

参赛密码 _____
(由组委会填写)



第十二届“中关村青联杯”全国研究生 数学建模竞赛

题 目 面向节能的单/多列车优化决策问题研究

摘 要:

铁路运输消耗的总能量巨大,研究列车节能操作运行具有重要的理论意义和实际应用价值。

针对问题一:分析了单列车运行过程中的能量转换机制。得到列车的牵引力做功与制动力、阻力做功之间的关系。据此,建立了牵引力做功最小的耗能最低优化模型。考虑到模型约束条件的复杂性,提出了基于模拟-优化思想的模型求解方法:首先,通过模拟方法找到列车的可行运行工况;其次,采用布谷鸟优化算法优化了列车运行工况时间切换点;最后,确定了列车最优运行速度距离曲线。所求结果显示,列车从 A_6 - A_7 站以及 A_6 - A_8 站的最低能耗分别为 $3.4 \times 10^7 \text{J}$ 和 $6.7 \times 10^7 \text{J}$ 。

针对问题二:分析了多列车节能优化控制中列车运行时间以及列车制动牵引重叠时间对能耗的影响。首先,基于“列车运行时间与耗能成反比”的基本规律,提出了缩短列车停站时间以及采用单站最优速度距离曲线的基本节能控制策略;其次,通过控制列车发车间隔实现了列车牵引制动重叠时间的最大化;最后,建立了多列车能量交换重叠时间最大优化模型,并采用动态搜索方法结合布谷鸟优化算法对该模型进行了求解,分别求得 100 列和 240 列列车总耗能最低的发车间隔,进而得到对应的发车时刻图(如图 11 和图 12 所示)、列车速度距离曲线图(如图 10 和图 13~14 所示)。

针对问题三：首先，通过分析列车延误后优化控制问题中尽快恢复正点以及恢复期间耗能最低两个基本目标，建立了列车延误时间最小以及能耗最低的多目标优化控制模型；其次，以单列车从 A_2 到 A_3 站的运行过程为研究对象，以延误 10s 为模型输入，基于模拟优化方法求解了列车延误后尽快恢复正点运行的最优速度距离曲线（如图 15 所示）；最后，对于延误时间为随机变量问题，建立了一个随机模拟模型，生成了大量样本数据，以样本数学期望 7.5s 作为模型输入，重新拟定列车的最优速度距离曲线（如图 16 所示），并与延误之前的控制方案进行了对比。

关键词：列车节能优化；模拟优化；布谷鸟优化算法；动态搜索；多目标优化

目 录

1	问题重述.....	- 1 -
2	问题假设.....	- 2 -
3	符号说明.....	- 2 -
4	问题分析.....	- 4 -
5	模型的建立与求解.....	- 5 -
5.1	单列车能耗最低优化模型.....	- 5 -
5.1.1	模型建立.....	- 5 -
5.1.2	模型求解.....	- 7 -
5.1.3	结果分析.....	- 10 -
5.2	多列车节能优化模型.....	- 11 -
5.2.1	模型建立.....	- 11 -
5.2.2	模型求解.....	- 15 -
5.2.3	结果分析.....	- 16 -
5.3	列车延误优化控制模型.....	- 21 -
5.3.1	模型建立.....	- 21 -
5.3.2	模型求解.....	- 23 -
5.3.3	结果分析.....	- 23 -
6	模型的评价与推广.....	- 24 -
6.1	模型的优点.....	- 24 -
6.2	模型的缺点.....	- 25 -
6.3	模型的推广.....	- 25 -
	参考文献.....	- 25 -
	附 录.....	- 27 -
	附录 1 问题一（1）结果数据.....	- 27 -
	附录 2 问题一（2）结果数据.....	- 30 -
	附录 3 求解过程中主要程序.....	- 35 -
	附录 3-1 第一题主要程序.....	- 36 -
	附录 3-2 第二题主要程序.....	- 46 -
	附录 3-3 第三题主要程序.....	- 49 -

1 问题重述

轨道交通系统的能耗是指列车牵引、通风空调、电梯、照明、给排水、弱电等设备产生的能耗。根据统计数据，列车牵引能耗占轨道交通系统总能耗 40% 以上。在低碳环保、节能减排日益受到关注的情况下，针对减少列车牵引能耗的列车运行优化控制近年来成为轨道交通领域的重要研究方向。

请从数学建模的角度进行探索，研究以下问题：

一. 单列车节能运行优化控制问题

(1) 请建立计算速度距离曲线的数学模型，计算寻找一条列车从 A_6 站出发到达 A_7 站的最节能运行的速度距离曲线，其中两车站间的运行时间为 110 秒，列车参数和线路参数详见文件“列车参数.xlsx”和“线路参数.xlsx”。

(2) 请建立新的计算速度距离曲线的数学模型，计算寻找一条列车从 A_6 站出发到达 A_8 站的最节能运行的速度距离曲线，其中要求列车在 A_7 车站停站 45 秒， A_6 站和 A_8 站间总运行时间规定为 220 秒（不包括停站时间），列车参数和线路参数详见文件“列车参数.xlsx”和“线路参数.xlsx”。

注：请将本问 (1) 和 (2) 得到的曲线数据按每秒钟一行填写到文件“数据格式.xlsx”中红色表头那几列，并将该文件和论文一并提交。（请只填写和修改数据，一定不要修改文件“数据格式.xlsx”的格式。其中计算公里标 (m) 是到起点的距离，计算距离 (m) 是到刚通过的一站的距离）

二. 多列车节能运行优化控制问题

(1) 当 100 列列车以间隔 $H = \{h_1, h_2, \dots, h_{99}\}$ 从 A_1 站出发，追踪运行，依次经过 A_2 , A_3 , 最终到达 A_{14} 站，中间在各个车站停站最少 D_{\min} 秒，最多 D_{\max} 秒。间隔 H 各分量的变化范围是 H_{\min} 秒至 H_{\max} 秒。请建立优化模型并寻找使所有列车运行总能耗最低的间隔 H 。要求第一列列车发车时间和最后一列列车的发车时间之间间隔为 $T_0 = 63900$ 秒，且从 A_1 站到 A_{14} 站的总运行时间不变，均为 2086s（包括停站时间）。假设所有列车处于同一供电区段，各个车站间线路参数详见文件“列车参数.xlsx”和“线路参数.xlsx”。

补充说明：列车追踪运行时，为保证安全，跟踪列车（后车）速度不能超过限制速度 V_{limit} ，以免后车无法及时制动停车，发生追尾事故。其计算方式可简化如下：

$$V_{limit} = \min(V_{line}, \sqrt{2LB_e})$$

其中, V_{line} 是列车当前位置的线路限速 (km/h), L 是当前时刻前后车之间的距离(m), B_e 是列车制动的最大减速度(m/s²)。

(2) 接上问, 如果高峰时间 (早高峰 7200 秒至 12600 秒, 晚高峰 43200 至 50400 秒) 发车间隔不大于 2.5 分钟且不小于 2 分钟, 其余时间发车间隔不小于 5 分钟, 每天 240 列。请重新为它们制定运行图和相应的速度距离曲线。

三. 列车延误后运行优化控制问题

接上问, 若列车 i 在车站 A_j 延误 DT_j^i (10 秒) 发车, 请建立控制模型, 找出在确保安全的前提下, 首先使所有后续列车尽快恢复正点运行, 其次恢复期间耗能最少的列车运行曲线。

假设 DT_j^i 为随机变量, 普通延误 ($0 < DT_j^i < 10s$) 概率为 20%, 严重延误 ($DT_j^i > 10s$) 概率为 10% (超过 120s, 接近下一班, 不考虑调整), 无延误 ($DT_j^i = 0$) 概率为 70%。若允许列车在各站到、发时间与原时间相比提前不超过 10 秒, 根据上述统计数据, 如何对第二问的控制方案进行调整。

2 问题假设

假设 1: 列车运行过程中, 无意外事故发生, 不会出现非人为停车事故;

假设 2: 列车运行过程中, 忽略重力做功;

假设 3: 将列车当作质点, 不考虑列车的自身长度;

假设 4: 单列车节能优化控制问题, 不考虑再生能源利用, 与其他列车无能量交换;

假设 5: 多列车节能运行优化控制问题, 各列车在同一站点区间内采用相同的运行策略;

3 符号说明

符号	符号含义
E_F	牵引力所做功

W_F	总阻力所做功
B_F	制动力所做功
L	列车站间间距
T_{ol}	列车 $i+1$ 制动的时间与列车 i 加速时间的重叠时间
T_{br}	列车 $i+1$ 的制动时间
T_m	列车从 A_m 站点到 A_{m+1} 站的运行时间
D_m	列车在第 A_m 站点停站时间
$TQ_{i,m}$	列车在第 A_m 站的牵引终止时刻
$TZ_{i,m}$	列车在第 A_m 站的制动起始时刻
n	列车 $i+1$ 与列车 i 的间距车站数
m	站点编号
T_{sum}	第一列列车发车时间和最后一列列车的发车时间差
D_{sum}	从 A_1 站到 A_{14} 站的总运行时间
DT_j^i	列车 i 在车站 A_j 的延误时间
$t_{m,k}$	未延误时第 m 辆列车在第 A_k 站点的时刻
$t'_{m,k}$	延误后第 m 辆列车在第 A_k 站点的时刻
$F_{A_i-A_j}(t)$	列车未延误时从第 A_i 站到 A_j 站的牵引力函数
$F'_{A_i-A_j}(t)$	列车延误调整时从第 A_i 站到 A_j 站的牵引力函数
$V_{A_i-A_j}(t)$	列车未延误时从第 A_i 站到 A_j 站的速度函数
$V'_{A_i-A_j}(t)$	列车延误调整时从第 A_i 站到 A_j 站的速度函数

注：其他符号见正文具体说明。

4 问题分析

列车在运行过程中的时间与路程固定，根据相关研究成果，列车节能操作需遵循如下四个基本原则^[1-4]：

第一：在列车牵引阶段，使用最大牵引力牵引，有利于快速使列车获得较大的动能，为列车的惰行提供足够的能量；

第二：列车在站间运行时尽量减少惰行的次数，一是惰行和牵引的频繁转换会降低乘客的舒适性，二是惰行次数越多证明牵引次数越多，而站间距离是固定的，牵引的时间越长能量的消耗越大；

第三：在列车制动停车前的站间运行不采取制动措施，因为不必要的制动会带来巨大的能量损失。若没有突发情况，在一般线路上经过合理的操纵设计完全可以避免站间采取不必要的制动；

第四：列车到达终点停车时根据列车运行站点时刻表采取合适的制动力。

对于问题一（1），要求建立单列车节能优化数学模型，求解列车的速度距离曲线，使得列车从 A_6 站出发到达 A_7 站的能耗最低。列车行驶过程中，牵引力做正功，制动力和阻力做负功。基于能量守恒定律，列车的耗能最低可转化为牵引力做功最小。从节能角度出发，当车站间距离较短时，列车一般采用“牵引-惰行-制动”的策略运行。当站间距离较长时，则会采用牵引到接近限制速度后，交替使用惰行、巡航、牵引三种工况，直至接近下一车站采用制动进站停车，具体运行工况由路况以及列车自身特性决定。因此，该问题可抽象为如下优化问题：在满足列车运行速度、时刻、站点间距等约束条件下，首先根据路况条件初步判定列车的运行工况，在此基础上优选运行工况；其次确定各工况之间的最佳切换时间，使得列车运行期间牵引力做功最小。

对于问题一（2），要求重新建立单列车节能优化数学模型，求解列车的速度距离曲线，使得列车从 A_6 站出发到达 A_8 站的能耗最低。将运行总时间根据路况分配至每个站点区间，该问题优化目标不变，优化规模相对扩大，约束条件略有改变，求解策略与问题一（1）基本相同。

