

# 基于变邻域下降的列车运行时刻调整算法<sup>①</sup>



左义盟

(重庆交通大学 信息科学与工程学院, 重庆 400074)

通讯作者: 左义盟, E-mail: yanuoymeng@163.com

**摘要:** 高速铁路区间失效可能严重影响列车正常运行, 区间失效后调度员需要及时调整列车运行时刻. 本文主要针对区间失效后不改变列车顺序下的时刻调整问题进行研究, 建立了以所有列车在各站晚点时间之和为目标的列车运行调整模型, 模型中各约束条件保证列车安全运行. 针对目前常见的获取最优解或次优解需花费较长时间的问题, 提出一种基于变邻域下降算法的多阶段变邻域下降算法. 算法的第一、第二阶段使用变邻域下降算法结合禁忌表快速确定哪些列车经过调整后的时刻能与图定时刻相等, 第三阶段则调整未恢复到图定时刻的列车. 最后, 以西成客运专线与某日的列车时刻数据为算例, 求解多种区间失效场景下的列车运行时刻调整方案验证算法的有效性与实时性.

**关键词:** 区间失效; 列车运行调整; 变邻域下降; 禁忌表

引用格式: 左义盟. 基于变邻域下降的列车运行时刻调整算法. 计算机系统应用, 2020, 29(10): 167-172. <http://www.c-s-a.org.cn/1003-3254/7506.html>

## Train Rescheduling Algorithm Based on Variable Neighborhood Descent

ZUO Yi-Meng

(School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China)

**Abstract:** The blockage of the high-speed railway segment may seriously affect the original timetable of the train. If it happens, the dispatcher needs to adjust the timetable. This study focuses on how to reschedule the timetable without changing the sequence of trains that are affected by the segment blockage. A train rescheduling model is formulated with aiming at minimizing the sum of the delay time of all trains in each station, and all constraints of the model is used to ensure the safety of trains. And a multi-stage variable neighborhood descent algorithm is proposed to solve the problem that it takes a long time to get the optimal or suboptimal solution. In the first and second stages of the algorithm, variable neighborhood descent combined with tabu table is used to quickly ascertain the trains which can maintain the original timetable after several adjustments, and then adjust other trains in the third stage. Finally, taking Xi'an-Chengdu Passenger Dedicated Line and the timetable of a single day as an example, the validity and real-time performance of the algorithm is verified by getting the timetable adjusted under various interval failure scenarios.

**Key words:** segment blockage; train rescheduling; variable neighborhood descent; tabu table

在铁路系统中, 经常有各类异常事件影响列车正常运行. 以异常事件发生的位置分类, 可以把它们分为发生在区间的异常事件与发生在车站的异常事件<sup>[1]</sup>. 这两类异常事件中发生在区间上发生的异常事件造成的

影响往往比在车站发生的异常事件更大, 而在发生在区间的异常事件中, 又以区间失效造成的影响更大. 为了避免异常事件带来更坏的影响, 调度员需及时做出最优或次优的调整. 调整涉及到的内容有列车时刻

