时空相关的混载校车路径问题邻域搜索

党兰学[1] 侯彦娥[1,2] 孔云峰[2]
(河南大学计算机与信息工程学院 开封 475004)[1] (河南大学环境与规划学院 开封 475004)[2]

摘 要 为节约混载校车路径问题求解过程中邻域解搜索的时间,引入时空距离和时空相关度概念,将邻域搜索空间限定在合理的范围内。该算法首先计算站点间的时空距离,再附加上简单约束的预判断,从而得到时空相关度矩阵。然后对于任意学生乘车站点,将其他可能与之直接相连的站点按照时空相关度排序,形成一个邻接列表。在邻域搜索过程中,通过限定邻接列表长度,仅尝试最终接受概率较大的一部分移动操作,以此缩小邻域搜索空间,从而提高算法效率。在国际标准案例上的测试结果表明,基于时空相关度的搜索策略能在基本不降低求解质量的情况下,平均节省50%以上的求解时间。

关键词 校车路径问题,混载,邻域搜索,时空距离,时空相关度
中图法分类号 TP301.6 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.4.045

Spatiotemporal Neighborhood Search for Solving Mixed-load School Bus Routing Problem

DANG Lan-xue[1] HOU Yan-e[1,2] KONG Yun-feng[2]
(College of Computer and Information Engineering, Henan University, Kaifeng 475004, China)[1]
(College of Environment and Planning, Henan University, Kaifeng 475004, China)[2]

Abstract Neighborhood search is one of the key issues in the algorithm design for solving mixed-load school bus routing problem (SBRP). For saving the execution time of neighborhood search, this paper introduced the concepts of spatiotemporal distance and spatiotemporal connectivity index to reduce the size of searching space. The spatiotemporal distance between stop nodes was defined as the weighted sum of normalized spatial distance and temporal distance. Moreover, the spatiotemporal connectivity index, a number indicating the feasible connectivity between two SBRP nodes, is determined according to spatiotemporal distance and constraints such as school bus capacity, school time window and maximum riding time. The spatiotemporal connectivity indexes are sorted in a descending order for each node, and the neighborhood lists for all nodes are prepared for neighborhood search operators. In the following step of node moves, the neighborhood node with higher index value will be examined preferentially. Thus the effective node will be accepted as earlier as possible, and the size of neighborhood lists can be reduced substantially. The test of a set of large-scale mixed-load SBRP instances shows that the spatiotemporal neighborhood search is capable of reducing the neighborhood search space and saving the solution time dramatically.

Keywords School bus routing problem, Mixed-load, Neighborhood search, Spatiotemporal distance, Spatiotemporal connectivity index

1 引言

校车路径问题(SBRP)是指对于一批校车,确定适当的行驶路线,有序地将学生从居住地送达学校或从学校送回,并在一定条件下使其成本(或其他目标)最优,其本质是在保证服务质量的前提下尽可能地节约成本[1]。如果规划路径时允许一辆校车同时搭载不同学校的学生,则称为混载校车路径问题[2]。

文献[3]从不同角度对SBRP研究做了详尽的回顾。早期文献[2,4]对混载SBRP进行了讨论,指出允许混载是降低运营成本的有效途径。文献[5-7]针对混载SBRP给出了具体的启发式算法。但这些算法都属于构造式启发算法,即使有改进的阶段,也只是应用简单的策略对路径进行优化。

在车辆路径问题(VRP)启发式优化算法中,普遍利用邻域算子对路径进行改进。针对带有时间窗的装卸一体化问题(PDPTW)站点移动需要成对操作的特点,Nanry等[8]定义了3种邻域算子,分别是单个站点对插入(SPI)、路径间站点对交换(SBR)、路径内插入(WRI)。党兰学等[9]将SPI、SBR、WRI 3种邻域搜索算子引入到混载SBRP求解中,设计了以记录更新法(RRT)为框架的多邻域结构启发算法,其有效地

到稿日期:2014-06-03 返修日期:2014-08-26 本文受国家自然科学基金资助项目(41401461),河南省教育厅科学技术研究重点项目(13A520050)资助。

