

# 一种求解多校多车型校车路径问题的元启发算法

侯彦娥<sup>1,2</sup> 孔云峰<sup>2</sup> 党兰学<sup>1</sup> 王玉璟<sup>1,2</sup>

(河南大学计算机与信息工程学院 开封 475004)<sup>1</sup>

(河南大学黄河中下游数字地理技术教育部重点实验室 开封 475004)<sup>2</sup>

**摘要** 针对多种车型可用的多校校车路径问题(SBRP),建立数学模型,并提出了一种迭代局部搜索(ILS)元启发算法进行求解。该算法引入并改进了带时间窗的装卸一体化问题(PDPTW)求解中的点对邻域算子,并使用可变邻域下降搜索(VND)完成局部提升。局部提升过程中,设计一种基于路径段的车型调整策略,尽可能地调整车型,降低成本,并允许接受一定偏差范围内的邻域解以保证搜索的多样性。对于局部提升得到的最好解,使用多点移动方法对其进行扰动,以避免算法过早陷入局部最优。在国际基准测试案例上分别测试多校混载和不混载模式下算法的性能,实验结果验证了设计算法的有效性。进一步使用提出的算法求解单车型多校 SBRP 问题,并与后启发算法、模拟退火算法和记录更新法等算法进行比较,实验结果表明该算法仍然能够获得较好的优化效果。

**关键词** 多车型校车路径问题,多校,迭代局部搜索,可变邻域下降,车型调整策略

**中图分类号** TP301.6 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.08.037

## Metaheuristic Algorithm for Solving Multi-school Heterogeneous School Bus Routing Problem

HOU Yan-e<sup>1,2</sup> KONG Yun-feng<sup>2</sup> DANG Lan-xue<sup>1</sup> WANG Yu-jing<sup>1,2</sup>

(College of Computer and Information Engineering, Henan University, Kaifeng 475004, China)<sup>1</sup>

(Key Laboratory of Geospatial Technology for the Middle and Lower Yellow River Regions, Ministry of Education, Henan University, Kaifeng 475004, China)<sup>2</sup>

**Abstract** This paper aimed to deal with the multi-school bus routing problem (SBRP) with different types of school buses. Based on the iterated local search (ILS), we proposed a metaheuristic algorithm, which employs the neighborhood operators originally designed for pickup and delivery vehicle routing problem with time windows (PDPTW) to improve the route solution. In the local search phase, the algorithm adopts the variable neighborhood descent search (VND) to search neighborhood solutions by the neighborhood operators. Guided by an adjustment strategy based on route segment, these operators adjust bus type as much as possible in order to reduce the cost. Our algorithm accepts some worse neighborhood solutions within the scope of cost deviation to keep the search diversification, and it uses the perturbation method with multiple points shift to avoid trapping in the local optima. The results of benchmark datasets show that the proposed algorithm is effective to solve both the mixed-load SBRP and single-load SBRP. In addition, for the multi-school homogeneous SBRP, our algorithm outperforms existing algorithms, such as post-improvement heuristic, simulation annealing and record-to-record travel algorithm.

**Keywords** Heterogeneous school bus routing problem, Multi-school, Iterated local search, Variable neighborhood descent, Bus type adjustment strategy

## 1 引言

校车路径问题(School Bus Routing Problem, SBRP)是在满足既定约束的条件下,合理地规划校车线路,将学生从乘车站点送到学校(或从学校送回乘车站点),并达到特定目标的组合优化问题<sup>[1-3]</sup>。SBRP属于车辆路径问题(Vehicle Routing Problem, VRP)的应用范畴<sup>[2]</sup>,是一类求解难度极高的NP-hard问题。根据服务学校的数量,SBRP分为单校SBRP

和多校SBRP两类。多校SBRP又分为不混载和混载两种。前者是按照学校开学时间依次访问学校,即服务完一个学校之后再服务另外一个学校,也称为校车调度问题;后者是允许同一时刻校车上搭乘不同学校的学生,也称为混载SBRP。SBRP自提出后,国内外学者便进行了大量的研究,其中以单校SBRP的研究居多<sup>[3]</sup>。在实际运营过程中,多个学校联合运营能够有效地降低成本,提高校车的利用率<sup>[4]</sup>。此外,受学区地理环境、道路等级、学生分布状况和规划目标等多种因素

到稿日期:2016-07-06 返修日期:2016-10-02 本文受国家自然科学基金(41401461),河南省教育厅自然科学重点项目(15A520009)资助。

侯彦娥(1980-),女,博士,讲师,主要研究方向为智能算法、空间优化, E-mail:houyane@henu.edu.cn;孔云峰(1967-),男,博士,教授,博士生导师,主要研究方向为空间分析与应用、GIS分析与设计;党兰学(1980-),男,博士,副教授,主要研究方向为智能算法、空间分析与优化, E-mail:danglx@foxmail.com(通信作者);王玉璟(1980-),女,博士生,主要研究方向为空间优化。

的影响,校车服务车队通常由一组装载容量、成本各异的车辆组成。因此,研究在多校联合运营模式下使用多种车型的校车路径问题更加符合实际应用的需求。

目前,已有一部分文献对多校 SBRP 进行了研究。Braca 等<sup>[4]</sup>首次提出求解混载 SBRP 问题的算法,采用一种基于位置的启发算法(RLBH)求解纽约市曼哈顿区 73 所学校的校车路径安排问题。党兰学<sup>[5]</sup>针对混载 SBRP 问题,以车辆数最少为优化目标,提出了一种记录更新法(RRT),其混合使用带时间窗的装卸一体化问题(PDPTW)中的点对算子进行寻优。Bogl 等<sup>[6]</sup>规划总运营时间最短的校车行驶路线,涵盖乘车站点选择、学生分配到站点、路径生成和调度等 4 个子问题,并在求解过程中允许学生到达学校之前在不同线路间进行中转。陈小潘等<sup>[7]</sup>研究了多个学校不混载 SBRP 问题,在给定单校路径段的基础上,使用模拟退火算法进行路径合并,以获得最少的路径数。以上研究集中在单一车型多校 SBRP 问题的路径规划。

也有一些学者研究多种车型可用的情况下多校 SBRP 的路径安排。Spada 等<sup>[8]</sup>在给定校车数量的前提下,以最小化学生的损失时间为优化目标,提出了一种先构造初始解而后局部提升的两阶段算法,在案例研究中考虑了多种类型的校车,但算法实现过程中并未对如何实现进行详细说明。文献<sup>[9]</sup>设计了一个多阶段构造启发式算法来求解农村地区的 SBRP 问题,算法包括学生分配到站点、车辆分配到学校、估算学生插入成本和交换学生等多个步骤进行解的构造和提升,在提升过程中考虑使用“两小换一大”的策略调整车型。文献<sup>[10]</sup>以总运营里程最小为优化目标,设计了一种基于位置的自适应启发算法(ALBH)来求解混载 SBRP。车辆安排时以大车优先,所有路径构造完成之后再根据车的实际载重调整车型。Park 等<sup>[11]</sup>以车辆数为优化目标,设计了一种“后启发”算法来求解混载 SBRP,求解过程中仅对不同车型的校车进行分配,并不考虑不同车型间的成本差异。Kim 等<sup>[12]</sup>研究了给定路径段和车型的多校不混载 SBRP,以车辆数最小为目标,使用启发式算法完成多校路径的生成。Chen 等<sup>[13]</sup>针对相同的问题,提出一种模拟退火算法进行求解,实验结果优于文献<sup>[12]</sup>的结果。文献<sup>[12-13]</sup>均是在单校路径已知的基础上使用不同的车型完成多校路径的安排,在单校路径生成时不关注不同车型成本的差异。