对于问题二（1），多列车的节能运行优化控制，应从两个方面进行考虑：其一，应满足单列车站间运行期间的能耗最小；其二，应使相邻列车间产生的可再生能源利用的最大化。基于问题一的分析结论以及列车节能操作的基本原则可知，列车在站间运行时间越长，越有利于降低能耗。由于多列车总运行时间固定，应使得列车停站时间尽量

短，以确保多列车的运行时间相对较长。其次，通过调控列车的发车时间，使得相邻列车制动和牵引重叠的时间尽量最大。最后，根据列车总发车时间及相邻列车之间的间距，确定限制速度，重新调整列车的速度距离曲线。

对于问题二（2），在问题二（1）的基础上，考虑特定时段的发车间隔时间限制，同时增多发车数量，需重新制定多列车的运行图以及相应的速度距离曲线。相比于问题二（1），问题二（2）中对于特定时段的发车时间约束改变，同时由于发车数量增多，将导致相邻列车间距变小，列车的限制速度发生改变，需要据此重新计算列车的速度距离曲线。

对于问题三（1），在问题二的基础上，当列车发生延误时，须建立控制模型，找出安全运行前提下，列车尽快恢复正点运行，同时恢复期间的能耗最低，属于多列车延误追踪的多目标优化控制问题。延误发生之后，恢复正点运行即通过改变列车的运行策略，使得延误后的各列车在各站点的到达时间与正点时间的差值最小、能耗最低，即求解保证各列车安全运行时的最优速度距离曲线。

对于问题三（2），在问题三（1）的基础上，当列车延误时间为随机变量时，根据随机样本，重新建立模型计算，对第二问的控制方案进行调整，同时允许列车在各站到、发时间与正点时间相比不超过 10s。考虑到随机变量样本数据较多，且计算机运算能力有限，单次优化耗时较长。以随机变量的数学期望作为代表，分析随机样本对控制方案的影响是一种有效的途径。因此，首先可建立随机模拟模型生成大量样本，并计算统计样本的数学期望。其次以该数学期望作为多列车延误追踪能耗最低数学模型的输入，重新对第二问的控制方案进行调整。

5 模型的建立与求解

5.1 单列车能耗最低优化模型

5.1.1 模型建立

问题一（1）中要求寻找一条列车从 A_6 站出发到达 A_7 站的最节能运行的速度距离曲线，列车从启动到停止的整个过程遵循能量守恒定律：

$$E_F + mg\Delta h + E_W + E_B = \frac{1}{2}m(V_T^2 - V_0^2) \quad (1)$$

由上式可知列车总能耗为：

$$E_W + E_B = -E_F - mg\Delta h \approx -E_F \quad (2)$$

分析（2）式可知，能耗与牵引力做功以及重力势能的变化有关，当不考虑重力势能的该变量时，列车能耗与牵引力做功大小相等。最节能即可转为为求牵引力在整个过程中做功最小。牵引力做功大小与牵引力大小以及沿牵引力方向的位移有关。为此，需要对列车作受力分析，如图 1 所示。

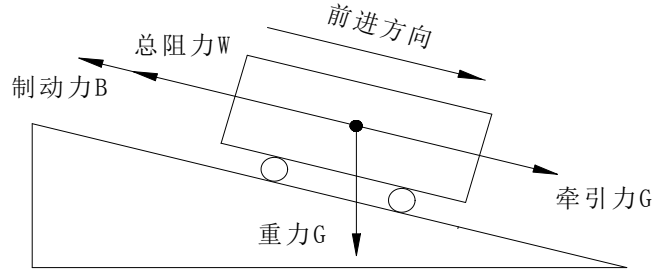


图 1 单质点列车受力分析示意

根据牛顿第二定律，可得到：

$$F + G \cdot \sin \theta - W - B = ma \quad (3)$$

列车在牵引阶段作变加速运动，所受阻力与运行速度有关。根据已知条件，列车总阻力由两部分构成，其一为基本阻力：

$$w_0 = A + Bv + Cv^2 \quad (4)$$

其二为附加阻力：

$$w_1 = i + c / R \quad (5)$$

总阻力为：

$$W = (w_0 + w_1) gM / 1000 \quad (6)$$

列车 t 时刻的瞬时速度为：

$$v_t = \int_t^{t+\Delta t} \frac{F + G \sin \theta - W}{m} dt \quad (7)$$

t 时刻对应的路程为：

$$s_t = \int_t^{t+\Delta t} v dt \quad (8)$$

综上所述，建立了单列车能耗最低优化模型，记为模型①。如下所示：

模型①

$$\begin{aligned}
Obj \quad \min E_F &= \int_{t_0}^{t_1} F_{A_6-A_7}(t) V_{A_6-A_7}(t) dt \\
s.t. \quad &\begin{cases} v_{\min} \leq V_{A_6-A_7} \leq v_{\max} \\ a_{\min} \leq a_{A_6-A_7} \leq a_{\max} \\ 0 \leq B_{A_6-A_7} \leq B_{\max} \\ 0 \leq F_{A_6-A_7} \leq F_{\max} \\ v_{A_6} = v_{A_7} = 0 \\ \sum_{t=0}^T s_t = L_{A_6-A_7} \end{cases} \quad (9)
\end{aligned}$$

在问题一（2）中，列车从 A_6 行驶至 A_8 的总路程以及总时间固定，经过 A_7 时速度降为 0，要求总能耗最小。该问题优化的实质与问题一（1）相同，但约束条件相对增多，求解规模相对扩大。为了降低优化难度，须将两站联合优化问题化为两个单站优化问题。首先，根据各站间的路段距离、坡度、曲率，合理分配各个站间区间列车运行时间^[5-6]。进而在前问的基础上建立单列车两站能耗最低优化模型，记为模型②。

模型②

$$\begin{aligned}
Obj \quad \min E_F &= \int_{t_0}^{t_1} F_{A_6-A_7}(t) V_{A_6-A_7}(t) dt + \int_{t_1+t_{\text{停站}}}^{t_2} F_{A_7-A_8}(t) V_{A_7-A_8}(t) dt \\
s.t. \quad &\begin{cases} v_{\min} \leq V_{A_6-A_8}(t) \leq v_{\max} \\ a_{\min} \leq a_{A_6-A_8} \leq a_{\max} \\ 0 \leq B_{A_6-A_8}(t) \leq B_{\max} \\ 0 \leq F_{A_6-A_8}(t) \leq F_{\max} \\ v_{A_6} = v_{A_7} = v_{A_8} = 0 \\ \sum_{t=0}^T s_t = L_{A_6-A_8} \end{cases} \quad (10)
\end{aligned}$$

5.1.2 模型求解

（1）模拟优化算法

本文采用模拟-优化方法对问题（一）对其进行求解。通常，当求解问题约束条件较多、求解规模较大时，直接采用优化算法寻优，往往难以找到最优解，有时甚至出现算法不收敛的情况。因此，结合模拟手段，在优化之前先寻找到可行解，然后再进行优化成为求解复杂问题的一种有效的途径。模拟优化算法的基本步骤如下：

步骤 1：根据节能的基本原则，采用最大牵引力牵引，有利于使列车获得较大的动能，为列车的惰行提供足够的能量。

步骤 2：根据站点路况，确定列车运行工况组合。从节能角度出发，当车站间距离较短时，即采用“牵引-惰行-制动”工况；当站间距离较长时，则先采用牵引到接近限制速度后，然后交替使用惰行、巡航、牵引三种运行工况，直至接近下一站点采用制动进站停车。

步骤 3：采用布谷鸟优化算法优化各工况的切换点时间点，优化目标为牵引力做功最小，采用罚函数处理等式和不等式约束条件，不断迭代，当满足算法终止准则时，输出最优速度距离曲线。

模拟-优化算法的计算流程图如图 2 所示：

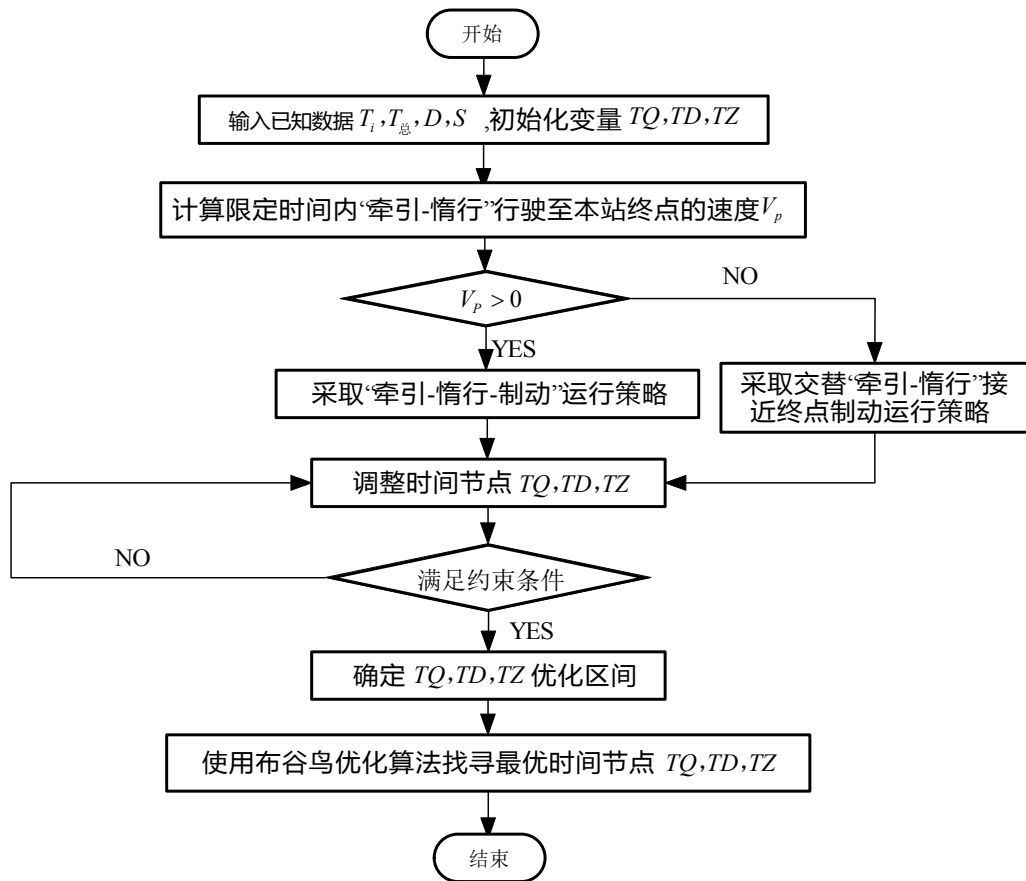


图 2 模拟优化算法求解流程图

(2) 布谷鸟搜索算法

布谷鸟搜索（Cuckoo Search, CS）算法是由英国学者 Xin-She Yang 和 Suash Deb 于 2009 年提出的一种新颖的启发式全局搜索算法^[7]。该算法通过模拟布谷鸟的寄生育雏行为以及莱维飞行（Lévy flight）特征以寻求优化问题的最优解。由于 CS 参数少、鲁棒

性强、搜索效率高，已被成功应用于各领域，成为既 GA 和 PSO 之后群智能算法的一个新亮点^[8-9]。CS 通过两个组件不断生成新的个体：Lévy flight 随机游动以及偏好随机游动，二者平衡算法的全局搜索以及局部搜索。采用 Lévy flight 随机游动更新个体的方式如下：

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus L(\lambda) \quad (11)$$

式中： $x_i^{(t+1)}$ 表示第 $t+1$ 代中个体 i ； α 为步长控制量，用于控制随机搜索的范围，默认值为 0.01； \oplus 为点对点乘法； $L(\lambda)$ 为 Lévy 随机搜索步长，服从 Lévy 分布。

$$L(\lambda) \sim u = t^{-\lambda}, 1 < \lambda \leq 3 \quad (12)$$

另外一种生成新个体的方式是通过一个固定的发现概率 P_a ：用一个随机数 ε 与发现概率 P_a 相比较以确定是否生成新个体。新个体生成方式如下：

$$x_i^{(t+1)} = x_i^{(t)} + \gamma \times H(p_a - \varepsilon) \otimes (x_j^{(t)} - x_k^{(t)}) \quad (13)$$

