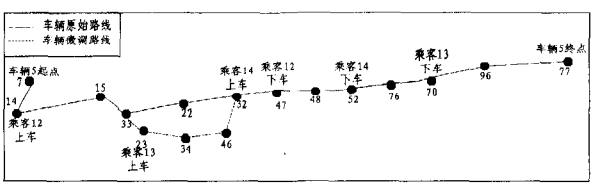

图 2　车辆 5 路径对比

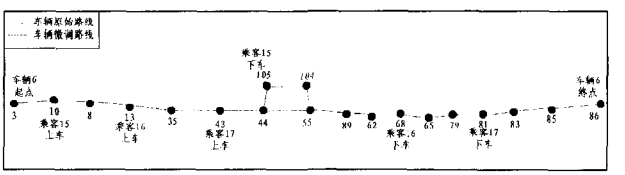

图 3　车辆 6 路径对比

车辆 5 的原初始路径为 7→14→15→33→22→32→47→48→52→76→96→77(见表 5)，如图 2 所示，由于车辆在站点 23 搭载了乘客 13，因此选择微调路线，车辆 5 的路线因微调搭乘乘客 13 变为 7→14→15→33→23→34→46→32→47→48→52→76→96→77。

车辆 6 的原初始路径为 3→10→8→13→35→43→44→55→89→62→68→65→79→81→83→85→86(见表 5)，如图 3 所示，由于车辆搭载了乘客 15，须将乘客 15 送往下车点 105，因此选择微调路线，车辆 6 的路线变为 3→10→8→13→35→43→44→105→104→55→89→62→68→65→79→81→83→85→86。

另外，29 号乘客由于下车点时间窗为[60,66](见表 6)，

其时间窗口不符合任何一辆车的时间窗口，因此将其排除；乘客 30 号搭载距离比较长，搭载需求从站点 6 至站点 61(见表 6)，同时需要跨越多辆车的运行区域，某一辆车如果搭乘该乘客会使得总的花费大幅度提高，导致该乘客不能搭乘。

图 4、图 5 是通过实验运行得到的 17 组解，通过运行迭代，得到最后的车辆-乘客路线总花费从最初的 30991 下降到 29773，搭乘率从一开始的 82.8%上升至 96.6%。

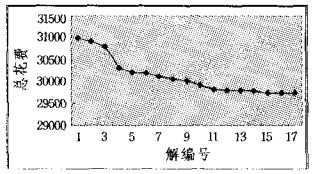

图 4　总花费

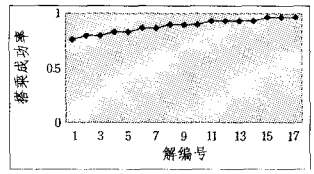

图 5　搭载成功率

由于乘客 29 的时间窗口不符合要求，因此本文在计算搭乘成功率时取底数为 30−1=29。

另外，乘客 30 的搭乘需求要跨越多个车辆运行区域，由于本文未涉及换乘概念，任何一辆车搭载之后都会大幅增加总花费，导致没有搭乘成功，最终只有 28 名乘客成功搭乘。最后 3 组解由于结果相同导致程序终止，经分析，符合实验要求，将最后 3 组解(3 组解相同)作为最优解输出。

结束语　本文通过详细的多车辆模型构建、相对贴近实际的实验及分析对目前关于车辆合乘匹配问题研究中存在的模型单一、适用性低等特点进行了改进，通过吸引粒子群算法结合先验聚类相关思想实现车辆-乘客之间的匹配、排序，同时通过匹配再优化策略对得到的结果进行再优化，避免了因为初次误匹配对后期实验结果产生重大影响。实验结果表明，该算法能以较高的效率和搭载成功率求解车辆合乘匹配问题。

由于针对车辆合乘问题的研究比较少，关于该问题的基础数据也比较少，因此在本文中基础数据生成过程需要较长时间，下一步将现实中的数据用于实验效果或许会更好。同时，根据实验结果，如果某乘客的搭载过程需要跨越多辆车的运行区域，则在考虑总花费的情况下会使得该需求长时间搭载不成功，此时可以考虑增加换乘概念，通过多辆车来接力搭乘该乘客。