以上多校多车型 SBRP 的研究存在以下不足:1)现有文献中,优化目标大多关注车辆数和总路径长度,并未考虑不同车型成本间的差异。因不同类型的校车购置、维修成本不同,单纯以车辆数或总运营里程进行衡量并不能反映实际运营成本。2)求解算法以构造启发式或改进启发式算法为主,元启发求解策略未得到充分利用,较为有效的求解 NP-hard 难题的元启发算法的相关研究成果较少。

鉴于此,本文以多校多车型校车路径问题为研究对象,并考虑多种车型、多个学校、不同开学时间、学生最大乘车时间等多种问题特征,寻找混载和不混载运营模式下总成本最小的校车路径安排方案。首先建立多校多车型校车路径问题的数学模型,然后提出一种迭代局部搜索算法(ILS)进行求解。在 ILS 算法内引入 PDPTW 问题求解的邻域算子,并混合使用可变邻域下降(VND)算法进行寻优。在搜索过程中采用

车型调整、接受一定范围内的较差解等策略来提高算法的寻优能力。在国际基准案例上对混载和不混载两种运营模式下算法的性能进行测试,实验结果验证了本文算法的有效性。

## 2 问题描述及定义

假设某个学区内已有校车场站、若干所学校、若干个乘车站点、学生和道路网络等数据。某校车服务公司有一组容量和成本各异的校车车队停放在校车场站,每个乘车站点均有一定数量的乘车学生,这些学生可以由到达不同学校的学生组成。场站、学校、乘车站点等任意两个站点之间的运行距离和运行时间已知。基于以上假设,要求:校车从场站出发经过乘车站点并在学校开学时间内将学生送到学校,每个学生在乘车站点上车有最早时间和最晚时间的限制。若一辆校车能够为多个学校提供服务,需要在乘车站点停留一段时间以便学生能够上车,校车到达某个学校后要停留一段时间以便学生下车。目标是找到总成本最小的路径安排。约束条件为:所有校车必须从场站出发,将乘车站点的学生送到其对应的学校即完成任务配送;每个乘车站点必须被访问且允许仅访问一次;任意时刻校车上的学生人数之和不能超过校车的容量;任何学生在校车上的时间不能超过最大乘车时间的限制;校车必须在学校规定的时间窗内到达;每种类型的校车具有不同的容量和固定成本。

根据以上问题描述,定义建立问题模型所需要的符号和决策变量如下: $D$  表示场站, $P^+$  是所有乘车站点的集合, $P^-$  是所有学校的集合,站点和学校集合  $P = P^+ \cup P^-$ , 全部站点集合  $V = D \cup P^+ \cup P^-$ 。任意两个站点之间的距离和行驶时间分别为  $d_{ij}$  和  $t_{ij}$ 。每个站点都有校车最早到达时间  $e_i$  和最晚到达时间  $l_i$  的限制。对于集合  $P$  中的每一个站点, $q_i$  表示站点  $i$  的上车或下车的人数; $st_i$  为站点  $i$  的服务时间,与站点上车或下车的人数有关。对于任何一个乘车站点  $i$ ,均有对应的学校站点  $s(i)$ ;  $T$  为学生的最大乘车时间。 $M$  为校车类型集合, $M = \{1, 2, 3, \dots, m\}$ ;  $Q_k$  为  $k$  类型的校车的容量, $k = 1, 2, 3, \dots, m$ ;  $k$  类型的校车的固定成本和单位可变成本记为  $f_k$  和  $v_k$ ;  $x_{ijk}$  表示校车  $k$  访问站点  $i$  后是否直接访问站点  $j$ ,若是则  $x_{ijk} = 1$ ,否则  $x_{ijk} = 0$ ;  $y_{ik}$  表示校车  $k$  是否经过站点  $i$ ,若是则  $y_{ik} = 1$ ,否则  $y_{ik} = 0$ 。 $T_{ik}$  表示校车  $k$  到达站点  $i$  的时间。 $L_{ik}$  表示校车  $k$  访问站点  $i$  之后,校车车上的人数。

借助以上符号和决策变量的定义,在文献<sup>[5, 11]</sup>的基础上定义本文所要研究的问题的数学模型:

$$\text{Min } f(x) = \sum_{j \in P^+} \sum_{k \in M} f_k x_{0jk} + \sum_{i \in V} \sum_{j \in V} \sum_{k \in M} x_{ijk} d_{ij} v_k \quad (1)$$

$$\text{s. t. } \sum_{k \in M} y_{ik} = 1, \forall i \in P^+ \quad (2)$$

$$\sum_{j \in V} x_{ijk} - \sum_{j \in V} x_{jik} = 0, \forall i \in P, k \in M \quad (3)$$

$$\sum_{j \in V} x_{jik} - \sum_{j \in V} x_{j(i)k} = 0, \forall i \in P^+, s(i) \in P^-, k \in M \quad (4)$$

$$T_{ik} + st_i + t_{ik(i)} \leq T_{s(i)k}, \forall i \in P^+, s(i) \in P^-, k \in M \quad (5)$$

$$\sum_{j \in P} x_{jok} = \sum_{j \in P} x_{0jk} = 1, \forall k \in M \quad (6)$$

$$L_{ik} = 0, \forall k \in M, i = D \quad (7)$$

$$q_i \leq L_{ik} \leq Q_k, \forall i \in P^+, k \in M \quad (8)$$

$$0 \leq L_{s(i)k} \leq Q_k - q_i, \forall i \in P^+, s(i) \in P^-, k \in M \quad (9)$$

$$0 \leq L_{jk} \leq Q_k - \sum_{s(i)=j} x_{ijk} q_i, \forall i \in P^+, k \in M \quad (10)$$

$$T_{s(i)k} - T_{ik} \leq T, \forall i \in P^+, k \in M \quad (11)$$

$$e_i \leq T_{ik} \leq l_i, \forall i \in V, k \in M \quad (12)$$

$$x_{ijk} \in \{0, 1\}, \forall i, j \in P, k \in M \quad (13)$$

$$y_{ij} \in \{0, 1\}, \forall i, j \in P \quad (14)$$

函数(1)是问题的优化目标,即总成本最小。约束(2)限定所有的学生都必须有且仅有一辆校车提供服务。约束(3)保证一辆校车驶入站点*i*之后必须离开。约束(4)确保经过某乘车站点的校车必须要经过该站点对应的学校。约束(5)限制校车的访问顺序,即校车必须先访问乘车站点,然后才能访问其对应的学校站点。约束(6)限定所有的校车从场站出发,最后又回到场站。约束(7)表示校车从场站出发时车上没有学生。约束(8)表示任何时刻校车上的学生人数均不能超过当前使用车型的容量。约束(9)表示车辆经过学校站点时车上学生人数的变化。约束(10)表示经过连续站点时校车容量的变化。若站点*j*是学校,则*q<sub>j</sub>*为负值,即表示到该学校下车的学生人数之和。约束(11)表示任何学生从其乘车站点到学校的时间均不超过最大乘车时间*T*。约束(12)要求校车必须在指定的时间范围内到达站点。约束(13)和约束(14)限定决策变量*x<sub>ijk</sub>*和*y<sub>jk</sub>*的取值范围。

### 3 算法设计

#### 3.1 算法描述

多校 SBRP 中,乘车站点和学校站点构成一个集货和卸货的站点对,这与装卸一体化问题(PDP)类似。再者,每个乘车站点和学校站点有最早到达时间和最晚到达时间的限制。因此,多校 SBRP 也可认为是一种带时间窗的装卸一体化问题(PDPTW)<sup>[5,11]</sup>。鉴于多校 SBRP 与 PDPTW 的相似性,本文将多校多车型 SBRP 转换为一类特殊的多车型 PDPTW 问题,并采用迭代局部搜索算法(ILS)统一求解多校混载和不混载两种运营模式下的校车路径安排方案。