式中： $\varepsilon, \gamma \in [0, 1]$ ，二者均服从均匀分布； $x_i^{(t)}, x_j^{(t)}, x_k^{(t)}$ 分别表示第 t 代中的 3 个随机个

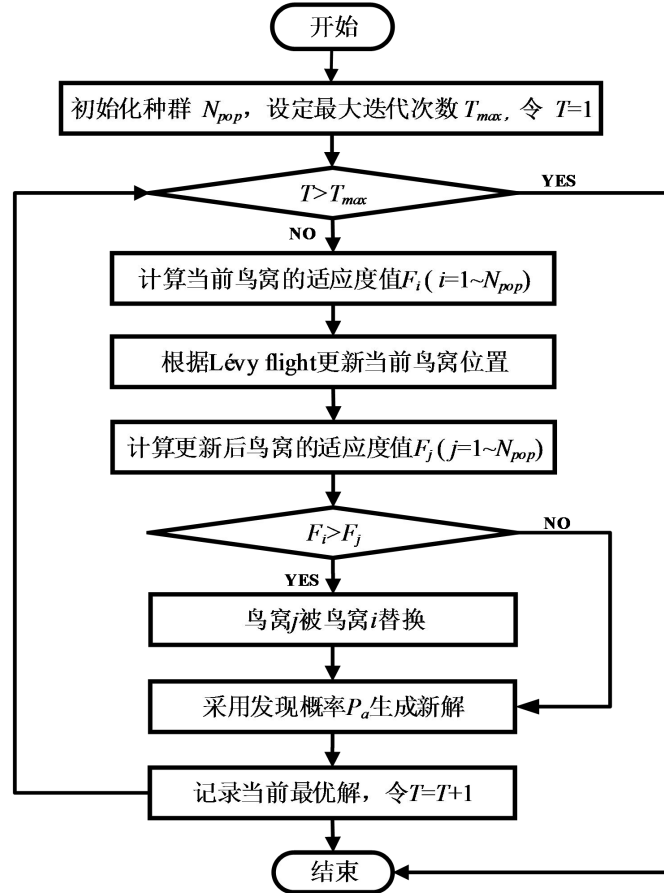


图 3 布谷鸟搜索算法流程图

体。 $H(p_a - \varepsilon)$ 为赫维赛德函数，当 $p_a - \varepsilon > 0$ 时，函数值为 0； $p_a - \varepsilon < 0$ ，函数值为 1； $p_a - \varepsilon = 0$

时，函数值为 0.5。

CS 算法通过以上两种方式对个体不断更新，当满足终止准则时（一般采用最大迭代次数）停止迭代，并输出最优个体。CS 算法计算流程图如图 3 所示。

5.1.3 结果分析

采用模拟-优化方法分别对问题一中模型①、②进行求解，得到列车在 A₆ 站至 A₇ 站以及 A₆ 站至 A₈ 站间列车处于各工况的运行时间以及牵引力做功大小，如表 1 所示。从表 1 可知，从 A₆ 站至 A₇ 站，列车总运行时间约为 110s，从 A₆ 站至 A₈ 站，列车运行时间约为 220s，满足约束条件；此外，列车从 A₆ 站至 A₇ 站牵引力做功为 3.4×10⁷J，列车从 A₆ 站至 A₈ 站牵引力做功为 6.7×10⁷J，二者之比约为对应时间之比；值得指出的是，针对同一站点，列车运行时间越长，牵引力做功越小，即能耗越小，该结论与题目所给出的基本控制策略一致。

表 1 列车各阶段运行时间及牵引力做功表

情景	牵引时长 (s)	惰行时长(s)	制动时长(s)	总时长(s)	做功 (10 ⁷ J)
问题一（1）A6-A7	21	75	14	110	3.4
问题一（2）A6-A7	19	80	14	113	3.2
问题一（2）A7-A8	21	73	13	107	3.5

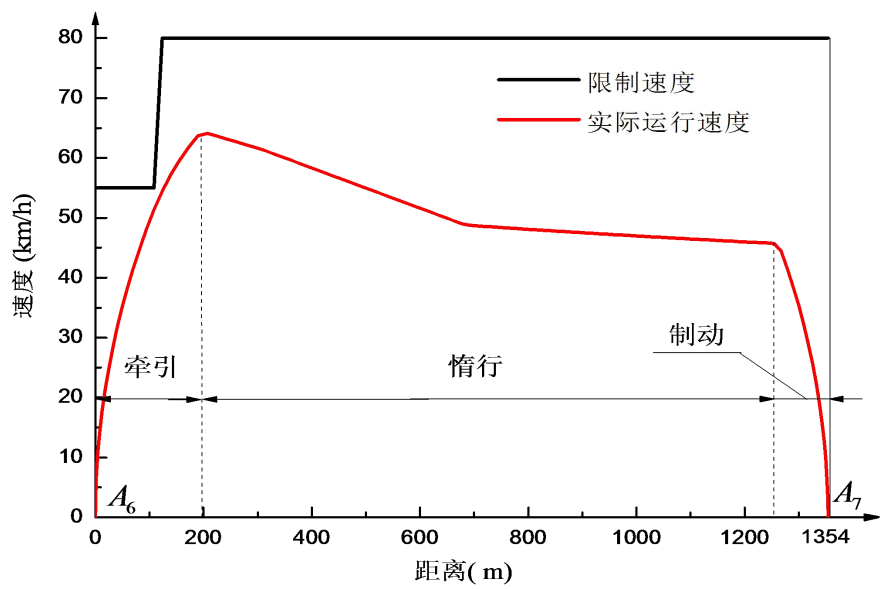


图 4 A₆ 站至 A₇ 站列车速度距离曲线

图 4 和图 5 所示为列车 A₆ 站至 A₇ 站、A₆ 站至 A₈ 站的速度距离曲线。从图中可知，列车在 A₆ 站至 A₇ 站、A₇ 站至 A₈ 站点间运行均采用“牵引—惰行—制动”三种运行工况，

且在达到站点末期，列车速度均降为 0；图 6 所示为列车从 A_6 站至 A_8 站加速度随距离的变化曲线，从图中可以看出，列车在各点的加速度均在限制范围内。综合可知，采用模拟优化方法求解模型①、②合理、可行。

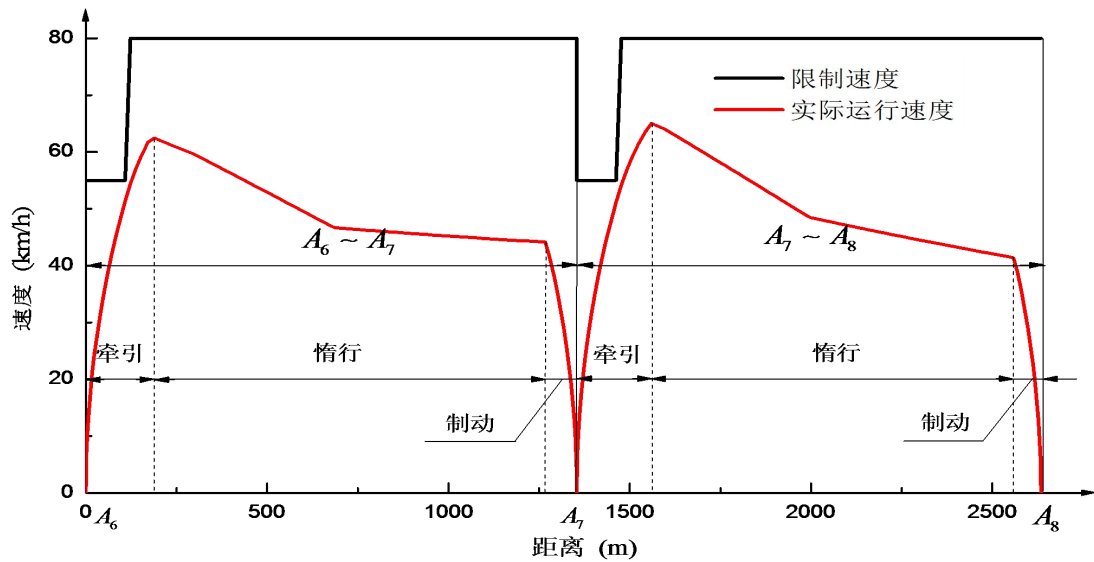


图 5 A_6 站至 A_8 站列车速度距离曲线

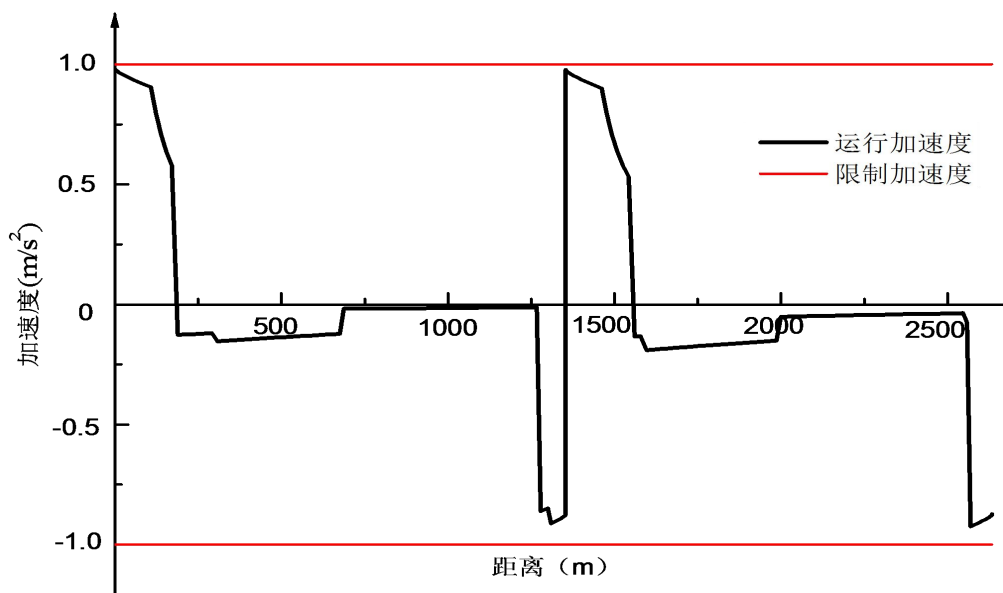


图 6 A_6 至 A_8 站加速度随距离变化示意变化图

5.2 多列车节能优化模型

5.2.1 模型建立

多列车的节能运行优化控制，应从两个方面进行考虑：其一，应满足单列车站间运行期间的能耗最小；其二，应使相邻列车间产生的可再生能量利用的最大化。根据已知

条件可知，列车在站间运行时间与能耗基本存在反比关系，如图 7 所示，列车在站间运行时间越长，能耗最低。

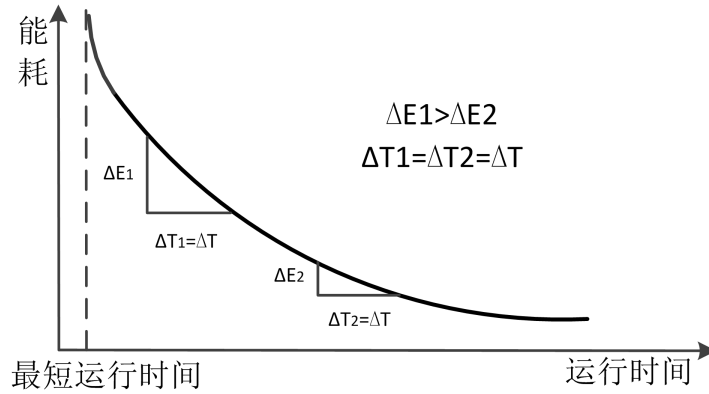


图 7 站间运行时间与耗能关系

由于多列车总运行时间固定，要使得能耗最低，应使列车停站时间尽量短，运行时间尽量长。在单列车最节能运行策略的基础之上，同时使得列车制动时产生的可再生能量利用的最大化，可再生能量利用示意图如图 8 所示。