党兰学(1980-),男,博士,讲师,CCF会员,主要研究方向为人工智能、空间分析与优化等,E-mail:danglx@foxmail.com;侯彦娥(1980-),女,博士生,主要研究方向为空间分析与优化;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,E-mail:yfkong@henu.edu.cn(通信作者)。

提升了求解质量，但在扩展邻域搜索策略的同时增加了求解时间。局部搜索的时间复杂度与搜索空间、约束条件、启发策略相关[10]。为改善邻域搜索耗时问题，Fang 等[10]在求解一般优化问题时，将约束条件分为几个相对独立的集合，通过建立约束集的交集，提出“岛约束方法”减小搜索空间。Chen 和 Smith[11]针对具有时序特征的优化问题，通过预先简单的顺序约束判断，减小搜索规模。

混载 SBRP 约束条件多且复杂，例如最大乘车时间无法在站点移动之前评估是否满足，因此基于邻域算子进行局部搜索相当耗时，是影响算法效率的重要因素。控制搜索空间从而减少大邻域空间搜索耗时的问题是提高算法效率的有效途径之一。鉴于此，本文在元启发算法中引入时空距离，结合简单约束预判断，定义时空相关度，在邻域搜索过程中通过相关度来限定搜索有限范围，提高算法效率。

2 混载校车路径问题及求解算法

2.1 混载校车路径问题描述

假定在场站 d 内有 K 辆校车，容量为 Q。站点 i 和 j 间的运行距离和时间分别为 c_{ij} 和 t_{ij}。学校具有严格的时间窗 $[e_i, l_i]$，校车到达学校的时间应该处于时间窗内；学生站点的时间窗则相对宽裕，以场站开放时间为最早时间 e_i，以对应学校的 l_i 为最晚时间。

本文所求解的混载 SBRP 以车辆数最少为目标。主要的约束条件有：

(1)任一校车 k 从场站 d 出发，送完学生以后回到场站。

(2)学生站点和其对应学校站点在同一条线路上，校车先经过学生站点后经过其对应的学校。即校车对站点的访问是成对并且有顺序要求的。

(3)任何时刻校车上学生人数都不超过 Q。

(4)校车到达学校的时间 T_i 必须在学校的时间窗内，即 $e_i \leqslant T_i \leqslant l_i$。

(5)对于任何站点的学生，其最大的乘车时间不超过最大乘车时间 R。

2.2 求解算法

本文的算法从初始解出发，以 RRT 作为基础，通过邻域操作算子优化线路[9]。算法整体描述如下：

(1)数据准备。接收算法所需参数，读入站点、道路网、车辆等信息，计算站点 O-D 矩阵等。

(2)构造初始解。将所有学生站点直接和它所对应的学校连接成一条路径，所有路径的集合即为初始解。

(3)组合使用 SPI、WRI、SBR 邻域算子[8,9]搜索邻域解，依据启发策略逐步改进当前解。使用每个算子搜索邻域时，先确定参与移动的站点(对)，然后构造邻域搜索空间，逐一搜索邻域解。搜索过程中，需要判断新解是否满足约束，按照最先或最优的策略决定是否接受新解。

(4)输出路径方案，整体上需要的路径(车辆)数，总的行驶里程、时间，车辆在每个站点服务的开始时间、终止时间、等待时间等。

3 时空相关的邻域搜索

3.1 算法思想

优化问题可以形式化地表述为：$\min\{f(x) \mid x \in X, X \subseteq S\}$。其中 $f(x)$ 是目标函数，S 是问题的解空间，x 是可行解，X 是可行解的集合。当 S 是一个有限但非常大的集合时，问题通常属于 NP 难题。针对这一类问题，通常是在当前解的基础上，基于特定的邻域结构迭代多次发现新解，逐步进行改进。假设 $N(x) \subseteq S$ 表示当前解 x 通过邻域算子 N 发现的新解的集合，判定集合中新解的目标值根据接受规则确定接受的新解。从解 $x \to x'$ 是通过邻域算子 N 操作完成的。具体到本文混载 SBRP 的算法，即应用 SPI、SBR、WRI 这 3 个算子移动站点对的位置而产生新解。因此确定邻域解搜索的空间可以转换为确定待移动站点可选择的插入位置。