① 收稿时间: 2019-12-16; 修改时间: 2020-01-14; 采用时间: 2020-01-21; csa 在线出版时间: 2020-09-30

表、乘务计划与列车交路计划,其中列车时刻调整是首要调整任务<sup>[2]</sup>。

区间失效后的列车时刻调整问题及相关研究中使用的方法主要有启发式算法、分支定界法、元启发式算法与专业数学优化软件等,其中启发式算法与分支定界法是最常使用的两种方法<sup>[3]</sup>。Josyula 等人把解空间表示为树,并提出以深度优先的搜索策略为基础的顺序启发算法来快速遍历树<sup>[4]</sup>。兰泽康等人提出使用组合列生成算法与分支定界法的分支定价法求解基于列车路径的整数规划模型<sup>[5]</sup>。Zhou 等人分别使用进化算法与混沌萤火虫算法获得列车时刻调整方案<sup>[6]</sup>。邓念和占曙光等人主要研究了高速铁路列车运行实时调整问题,然后直接使用 CPLEX 求解其构建的模型<sup>[7,8]</sup>; Hou 等研究地铁系统发生异常状况时如何调整列车,同样使用 CPLEX 求解<sup>[9]</sup>。彭其渊等人则指出随着列车和车站数的增加,模型求解呈指数型增长,直接求解十分困难,因此他们利用二阶段启发式算法结合 CPLEX 求解<sup>[10]</sup>。这些经典的方法存在的问题是面对较大规模问题时,求解时间较长,算法的实时性较低。为了提高算法的实时性,引入变邻域下降算法 (Variable Neighborhood Descend, VND) 快速解决区间失效后的列车调整问题。

VND 是由变邻域搜索算法 (Variable Neighborhood Search, VNS) 变化而来的一个算法框架,它以某种方式搜索解的邻域。Shakibayifar 与 Sheikholeslami 使用多目标变邻域搜索算法寻找列车调整问题的帕累托最优集<sup>[11]</sup>,本文则利用 VND 的局部搜索能力,并定义适用于列车时刻调整的邻域结构,结合禁忌表提出多阶段变邻域下降算法 (Multi Stage Variable Neighborhood Descend, MSVND),高效地获得不改变列车顺序时的调整后的列车运行时刻。

### 1 高速铁路区间失效的列车时刻调整模型

双线高速铁路上的列车分向运行,两个运行方向上的列车互不干扰。为了简便起见,本文研究双线高速铁路某一个方向上区间失效后的列车时刻调整问题,线路上除了全程运行的列车以外,还有只在线路上部分区间运行的列车。

#### 1.1 模型假设

- (1) 区间失效之前的所有列车活动都正常进行;
- (2) 一旦区间失效,失效持续时间可以预测;

- (3) 刚好进入失效区间的列车在区间等待区间恢复;
- (4) 时间的最小单位为分钟。

#### 1.2 符号说明

某高速铁路的运行指定方向列车的轨道与车站组成有向图  $GA = (S, R)$ , 其中  $S = \{s_1, s_2, \dots, s_N\}$  是按指定方向排序的车站集合,  $R = \{(s_1, s_2), (s_2, s_3) \dots, (s_{N-1}, s_N)\}$  是轨道区间的集合,  $(s_i, s_{i+1})$  表示车站  $s_i$  至车站  $s_{i+1}$  的轨道区间,  $N$  是线路上的办理客运业务的车站数量。

在  $GA$  上运行的列车集合为  $T = \{t_1, t_2, \dots, t_M\}$ , 其中  $M$  是列车的数量。  $T$  中的任意一列列车  $t$  在  $GA$  上运行的第一个车站与最后一个车站分别为  $s_t^f$  与  $s_t^l$ , 其图定到发时刻与调整后的到发时刻分别为:

$$O_t = \{o_1^t, o_2^t, o_3^t, o_4^t, \dots, o_{2N-1}^t, o_{2N}^t\},$$

$$X_t = \{x_1^t, x_2^t, x_3^t, x_4^t, \dots, x_{2N-1}^t, x_{2N}^t\}.$$

$O_t$  与  $X_t$  都有  $2N$  个值, 从索引 1 至索引  $2N$  依次为车站  $s_1$  至车站  $s_N$  的到站时刻与出站时刻。如果列车  $t$  不在某车站运行, 那么其对应的到站与出站时刻的值为 0。相应地,  $E = \{e_1, e_2, e_3, e_4, \dots, e_{2N-1}, e_{2N}\}$  依次是列车在车站  $s_1$  至车站  $s_N$  的进站附加时间、出站附加时间;  $G = \{g_1, g_2, g_3, g_4, \dots, g_{2N-1}, g_{2N}\}$  依次是列车在车站  $s_1$  至车站  $s_N$  的最小到站间隔时间、最小出站间隔时间。