ILS 算法从初始解出发,经过局部搜索、车型调整和扰动等步骤,逐步进行寻优。算法的基本流程如图 1 所示。

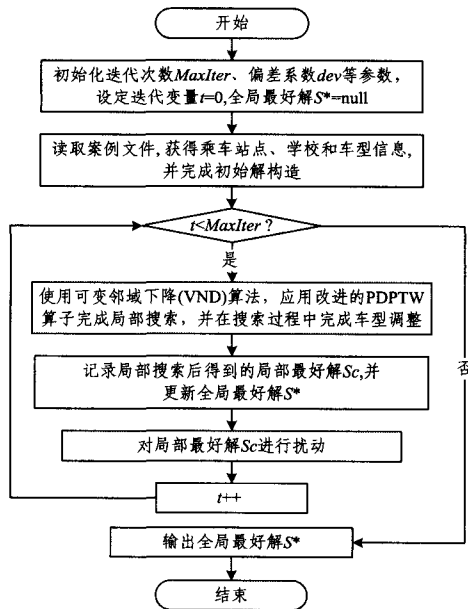


图 1 ILS 算法流程

由图 1 可知,ILS 算法包含以下 3 个部分:

(1)初始解的构造。根据乘车站点、学校和车型等信息,采用分段构造的方法(见 3.2 节)快速得到问题的初始解。

(2)解的改进。引入 PDPTW 问题求解应用的点对移动、交换和重定位邻域算子<sup>[14]</sup>(见 3.3 节),对其进行改进以使其能够适应多校多车型 SBRP 的求解。使用可变邻域下降(VND)过程完成局部搜索(见 3.4 节),并在搜索过程中进行车型调整(见 3.5 节),执行解的接受等算法策略。局部搜索完成后,使用扰动规则(见 3.6 节)对当前局部最好解完成扰动。

(3)更新全局最好解,并完成最好解的输出。

#### 3.2 初始解的构造

初始解的构造包含单个学校的路径构造和路径合并两个阶段。首先,针对每个学校,考虑场站、目标学校为该学校的乘车站点、可用车型信息和最大乘车时间等约束条件,使用广义插入法(GENI)构造若干条到达该学校的路径;然后,将针对每个学校构造的路径进行合并,即得到问题的初始解。构造初始解的关键在于针对单个学校构造若干条路径,具体构造过程如下:

(1)随机选择 3 个乘车站点,使用场站和学校构成一个封闭的环路。

(2)针对所有未插入到封闭环路上的站点,找到封闭环路内距离它最近的 *p* (*p* 是一个 2~7 之间的正整数)个站点。若环路内的所有站点的个数不足 *p*,则包含环路内所有的站点。

(3)随机选择一个不在环路中的站点,评估将该站点顺时针插入和逆时针插入的成本,寻找能够以最小成本插入的位置及插入方式。

(4)完成站点的插入,更新所有站点的 *p* 邻域。跳转到(3)继续执行,直到所有的站点均已经加入到环路中。

(5)根据得到的包含场站、所有乘车站点和学校的封闭环路,应用拆分过程划分为单个路径。

(6)从场站出发,顺序地访问环路中的站点,将站点加入到一条包含场站和学校的初始路径中。对于该路径,分配能够适应该站点的最小车型,不断地将站点加入到路径中,同时移除环路中对应的站点。若违反车型容量约束或最大乘车时间约束,则重新构造一条新的初始路径,跳转到(6)继续执行,直到环路中不存在站点为止。

这种构造初始解的方法尽管给出的是一组单校路径的集合,但能够降低初始解构造的难度。

#### 3.3 邻域操作算子

多校 SBRP 中移动的不是某个乘车站点,而是一个乘车站点及其对应学校站点组成的点对。因此,本文使用以下 3 种点对算子完成局部提升。

(1)单点对插入算子(Single Pair Insertion, SPI):将一条路径上的乘车站点和学校点对移动到另外一条路径上。图 2 给出了 SPI 的操作示意,对于路径 *r<sub>1</sub>* 上的学生站点 *P* 和其对应的学校站点 *S*,一次 SPI 操作即是将 *P* 放到路径 *r<sub>2</sub>* 上的站点 *u* 之后,同时将 *S* 放到站点 *v* 之后。

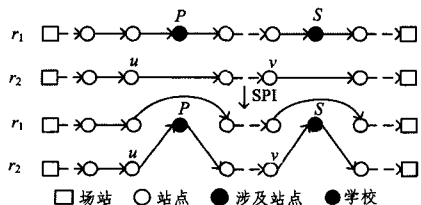


图 2 SPI 操作示意

(2)路径间点对交换(Swapping Pairs Between Routes, SBR):在两条路径之间完成两个站点和其学校组成的点对交换。SBR 的操作示意如图 3 所示。路径  $r_1$  上的站点对  $(P_1, S_1)$ 和路径  $r_2$  上的站点对  $(P_2, S_2)$  进行交换,即将站点对  $(P_1, S_1)$ 中的  $P_1$  插入到路径  $r_2$  上的站点  $u_2$  之后,且  $S_1$  放到站点  $v_2$  后;同样,站点对  $(P_2, S_2)$ 中  $P_2$  插入到路径  $r_1$  上的站点  $u_1$  之后,且  $S_2$  放到站点  $v_1$  后。

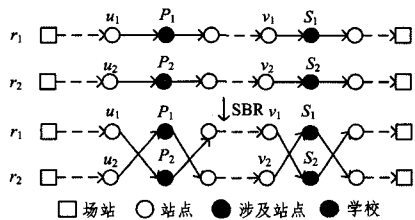


图 3 SBR 操作示意

(3)单点对路径内调整(Within Route Insertion, WRI):在同一条路径上改变站点及其对应学校的位置。其操作示意如图 4 所示。在图 4 中,将学生站点  $P$  移动到站点  $u$  之后,将  $P$  对应的学校站点  $S$  移动到站点  $v$  之后。

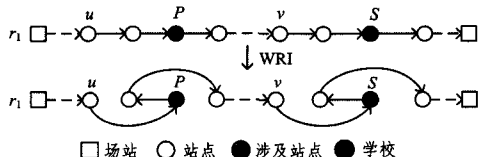


图 4 WRI 操作示意

在这 3 种点对算子的执行过程中,要求每次参与移动的必需是乘车站点和学校组成的站点对。由于在多校 SBRP 的一条路径上,乘车站点和学校之间的关系不是 PDP 中的一对一的关系,而是一对多的关系,因此每次移动学校站点时要避免出现重复的学校站点。即,仅当目标路径上不存在待移动学校站点时,才需要执行学校站点的插入操作。

点对邻域算子的实现过程由邻域空间构造、评估移动和车型调整 3 部分组成。其中,邻域空间构造是根据当前待操作的学生乘车站点,构造该站点及其对应学校站点可能插入位置的集合;评估移动是对于每一个可能插入的位置,评估移动一个点对是否合法、是否满足车型调整条件,并对满足条件的移动执行移动操作;车型调整则是按照车型调整策略实现路径上车型的改变。以 SPI 为例,其实现过程如算法 1 所示。