站点最终移动的位置并不是事先可以直接确定的，需要对所有可能的位置进行尝试。但有一部分因为不满足约束条件的位置经过检验后需要排除。在 VRP 的邻域搜索算法中，移动站点时会事先评估插入站点与当前移动站点直接相连是否违反距离约束，即通过空间距离来判断是否可以移动到指定位置。因此对于一个待移动的站点，通过站点间的距离可以划定一个范围，范围内的站点可以与指定站点直接相连形成路径上的边，除此之外的其他站点则不允许与之连接，如图 1(a)所示。

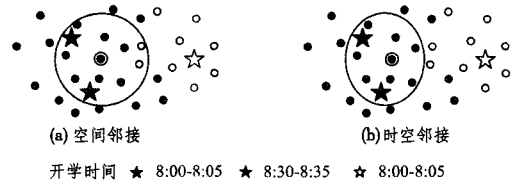


图 1 空间邻接和时空邻接

混载 SBRP 带有时间窗约束、最大乘车时间的约束，这些与时间有关的约束会进一步减小实际可行的插入位置，把一些时间窗口过近、不能准时达到的站点排除在外，如图 1(b)所示。基于以上认识和分析，在邻域搜索过程中引入时空相关度来限定搜索有限范围，提高算法效率。

3.2 时空相关度的定义

在混载 SBRP 中，两个站点是否能够连接形成一个路段弧线不仅与两点间的距离有关，也与站点的时间窗有关。为了方便度量两个站点之间的连接关系，引入时空相关度的概念，定义如下：

$$Related_{ij} = 1/(ST_{ij} + u) \tag{1}$$

其中，时空距离 $ST_{ij} = \alpha c'_{ij} + \beta d'_{ij}$。$c'_{ij}$ 和 d'_{ij} 分别为空间距离 c_{ij} 和时间窗距离 d_{ij} 的归一化值，介于[0,1]之间。α 和 β 为权重系数，用来体现空间和时间因素在时空距离中的比重。

两个站点间的时间窗距离 d_{ij} 用它们的时间窗来刻画[12]。由于学生站点并没有严格意义上的时间窗，对其服务的时间受其对应学校的开学时间制约，因此两个站点的时间距离使用其所对应学校的时间距离来代替。站点 i 和站点 j 的时间窗距离为：

$$d_{ij} = \begin{cases} e_{s(j)} - l_{s(i)}, & e_{s(j)} > l_{s(i)} \\ e_{s(i)} - l_{s(j)}, & e_{s(i)} > l_{s(j)} \\ 0, & e_{s(j)} \leqslant l_{s(i)} \wedge l_{s(j)} \geqslant l_{s(i)} \\ 0, & e_{s(i)} \leqslant l_{s(j)} \wedge l_{s(i)} \geqslant l_{s(j)} \end{cases}$$

其中，$s(i)$ 和 $s(j)$ 分别是站点 i 和站点 j 对应的学校。

式(1)中的 u 为附加因子，是对约束条件的简单判断[11]，比如在两个站点的运行时间超过最大乘车时间、两个站点间

(或对应的两个学校间)的行驶时间大于可用时间、两个站点容量之和超过车辆容量等情况下,直接令 $u=+\infty$(这里设置 $u=10^9$),此时 $Related_{ij}\approx 0$。虽只进行简单的约束检测,但能把一些明显不符合要求的邻接点过滤掉,在有限的邻接表长度内保留更多可能的邻接站点。

将其他站点按与站点 i 的时空相关度从大到小排序后构成的一个列表,称为站点 i 的邻接表,列表的长度称为邻接度。算法的思想是通过限定站点的邻接度来减小邻域搜索空间,从而提高算法的执行效率。