其余的参数与变量及其说明如表 1 所示, 其中  $\beta_i^s$ ,  $\lambda_{it}^s$ ,  $\varphi_{it}^s$  与  $Q_{it}^s$  都是变量符号。

表 1 参数与变量符号及说明

符号	说明	符号	说明
$A$	到站时刻的下标集合	$D$	出站时刻的下标集合
$s^+$	按指定方向车站 $s$ 的前方车站	$(s_0, s_0^+)$	失效的区间, 称为区间 $s_0$
$r_1$	区间失效开始时刻	$r_2$	区间失效结束时刻
$h_s$	车站 $s$ 的最小停站时间	$C_s$	车站 $s$ 的到发线数量
$\alpha_i^s$	原时刻列车 $t$ 在车站 $s$ 停车: 真为 1, 假为 0	$\beta_i^s$	调整后列车 $t$ 在车站 $s$ 停车: 真为 1, 假为 0
$\lambda_{it}^s$	列车 $t'$ 是否在列车 $t$ 到达之前到站	$\varphi_{it}^s$	列车 $t'$ 是否在列车 $t$ 到达之前出站
$Q_{it}^s$	列车 $t$ 是否在列车 $t'$ 到达车站 $s$ 之前从车站 $s$ 出发		

### 1.3 模型构建

#### 1.3.1 基本模型

以最小化列车在各车站的总晚点时间为目标, 可以构建列车运行调整的基本模型如下:

$$\min \sum_{i \in D} \sum_{t \in T} (x_i^t - o_i^t) \tag{1}$$

$$\text{s.t. } x_i^t = o_i^t \quad \forall t \in T, \forall t \in D \cup A : x_i^t \leq r_1 \quad (2)$$

$$x_i^t \geq o_i^t \quad \forall t \in T, \forall t \in D : o_i^t \neq o_{i-1}^t \quad (3)$$

$$\begin{aligned} o_{i+1}^t - o_i^t + \theta_i^{s^+} e_{i+1} + \theta_i^{s^-} e_i &\leq x_{i+1}^t - y_i^t \\ s = i \div 2, \forall i \in D, \forall t \in T : s_i^f &\leq s \leq s_i^l \end{aligned} \quad (4)$$

$$\begin{aligned} h_s \beta_i^s &\leq x_{i+1}^t - x_i^t \leq M^+ \beta_i^s \\ s = (i+1) \div 2, \forall i \in A, \forall t \in T : s_i^f &\leq s \leq s_i^l \end{aligned} \quad (5)$$

$$\alpha_i^s \leq \beta_i^s \quad \forall s \in S, \forall t \in T : s_i^f \leq s \leq s_i^l \quad (6)$$

式(1)是列车时刻调整的目标;式(2)表示区间失效之前;列车按照计划时刻运行;式(3)表示列车在计划停站的车站的出站时刻不早于其计划出站时刻;式(4)限定列车的区间运行时分;式(5)限定若列车在车站  $s$  停站,则其停站时间不能小于最小停站时间,若列车在车站  $s$  不停车,则列车的停站时间为0;式(6)表示列车的停站选择约束,在其计划停站的车站必须停站。

### 1.3.2 其他约束

除了单列列车的约束条件以外,还有列车之间因为互相影响而需遵守的一些约束条件。

(1) 出发间隔时间与到达间隔时间约束

$$\begin{cases} |x_i^t - x_i^{t'}| \leq g_i \\ s = [i \div 2], \forall i \in A \cup D, \forall t \in T, t' \in T : t \neq t' \end{cases} \quad (7)$$

式(7)表示任意两列列车在车站的到站与出站间隔时间都满足最小间隔时间约束。

(2) 出发与到达间隔时间约束

$$\begin{cases} lQ_{i'}^s(x_{i+1}^t - x_i^t) + Q_{i'}^s(x_{i+1}^{t'} - x_i^{t'}) \geq (Q_{i'}^s + Q_{i'}^s)G_s^{d,a} \\ \forall s \in S; \forall t, t' \in T : t \neq t', s_i^f \leq s \leq s_i^l, s_{i'}^f \leq s \leq s_{i'}^l \end{cases} \quad (8)$$