算法 1 SPI 邻域算子的实现

输入:当前解实例 sol,最好解  $S^*$ ,乘车站点  $j$ ,搜索规则 rules

输出:最好解  $S^*$

```
1. sol. CreateNeighborSpace(j, rules); //构造邻域列表
2. for (int i=0; i<sol. searchSpace. Count; i++){
```

```
3. MNeighborSpace m=sol. searchSpace[i]; //获得一个邻域元素
4. M=new MHSBRPMove();
5. flag=false; //能否调整车型
6. bool result=EvaluateShift(sol,j,m,M,rules,out flag); //评估是否满足移动条件
7. if(result || (!result && flag)) {
8.     ChangeFleet(sol,M); //调整车型
9.     MoveShift(sol,j,m,M); //执行移动操作
10. } //end if
11.  $S^* = \text{getBetter}(\text{sol}, S^*)$ ; //更新当前最好解
12. } //end for
```

在算法 1 中,第 1 行根据当前的搜索规则生成乘车站点  $j$  的邻域搜索列表,每个邻域列表中的元素用一个三元组表示。例如,三元组  $(u, v, t)$  表示的是乘车站点  $j$  及其对应的学校站点将可能插入到路径  $t$  上的站点  $u$  和  $v$  后。第 2—12 行依次搜索乘车站点  $j$  邻域列表中的每个元素,找到可行的插入位置并完成站点  $j$  的移动;同时,更新移动过程中发现的最好解。第 3 行获得邻域列表中的一个邻域元素  $m$ , MNeighborSpace 是描述邻域三元组对象的结构。第 4 行构造 MHSBRPMove 类型的对象  $M$  记录点对移动时路径的变化信息。第 5 行初始化变量  $flag$  为不调整车型。第 6 行评估将乘车站点  $j$  移动到邻域元素  $m$  后是否满足条件,并返回不合法时是否允许车型调整的结果。第 7—10 行表示满足条件时执行车型调整和移动操作。当评估移动合法或者仅违反容量约束时,允许进行车型调整。在车型调整 ChangeFleet 方法中,判定能够找到合适调整的车型并进行调整,同时将调整后的车型信息记录到对象  $M$  中。MoveShift 方法则根据  $M$  对象中记录的路径信息和车型信息,确定是否执行站点  $j$  及其对应学校站点的移动。第 11 行对比移动后的解与当前最好解,并更新当前最好解。

3.4 局部搜索

变邻域下降搜索(VND)是变邻域搜索算法的一种变体,能够有效地完成局部搜索。本文采用 VND 完成局部搜索,并在搜索过程中采用类似记录更新法(RRT)的偏差接受策略<sup>[5]</sup>接受一定范围内的邻域解。在 VND 中,邻域算子按照前期实验确定的最佳顺序(SPI, WRI, SBR)进行执行。VND 的基本实现流程如下:

- (1)设置邻域变量  $k=1$ ,邻域最大数  $k_{\max}=3$ ,初始化邻域列表  $NL=\{SPI, WRI, SBR\}$  及当前解  $S_c$ 。
- (2)获得当前解  $S_c$  中所有乘车站点的集合  $perm$ ,并根据搜索规则对  $perm$  集合中的站点进行排序。
- (3)根据当前的  $k$  值选择对应的邻域算子,并应用该邻域算子对当前解  $S_c$  进行提升,从而得到新解  $S_n$ 。
- (4)如果  $S_n$  优于  $S_c$  或  $S_n$  与  $S_c$  的目标值之差的绝对值在一定的偏差范围内,则置  $S_c = S_n, k=1$ ; 否则,  $k++$ 。
- (5)跳转到步骤(3)继续执行,直到  $k=k_{\max}$ 。

3.5 车型调整策略

合理的车型调整不仅能有效地降低一条路径的成本,还能增加邻域搜索过程中解的多样性。多校 SBRP 中,一条路径上乘车站点和对应的学校站点交替出现,容易出现校车行

驶过程中空载的现象。因此,多校 SBRP 中无法像单校 SBRP 那样直接根据整条路径上的校车剩余容量进行车型调整。鉴于学生站点和学校站点的对应关系,本文设计了一种基于路径段的车型调整方法:首先,将一条路径划分成若干个路径段;然后,针对每一个路径段找到实际校车负载的最大值,并根据该最大值与当前车型容量之间的关系进行车型调整。

### (1) 路径段划分

路径段是一系列乘车站点及其对应学校站点的集合,它以学校站点进行划分。路径段的数目等于一条路径上学校站点的个数。路径段的结束是学校站点,开始则是到达该学校的第一个乘车站点。图 5 给出了一条路径在多校不混载和混载两种模式下的路径段划分示意图。

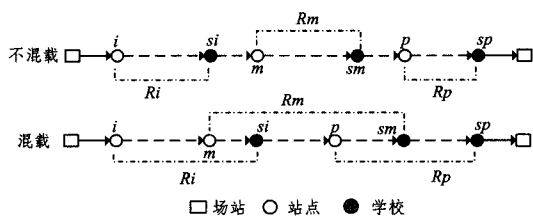


图 5 一条路径在两种运营模式下的路径段划分示意图

在图 5 中, $si, sm$  和  $sp$  是 3 个学校站点,到达该学校的第一个乘车站点分别是  $i, m$  和  $p$ 。由此,可将该路径划分成 3 个路径段,即  $Ri, Rm$  和  $Rp$ 。在多校不混载模式下,由于校车按照顺序依次访问学校,因此每个路径段仅包含到达同一个学校的乘车站点。而在多校混载模式下,允许同一时刻到达不同学校的学生搭乘校车,也就意味着每一个路径段上还可能包含到达其他学校的学生,此时路径段之间就可能存在交叉。

### (2) 车型调整条件和规则

在车型调整时,仅对一条合法路径或者仅违反容量约束的路径进行调整。这两种情况下均需要考虑每个路径段的实际负载与车型容量的关系。假设路径  $r_1$  当前的车型为  $k$ ,车型容量为  $Q_k$ ,该路径上的成本记为  $C_k^r$ ,按车型容量从小到大排列的车型集合  $M = \{1, 2, \dots, m\}$ ;路径  $r_1$  包含的路径段集合  $R = \{R_1, R_2, \dots, R_n\}$ ,每个路径段  $R_i$  的校车实际载重为  $Q_{R_i}$ ,所有路径段上校车实际载重的最大值  $Q_{\max} = \max\{Q_{R_1}, Q_{R_2}, \dots, Q_{R_n}\}$ 。车型调整规则定义为:

1) 当  $Q_k = Q_{\max}$  时,存在某个路径段上校车的实际载重达到当前车型的容量,此时不进行车型调整。

2) 当  $Q_{\max} < Q_k$  时,尝试调小车型。若  $k$  已经是最小车型(即  $k=1$ )则不予调整;否则寻找  $\min\{C_l^r - C_k^r\}$  对应的车型  $l$ ,此时  $l \in \{1, 2, \dots, k\}$ 。

3) 当  $Q_{\max} > Q_k$  时,表示路径  $r_1$  容量约束违反,尝试调大车型。若  $k$  已经是最大车型(即  $k=m$ )则不予调整;否则寻找  $\min\{C_l^r - C_k^r\}$  对应的车型  $l$ ,此时  $l \in \{k, k+1, \dots, m\}$ 。

### 3.6 扰动规则

为了避免算法过早陷入局部最优,本文采用多点移动的方式完成扰动。随机选择若干个学生站点,然后将该学生站点和对应的学校站点移动到其他路径上。由于多校 SBRP 的约束条件紧,扰动站点数太多,不易找到可行解,因此将扰动的学生站点个数控制在 1~5 之间。