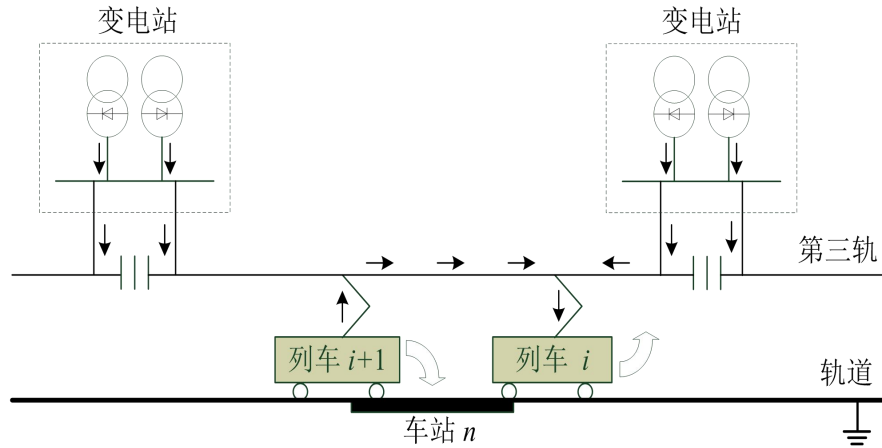


图 8 再生能量利用示意图

相邻列车可再生能量数学表达式如下：

$$E = \eta \frac{T_{ol}}{T_{br}} \left(\frac{1}{2} m v^2 + m g \Delta H - E_f \right) \quad (14)$$

由上式可知，可再生能量与相邻列车的制动、加速重叠时间基本成正比关系。因此，多列车耗能最低模型最终转化为相邻列车制动和牵引的重叠时间最长，即为：

$$\max T_{ol} \quad (15)$$

为不失一般性，以列车 i 和列车 $i+1$ 为研究对象，分析其速度随时间的变化规律，

如图 7 所示为两车在行驶过程中的关键时间节点。

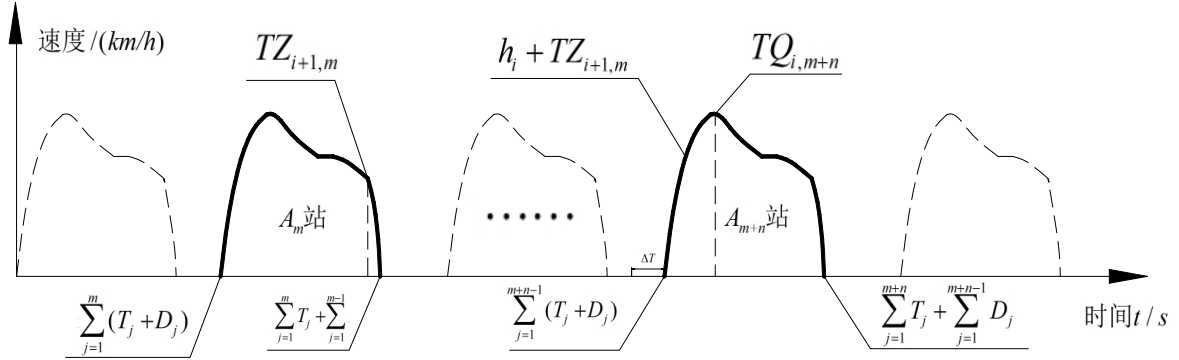


图 9 列车行驶关键时间节点示意图

由图 9 可知，列车 $i+1$ 与列车 i 发车的时间间隔为 h_i ，当列车 $i+1$ 处于制动的起始时刻 $TZ_{i+1,m}$ ，此时列车 i 的运行时间为 $h_i + TZ_{i+1,m}$ ，通过判断列车 i 此刻处于速度—距离（时间）曲线中的哪一阶段，对可能产生再生能量的过程进行分析。具体来讲，可分为两种情景：

情景一：当列车 i 处于牵引阶段，列车 $i+1$ 处于制动阶段，二者产生能量交换。即当 $\sum_{j=1}^{m+n-1} (T_j + D_j) \leq h_i + TZ_{i+1,m} \leq TQ_{i,m+n}$ 时，列车 i 的剩余牵引时间为 $TQ_{i,m+n} - (h_i + TZ_{i+1,m})$ ，

列车 $i+1$ 的制动时间为 $\sum_{j=1}^m T_j - TZ_{i+1,m}$ ，重叠时间取二者中的较小值，即为：

$$T_{ol} = \min \left\{ TQ_{i,m+n} - (h_i + TZ_{i+1,m}), \sum_{j=1}^m T_j - TZ_{i+1,m} \right\} \quad (16)$$

情景二：当列车 i 处于将要发车阶段，列车 $i+1$ 未完成制动时，二者产生能量交换。

记为 $h_i + TZ_{i+1,m} \leq \sum_{j=1}^{m+n-1} (T_j + D_j)$ 。定义为距离发车时刻为：

$$\Delta T = \sum_{j=m}^{m+n-1} (T_j + D_j) - (h_i + TZ_{i+1,m}) \quad (17)$$

当列车 i 的剩余牵引时间为 $TQ_{i,m+n} - \sum_{j=1}^{m+n-1} (T_j + D_j)$ ，列车 $i+1$ 剩余制动时间为

$\sum_{j=1}^m T_j - TZ_{i+1,m} - \Delta T$ ，取二者较小值。即有：

$$T_{ol} = \min \left\{ \sum_{j=1}^m T_j - TZ_{i+1,m} - \Delta T, TQ_{i,m+n} - \sum_{j=1}^{m+n-1} (T_j + D_j) \right\} \quad (18)$$

综上所述，建立多列车能量交换重叠时间最大数学模型（记为模型③），具体表达式如下：

模型③

$$\begin{aligned} Obj \quad & \max \sum_{i=1}^{99} T_{ol} \\ T_{ol} = & \begin{cases} 0, & \sum_{j=1}^{m+n} T_j + \sum_{j=1}^{m+n-1} D_j \geq h_i + TZ_{i+1,m} \geq TQ_{i,m+n} \\ \min \left\{ TQ_{i,m+n} - (h_i + TZ_{i+1,m}), \sum_{j=1}^m T_j - TZ_{i+1,m} \right\}, & \sum_{j=1}^{m+n-1} (T_j + D_j) \leq h_i + TZ_{i+1,m} \leq TQ_{i,m+n} \\ \min \left\{ \sum_{j=1}^m T_j - TZ_{i+1,m} - \Delta T, TQ_{i,m+n} - \sum_{j=1}^{m+n-1} (T_j + D_j) \right\}, & h_i + TZ_{i+1,m} \leq \sum_{j=1}^{m+n-1} (T_j + D_j) \end{cases} \quad (19) \end{aligned}$$

$$S.t. \quad \begin{cases} \sum_{i=1}^{99} h_i = T_{sum} \\ h_{\min} \leq h_i \leq h_{\max} \\ \sum_{m=1}^{13} T_m + \sum_{m=1}^{12} D_m = T_{sum} \\ D_{\min} \leq D_i \leq D_{\max} \\ a_{\min} \leq a_{A_m - A_{m+1}} \leq a_{\max} \\ v_{\min} \leq V_{A_m - A_{m+1}}(t) \leq \min \{ v_{\max}, \sqrt{2LB_e} \} \\ 0 \leq B_{A_m - A_{m+1}}(t) \leq B_{\max} \\ 0 \leq F_{A_m - A_{m+1}}(t) \leq F_{\max} \\ \sum_{t=0}^T s_t = L_{A_1 - A_{14}} \end{cases} \quad (20)$$

在问题二（2）中要求在高峰时段（早高峰 7200 秒至 12600 秒，晚高峰 43200 至 50400 秒）发车间隔 h_i 不大于 2.5 分钟且不小于 2 分钟，其余时间发车间隔不小于 5 分钟，每天 240 列。该问题优化的实质与问题二（1）相同，不同之处在于在特定时段内改变发车时间的约束条件，且运行的列车总数量增多。为此，在问题二（1）的基础上建立了如下数学模型，记为模型④，如下所示：

模型④

$$\begin{aligned}
Obj \quad & \max \sum_{i=1}^{239} T_{ol} \\
T_{ol} = & \begin{cases} 0, & \sum_{j=1}^{m+n} T_j + \sum_{j=1}^{m+n-1} D_j \geq h_i + TZ_{i+1,m} \geq TQ_{i,m+n} \\ \min \left\{ TQ_{i,m+n} - (h_i + TZ_{i+1,m}), \sum_{j=1}^m T_j - TZ_{i+1,m} \right\}, & \sum_{j=1}^{m+n-1} (T_j + D_j) \leq h_i + TZ_{i+1,m} \leq TQ_{i,m+n} \\ \min \left\{ \sum_{j=1}^m T_j - TZ_{i+1,m} - \Delta T, TQ_{i,m+n} - \sum_{j=1}^{m+n-1} (T_j + D_j) \right\}, & h_i + TZ_{i+1,m} \leq \sum_{j=1}^{m+n-1} (T_j + D_j) \end{cases} \quad (21)
\end{aligned}$$

$$\begin{aligned}
s.t. \quad & \begin{cases} \sum_{i=1}^{239} h_i = T_{sum} \\ h_{\min,1} \leq h_{i,1} \leq h_{\max,2}, \quad h_{\min,2} \leq h_{i,2} \leq h_{\max,2} \\ \sum_{m=1}^{13} T_m + \sum_{m=1}^{12} D_m = T_{sum} \\ D_{\min} \leq D_i \leq D_{\max} \\ a_{\min} \leq a_{A_m - A_{m+1}} \leq a_{\max} \\ v_{\min} \leq V_{A_m - A_{m+1}}(t) \leq \min \{ v_{\max}, \sqrt{2LB_e} \} \\ 0 \leq B_{A_m - A_{m+1}}(t) \leq B_{\max} \\ 0 \leq F_{A_m - A_{m+1}}(t) \leq F_{\max} \\ \sum_{t=0}^T s_t = L_{A_1 - A_{14}} \end{cases} \quad (22)
\end{aligned}$$

5.2.2 模型求解

本文采用动态搜索算法结合布谷鸟优化算法对问题二进行求解。首先，为了使得多列车总能耗最低，将列车停站时间取最小值，即 $D_{\min}=30s$ ，以此保证列车运行时间的最大化，同时，各列车均按照问题一中所提供的速度距离曲线运行，从而在整体上保证列车运行能耗相对较低。其次，通过调整发车间隔，使得相邻两列车制动、牵引重叠时间的最大化，从而保证列车生成的再生能量的相对较大。最后，判断在上述发车策略下，否有可能发生追尾事故，如果可能发生追尾事故，更新限制速度，同时调整速度距离曲线以及发车策略；如果不可能发生追尾事故，则直接输出最优发车策略。以相邻两列车为研究对象，进行优化，具体计算步骤如下：

步骤 1：首先将总时间分配至各站点区间，基于模型①，计算单辆列车从 A_1-A_{14} 之间各站间的速度距离曲线；

步骤 2: 将各列车停站时间取最小值 $D_{\min}=30s$, 以确定列车运行总时间;

步骤 3: 以相邻两辆车为研究对象, 对于不同的发车间隔分量 h_i , 其限定的范围内, 以一定的步长进行动态搜索, 找出不同 h_i 值时两列车的重叠时间, 同时建立重叠时间与 h_i 的函数关系;

步骤 4: 以 h_i 为优化变量, 以列车重叠时间最大为目标, 采用布谷鸟算法对其进行优化, 并采用惩罚函数法处理等式约束条件 (总时间为定值)。

步骤 5: 输出最优发车间隔, 根据速度限制公式并判断是否需要对限制速度进行更新, 需要, 返回步骤 1; 不需要, 输出最优发车间隔。

模型④求解方法同上。

5.2.3 结果分析

基于以上分析, 由于列车总运行时间、停站时间均为定值, 可以求得列车在站点间的运行时间; 然后, 将此时间根据各站点间的距离以及坡度情况合理分配到各站点区间; 基于模型①以及模拟-优化算法逐一计算相邻两站点间的速度距离曲线。表 2 为列车从 A_1 站至 A_{14} 站的牵引与制动时刻表, 利用该表可计算相邻两列车制动牵引的重叠时间。