3.3 邻域搜索算子

混载 SBRP 的解是由站点构成的一个序列,利用邻域算子产生邻域解的过程实质是对站点顺序的调整。调整站点的顺序可以通过点移动或边调整来完成。本文算法使用 SPI、SBR 和 WRI 这 3 个邻域操作算子对路径进行优化,它们对应的操作如图 2—图 4 所示[9]。

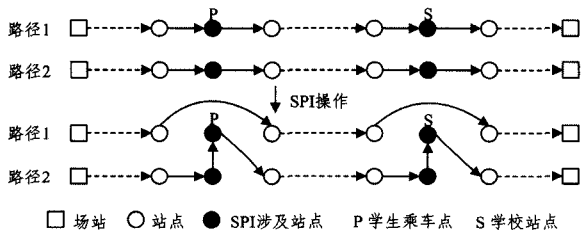


图 2 SPI 操作

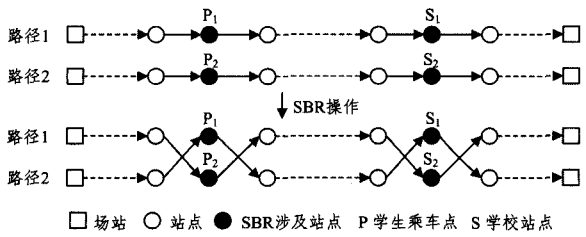


图 3 SBR 操作

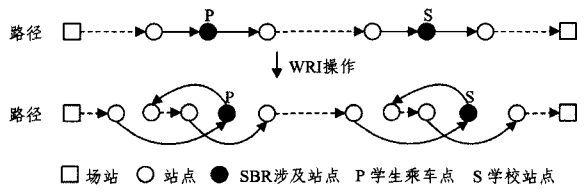


图 4 WRI 操作

SPI 尝试将一个站点对(学生站点,学校站点)从一条路径移到另外一条路径,这种移动能够减少路径数目,从而最大限度地提高目标函数。SBR 对现有两个不同路径上的站点对进行交换。通过一段时间的 SPI 移动达到局部最优,无法再找到能够改进目标函数的 SPI 移动。而 SBR 通过交换点对改变搜索策略来发现新的可行解。WRI 是通过在路径内调整学生站点和学校站点来优化单条路径,站点调整可以单独进行也可以成对进行,但调整必须在满足各种约束的前提下进行。虽然 WRI 不能减少路径,但 WRI 通过改变路径上的站点顺序可以使单条路径达到最优,为继续进行 SPI、SBR 等操作提供可能。

3.4 基于时空相关度的邻域搜索

用三元组 (u,d,r) 表示当前解 x 中待移动的站点对,其中 u 为学生乘车站点号,d 为其对应的学校站点号,r 为点对所在的路径编号。用三元组 (i,j,k) 指定移动站点对时 u 和 d 对应的新位置,即将 u 和 d 从路径 r 移动到路径 k 上,分别插入到站点 i 和站点 j 之后。执行移动后,站点 i 和站点 u 则直接相连构成了路径上的一条路段。以 SPI 算子为例,站点对 (u,d,r) 的移动空间 $M=\{(i,j,k)\mid i\in k,j\in k,k!=r\}$,即将原站点对 (u,d,r) 移动到除路径 r 之外的任意一条路径 k 上的两个位置。当前解 x 中所有可移动站点对的移动空间构成了 SPI 的邻域搜索空间。

在构造邻域搜索空间时,使用基于时空相关度排序的邻接列表,按照表中的顺序构造一个站点对的移动空间。这样便可以通过限定邻接表长度减小邻域搜索空间。基于时空相关度邻域搜索空间构造算法如算法 1 所示。

算法 1 基于时空相关度构造邻域搜索空间

```
输入:u,d,r,neighborhoodlist
输出:Searchspace 数组
for(int i=0;i<neigborhoodlist. len;i++){
    r1=getroutenumber(neigborhoodlist[i]);
    if(r1==r) continue;
    pos1=route[r1]. firststop;
    while(pos1 != NULL){
        searchspace[index]. i=i;
        searchspace[index]. j=pos1;
        searchspace[index++]. k=r1;
        pos1=next(pos1);
    }
    SearchSpace_Deopt(u,d,r);
    SearchSpace_School(u,d,r);
}
```