参 考 文 献

[1] Lauri H. 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

[2] Cordeau J-F, Laporte G. The dial-a-ride problem: models and algorithms[J]. Ann Oper Res, 2007, 4(1): 29-46

[3] 李相勇. 车辆路径问题模型及算法研究[D]. 上海：上海交通大学，2007

[4] 段凤华. 带软时间窗约束的开放式车辆路径问题及其应用[D]. 长沙：中南大学，2009

[5] Kennedy J, Eberhart R. Particle swarm optimization[A]// Proc IEEE Int Conf on Neural Networks[C]. Perth, 1995: 1942-1948

[6] 谢晓锋，等. 微粒群算法综述[J]. 控制与决策，2003，18(2)：129-134

[7] 魏明，靳文舟. 求解车辆路径问题的离散粒子群算法[J]. 计算机科学，2010，37(4)：187-191

[8] 蒋忠中，汪定伟. 物流配送车辆路径优化的模糊规划模型与算法[J]. 系统仿真学报，2006，18(11)：3301-3306

[9] 刘云忠，宣慧玉. 车辆路径问题的模型及算法研究综述[J]. 管理工程学报，2005，19(1)：124-130

[10] 李琳，等. 改进的蚁群算法求解带时间窗的车辆路径问题[J]. 控制与决策，2010，25(9)：1379-1383

[11] 黄敏芳. 物流配送车辆路径方案的智能生成方法研究[D]. 大连：大连理工大学，2008

[12] 吴耀华，张念志. 带时间窗车辆路径问题的改进粒子群算法研究[J]. 计算机工程与应用，2010，46(15)：230-234

(上接第 194 页)

结束语　本文基于高阶逻辑定理证明器 HOL4，实现了 Gauge 积分的运算性质和相关定理的形式化，以反相积分器的形式化验证为例证明了本文提出的定理库的有效性。本文的积分定理库可以在 HOL4 中直接加载使用，它为使用 HOL4 对积分相关的系统进行形式化分析和验证奠定了基础。

参 考 文 献

[1] 韩俊刚，杜慧敏. 数字硬件的形式化验证[M]. 北京：北京大学出版社，2001

[2] Richter S. Formalizing integration theory, with an application to probabilistic algorithms [D]. Technische Universitat Munchen, Department of Informatics, Germany, 2003

[3] Fleuriot J D. On the mechanization of real analysis in isabelle/hol[C]// Theorem Proving in Higher Order Logics: 13th International Conference, TPHOLs 2000, Lecture Notes in Computer Science. volume 1869, 2000

[4] Cruz-Filipe L. Constructive Real Analysis: a Type-Theoretical Formalization and Applications [D]. University of Nijmegen, April 2004

[5] Butler R W. Formalization of the Integral Calculus in the PVS Theorem Prover [J]. Journal of Formalized Reasoning, 2009, 2(1): 1-26

[6] Harrison J. Theorem Proving with the Real Numbers [R]. Technical Report number 408. University of Cambridge Computer Laboratory, December 1996

[7] Henstock-Kurzweil integral [EB/OL]. http://en.wikipedia.org/wiki/Henstock%E2%80%93Kurzweil_integral

[8] Gordon R A. The Integrals of Lebesgue, Denjoy, Perron, and Henstock [M]. American Mathematical Society, 1994: 150-169

[9] Abdullah N, Akbarpour B, Tahar S. Error Analysis and Verification of an IEEE 802.11 OFDM Modem using Theorem Proving [J]. Electronic Notes in Theoretical Computer Science, Elsevier B. V. Pub., 2009, 242(2): 3-30

[10] Abdullah A N M. Formal Analysis and Verification of an OFDM Modem Design [D]. Department of Electrical and Computer Engineering, Concordia University, Montreal, Quebec, Canada, February 2006

[11] Akbarpour B. Modeling and Verification of DSP Designs in HOL [D]. Department of Electrical and Computer Engineering, Concordia University, Montreal, Quebec, Canada, April 2005

[12] 李黎明，关永，吴敏华，等. 运用定理证明的形式化方法验证 SpaceWire 编码电路[J]. 小型微型计算机系统，2011，30(6)：1372-1376