### 3.7 算法复杂度

假设 ILS 算法的迭代次数为  $M$ ,问题规模为  $N$ 。采用 GENI 算法构造算法的初始解,由于引入了  $p$  ( $p$  是常量)邻域,因此初始解构造的复杂度为  $O(N * p^4 + N^2)^{[15]}$ 。ILS 算法一次迭代需要执行局部搜索、扰动和解的更新操作。局部搜索使用 SPI, WRI 和 SBR 3 个点对邻域算法完成,其中 SPI 和 WRI 的复杂度为  $O(N^3)^{[4,14]}$ ,SBR 的复杂度为  $O(N^2)^{[4]}$ ,则局部搜索过程的复杂度为  $O(N^3)$ 。扰动采用的是多点移动,本质上仍是使用 SPI 算子,其复杂度亦为  $O(N^3)$ 。解更新操作的复杂度为常量。因此,ILS 算法一次迭代的复杂度是  $O(N^3)$ ,则整个迭代部分的复杂度为  $O(M * N^3)$ 。由于初始解构造的复杂度为  $O(N * p^4 + N^2)$ ,其复杂度约为  $O(N^2)$ ,因此 ILS 算法的总复杂度为  $O(N^2) + O(M * N^3) \approx O(M * N^3)$ 。

## 4 实验与结果分析

本文算法使用 Visual Studio 2010 环境中的 C# 实现。测试环境为 PC 机,机器配置为 Intel core i7-4790 3.60GHz CPU,8GB 内存、Windows 7 64 位操作系统。算法参数设置如下:迭代次数为 50,邻域大小为  $\min\{0.5 * n, 150\}$ ,GENI 算法的参数  $p$  取值为 5,偏差系数取  $10^{-4}$ ,邻域算子的执行顺序为 {SPI, WRI, SBR},搜索规则采用最先接受。测试案例集采用文献 [11] 公开的 CSCB01 - CSCB04 和 RSRB01 - RSRB04 两类多校 SBRP 基准测试案例,其中车辆的行驶速度、学生站点和学校站点的服务时间的设置同文献 [11],距离使用曼哈顿距离进行计算,且以英里为单位。最大乘车时间 (MRT) 设定为 2700s 和 5400s。假设有 A, B, C 3 种车型,容量分别为 50, 60, 70,固定成本分为 2500, 2800 和 3000,所有车型的单位可变成本相同,均为 1。

### 4.1 算法测试

使用 ILS 算法求解 RSRB 和 CSCB 两类测试案例,且运营模式包含多校不混载和多校混载两种。每个案例随机运行 10 次,并统计其最好解、平均解、标准差系数 (%) 和平均运行时间 (s),结果如表 1 所列。表中“Best”,“Average”,“Std”和“AvgTime”分别代表案例 10 次运行的最好解、平均解、标准差系数和平均运行时间。

由表 1 的实验结果可以看出:1) ILS 算法能够在合理的时间内求解多校不混载和混载两种模式的多车型 SBRP 问题。ILS 算法在所有案例上的平均运行时间为 106.62s 和 163.13s;对于最大规模案例(500 个站点),ILS 算法的运行时间在 300s 左右,算法的运算效率相对较高。2) ILS 算法的稳定性较好,ILS 在多校不混载和混载两种模式下的平均标准差在 3.54% 和 3.49%。3) 最大乘车时间 MRT 为 5400 时,ILS 算法的求解结果整体较好,原因在于 MRT 的值变大相当于问题约束更加宽松,算法能够在较短的时间内找到更好的解。4) ILS 算法在多校混载模式的总成本较低。这说明多校运营模式中允许不同学校的学生搭乘同一辆车可以进一步地降低运营成本。因多校混载模式约束多,求解复杂,因此比不混载模式需要更长的运行时间。

表 1 两种模式下案例的测试结果

案例	站点数目	学校数目	MRT	多校不混载模式				多校混载模式			
				Best	Average	Std/%	AvgTime/s	Best	Average	Std/%	AvgTime/s
CSCB01	250	6	2700	80856.87	85673.80	3.07	55.19	78206.83	81963.76	2.60	78.96
CSCB02	250	12	2700	74371.82	78141.87	1.93	61.91	70096.54	74876.08	3.56	90.45
CSCB03	500	12	2700	152354.28	163223.23	2.09	178.63	141227.16	148782.98	1.65	316.42
CSCB04	500	25	2700	162594.13	172144.24	2.31	173.07	157246.92	163395.87	2.48	252.84
RSRB01	250	6	2700	81670.15	88332.56	4.58	49.53	80059.15	85024.96	3.71	59.61
RSRB02	250	12	2700	78032.18	81694.04	3.68	51.93	75811.45	79991.72	3.06	62.29
RSRB03	500	12	2700	151705.60	169092.69	5.72	155.82	153119.55	174196.38	3.49	197.6
RSRB04	500	25	2700	160816.51	165525.10	2.02	166.82	153601.61	161501.74	1.68	225.94
CSCB01	250	6	5400	68698.92	72590.23	4.17	55.61	66911.91	72086.74	8.97	97.36
CSCB02	250	12	5400	57707.51	62262.367	4.12	58.13	56834.79	60285.96	4.06	94.82
CSCB03	500	12	5400	119835.28	125401.31	1.55	164.34	112928.49	114522.38	1.97	311.61
CSCB04	500	25	5400	122147.47	128493.44	3.54	163.43	113756.89	121735.01	1.28	273.37
RSRB01	250	6	5400	72231.21	86489.16	5.45	37.92	72290.94	81602.47	5.72	51.68
RSRB02	250	12	5400	63919.06	67283.80	3.60	47.25	61349.26	63792.19	2.06	73.35
RSRB03	500	12	5400	134331.26	154683.54	6.87	138.93	137510.73	145548.21	6.58	179.46
RSRB04	500	25	5400	127013.53	131986.66	1.96	147.48	121234.66	128363.54	2.99	244.34
平均	—	—	—	106767.86	114563.63	3.54	106.62	103261.68	109854.37	3.49	163.13

4.2 算法比较

将本文设计的 ILS 算法与文献[4]和文献[10]提出的 RLBH 算法和 ALBH 算法进行比较。ALBH 是文献[10]提出的自适应基于位置的启发算法。根据文献[4]和文献[10]公开的算法实现流程,模拟实现这两个算法,并假设 RLBH 算法和 ALBH 算法使用相同的车型调整策略。在相同的测试条件下,分别使用这 3 种算法求解本文的测试案例,统计的最好解及其运行时间如表 2 所列。表中“ $C_{ILS}$ ”,“ $C_{RLBH}$ ”,“ $C_{ALBH}$ ”列分别对应 ILS,RLBH 和 ALBH 3 种算法获得的最佳解的目标值,“ $T_{ILS}$ ”,“ $T_R$ ”,“ $T_A$ ”表示 3 种算法对应的运算时间(s)。

由表 2 的实验结果可知:1)ILS 算法在不混载和混载两

种模式下均优于 RLBH 算法和 ALBH 算法,并且在所有的案例上均找到了最好解。在不混载模式下,ILS 算法相比 RLBH 算法和 ALBH 算法分别平均改进了 29.01%和 34.84%;而在多校混载模式下,ILS 算法相比 RLBH 和 ALBH 分别平均改进了 28.93%和 34.66%。2)在多校混载模式下,校车运营的总成本较低。多校混载模式比不混载模式的成本平均降低了 3.28%。3)ILS 算法需要比 RLBH 算法和 ALBH 算法更多的运算时间,这是因为 RLBH 和 ALBH 算法均是构造启发式算法,拥有较好的时间优势,但求解质量有限;而 ILS 算法在适当增加运算时间的基础上,获得了更高质量的解;并且,对于站点规模最大的案例 CSCB03—CSCB04 和 RSRB03—RSRB04 而言,其将运算时间仍控制在合理的范围内(300s 左右)。