式(8)表示两列列车占用车站相同轨道需要满足最小发到间隔时间。

(3) 车站能力约束

$$\begin{cases} l \sum_{t, t' \in T} \lambda_{i'}^s - \sum_{t, t' \in T} \varphi_{i'}^s \leq C_s - 1 \\ \forall s \in S : t \neq t', s_i^f \leq s \leq s_i^l, s_{i'}^f \leq s \leq s_{i'}^l \end{cases} \quad (9)$$

式(9)表示列车  $t$  在到达车站  $s$  的时候,车站  $s$  应有能力为列车  $t$  提供服务。

(4) 列车越行约束

$$\begin{cases} l(x_{i+1}^t - x_{i+1}^{t'})(x_i^t - x_i^{t'}) > 0 \\ \forall i \in D; \forall t, t' \in T : t \neq t', s_i^f \leq s \leq s_i^l, s_{i'}^f \leq s \leq s_{i'}^l \end{cases} \quad (10)$$

式(10)表示列车  $t$  与  $t'$  在某站的出站时刻之差与在前一车站的到站时刻之差应同正负,其意义是列车不能在区间越行。

## 2 多阶段变邻域下降算法

### 2.1 初始解

在获得初始解之前,根据列车的所有图定时刻是否在前一车站的到站时刻之差应同正负,其意义是列车不能在区间越行。在获得初始解之前,根据列车的所有图定时刻是否在前一车站的到站时刻之差应同正负,其意义是列车不能在区间越行。在需调整的列车集合  $T_1$  中有两类车直接导致线路上的列车受到影响,一类是在  $r_1$  及之前的某时刻进入区间  $(s_0, s_0^+)$  且未能在  $r_2$  前离开的列车,另一类是在时间段  $(r_1, r_2)$  到达车站  $s_0$  的列车,分别如图1中实斜线与虚斜线所示,并令它们分别组成集合  $T_0$  与  $T_0'$ 。如果没有上述两类列车,所有的列车能够按照各自的图定时刻运行。如果有这两类列车的任意一种,则需要调整列车集合  $T_1$  中列车的时刻,并且需调整的时刻是大于  $r_1$  的时刻。令  $(I_t, J_t)$  为列车集合  $T_1$  中任意一列列车  $t$  的图定到发时刻  $O_t = \{o_1^t, o_2^t, o_3^t, o_4^t, \dots, o_{2N-1}^t, o_{2N}^t\}$  中大于  $r_1$  的时刻的起止索引,  $I_t$  与  $J_t$  分别是大于  $r_1$  的时刻的最小索引与最大索引。例如某列车在车站  $s_0$  与车站  $s_N$  之间的所有时刻都需调整,它的起止索引为  $(1, 2N)$ 。

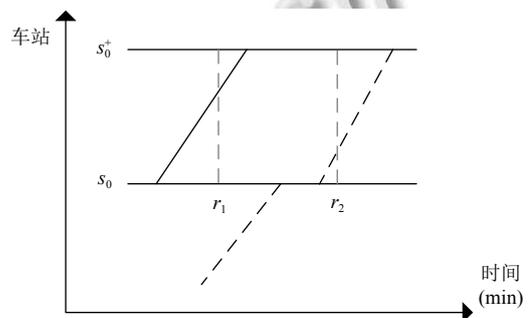


图1 直接导致线路上列车受影响的两类列车

本文使用右移法获得算法的初始解,该方法把图定的大于  $r_1$  的所有时刻加上某时间长度,相当于大于  $r_1$  的所有时刻在列车运行图上整体向右移动了某时间长度。假设集合  $T_0$  中的列车到达车站  $s_0^+$  的时刻最早为  $r_1$ ,集合  $T_0'$  中的列车从  $s_0$  出站的时刻最早为  $r_1$ ,则右移的时间长度如下:

$$\max(\max(r_2 - o_{2*s_0^+ - 1}^{t_1}), \max(r_2 - o_{2*s_0}^{t_2})) \quad t_1 \in T_0, t_2 \in T_0'$$