算法 1 基于站点的邻接表来构造。neighborhoodlist 是经过排序的一个列表,相关度较高的站点位置靠前,因此可以通过限定邻接表长度来缩减邻域搜索规模。函数 SearchSpace_Deopt 和 SearchSpace_School 是针对场站和学校单独构造搜索空间,因为与这两类站点相连比较容易满足约束条件。

4 实验结果及分析

4.1 实验数据及环境

为测试本文算法的有效性,采用国际上已有测试案例运行程序。数据集来自文献[7],是专门针对混载 SBRP 设计的案例库,每个案例在长宽均为 32.2km(20 英里)的范围内生成了若干学校和学生站点,其中最大规模的案例包含 100 个学校和 2000 个站点。案例分为 RSRB 和 CSCB 两种类型,RSRB 类测试案例中学校和乘车站点分布是随机的,CSCB 类测试案例中学校和乘车点则是相对集中地分布在若干个积聚区。为了便于对比,问题约束的设置与文献[7]一致,校车容量限制为 66,最大乘车时间设置为 2700s(45min)。车辆的平均行驶速度为 32.2km/h(20 英里/小时)。各个站点的服务时间[5,7]为:

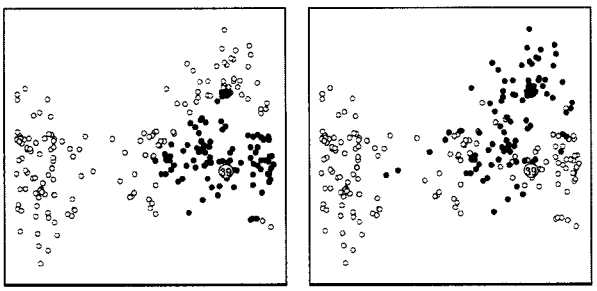
$$t_i=\begin{cases}19.0+2.6n_i, & \text{站点 } i \text{ 为乘车站点}\\ 29.0+1.9n_i, & \text{站点 } i \text{ 为学校}\end{cases}$$

n_i 为在站点 i 上车或下车的学生数,t_i 的单位为 s。

实验是在安装了 Win7 操作系统的个人计算机上进行的,CPU 为 intel core i7-3400,内存为 2G。算法采用 Visual Studio 2010 下的 C++实现。程序运行时循环次数为 30,待移动站点的选择策略为短路径优先。

4.2 基于时空搜索的优势

为了说明基于时空相关度对邻接表排序的优点，将之与基于空间距离的排序方式进行对比。图5给出了RSRB01案例中站点39按照不同方式对邻接表进行排序时的情况。以空间关系(曼哈顿距离)划定100个邻接站点时，黑色的邻接站点分布在以站点39为中心的一个区域内(见图5(a))。以时空距离划定的邻接站点(黑色)分布不是规则的，包含了一些相对较远的站点，但排除了一些空间距离较近的点。这是因为某些站点虽然与选定站点之间的距离近，但是时间窗相差较大，无法直接连接，或者其他约束直接将之排除在邻接关系之外(见图5(b))。



(a)空间邻接　　(b)时空邻接

图5　RSRB01案例站点39对应的100个邻接点

在不限定邻接表长度的情况下，记录下所有SPI操作产生的(u,d,r)到(i,j,k)的移动。每次移动都将站点u插入到站点i之后，即将站点i和站点u相连。由于没有限定邻接表长度，因此除u之外的所有站点都在邻接表中，这必然包括站点i。按照空间距离和时空距离对u邻接表中的站点进行排序时，i出现的位次是不一样的。对所有SPI移动中的u，统计与之相连的i在其邻接表中按不同排序方式的平均位序，结果如表1所列。可以看出，采用时空距离排序时连接站点的平均位序较小，即可以在较小的邻接表长度内划定连接站点。由于只以距离排序邻接点的平均位序较大，必须将邻接表长度放大才能包含这些可能的邻接站点。因此采用时空距离比单用距离排序更能体现邻接的相关度。