表 2 两种模式下 3 种多车型 SBPR 算法的比较

案例	MRT	多校不混载模式						多校混载模式					
		$C_{ILS}$	$C_{RLBH}$	$C_{ALBH}$	$T_{ILS}/s$	$T_R/s$	$T_A/s$	$C_{ILS}$	$C_{RLBH}$	$C_{ALBH}$	$T_{ILS}/s$	$T_R/s$	$T_A/s$
CSCB01	2700	80856.87	112509.97	124607.23	55.19	1.87	1.62	78206.83	112311.41	116161.55	78.96	2.84	4.63
CSCB02	2700	74371.82	109565.28	130979.77	61.91	1.57	1.64	70096.54	110366.03	125300.56	90.45	2.98	2.93
CSCB03	2700	152354.28	211797.91	225501.18	178.63	9.04	12.05	141227.16	198958.11	218822.98	316.42	20.05	29.69
CSCB04	2700	162594.13	255850.64	280267.22	173.07	8.46	7.88	157246.92	239906.87	279074.81	252.84	19.91	19.40
RSRB01	2700	81670.15	110654.13	114979.21	49.53	1.97	2.42	80059.15	108350.88	112018.29	59.61	3.53	8.83
RSRB02	2700	78032.18	117115.79	134013.91	51.93	2.15	1.74	75811.45	108812.27	128807.63	62.29	4.98	5.57
RSRB03	2700	151705.60	221803.92	217608.32	155.82	9.44	12.16	153119.55	204734.33	212506.22	197.60	18.13	33.29
RSRB04	2700	160816.51	238385.14	246312.13	166.82	8.62	12.02	153601.61	234248.48	239737.12	225.94	19.56	36.02
CSCB01	5400	68698.92	92915.59	111013.46	55.61	2.90	3.25	66911.91	87660.03	104670.31	97.36	3.85	11.71
CSCB02	5400	57707.51	84636.99	92570.66	58.13	2.45	2.93	56834.79	80848.90	89640.51	94.82	4.67	7.77
CSCB03	5400	119835.28	158059.87	179243.10	164.34	13.35	17.89	112928.49	152381.01	168814.99	311.61	27.79	66.05
CSCB04	5400	122147.47	178241.57	206143.52	163.43	12.93	13.24	113756.89	176513.93	215259.98	273.37	28.92	33.10
RSRB01	5400	72231.21	88869.81	105261.63	37.92	2.45	4.25	72290.94	85425.43	101184.23	51.68	3.97	17.10
RSRB02	5400	63919.06	94429.38	103542.05	47.25	3.08	4.02	61349.26	93232.15	96880.95	73.35	6.52	12.88
RSRB03	5400	134331.26	169555.27	159268.47	138.93	12.14	24.94	137510.73	170795.19	162315.25	179.46	26.50	90.33
RSRB04	5400	127013.53	182363.83	192606.79	147.48	12.21	18.14	121234.66	180046.82	177650.78	244.34	26.49	58.09
平均	—	106767.86	151672.19	163994.92	106.62	6.54	8.76	103261.68	146536.99	159302.89	163.13	13.79	27.34

ILS 算法优于 RLBH 算法和 ALBH 算法的原因在于: ILS 算法能够借助邻域搜索和车型调整等策略提高算法的寻优能力,同时允许接受一定偏差范围内的解以保证算法的多样性。而 RLBH 和 ALBH 算法属于构造启发式算法,二者的不同之处在于每次构造一条新路径时点的选择方式不同: RLBH 算法使用的是随机选择,而 ALBH 每次选择距离学校最远的站点。RLBH 算法和 ALBH 算法都是贪婪地寻找成

本最廉价的站点完成站点插入,并且一开始路径上使用大车,导致后期车型调整比较困难。

为了进一步验证 ILS 算法的有效性,将其与现有的多校 SBPR 元启发算法进行比较。由于目前多校 SBPR 元启发算法求解的是单车型问题,并且以车辆数为最小目标,因此本文采用与文献[11,13,16]相同的校车容量和优化目标(最小化车辆数)、相同的站点服务时间和站点间距离计算方式。在多

校不混载和混载两种模式下,ILS算法与文献[11]提出的后启发算法(PH)、文献[13]提出的不混载 SBRP 的 SA 算法、文献[16]提出的混载 SBRP 的 RRT 算法的比较结果如表 3 所列。表中“ $N_{ILS}$ ”表示 ILS 算法获得的最小车辆数,“ $N_{PH}$ ”,“ $N_{SA}$ ”和“ $N_{RRT}$ ”的值分别取自文献[11,13]和文献[16]中给出的结果。

表 3 ILS 算法与其他 3 种算法的比较

案例	MRT	多校不混载模式			多校混载模式		
		$N_{ILS}$	$N_{PH}$	$N_{SA}$	$N_{ILS}$	$N_{PH}$	$N_{RRT}$
CSCB01	2700	28	39	31	26	30	27
CSCB02	2700	25	33	26	25	30	25
CSCB03	2700	53	66	59	48	55	50
CSCB04	2700	57	72	61	54	62	55
RSRB01	2700	23	35	29	23	24	23
RSRB02	2700	20	27	23	19	22	19
RSRB03	2700	40	52	42	39	41	40
RSRB04	2700	41	57	45	39	43	37
CSCB01	5400	25	35	26	26	30	26
CSCB02	5400	27	32	27	26	29	25
CSCB03	5400	51	66	47	51	56	51
CSCB04	5400	54	68	58	52	59	53
RSRB01	5400	24	31	28	23	27	25
RSRB02	5400	22	30	23	22	23	23
RSRB03	5400	46	61	46	47	47	46
RSRB04	5400	41	56	41	41	46	40
平均	—	36.06	47.50	38.25	35.06	39.00	35.12

由表 3 可知:1)本文设计的 ILS 算法同样适用于求解多校单车型 SBRP 问题,并且具有较好的优化性能。无论是混载还是不混载运营模式,ILS 算法均能获得较少的平均车辆数。2)ILS 算法显著优于后启发算法,不混载和混载模式下平均车辆数分别下降了 24.53%和 10.13%。3)不混载模式下,ILS 算法明显优于 SA 元启发算法。相比于 SA 算法,ILS 算法在所有案例上平均改进了 6.16%,单个案例上最大改

进 20.69%。与混载 SBRP 的 RRT 算法相比,ILS 算法在所有案例上平均改进了 0.76%。

ILS 算法优于后启发算法、SA 算法的原因在于:后启发算法和 SA 算法都是采用分段求解模式,并没有从全局的角度进行优化。后启发算法和 SA 算法均是在先得到单校路径之后再行路径的合并。二者的不同之处在于:后启发算法合并路径时使用简单的改进启发策略,而 SA 算法则在模拟退火框架下完成路径的合并。ILS 算法和 RRT 算法均是从全局的角度进行优化,但 ILS 算法略优于 RRT 算法。原因在于:RRT 仅通过偏差系数来增强搜索的多样性,而 ILS 算法则引入可变邻域下降、扰动机制和允许接受较差解等多种算法策略,整体上能够获得较好的优化性能。