## 2.2 多阶段变邻域下降算法

### 2.2.1 第1阶段变邻域下降算法

#### (1) 邻域结构

为列车集合 $T_1$ 中的每列列车 $t$ 定义一个与其图定时刻 $O_t$ 等长的整体平移邻域结构 $N_t$ ,在其起止索引( $I_t, J_t$ )范围内的值为-1,其余值为0,例如某列车从车站 $s_0$ 到车站 $s_N$ 的全部时刻都需调整,其邻域结构为 $\{-1, -1, -1, -1, \dots, -1, -1\}$ ,其意义是列车 $t$ 所有需调整的时刻在列车运行图中都整体向左平移1分钟.

因为运行时刻在前的列车影响运行时刻在后的列车、运行位置在前方的列车影响运行位置在后方的列车,优先平移运行时刻与运行位置在前方的列车可以加快邻域搜索.因此列车集合 $T_1$ 对应的整体平移邻域结构集合 $N_{T_1}$ 中邻域结构按照列车的起始索引与索引对应的时刻排序.

#### (2) 禁忌表

在对列车集合 $T_1$ 中的列车进行邻域搜索的过程中,当某列车恢复到其图定时刻时,将该列车加入到禁忌表中,在禁忌表中的列车不再进行邻域搜索.

#### (3) 约束条件

列车图定时刻 $O_t$ 隐式地包含了列车安全运行的约束条件,在列车时刻整体平移的过程中不需要检验邻域中的解是否满足模型中的所有约束条件,只需遵守式(3),式(7),式(9)就能保证列车安全运行.

基于前述邻域结构、禁忌表与约束条件设计的第1阶段多邻域下降算法的伪代码如算法1.

算法1. 第1阶段多邻域下降算法

输入:  $X=(X_{t_1}, X_{t_2}, \dots, X_{t_{M_1}})$ ,  $N_{T_1}=(N_{t_1}, N_{t_2}, \dots, N_{t_{M_1}})$ ,

```

1 Tabu: =  $\emptyset$ 
2 repeat
3   for  $k=1$  to  $M_1$ 
4     change = True
5     while change do
6        $X_{t_k}' := N_{t_k}(X_{t_k})$ 
7       if  $X_{t_k}'$  满足各种约束条件 then
8          $X_{t_k} := X_{t_k}'$ 
9       else then
10        change = False
11      end
12      if  $X_{t_k} = O_{t_k}$  then
13        Tabu: = Tabu  $\cup$   $t_k$ 
14      end
15    end
16  $M_1$ : = 在列车集合 $T_1$ 中但不在禁忌表 Tabu 中的列车数量;
```

17 until 某列车在第1次没有更优解与第2次没有更优解之间其他列车也没有更优解.

算法1中步骤2至17调整 $T_1$ 中所有列车直至它们在对应的邻域结构中没有更优解.其中步骤3至13按照指定规则依次调整 $T_1$ 的一列列车,步骤5至11循环调整列车 $t_k$ ,如果列车的新时刻 $X_{t_k}'$ 能满足约束条件就一直搜索其邻域,否则就跳出循环;步骤12至14表示列车 $t_k$ 调整后的时刻与原计划时刻相等时,将列车加入到禁忌表中.

### 2.2.2 第2阶段变邻域下降算法

第2阶段的算法流程与第1阶段一致,区别只是第2阶段调整区间失效后仍在车站 $s_1$ 与 $s_0$ 之间运行的列车.假设这些列车组成集合 $T_2'$ ,令列车集合 $T_2'$ 中的列车调整的起止索引为 $(I_t^2, J_t^2)$ ,其中 $I_t^2 = I_t$ ,  $J_t^2 = \min(2 * s_0 - 1, J_t)$ .