表1　邻接站点平均分布位次统计表

实例	最大乘车时间/s	时空排序	空间排序	实例	最大乘车时间/s	时空排序	空间排序
RSRB01	2700	10.54	14.84	RSRB01	5400	12.73	19.75
RSRB02	2700	5.71	10.84	RSRB02	5400	6.78	11.51
RSRB03	2700	8.48	15.25	RSRB03	5400	10.66	20.34
RSRB04	2700	7.86	18.03	RSRB04	5400	8.94	18.95
RSRB05	2700	11.29	28.38	RSRB05	5400	14.72	34.80
RSRB06	2700	8.24	24.94	RSRB06	5400	11.40	30.33
RSRB07	2700	10.71	43.01	RSRB07	5400	15.53	55.95
RSRB08	2700	10.71	50.30	RSRB08	5400	17.14	56.78
平均	2700	9.19	25.70	平均	5400	12.24	31.05

4.3 基于时空邻域搜索的结果

为测试按照时空相关邻域搜索对结果的影响，针对RSRB和CSCB两种类型站点分布案例，分别在邻接表长度不受限(所有站点均在邻接列表中)和受限(邻接表长度为站点数的20%且不小于100)的情况下运行程序。算法中的时空权重参数α和β均为0.5(实验得到的最优设置)。具体实验结果如表2和表3所列，V_1和T_1为邻接表长度受限时的车辆数和执行时间，V_2和T_2为不受限时的执行结果，V_3和T_3为Park算法[7]的结果。

表2　RSRB类案例实验结果

实例	最大乘车时间/s	车辆数			执行时间/s		
		20%邻接表	全邻接表	Park	20%邻接表	全邻接表	Park
RSRB01	2700	27	27	30	14.94	20.45	0.9
RSRB02	2700	27	27	29	23.88	30.07	0.9
RSRB03	2700	53	53	56	81.03	135.13	3.2
RSRB04	2700	53	53	59	115.57	168.73	2.2
RSRB05	2700	92	92	98	477.98	863.96	5.0
RSRB06	2700	78	78	89	724.79	1176.85	5.8
RSRB07	2700	141	141	154	3044.05	6722.54	40.7
RSRB08	2700	144	144	157	3919.83	7694.12	18.2
RSRB01	5400	25	25	27	16.36	26.21	0.9
RSRB02	5400	23	23	23	30.11	40.83	0.7
RSRB03	5400	47	47	47	81.26	172.32	3.4
RSRB04	5400	41	41	46	124.27	234.02	2.1
RSRB05	5400	75	75	79	457.88	1102.36	5.6
RSRB06	5400	63	62	72	717.26	1516.45	5.8
RSRB07	5400	124	124	134	2800.53	7912.88	43.1
RSRB08	5400	120	120	142	3733.02	9221.09	26.4
平均	—	70.81	70.75	77.63	1022.67	2314.88	10.3

表3　CSCB类案例实验结果

实例	最大乘车时间/s	车辆数			执行时间/s		
		20%邻接表	全邻接表	Park	20%邻接表	全邻接表	Park
CSCB01	2700	27	27	30	19.17	30.11	1.0
CSCB02	2700	26	26	30	33.34	41.57	0.9
CSCB03	2700	49	49	55	145.83	235.13	4.4
CSCB04	2700	59	59	62	131.68	188.13	1.7
CSCB05	2700	101	101	116	637.95	1075.73	10.6
CSCB06	2700	106	106	120	773.86	1200.89	10.4
CSCB07	2700	162	162	175	4949.12	8649.36	12.0
CSCB08	2700	156	156	175	5027.52	8773.59	24.1
CSCB01	5400	23	23	24	21.31	35.45	1.1
CSCB02	5400	20	20	22	28.85	45.70	0.8
CSCB03	5400	40	40	41	119.80	277.33	3.9
CSCB04	5400	38	38	43	144.89	274.12	1.7
CSCB05	5400	88	88	97	523.99	1277.65	9.9
CSCB06	5400	91	91	102	700.88	1460.23	13.2
CSCB07	5400	125	125	133	3967.63	10165.92	16.7
CSCB08	5400	127	128	141	4687.79	10633.21	27.8
平均	—	77.38	77.44	85.38	1369.60	2772.76	8.8