表 2 列车运行过程牵引与制动时刻表

站点名称	牵引		制动	
	起始时刻 (S)	终止时刻 (S)	起始时刻 (S)	终止时刻 (S)
A_1-A_2	0	28	93.2	111.5
A_2-A_3	141.5	163.2	228.35	240
A_3-A_4	270	295	334	363
A_4-A_5	447.35	476.51	397.85	417.35
A_5-A_6	665.57	689.57	630.67	635.57
A_6-A_7	890.47	917.47	852.47	860.47
A_7-A_8	1016.27	1042.27	968.57	986.27
A_8-A_9	1140.57	1166.57	1094.77	1110.57
A_9-A_{10}	1275.97	1298.97	1231.22	1245.97
$A_{10}-A_{11}$	1382.27	1407.27	1334.49	1352.27
$A_{11}-A_{12}$	1578.11	1606.61	1537.8	1548.11
$A_{12}-A_{13}$	1682.61	1719.61	1738.91	1748.56
$A_{13}-A_{14}$	1778.56	1806.56	1845.01	1862.51
	1892.51	1921.06		
	1951.36	1954.86	2073.5	2086

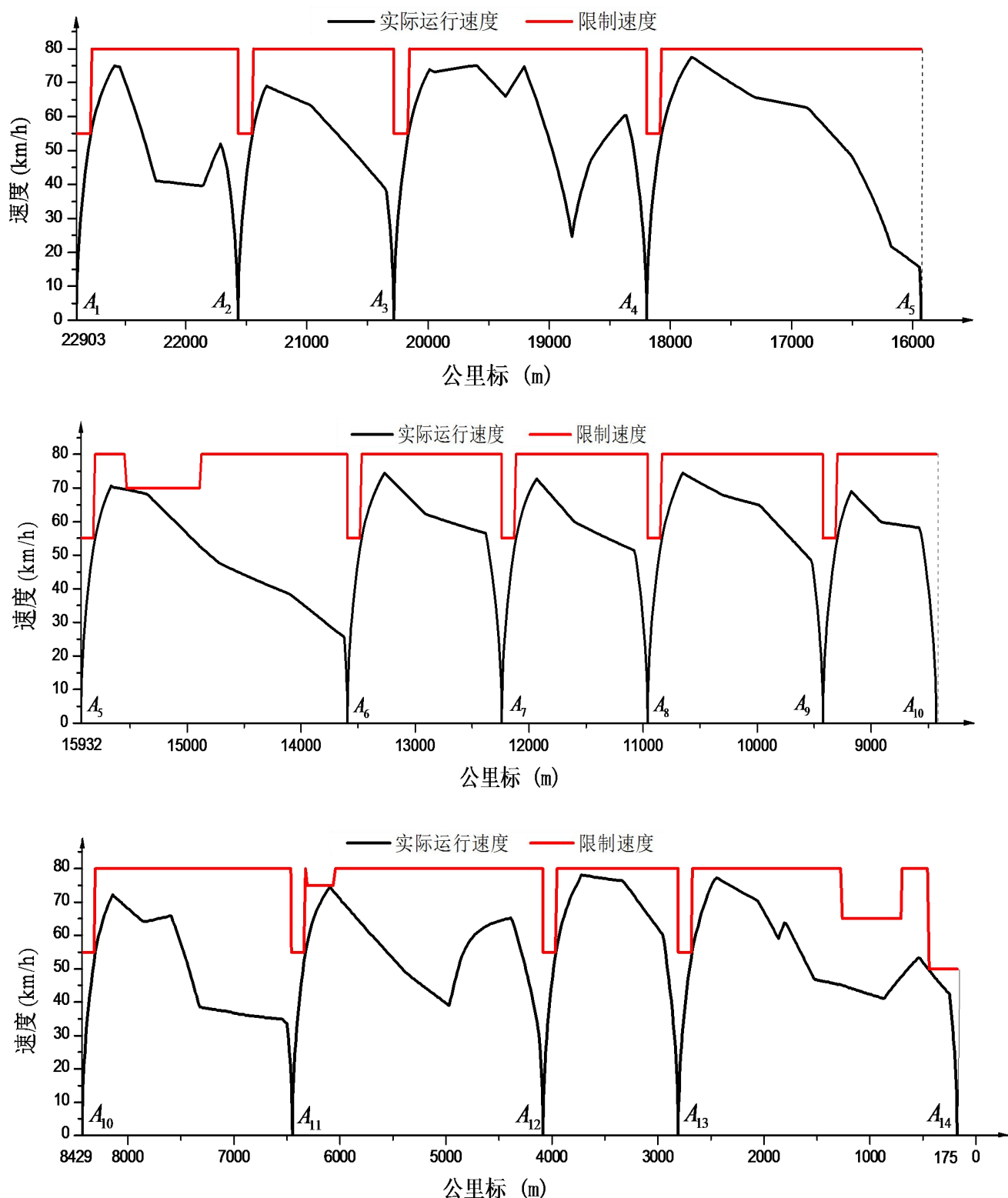


图 10 列车从 A₁ 至 A₁₄ 站的速度-公里标运行曲线

图 10 为列车行驶在 A₁ 站至 A₁₄ 站过程中相邻两站点的速度距离曲线（距离采用公里标表示）。由于该速度距离曲线是基于模型①优化所得，因此，采用该速度距离曲线的运行策略能在局部上确保多列车运行耗能较低。

采用如上提出的动态搜索优化策略对模型③进行求解,可得到 100 列列车的发车时间间隔,见表 3。可知,从 99 次发车时间间隔中,有 86 次为 660s,有 1 次为 444s,有 12 次为 558s,总时长为 63900s,结果合理。结合表 2,可以得到列车 $i+1$ 制动与列车 i 加速最大重叠时间总和为 3760s,约占总运行时间的 5.9%。

表 3 100 辆列车发车时刻表

列车序号		发车时刻 （单位：秒）								
1-10	0	660	1320	1980	2640	3300	3960	4620	5280	5940
10-20	6600	7260	7920	8580	9240	9900	10560	11220	11880	12540
20-30	13200	13860	14520	15180	15840	16500	17160	17820	18480	19140
30-40	19800	20460	21120	21780	22440	23100	23760	24420	25080	25740
40-50	26400	27060	27720	28380	29040	29700	30360	31020	31680	32340
50-60	33000	33660	34320	34980	35640	36300	36960	37620	38280	38940
60-70	39600	40260	40920	41580	42240	42900	43560	44220	44880	45540
70-80	46200	46860	47520	48180	48840	49500	50160	50820	51480	52140
80-90	52800	53460	54120	54780	55440	56100	56760	57204	57762	58320
90-100	58878	59436	59994	60552	61110	61668	62226	62784	63342	63900

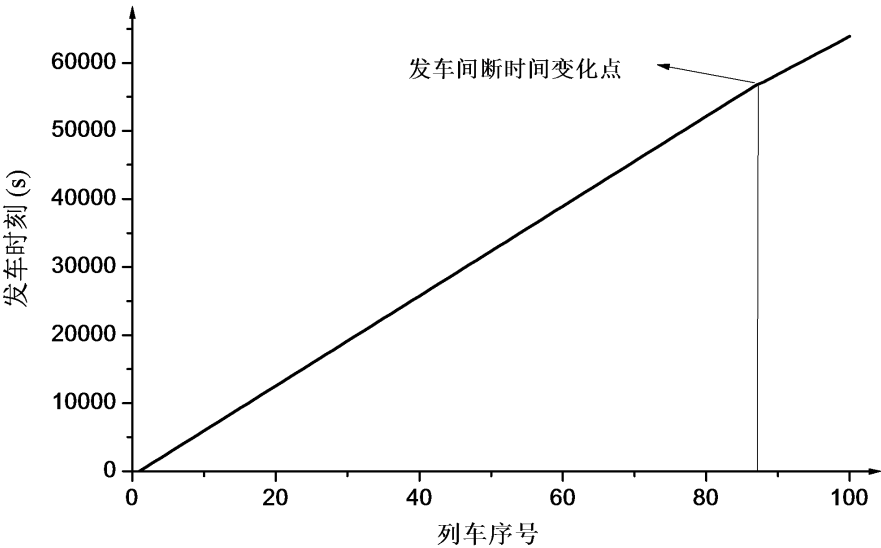


图 11 100 列列车发车时刻图

由于相邻列车间的发车时间较长,最短为 444s,根据列车追尾速度限制公式可得到

列车的限制速度不会超过原来的限制速度。因此，预先确定的列车速度距离曲线仍为最优，无需进行调整。

同理，采用如上方法对模型④进行求解，得到 240 列列车的发车时刻表，如表 4 所示。可知，列车运行过程中，发车时间间隔中有 35 次为 120s，有 61 次为 150s，有 84 次为 354s，有 46 次为 353s，有 13 次为 352s；所有列车牵引制动的最大重叠时间总和约为 11139s，约占总时间的 17.4%。

表 4 240 辆列车发车时刻表

列车序号		发车时刻 （单位：s）								
1-10	0	354	708	1062	1416	1770	2124	2478	2832	3186
10-20	3539	3892	4245	4598	4951	5304	5657	6010	6363	6716
20-30	7069	7189	7309	7429	7549	7669	7789	7909	8029	8149
30-40	8269	8389	8509	8629	8749	8869	8989	9109	9229	9349
40-50	9499	9649	9799	9949	10099	10249	10399	10549	10699	10849
50-60	10999	11149	11299	11449	11599	11749	11899	12049	12199	12349
60-70	12499	12649	12799	13152	13505	13858	14211	14564	14917	15270
70-80	15623	15976	16329	16682	17035	17388	17741	18094	18447	18800
80-90	19153	19506	19859	20212	20565	20918	21271	21624	21977	22330
90-100	22683	23036	23389	23742	24095	24448	24801	25154	25507	25860
100-110	26213	26566	26919	27272	27626	27980	28334	28688	29042	29396
110-120	29750	30104	30458	30812	31166	31520	31874	32228	32582	32936
120-130	33290	33644	33998	34352	34706	35060	35414	35768	36122	36476
130-140	36830	37184	37538	37892	38246	38600	38954	39308	39662	40016
140-150	40370	40724	41078	41432	41786	42140	42494	42848	43202	43322
150-160	43442	43562	43682	43802	43922	44042	44162	44282	44402	44522
160-170	44642	44762	44882	45002	45152	45302	45452	45602	45752	45902
170-180	46052	46202	46352	46502	46652	46802	46952	47102	47252	47402
180-190	47552	47702	47852	48002	48152	48302	48452	48602	48752	48902
190-200	49052	49202	49352	49502	49652	49802	49952	50102	50252	50402
200-210	50552	50702	50852	51204	51556	51908	52260	52612	52964	53316
210-220	53668	54020	54372	54724	55076	55428	55781	56134	56487	56840
220-230	57193	57546	57899	58252	58605	58958	59311	59664	60017	60370
230-240	60723	61076	61429	61782	62135	62488	62841	63194	63547	63900