#### (1) 邻域结构

为列车集合 $T_2'$ 中的每列列车 $t$ 定义一个与 $O_t$ 等长的部分平移邻域结构,在索引范围 $(I_t^2, J_t^2)$ 内的值为-1,其余则为0.

#### (2) 禁忌表

如果 $T_2'$ 中的列车 $t$ 在其调整的起止索引范围 $(I_t^2, J_t^2)$ 内恢复图定时刻,将列车 $t$ 加入到第2阶段禁忌表中,在第2阶段邻域搜索中不再搜索该列车.

#### (3) 约束条件

第2阶段变邻域下降算法需要遵守式(3),式(5),式(7)~式(9)表示的约束条件.

### 2.2.3 第3阶段变邻域下降算法

第3阶段调整的列车是 $T_2$ 中不在第2阶段禁忌表中的列车或在禁忌表中而其终止索引 $J_t \geq 2 * s_0 - 2$ 的列车,假设它们组成集合 $T_3$ , $T_3$ 中列车数量为 $M_3$ ,其中的列车 $t$ 需要调整时刻的起止索引为 $(I_t^3, J_t^3)$ .如果列车 $t$ 在第2阶段禁忌表中而其终止索引 $J_t \geq 2 * s_0 - 1$ ,那么列车 $t$ 的起止索引更新为 $(2 * s_0 - 1, J_t)$ ,其余列车的起止索引不变.

#### (1) 邻域结构

为列车集合 $T_3$ 中的每列列车 $t$ 定义一系列与 $O_t$ 等长的邻域结构.如果列车 $t$ 的起始索引 $I_t^3$ 对应的是到达时刻,为该列车定义一个到达平移邻域结构,该索引 $I_t^3$ 对应的值为-1,其余值为0;然后定义 $(J_t^3 - I_t^3 - 1) \div 2$ 个发到平移邻域结构,这些邻域结构分

别在索引 $(I_t^3 + 1, I_t^3 + 2)$ 、 $(I_t^3 + 2, I_t^3 + 3)$ 等对应位置取值为-1, 其余取0. 如果列车 $t$ 的起始索引 $I_t^3$ 对应的是出发时刻, 则直接定义 $(J_t^3 - I_t^3) \div 2$ 个发到平移邻域结构, 它们分别在索引 $(I_t^3, I_t^3 + 1)$ 、 $(I_t^3 + 2, I_t^3 + 3)$ 等对应位置取值为-1, 其余取值为0. 因此列车集合 $T_3$ 中的列车 $t$ 具有一个邻域结构集合 $\mathcal{N}_t$ , 集合中至少有一个邻域结构.

### (2) 约束条件

第3阶段变邻域下降算法需要遵守式(3), 式(5), 式(7)–式(9)表示的约束条件.

第3阶段变邻域下降算法的伪代码如算法2.

#### 算法2. 第3阶段多邻域下降算法

输入:  $X=(X_{t_1}, X_{t_2}, \dots, X_{t_{M_3}})$ ,  $\mathcal{N}_T=(\mathcal{N}_{t_1}, \mathcal{N}_{t_2}, \dots, \mathcal{N}_{t_{M_3}})$

1  $k:=\min(I_{t_1}^3, I_{t_2}^3, \dots, I_{t_{M_3}}^3)$ ,  $t_1, t_2, \dots, t_{M_3} \in T$

2 **repeat**

3 **for**  $s=[(k+1) \div 2]$  **to**  $S_N$

4  $T_3^s:=T_3$ 中索引 $I_t^3 \leq 2 \times s \leq I_t^3$ 的列车组成的集合

5  $M_3^s:=T_3^s$ 中元素个数

6 **for**  $m=1$  **to**  $M_3^s$

7 **change** = **True**

8 **while** **change** **do**

9  $X_{t_m}' := \mathcal{N}_{t_m}^s(X_{t_m})$ ,  $t_m \in T_3^s$ ,

$\mathcal{N}_{t_m}^s$ 为 $\mathcal{N}_{t_m}$ 中车站 $s$ 对应的邻域结构

10 **if**  $X_{t_m}'$ 满足各种约束条件 **then**

11  $X_{t_m} := X_{t_m}'$

12 **else** **then**

13 **change** = **False**

14 **end**

15 **end**

16 **end**

17 **until** 在列车集合 $T_3$ 中的任意列车的任意邻域中搜索都不能获得更优解.