综合考虑表 2 和表 3 可知,本文设计的 ILS 算法能够有效求解多校不混载与混载、单车型与多车型的 SBRP 问题,并且在求解性能上优于现有的算法。

### 4.3 邻域算子执行策略的分析

为了评估邻域算子执行策略对算法的影响,另外设计 3 种邻域算子执行策略,并与本文采用的 VND 邻域算子执行策略进行比较。这 3 种邻域执行方式包括:1)固定顺序(FLS),即按照 SPI, WRI 和 SBR 的固定顺序依次执行;2)随机顺序(RLS),即按照随机顺序完成 SPI, WRI 和 SBR 算子的执行;3)随机选择邻域执行的 VND(RVND)<sup>[17]</sup>,它与 VND 的不同之处在于每次随机选择邻域算子执行。基于相同的测试环境和参数设置,在 ILS 算法框架内使用这 3 种执行方式分别求解本文的测试案例。每个案例运行 10 次,并记录其最好解、标准差、执行时间等信息。多校不混载和混载两种模式下不同算子执行策略的平均成本、平均标准差和平均执行时间的比较结果如表 4 所列。

表 4 两种模式下邻域算子不同执行策略的比较

衡量指标	多校不混载模式				多校混载模式			
	FLS	RLS	RVND	VND	FLS	RLS	RVND	VND
平均成本	107908.79	115771.16	114295.53	106767.86	104259.64	111022.93	109575.98	103261.68
平均标准差/%	4.08	6.71	6.94	3.54	3.49	6.58	7.12	3.49
平均执行时间/s	70.40	94.77	97.69	106.62	105.05	138.24	412.79	161.13

由表 4 可知:1)邻域算子使用 VND 方式执行能够获得更低的平均成本,在求解质量上显著优于多个邻域算子执行,即 VND 优于 FLS, RVND 优于 RLS。这也证明,在 ILS 算法中混合使用 VND 能够提高算法的寻优能力。2)邻域算子按照事先确定的最佳顺序执行比随机执行效果更好,即 VND 优于 RVND, FLS 优于 RLS。3)邻域算子按照固定顺序执行时,算法能够得到较小的标准差,当邻域算子执行过于随机时算法标准差较大,稳定性较差。由此可知,使用固定顺序执行邻域算子时,算法的稳定性较好。4)邻域算子采用 VND 或 RVND 方式执行需要更多的运算时间。

综合考虑表 4 中的衡量指标,可以得出以下结论:邻域算子采用 VND 方式执行在寻优能力上优于直接执行邻域算子。尽管使用该方式需要更多的执行时间,但并未显著提高算法的复杂度。邻域算子按照固定顺序执行时,算法的稳定性较好。

### 4.4 参数对算法影响的分析

#### (1)迭代次数对算法的影响

ILS 算法是通过迭代执行 SPI, WRI 和 SBR 3 个点算子完成局部提升的,因此迭代次数会影响 ILS 算法的求解质量和运算时间。为了测试迭代次数对算法的影响,在其他参数取值不变的情况下,允许算法最多执行 100 次,同时记录不混载和混载两种模式下所有案例最好解的平均值和平均运算时间(s),如图 6 和图 7 所示。

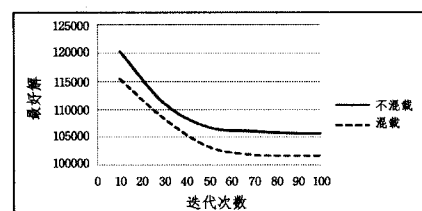


图 6 不同迭代次数下算法最好解的平均值

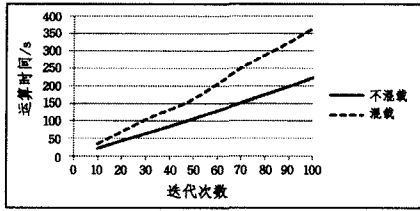


图7 不同迭代次数下算法的平均运算时间

由图6可知,无论是多校不混载还是多校混载模式,随着算法迭代次数的增加,最好解的平均值都有下降的趋势。当迭代次数从10增加到50时,最好解的平均值下降较快;当迭代次数继续增加时,后续最好解的平均值下降得极其缓慢。而由图7可知,随着迭代次数的增加,算法的平均运算时间增加较快。由此可知,增加迭代次数能够提高算法的求解质量,但当增加到一定值后,算法的求解质量不再显著提升;同时,迭代次数的增加会导致算法运算时间快速增加。

### (2) 偏差系数对算法的影响

本文的ILS算法允许接受一定偏差范围内的解,以增强算法局部搜索的多样性。偏差范围表示为当前解的成本值与偏差系数的乘积。偏差系数取值为0时,表示算法总是接受有改进的邻域解;偏差系数的大小决定了接受的邻域解与当前解的差异程度。为了测试不同偏差系数的取值对算法的影响,在算法其他参数保持不变的情况下,测试偏差系数取不同值时算法所有案例最好解的平均值,统计结果如表5所列。

表5 偏差系数取不同值时对算法的影响

偏差系数的值	所有案例最好解的平均值	
	不混载模式	混载模式
0	109200.60	105471.14
$10^{-1}$	110798.11	107859.16
$10^{-2}$	109058.44	106758.30
$10^{-3}$	110079.90	107304.01
$10^{-4}$	106767.86	103261.68
$10^{-5}$	109443.60	105276.90
$10^{-6}$	110998.57	107602.70
$10^{-7}$	108805.23	105394.45

从表5可以看出,偏差系数的不同取值会影响算法的求解质量。当偏差系数的取值为 $10^{-4}$ 时,ILS算法在多校不混载和多校混载两种模式下的求解质量最好。当偏差系数取0时,算法总接受有改进的解,算法的多样性不足,并不能获得更好的解。偏差系数太小时也不能提高算法的求解质量,其原因在于偏差系数太小,所得的偏差值过小,并不能增强邻域解的多样性。

**结束语** 本文针对多校不混载和混载两种联合运营模式下的多车型校车路径问题,提出了一种迭代局部搜索算法(ILS)进行求解。算法中引入了PDPTW问题求解中的邻域算子,并使用可变邻域下降(VND)完成算子的执行,同时允许进行车型调整。在国际基准测试案例上进行了实验,结果表明本文设计的ILS算法能够有效地求解多车型和单车型多校SBRP问题。进一步测试邻域算子的执行策略,结果发现使用VND方式执行邻域算子能够在适当增加运算时间的情况下显著提高算法的寻优能力。当邻域算子按照固定顺序执

行时,算法的寻优能力优于邻域算子随机执行的寻优能力,且算法的稳定性较好。对算法的参数进行分析发现,迭代次数和偏差系数的取值会影响算法的性能,适当地设置参数的取值能够提高算法的性能。由于本算法中使用的点对邻域算子的时间复杂度较高,下一步将尝试从邻域空间构造、点对移动评估方面尽可能地排除点对无效的移动,从而减少算子的执行时间。