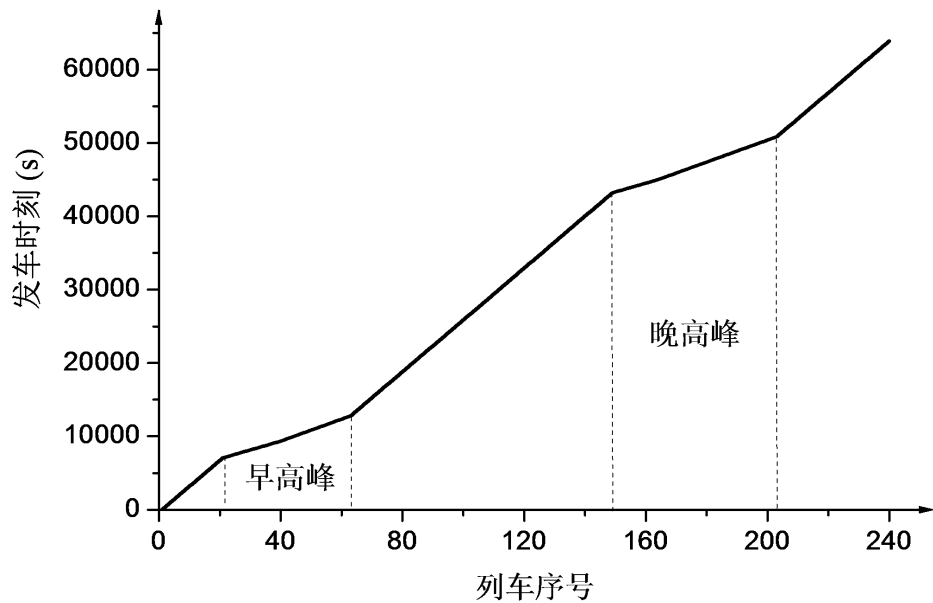


图 12 240 列列车发车时刻图

图 12 为考虑了早高峰和晚高峰的 240 列列车发车时刻图，当列车处于早高峰和晚高峰时，列车发车间隔变小，导致列车的发车更加频繁，图中表现为斜率变小；对于其他时段，列车发车间隔相对较大，因此，发车次数相对较少，图中表现为斜率变大。此外，由于列车发车数量增多，导致相邻列车发车间距大大缩短，发车间隔最短为 120s，经过计算发现，按照线路限速可能会发生追尾事故。由于列车的限制速度发生改变时，需要重新拟定列车的速度距离曲线。因此，本文以 A_2 - A_3 和 A_5 - A_6 区间段为例，结合限制速度计算公式，重新拟定列车的速度距离曲线，分别如图 13~14 所示。

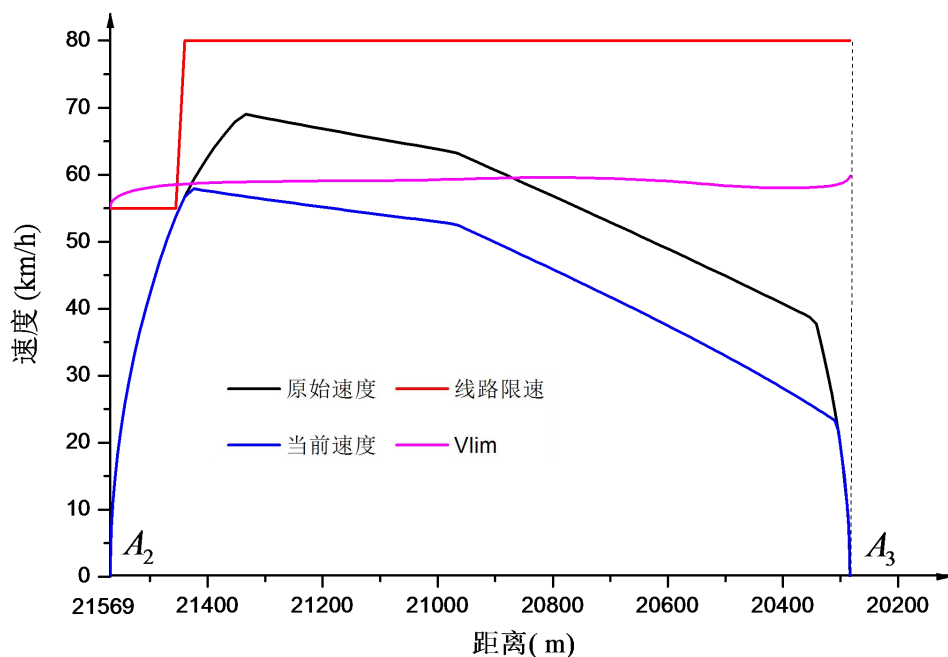


图 13 A_2 - A_3 站列车速度距离曲线调整前后对比图

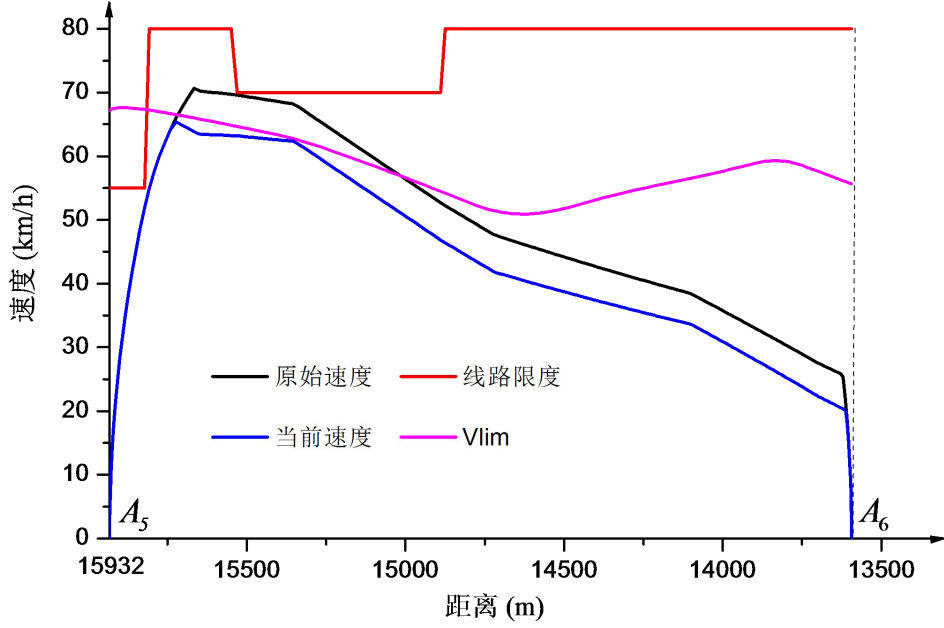


图 14 A₅-A₆站速度距离曲线调整前后对比图

5.3 列车延误优化控制模型

5.3.1 模型建立

在问题三（1）中，列车 i 在车站 A_i 延误 10s 发车，则会导致后续列车的运行时刻发生变化，需对原始列车的运行方案进行调整，在确保安全的前提下，使后续所有列车尽快恢复正点运行，其次，使恢复期间列车耗能最低。该问题的实质是对以下两个目标作优化：① 列车 i 在车站 A_i 发生延误后，需要对随后通过之后的 n 辆列车时刻进行调整，使 $n+1$ 辆列车在调整后到达正点时间与未延误之前；② 在 n 辆列车恢复正点运行的过程中，控制列车的速度距离曲线，使得列车的总耗能最少。综上，建立了列车延误后运行优化控制多目标优化模型如下：

模型⑤

$$\begin{aligned}
 Obj1 \quad & \min \sum_{m=i}^{i+n} \sum_{k=j+1}^{14} (t_{m,k} - t'_{m,k})^2 \\
 Obj2 \quad & \min \int_{t_k}^{t_{14}} [F_{A_k-A_{14}}(t) V_{A_k-A_{14}}(t) dt - F'_{A_k-A_{14}}(t) V'_{A_k-A_{14}}(t)] dt
 \end{aligned} \tag{23}$$

$$s.t. \begin{cases} t'_{i,j} - t_{i,j} = 10 \\ v_{\min} \leq V_{A_m - A_{m+1}}(t) \leq \min \{v_{\max}, \sqrt{2LB_e}\} \\ 0 \leq B_{A_m - A_{m+1}}(t) \leq B_{\max} \\ 0 \leq F_{A_m - A_{m+1}}(t) \leq F_{\max} \\ a_{\min} \leq a_{A_m - A_{m+1}} \leq a_{\max} \\ h_{\min} \leq h_i \leq h_{\max} \\ D_{\min} \leq D_i \leq D_{\max} \\ v_{A_m} = 0 \\ \sum_{t=0}^T s_t = L_{A_1 - A_{14}} \end{cases} \quad (24)$$

在问题三（2）中列车 i 在车站 A_i 延误时间是一个随机变量，各延误程度服从一定的概率分布。该问题优化的实质是根据延误时间的概率分布，随机生成 DT_j^i ，即延误时间为随机变量条件下，调整列车运行控制方案，使其运行工况达到最优。

考虑到随机变量样本数据较多，且计算机运算能力有限，单次优化耗时较长。因此，以随机变量的数学期望作为代表分析随机样本对控制方案的影响。为此，首先建立了随机模拟模型，据此计算随机样本的数学期望。建立随机模拟模型如下：

模型⑥：

$$\overline{DT} = \frac{1}{10M} \sum_{i=1}^{10M} DT_i$$

$$\begin{cases} DT_1 = \text{run}(2M, \text{rnd}[a_1, b_1]) \\ DT_2 = \text{run}(M, \text{rnd}[a_2, b_2]) \\ DT_3 = \text{run}(7M, \text{rnd}[a_3, b_3]) \end{cases} \quad (25)$$

式中： $\text{run}(2M, \text{rnd}[a_1, b_1])$ 表示生成 $2M$ 个随机数，且该随机数服从 $[a_1, b_1]$ 的均匀分布

由上述随机模拟模型将该问题中延误时间随机变量 DT_j^i 转变为一个确定的延误期望值，此期望值为出现概率最大的延误值，至此该问题与问题三（1）本质相同，不同的是允许列车在各站到、发时间与原时间相比提前不超过 10 秒，则考虑延误期间耗能最低，将模型⑥的输出结果作为输入，建立列车延误恢复耗能最低单目标优化模型：

模型⑦：

$$Obj \min \int_{t_k}^{t_{14}} [F_{A_k - A_{14}}(t) V_{A_k - A_{14}}(t) dt - F'_{A_k - A_{14}}(t) V'_{A_k - A_{14}}(t)] dt \quad (26)$$

$$S.t. \begin{cases} t'_{i,j} - t_{i,j} = \overline{DT} \\ |t_{m,k} - t'_{m,k}| \leq 10 \\ v_{\min} \leq V_{A_m - A_{m+1}}(t) \leq \min \{v_{\max}, \sqrt{2LB_e}\} \\ 0 \leq B_{A_m - A_{m+1}}(t) \leq B_{\max} \\ 0 \leq F_{A_m - A_{m+1}}(t) \leq F_{\max} \\ a_{\min} \leq a_{A_m - A_{m+1}} \leq a_{\max} \\ h_{\min} \leq h_i \leq h_{\max} \\ D_{\min} \leq D_i \leq D_{\max} \\ v_{A_m} = 0 \\ \sum_{t=0}^T s_t = L_{A_1 - A_{14}} \end{cases} \quad (27)$$

5.3.2 模型求解

多目标优化问题的求解思路主要有两种：其一，基于多目标优化算法直接进行优化计算，如多目标遗传算法（NSGA-II）、多目标粒子群算法（MPSO）等；其二，通过权重法将多目标优化问题转化为单目标优化问题，然后采用单目标优化算法进行求解，如动态规划（DP）、布谷鸟搜索算法（CS）等。考虑到模型⑤中优化目标较为复杂，且涉及到的约束条件众多，无论是采用传统规划类方法还是智能算法均难以找到最优解。

对于模型⑤，本文仍采用模拟优化方法进行求解，且优先考虑第一目标，即尽快恢复正点运行，其次考虑恢复期间的耗能最低。由于时间有限，本文仅以单列车在 A₂-A₃ 间的运行工况为实例进行模拟优化计算，即推求列车在运行时间缩短 10s 的情况下的速度距离曲线。

对于模型⑦，列车的延误时间为一随机变量。首先，基于模型⑥生成大量的随机样本；其次，求解然后以随机变量的数学期望作为模型输入；最后，基于模型⑤的求解方法对模型⑦进行求解。