### 2.2.4 调整阶段

本阶段针对列车 $t$ 在某车站 $s$ 原计划不停车而调整后到达时刻与出发时刻不同的情况. 若满足条件的列车 $t$ 在车站 $s$ 的出发时刻与到达时刻之差大于等于 $h_s + e_{s \times 2} + e_{s \times 2 - 1}$ 且满足其他约束条件, 令其在车站 $s$ 的停站计划 $\beta_t^s$ 为1; 否则, 令到达时刻与到达时刻等于它们两个时刻之间最小的且满足约束条件的时刻.

## 3 实例分析

以西成客运专线上行方向为例, 根据2019年7月某日上线运行的列车进行实例分析. 算法基于Python3.6实现, 所有实验都使用CPU为Intel(R) Core(TM) i7-

7500U 2.70 GHz, 内存为8 GB的计算机完成.

### 3.1 数据准备

西成客运专线上行方向办理客运业务的车站共有22个, 车站将线路划分为21个区间. 当日上午运营的列车有89列, 其中44列全程运行, 其余45列只在西成客运专线某段运行.

算例的其他参数按如下规则设置. 两列列车同一车站的到达间隔时间与出发间隔时间都为2分钟, 车站使用相同到发线的发到间隔约束时间为3分钟. 青川站的起停附加时间为3分钟, 其余车站的起停附加时间为2分钟.

### 3.2 实验及对比分析

实验设置4种基本失效场景, 4种场景由两类失效区间(绵阳至青莲, 宁强南至汉中)与两类失效开始时刻(10:00, 16:00)两两组合而成, 每种失效场景下又分别对应20, 60, 120分钟3类失效持续时间. 此外, 使用CPLEX获得上述多种失效场景下的列车停止结果并进行对比分析. CPLEX求解时间设置为3小时, 其余保持默认设置, 不包含约束条件(8). 由MSVND与CPLEX获得的列车调整结果如表2所示.

由表2可以看出, MSVND在获得列车调整结果上展示了优秀的效率. 相同失效区间与失效开始时刻下, 计算时间随着失效持续时间增加而增加. 在所有失效场景中(宁强南至汉中, 10:00, 120)的计算时间最长, 共20.765 s; 失效持续时间为20 s的场景的最长计算时间是3.322 s. 此外, 从表2可以看出第一阶段计算时间占比都大于60%, 且失效持续时间为20与60分钟时, 该占比大于80%.

由表2中目标值与CPLEX求解目标值进行比较可以看出, M-S VND得到的列车运行时调整结果与CPLEX得到的结果差别较小. 在计算时间方面, CPLEX计算时间只有(宁强南至汉中, 16:00, 20)的计算时间小于3小时, 其余场景在使用CPLEX求解时在3小时内都未完成求解. 与MSVND相比, CPLEX求解耗费的计算时间太多, 反应了MSVND在计算时间方面优秀的性能.

## 4 结论与展望

将多阶段变邻域下降算法应用到区间失效后的列车运行时刻调整问题, 实验证明算法调整的结果与全局最优结果相比没有差距, 同时其计算时间又有显著

的提升,表明了算法在列车运行时刻调整问题中的有效性与实时性. 第一阶段快速确定哪些列车恢复到图

定时刻,可以结合其他方法对未恢复到图定时刻的列车进行调整,减少其他方法直接调整的范围.