### 参考文献

- [1] NEWTON R M, THOMAS W H. Design of school bus routes by computer[J]. Socio-Economic Planning Sciences, 1969, 3(1): 75-85.
- [2] PARK J, KIM B I. The school bus routing problem: A review [J]. European Journal of Operational Research, 2010, 202(2): 311-319.
- [3] DANG L X, CHEN X P, KONG Y F. Review of School Bus Routing Problem: Concept, Model and Optimization Algorithms [J]. Journal of Henan University (Natural Science), 2013, 43(6): 682-691. (in Chinese)  
党兰学, 陈小潘, 孔云峰. 校车路径问题模型及算法研究进展 [J]. 河南大学学报(自然科学版), 2013, 43(6): 682-691.
- [4] BRACA J, BRAMEL J, POSNER B, et al. A Computerized Approach to the New York City School Bus Routing Problem [J]. IIE Transactions, 1997, 29(8): 693-702.
- [5] DANG L X. Optimization Algorithms for Large Scale Mixed Load School Bus Routing Problem [D]. Kaifeng: Henan University, 2014. (in Chinese)  
党兰学. 大规模混载校车路径问题优化算法研究[D]. 开封: 河南大学, 2014.
- [6] BOGL M, DOERNER K F, PARRAGH S N. The School Bus Routing and Scheduling Problem with Transfers [J]. Networks, 2015, 65(2): 180-203.
- [7] CHEN X P, DANG L X, KONG Y F. A Meta-heuristic Algorithm to Solve the Large-Scale School Bus Scheduling Problem [J]. Journal of Geo-Information Science, 2013, 15(6): 879-885. (in Chinese)  
陈小潘, 党兰学, 孔云峰. 一种求解大规模校车调度问题的元启发式算法[J]. 地球信息科学学报, 2013, 15(6): 879-885.
- [8] SPADA M, BIERLAIRE M, LIEBLING T M. Decision-Aiding Methodology for the School Bus Routing and Scheduling Problem [J]. Transportation Science, 2005, 39(4): 477-490.
- [9] THANGIAH S R, FERGANY A, WILSON B, et al. School Bus Routing in Rural School Districts [C] // Proceedings of the 9th International Conference on Computer-Aided Scheduling of Public Transport. Spring, 2008: 209-232.
- [10] DE SOUZA L, SIQUEIRA P H. Heuristic Methods Applied to the Optimization School Bus Transportation Routes-A Real Case [C] // 23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems. 2010: 247-256.
- [11] PARK J, TAE H, KIM B I. A Post-improvement Procedure for the Mixed Load School Bus Routing Problem [J]. European



- Journal of Operational Research, 2012, 217(1):204-213.
- [12] KIM B, KIM S, PARK J. A school bus scheduling problem[J]. European Journal of Operational Research, 2012, 218(2): 577-585.
- [13] CHEN X, KONG Y, DANG L, et al. Exact and Metaheuristic Approaches for a Bi-objective School Bus Scheduling Problem [J]. Plos One, 2015, 10(7): e0132600.
- [14] NANRY W P, BARNES J W. Solving the pickup and delivery problem with time windows using reactive tabu search [J]. Transportation Research Part B Methodological, 2000, 34(2): 107-121.
- [15] WU B. Particle Swarm Optimization for Vehicle Routing Problem and its Application[D]. Hangzhou: Zhejiang University of Technology, 2008. (in Chinese)
- 吴斌. 车辆路径问题的粒子群算法研究与应用[D]. 杭州: 浙江工业大学, 2008.
- [16] DANG L X, HOU Y E, KONG Y F. Spatiotemporal Neighborhood Search for Solving Mixed-load School Bus Routing Problem[J]. Computer Science, 2015, 42(4): 221-225. (in Chinese)
- 党兰学, 侯彦娥, 孔云峰. 时空相关的混载校车路径问题邻域搜索[J]. 计算机科学, 2015, 42(4): 221-225.
- [17] PENNA P H V, SUBRAMANIAN A, OCHI L S. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem[J]. Journal of Heuristics, 2013, 19(2): 201-232.

(上接第 197 页)

**结束语** 本文提出了一种利用马尔可夫模型对城市移动对象稀疏轨迹终点预测的方法。在此过程中, 提出并证明了比均匀划分网格更有优势的基于 K-d tree 网格划分样本运动空间的算法。然后, 本文详细地阐述了预测方法和步骤。最后, 本文通过实验对比了 Baseline, LSTM, SubSyn 和 Markov Model 4 种不同预测算法的精度、时间效率和轨迹覆盖率, 综合得出马尔可夫模型可以有效解决稀疏轨迹终点预测问题。

本文中研究的不足之处是仅仅使用了轨迹数据信息, 并没有考虑路况、天气、移动对象的运行速度等上下文信息, 因此轨迹终点预测还有待进一步研究。在未来工作中, 可以对上下文信息进行特征抽取并将其加入到预测模型中, 以进一步提高预测的精度。

### 参 考 文 献

- [1] GONZALEZ M C, HIDALGO C A, BARABASI A L. Understanding individual human mobility patterns[J]. Nature, 2008, 453(7196): 779-782.
- [2] SONG C, QU Z, BLUMM N, et al. Limits of predictability in human mobility[J]. Science, 2010, 327(5968): 1018-1021.
- [3] ZHENG Y, ZHOU X F, et al. Computing with spatial trajectories[M]. Springer Science & Business Media, 2011.
- [4] MAMOULIS N, CAO H, KOLLIOS G, et al. Mining, indexing, and querying historical spatiotemporal data[C]//Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2004: 236-245.
- [5] MORZY M. Mining frequent trajectories of moving objects for location prediction[M]//Machine Learning and Data Mining in Pattern Recognition. Springer Berlin Heidelberg, 2007: 667-680.
- [6] JEUNG H, LIU Q, SHEN H T, et al. A hybrid prediction model for moving objects[C]//IEEE 24th International Conference on Data Engineering, 2008(ICDE 2008). IEEE, 2008: 70-79.
- [7] YING J J C, LEE W C, WENG T C, et al. Semantic trajectory mining for location prediction[C]//Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, 2011: 34-43.
- [8] ZHENG Y, ZHANG L, XIE X, et al. Mining interesting locations and travel sequences from GPS trajectories[C]//Proceedings of the 18th International Conference on World Wide Web. ACM, 2009: 791-800.
- [9] XUE A Y, ZHANG R, ZHENG Y, et al. Destination prediction by sub-trajectory synthesis and privacy protection against such prediction[C]//2013 IEEE 29th International Conference on Data Engineering (ICDE). IEEE, 2013: 254-265.
- [10] XUE A Y, QI J, XIE X, et al. Solving the data sparsity problem in destination prediction[J]. The VLDB Journal, 2015, 24(2): 219-243.
- [11] GHAHRAMANI Z. An introduction to hidden Markov models and Bayesian networks[J]. International Journal of Pattern Recognition and Artificial Intelligence, 2001, 15(1): 9-42.
- [12] Handbook of Markov Chain Monte Carlo[M]. CRC Press, 2011.
- [13] 汪荣鑫. 随机过程[M]. 西安: 西安交通大学出版社, 2006.
- [14] KARLIN S. A first course in stochastic processes[M]. Academic Press, 2014.
- [15] BAKULA M K, NIKODEM K. On the converse Jensen inequality for strongly convex functions[J]. Journal of Mathematical Analysis and Applications, 2016, 434(1): 516-522.
- [16] STONE J V. Bayes' rule: a tutorial introduction to Bayesian analysis[M]. Sebtel Press, 2013.
- [17] BERNARDO J M, SMITH A F M. Bayesian theory[J]. Journal of the Royal Statistical Society, 2000, 15(19): 13-23.
- [18] LIU J W, LIU Y, LUO X L. Research and development on deep learning[J]. Application Research of Computers, 2014, 31(7): 1921-1930. (in Chinese)
- 刘建伟, 刘媛, 罗雄麟. 深度学习研究进展[J]. 计算机应用研究, 2014, 31(7): 1921-1930.
- [19] GREFF K, SRIVASTAVA R K, KOUTNÍK J, et al. LSTM: A search space odyssey[J]. IEEE Transactions on Neural Networks & Learning Systems, 2015, pp(99): 1-11.
- [20] WÖLLMER M, KAISER M, EYBEN F, et al. LSTM-Modeling of continuous emotions in an audiovisual affect recognition framework[J]. Image and Vision Computing, 2013, 31(2): 153-163.
- [21] MONNER D, REGGIA J A. A generalized LSTM-like training algorithm for second-order recurrent neural networks[J]. Neural Networks, 2012, 25(1): 70-83.