从表2可以看出，针对RSRB案例采用时空距离进行邻域搜索，取邻接表长度为站点数的20%(且不小于100)时，执行的结果仅有RSRB06在最大乘车时间为5400s时车辆数不同，其他的案例与不限邻接表长度完全一样，时间上平均节省了55.82%。表3显示在CSCB类型案例下，仅在最大乘车时间为5400s时，CSCB08案例与不限邻接表长度不同，总体上CSCB案例在求解时间上平均节省了50.61%。分析表2和表3中求解结果不同的案例可知，其原因在于当站点规模较大时，邻接表长度变化导致邻域搜索顺序改变，从而在有限次固定搜索顺序的情况下，会对结果产生一定的影响。

表2和表3同时给出了Park算法[7]的车辆数和求解时间。Park算法强调的是快速求解，具有很强的时间优势，但求解结果与本文设计的基于邻域搜索的元启发算法(不限邻接表长度)平均相差9.08%。

4.4 邻域表长度对结果的影响

基于时空邻域搜索算法的核心思想即为改变站点邻接表的排列顺序，从而通过限定邻接表的长度来减小搜索空间，提高算法的执行速度。从4.2节的分析可以看出，基于时空邻

接度的排序更能在有限的列表长度内包含最终选择的站点。为此，基于案例测试了不同邻接表长度对最终求解结果的影响。测试时设定邻接表长度分别为站点数的15%、20%、30%、40%和100%，但至少保证不少于100个邻接站点。

实验结果如表4、表5所列。从表4可以看出，与不限定邻接表长度(100%)的情况相比，在40%、30%、20%的情况下，平均执行时间显著下降；但从20%向下再继续缩减邻接表长度，执行时间下降得并不明显，并且开始对求解结果产生影响(见表5)。因此根据实验结果，将邻接表长度限定在20%的范围内是比较合理的，既能保证求解质量，又能显著地减少求解时间。

表4 邻接表长度对求解时间的影响

案例	最大乘车时间/s	不同邻接表长度下多个案例的平均求解时间/s				
		15%	20%	30%	40%	100%
RSRB01-RSRB08	2700	1029.48	1050.26	1124.79	1205.99	2101.48
RSRB01-RSRB08	5400	936.36	995.09	1213.63	1405.72	2528.27
CSCB01-CSCB08	2700	1437.90	1464.81	1568.88	1656.23	2524.31
CSCB01-CSCB08	5400	1214.73	1274.39	1509.35	1748.30	3021.20

表5 邻接表长度对车辆数的影响

案例	最大乘车时间/s	不同邻接表长度下多个案例的平均车辆数				
		15%	20%	30%	40%	100%
RSRB01-RSRB08	2700	76.88	76.88	76.88	76.88	76.88
RSRB01-RSRB08	5400	64.63	64.75	64.63	64.63	64.63
CSCB01-CSCB08	2700	86.00	85.75	85.75	85.75	85.75
CSCB01-CSCB08	5400	68.75	69.00	69.00	69.13	69.13

表5中，当邻接表长度从100%下降到30%时车辆数都没有增加，甚至在CSCB(5400)的情况下还有下降。再继续缩小邻接表长度到20%时，RSRB(5400)出现了偶然性的增加。当邻接表长度下降到15%时，平均车辆数变化不大，但是具体到案例，车辆数有增有减，对结果产生了一定的影响。这验证了基于时空邻接度排序邻接表能够在有限的长度下，尽可能地包含更多最终真正执行移动的站点，但邻接表过短时，它会限制移动站点的选择。通过对邻域搜索过程的监视，发现个别案例出现邻接表缩短时车辆数更少的情况，这主要是因为限定了一些站点之后，导致邻域搜索的轨迹发生了变化，出现了更有利于缩减路径数的特例。