表2 列车运行时刻调整结果

场景	(失效区间,开始时刻,持续时间(min))	目标值(min)	第一阶段计算时间(s)	总计算时间(s)	第一阶段计算时间占比(%)	CPLEX求解目标值(min)	CPLEX计算时间
1	(绵阳至青莲,10:00,20)	548	2.416	2.547	94.86	533	—
	(绵阳至青莲,10:00,60)	5092	7.429	8.542	86.97	3793	—
	(绵阳至青莲,10:00,120)	17272	13.256	17.268	76.77	14964	—
2	(绵阳至青莲,16:00,20)	152	0.473	0.514	92.02	192	—
	(绵阳至青莲,16:00,60)	3232	1.824	2.152	84.76	2666	—
	(绵阳至青莲,16:00,120)	12000	2.764	4.397	62.86	11107	—
3	(宁强南至汉中,10:00,20)	477	3.322	4.397	75.55	437	—
	(宁强南至汉中,10:00,60)	2921	10.042	10.614	94.61	2277	—
	(宁强南至汉中,10:00,120)	10893	17.270	20.765	83.17	8841	—
4	(宁强南至汉中,16:00,20)	241	0.764	0.777	98.33	270	1 h 06 min
	(宁强南至汉中,16:00,60)	1919	2.467	2.743	89.94	1855	—
	(宁强南至汉中,16:00,120)	7505	3.986	5.918	67.35	7072	—

## 参考文献

- 占曙光. 区间通过能力临时失效下高速铁路列车运行实时调整模型与算法研究 [博士学位论文]. 成都: 西南交通大学, 2016.
- Cacchiani V, Huisman D, Kidd M, *et al.* An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, 2014, 63: 15–37. [doi: 10.1016/j.trb.2014.01.009]
- Fang W, Yang SX, Yao X. A survey on problem models and solution approaches to rescheduling in railway networks. *IEEE Transactions on Intelligent Transportation Systems*, 2015, 16(6): 2997–3016. [doi: 10.1109/TITS.2015.2446985]
- Josyula SP, Krasemann JT, Lundberg L. A parallel algorithm for train rescheduling. *Transportation Research Part C: Emerging Technologies*, 2018, 95: 545–569. [doi: 10.1016/j.trc.2018.07.003]
- 兰泽康, 何世伟, 黎浩东. 铁路网络列车运行调整的优化模型及其分支定价算法. *交通运输系统工程与信息*, 2018, 18(1): 179–185.
- Zhou XZ, Xu W, Zhang Q, *et al.* Application research for high-speed railway traffic intelligent rescheduling based on

- chaotic firefly algorithm. *Proceedings of the 2018 33rd Youth Academic Annual Conference of Chinese Association of Automation*. Nanjing, China. 2018. 258–263.
- 邓念, 彭其渊, 占曙光. 干扰条件下高速铁路列车运行实时调整问题研究. *交通运输系统工程与信息*, 2017, 17(4): 118–123.
- 占曙光, 赵军, 彭其渊. 高速铁路区间能力部分失效情况下列车运行实时调整研究. *铁道学报*, 2016, 38(10): 1–13. [doi: 10.3969/j.issn.1001-8360.2016.10.001]
- Hou ZP, Dong HR, Gao SG, *et al.* Energy-saving metro train timetable rescheduling model considering ATO profiles and dynamic passenger flow. *IEEE Transactions on Intelligent Transportation Systems*, 2019, 20(7): 2774–2785. [doi: 10.1109/TITS.2019.2906483]
- 彭其渊, 陆柳洋, 占曙光. 基于突发故障的高速列车运行调整研究. *交通运输工程与信息学报*, 2018, 16(1): 1–8, 23. [doi: 10.3969/j.issn.1672-4747.2018.01.001]
- Shakibayifar M, Sheikholeslami A, Jamili A. A multi-objective decision support system for real-time train rescheduling. *IEEE Intelligent Transportation Systems Magazine*, 2018, 10(3): 94–109. [doi: 10.1109/MITS.2018.2842037]