5.3.3 结果分析

以 A₂-A₃ 站为例，当列车延误时间为 10s 时，分析列车延误前后的速度距离变化曲线，如图 15 所示（由于只考虑一列列车，故无追尾事故发生）。由图可知，列车牵引阶段速度增大，直到接近限制速度时开始惰行，速度下降；当列车延误时间为随机变量

时，基于随机模拟模型得到样本的数学期望值约为 7.5s，此时推求列车延误前后的速度距离曲线，如图 16 所示。可知该控制方案与延误时间为 10s 时的控制方案差别不大。

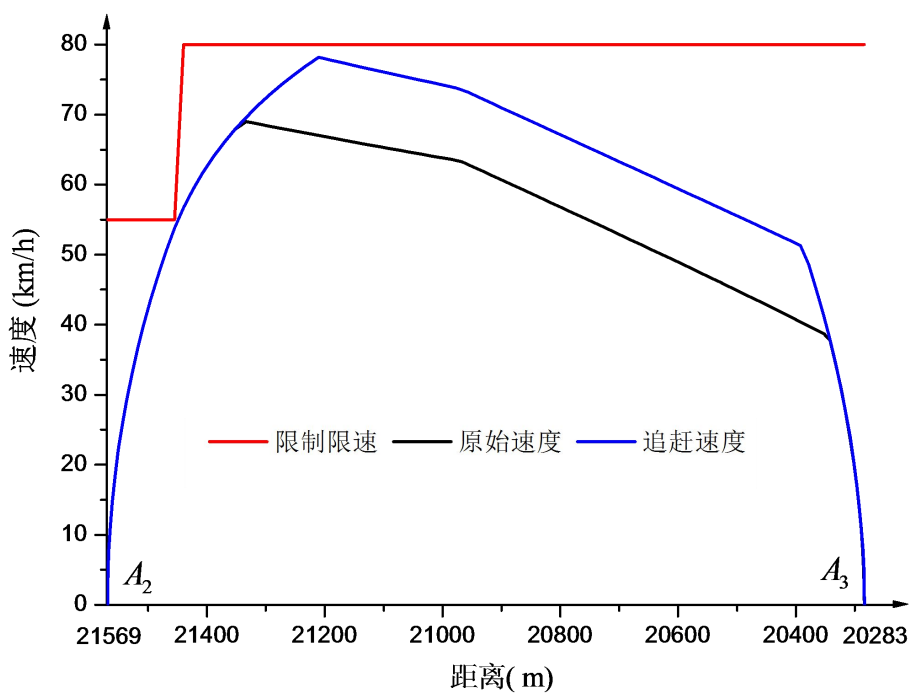


图 15 延误 10s 条件下列车延误前后速度距离曲线对比图

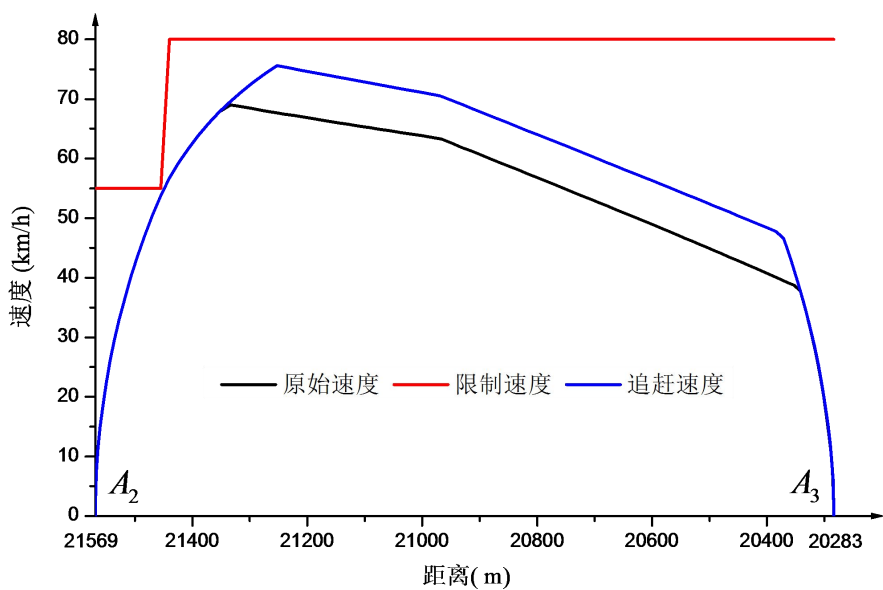


图 16 延误 7.5s 条件下列车延误前后速度距离曲线对比图

6 模型的评价与推广

6.1 模型的优点

1. 本文所建立的模型具有一定的普适性和代表性，可用于指导单列车、多列车的

实际操作运行，也可用于列车优化运行调度的理论研究；

2. 将布谷鸟优化算法引入到列车节能优化问题当中，利用算法较强的搜索能力，可获得列车节能优化问题的更稳定、更精确的计算结果，为列车节能优化问题提供一条可行思路；

3. 提出了基于模拟-优化思想的列车节能优化问题的求解思路，一方面可融合决策者的经验，另外一方面也结合了优化算法优化性能，二者相互配合，使得该方法具有更强的可操作性，更适合求解大规模复杂优化调度问题。

6.2 模型的缺点

1. 所建立的模型结构复杂、约束条件较多，难以求解；

2. 基于模拟-优化思想的模型求解方法虽然可以找到复杂优化问题的相对较优解，但并非理论上的最优解。

6.3 模型的推广

本文围绕列车节能控制问题建立了单列车耗能最低优化模型、多列车节能优化控制模型、列车延误优化模型，提出了基于模拟-优化、动态搜索的模型求解方法，得到了列车在不同运行工况下的最优运行控制策略。研究思路、模型及方法对于列车节能操作运行具有一定的借鉴意义。此外，本文提出的基于模拟-优化思想的模型求解方法在充分利用决策者经验的基础上优化，操作简单、易行，为求解约束条件复杂、优化规模庞大的优化决策问题提供一条新的思路，可广泛应用于调度、决策等实践工作中。

参考文献

- [1] 冯佳. 考虑节能目标的城市轨道交通列车运行行为优化研究[D]. 北京交通大学, 2014.
- [2] 顾青. 城市轨道交通列车节能优化驾驶研究[D]. 北京交通大学, 2014.
- [3] 陈涛. 高速列车运行能耗测算方法及其影响因素量化分析[D]. 北京交通大学, 2011.
- [4] 张燕, 王柄达. 定时条件下列车节能操纵研究综述[J]. 交通运输工程与信息学报, 2011, 9(1): 21-26.
- [5] 丁勇, 刘海东, 栢赞, 等. 地铁列车节能运行的两阶段优化模型算法研究[J]. 交通运

- 输系统工程与信息, 2011, 11(1): 96-101.
- [6] 金炜东, 靳蕃. 列车优化操纵速度模式曲线生成的智能计算研究[J]. 铁道学报, 1998, 20(5): 47-52.
- [7] Xin-she Yang, Deb Suash. Cuckoo search via Lévy Flights [C]. Proceedings of World Congress on Nature & Biologically Inspired Computing. Piscataway: IEEE Publications, 2009: 210-214.
- [8] G. Kanagaraj, S.G. Ponnambalam, N. Jawahar. A hybrid cuckoo search and genetic algorithm for reliability–redundancy allocation problems [J]. Computers & Industrial Engineering, 2013, 66(4): 1115-1124.
- [9] M.K. Marichelvama, T. Prabakaranb, X.S. Yang. Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan [J]. Applied Soft Computing, 2014, 19(6): 93-101

附 录

附录1 问题一（1）结果数据

车站名称	A6							
发车时刻	0:00:00							
站间实际长度(m)	1354							
停站时间(s)	XXX							
站间运行时间(s)	110							
满载率(%)	70%							
时刻 (hh:mm:ss)	实际速度 (cm/s)	实际速度 (km/h)	计算加速 度(m/s ²)	计算距离 (m)	计算公里 标(m)	当前坡度 (‰)	计算牵引力 (N)	计算牵引 功率(Kw)
0:00:00	0.000	0.000	0.983	0.000	13594.000	0.000	194880.000	0.000
0:00:01	98.197	3.535	0.981	0.491	13593.509	0.000	194880.000	191.366
0:00:02	196.134	7.061	0.978	1.963	13592.037	0.000	194880.000	382.227
0:00:03	293.769	10.576	0.975	4.413	13589.587	0.000	194880.000	572.498
0:00:04	391.060	14.078	0.971	7.837	13586.163	0.000	194880.000	762.097
0:00:05	487.963	17.567	0.967	12.233	13581.767	0.000	194880.000	950.942
0:00:06	584.437	21.040	0.962	17.595	13576.405	0.000	194880.000	1138.951
0:00:07	680.443	24.496	0.958	23.920	13570.080	0.000	194880.000	1326.046
0:00:08	775.939	27.934	0.952	31.202	13562.798	0.000	194880.000	1512.150
0:00:09	870.888	31.352	0.947	39.437	13554.563	0.000	194880.000	1697.186
0:00:10	965.250	34.749	0.941	48.618	13545.382	0.000	194880.000	1881.080
0:00:11	1058.990	38.124	0.934	58.740	13535.260	0.000	194880.000	2063.760
0:00:12	1152.071	41.475	0.927	69.796	13524.204	0.000	194880.000	2245.156
0:00:13	1244.459	44.801	0.920	81.779	13512.221	0.000	194880.000	2425.201
0:00:14	1336.119	48.100	0.913	94.682	13499.318	0.000	194880.000	2603.828
0:00:15	1427.019	51.373	0.905	108.499	13485.501	0.000	194880.000	2780.974
0:00:16	1511.976	54.431	0.796	123.203	13470.797	0.000	175060.149	2646.867
0:00:17	1586.868	57.127	0.707	138.705	13455.295	0.000	159168.397	2525.793
0:00:18	1653.846	59.538	0.636	154.914	13439.086	0.000	146679.407	2425.851
0:00:19	1714.442	61.720	0.578	171.760	13422.240	0.000	136638.721	2342.591
0:00:20	1769.780	63.712	0.530	189.186	13404.814	0.000	128406.100	2272.506
0:00:21	1780.770	64.108	-0.132	207.015	13386.985	0.000	0.000	0.000
0:00:22	1767.661	63.636	-0.130	224.757	13369.243	0.000	0.000	0.000
0:00:23	1754.687	63.169	-0.129	242.368	13351.632	0.000	0.000	0.000
0:00:24	1741.845	62.706	-0.128	259.851	13334.149	0.000	0.000	0.000
0:00:25	1729.133	62.249	-0.126	277.206	13316.794	0.000	0.000	0.000
0:00:26	1716.549	61.796	-0.125	294.434	13299.566	0.000	0.000	0.000