结束语 与传统构造式启发算法相比，基于邻域算子优化混载SBRP能显著提高解的质量，但大规模的邻域搜索增加了问题的求解时间。为解决邻域搜索耗时的问题，本文设计了一种基于时空相关的邻域搜索算法。该算法基于站点的邻接表构造邻域搜索空间，综合考虑了站点间的空间距离、时间距离，同时进行了简单约束的预处理。基于时空相关度对邻接表排序使得最终接受的邻接解通常位于搜索空间中比较靠前的位置，因此适当限定邻接表长度并不会对求解质量产生大的影响。实验表明，将邻接表长度限定为站点数20%(同时不小于100)的情况下，能在基本保证求解质量的同时节省50%以上的求解时间。

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] Bodin L D, Berman L. Routing and scheduling of school buses by computer[J]. Transportation Science, 1979, 13(2): 113-129

[3] Park J, Kim B-I. The school bus routing problem: A review[J]. European Journal of Operational Research, 2010, 202(2): 311-319

[4] Bodin L D, Golden B, Assad A, et al. Routing and scheduling of vehicles and crews: the state of the art[J]. Computers and Operations Research, 1983, 10(2): 63-211

[5] 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

[6] Vidal d S L, Siqueira P H. Heuristic Methods Applied to the Optimization School Bus Transportation Routes-A Real Case[C]// IEA/AIE 2010, Part II. Berlin: Springer Verlag, 2010: 247-256

[7] 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

[8] Nanry W, Barnes J. Solving the pickup and delivery problem with time windows using reactive tabu search[J]. Transportation Research Part B, 2000, 34(2): 107-21

[9] 党兰学，王震，刘青松，等. 一种求解混载校车路径的启发式算法[J]. 计算机科学，2013，40(7)：248-253

[10] Fang H, Kilani Y, Lee J H M, et al. Reducing search space in local search for constraint satisfaction[C]// AAAI/IAAI. 2002: 28-33

[11] Chen S Y, Smith S F. Improving Genetic Algorithms by Search Space Reductions (with Applications to Flow Shop Scheduling)[C]//GECCO. 1999: 135-140

[12] 戚铭尧，丁国祥，周游，等. 一种基于时空距离的带时间窗车辆路径问题算法[J]. 交通运输系统工程与信息，2011，11(1)：85-89

(上接第193页)

[13] Riedmiller M, Gabel T, Hafner R. Reinforcement Learning for Robot Soccer[J]. Autonomous Robots, 2009, 27(1): 55-73

[14] Gabel T, Riedmiller M. On Progress in RoboCup: the Simulation League Showcase in RoboCup 2010: Robot Soccer World Cup XIV[M]. Berlin: Springer Verlag, 2011: 36-47

[15] Kalyanakrishnan S, Stone P. Characterizing Reinforcement Learning Methods through Parameterized Learning Problems[J]. Machine Learning, 2011, 84(1/2): 205-247

[16] Sherstov A A, Stone P. Function Approximation via Tile Coding: Automating Parameter Choice in Abstaction, Reformulation and Approximation [M]. Berlin: Springer Verlag, 2005: 194-205

[17] Stone P, Sutton R S. Scaling Reinforcement Learning toward RoboCup Soccer[C]// the Eighteenth International Conference on Machine Learning. Massachusetts: Williamstown, 2001: 537-544

[18] 程显毅，朱倩. 一种改进的强化学习方法在RoboCup中的应用[J]. 广西师范大学学报：自然科学版，2010，28(3)：99-102

[19] 刘春阳，谭应清，柳长安. 多智能体强化学习在足球机器人中的研究与应用[J]. 电子学报，2010，38(8)：1958-1962

[20] 李瑾，刘全，杨旭东. 一种改进的平均奖赏强化学习方法在RoboCup训练中的应用[J]. 苏州大学学报，2012，28(2)：21-26