0:00:27	1702.543	61.292	-0.159	311.534	13282.466	-1.800	0.000	0.000
0:00:28	1686.714	60.722	-0.158	328.480	13265.520	-1.800	0.000	0.000
0:00:29	1671.042	60.158	-0.156	345.269	13248.731	-1.800	0.000	0.000
0:00:30	1655.524	59.599	-0.154	361.901	13232.099	-1.800	0.000	0.000
0:00:31	1640.157	59.046	-0.153	378.380	13215.620	-1.800	0.000	0.000
0:00:32	1624.939	58.498	-0.151	394.705	13199.295	-1.800	0.000	0.000
0:00:33	1609.867	57.955	-0.150	410.879	13183.121	-1.800	0.000	0.000
0:00:34	1594.938	57.418	-0.149	426.903	13167.097	-1.800	0.000	0.000
0:00:35	1580.151	56.885	-0.147	442.778	13151.222	-1.800	0.000	0.000
0:00:36	1565.504	56.358	-0.146	458.506	13135.494	-1.800	0.000	0.000
0:00:37	1550.992	55.836	-0.144	474.089	13119.911	-1.800	0.000	0.000
0:00:38	1536.616	55.318	-0.143	489.527	13104.473	-1.800	0.000	0.000
0:00:39	1522.372	54.805	-0.142	504.821	13089.179	-1.800	0.000	0.000
0:00:40	1508.258	54.297	-0.141	519.974	13074.026	-1.800	0.000	0.000
0:00:41	1494.272	53.794	-0.139	534.987	13059.013	-1.800	0.000	0.000
0:00:42	1480.412	53.295	-0.138	549.860	13044.140	-1.800	0.000	0.000
0:00:43	1466.676	52.800	-0.137	564.596	13029.404	-1.800	0.000	0.000
0:00:44	1453.062	52.310	-0.136	579.194	13014.806	-1.800	0.000	0.000
0:00:45	1439.569	51.824	-0.134	593.657	13000.343	-1.800	0.000	0.000
0:00:46	1426.194	51.343	-0.133	607.986	12986.014	-1.800	0.000	0.000
0:00:47	1412.935	50.866	-0.132	622.181	12971.819	-1.800	0.000	0.000
0:00:48	1399.790	50.392	-0.131	636.245	12957.755	-1.800	0.000	0.000
0:00:49	1386.759	49.923	-0.130	650.178	12943.822	-1.800	0.000	0.000
0:00:50	1373.839	49.458	-0.129	663.981	12930.019	-1.800	0.000	0.000
0:00:51	1361.028	48.997	-0.128	677.655	12916.345	-1.800	0.000	0.000
0:00:52	1353.818	48.737	-0.023	691.216	12902.784	3.500	0.000	0.000
0:00:53	1351.520	48.655	-0.023	704.743	12889.257	3.500	0.000	0.000
0:00:54	1349.241	48.573	-0.023	718.247	12875.753	3.500	0.000	0.000
0:00:55	1346.981	48.491	-0.023	731.728	12862.272	3.500	0.000	0.000
0:00:56	1344.740	48.411	-0.022	745.186	12848.814	3.500	0.000	0.000
0:00:57	1342.518	48.331	-0.022	758.622	12835.378	3.500	0.000	0.000
0:00:58	1340.314	48.251	-0.022	772.037	12821.963	3.500	0.000	0.000
0:00:59	1338.129	48.173	-0.022	785.429	12808.571	3.500	0.000	0.000
0:01:00	1335.962	48.095	-0.022	798.799	12795.201	3.500	0.000	0.000
0:01:01	1333.812	48.017	-0.021	812.148	12781.852	3.500	0.000	0.000
0:01:02	1331.681	47.941	-0.021	825.476	12768.524	3.500	0.000	0.000
0:01:03	1329.567	47.864	-0.021	838.782	12755.218	3.500	0.000	0.000
0:01:04	1327.470	47.789	-0.021	852.067	12741.933	3.500	0.000	0.000
0:01:05	1325.391	47.714	-0.021	865.331	12728.669	3.500	0.000	0.000
0:01:06	1323.329	47.640	-0.021	878.575	12715.425	3.500	0.000	0.000

0:01:07	1321.284	47.566	-0.020	891.798	12702.202	3.500	0.000	0.000
0:01:08	1319.256	47.493	-0.020	905.001	12688.999	3.500	0.000	0.000
0:01:09	1317.245	47.421	-0.020	918.183	12675.817	3.500	0.000	0.000
0:01:10	1315.250	47.349	-0.020	931.346	12662.654	3.500	0.000	0.000
0:01:11	1313.271	47.278	-0.020	944.488	12649.512	3.500	0.000	0.000
0:01:12	1311.308	47.207	-0.020	957.611	12636.389	3.500	0.000	0.000
0:01:13	1309.362	47.137	-0.019	970.714	12623.286	3.500	0.000	0.000
0:01:14	1307.432	47.068	-0.019	983.798	12610.202	3.500	0.000	0.000
0:01:15	1305.517	46.999	-0.019	996.863	12597.137	3.500	0.000	0.000
0:01:16	1303.618	46.930	-0.019	1009.909	12584.091	3.500	0.000	0.000
0:01:17	1301.734	46.862	-0.019	1022.935	12571.065	3.500	0.000	0.000
0:01:18	1299.866	46.795	-0.019	1035.943	12558.057	3.500	0.000	0.000
0:01:19	1298.013	46.728	-0.018	1048.933	12545.067	3.500	0.000	0.000
0:01:20	1296.174	46.662	-0.018	1061.904	12532.096	3.500	0.000	0.000
0:01:21	1294.351	46.597	-0.018	1074.856	12519.144	3.500	0.000	0.000
0:01:22	1292.543	46.532	-0.018	1087.791	12506.209	3.500	0.000	0.000
0:01:23	1290.749	46.467	-0.018	1100.707	12493.293	3.500	0.000	0.000
0:01:24	1288.970	46.403	-0.018	1113.606	12480.394	3.500	0.000	0.000
0:01:25	1287.205	46.339	-0.018	1126.487	12467.513	3.500	0.000	0.000
0:01:26	1285.454	46.276	-0.017	1139.350	12454.650	3.500	0.000	0.000
0:01:27	1283.718	46.214	-0.017	1152.196	12441.804	3.500	0.000	0.000
0:01:28	1281.995	46.152	-0.017	1165.024	12428.976	3.500	0.000	0.000
0:01:29	1280.287	46.090	-0.017	1177.836	12416.164	3.500	0.000	0.000
0:01:30	1278.592	46.029	-0.017	1190.630	12403.370	3.500	0.000	0.000
0:01:31	1276.910	45.969	-0.017	1203.408	12390.592	3.500	0.000	0.000
0:01:32	1275.242	45.909	-0.017	1216.168	12377.832	3.500	0.000	0.000
0:01:33	1273.588	45.849	-0.016	1228.913	12365.087	3.500	0.000	0.000
0:01:34	1271.947	45.790	-0.016	1241.640	12352.360	3.500	0.000	0.000
0:01:35	1270.319	45.731	-0.016	1254.352	12339.648	3.500	0.000	0.000
0:01:36	1236.285	44.506	-0.868	1266.985	12327.015	3.500	0.000	0.000
0:01:37	1149.829	41.394	-0.861	1278.915	12315.085	3.500	0.000	0.000
0:01:38	1064.016	38.305	-0.855	1289.984	12304.016	3.500	0.000	0.000
0:01:39	978.809	35.237	-0.849	1300.197	12293.803	3.500	0.000	0.000
0:01:40	890.061	32.042	-0.912	1309.549	12284.451	0.000	0.000	0.000
0:01:41	799.139	28.769	-0.907	1317.995	12276.005	0.000	0.000	0.000
0:01:42	708.749	25.515	-0.901	1325.534	12268.466	0.000	0.000	0.000
0:01:43	618.850	22.279	-0.897	1332.172	12261.828	0.000	0.000	0.000
0:01:44	529.402	19.058	-0.892	1337.913	12256.087	0.000	0.000	0.000
0:01:45	440.368	15.853	-0.888	1342.761	12251.239	0.000	0.000	0.000
0:01:46	351.708	12.661	-0.885	1346.721	12247.279	0.000	0.000	0.000

0:01:47	263.386	9.482	-0.882	1349.796	12244.204	0.000	0.000	0.000
0:01:48	175.364	6.313	-0.879	1351.990	12242.010	0.000	0.000	0.000
0:01:49	87.605	3.154	-0.876	1353.304	12240.696	0.000	0.000	0.000
0:01:50	0.000	0.000	-0.874	1354.000	12240.000	0.000	0.000	0.000

附录2 问题一（2）结果数据

车站名称	A6	A7
发车时刻	0:00:00	0:02:38
站间实际长度(m)	1354	1280
停站时间(s)	XXX	45
站间运行时间(s)	113	107
满载率(%)	70%	

时刻 (hh:mm:ss)	实际速度 (cm/s)	实际速度 (km/h)	计算加速 度(m/s ²)	计算距离 (m)	计算公里 标(m)	当前坡度 (‰)	计算牵引力 (N)	计算牵引 功率(Kw)
0:00:00	0.000	0.000	0.983	0.000	13594.000	0.000	194880.000	0.000
0:00:01	98.197	3.535	0.981	0.491	13593.509	0.000	194880.000	191.366
0:00:02	196.134	7.061	0.978	1.963	13592.037	0.000	194880.000	382.227
0:00:03	293.769	10.576	0.975	4.413	13589.587	0.000	194880.000	572.498
0:00:04	391.060	14.078	0.971	7.837	13586.163	0.000	194880.000	762.097
0:00:05	487.963	17.567	0.967	12.233	13581.767	0.000	194880.000	950.942
0:00:06	584.437	21.040	0.962	17.595	13576.405	0.000	194880.000	1138.951
0:00:07	680.443	24.496	0.958	23.920	13570.080	0.000	194880.000	1326.046
0:00:08	775.939	27.934	0.952	31.202	13562.798	0.000	194880.000	1512.150
0:00:09	870.888	31.352	0.947	39.437	13554.563	0.000	194880.000	1697.186
0:00:10	965.250	34.749	0.941	48.618	13545.382	0.000	194880.000	1881.080
0:00:11	1058.990	38.124	0.934	58.740	13535.260	0.000	194880.000	2063.760
0:00:12	1152.071	41.475	0.927	69.796	13524.204	0.000	194880.000	2245.156
0:00:13	1244.459	44.801	0.920	81.779	13512.221	0.000	194880.000	2425.201
0:00:14	1336.119	48.100	0.913	94.682	13499.318	0.000	194880.000	2603.828
0:00:15	1427.019	51.373	0.905	108.499	13485.501	0.000	194880.000	2780.974
0:00:16	1511.976	54.431	0.796	123.203	13470.797	0.000	175060.149	2646.867
0:00:17	1586.868	57.127	0.707	138.705	13455.295	0.000	159168.397	2525.793
0:00:18	1653.846	59.538	0.636	154.914	13439.086	0.000	146679.407	2425.851
0:00:19	1714.442	61.720	0.578	171.760	13422.240	0.000	136638.721	2342.591
0:00:20	1734.978	62.459	-0.127	189.095	13404.905	0.000	0.000	0.000
0:00:21	1722.335	62.004	-0.126	206.381	13387.619	0.000	0.000	0.000
0:00:22	1709.820	61.554	-0.125	223.542	13370.458	0.000	0.000	0.000
0:00:23	1697.430	61.107	-0.123	240.578	13353.422	0.000	0.000	0.000
0:00:24	1685.162	60.666	-0.122	257.491	13336.509	0.000	0.000	0.000
0:00:25	1673.016	60.229	-0.121	274.281	13319.719	0.000	0.000	0.000

0:00:26	1660.990	59.796	-0.120	290.951	13303.049	0.000	0.000	0.000
0:00:27	1648.340	59.340	-0.154	307.501	13286.499	-1.800	0.000	0.000
0:00:28	1633.043	58.790	-0.152	323.908	13270.092	-1.800	0.000	0.000
0:00:29	1617.893	58.244	-0.151	340.162	13253.838	-1.800	0.000	0.000
0:00:30	1602.889	57.704	-0.149	356.266	13237.734	-1.800	0.000	0.000
0:00:31	1588.026	57.169	-0.148	372.221	13221.779	-1.800	0.000	0.000
0:00:32	1573.305	56.639	-0.147	388.027	13205.973	-1.800	0.000	0.000
0:00:33	1558.721	56.114	-0.145	403.687	13190.313	-1.800	0.000	0.000
0:00:34	1544.273	55.594	-0.144	419.202	13174.798	-1.800	0.000	0.000
0:00:35	1529.958	55.078	-0.142	434.573	13159.427	-1.800	0.000	0.000
0:00:36	1515.775	54.568	-0.141	449.802	13144.198	-1.800	0.000	0.000
0:00:37	1501.721	54.062	-0.140	464.889	13129.111	-1.800	0.000	0.000
0:00:38	1487.794	53.561	-0.139	479.836	13114.164	-1.800	0.000	0.000
0:00:39	1473.992	53.064	-0.137	494.645	13099.355	-1.800	0.000	0.000
0:00:40	1460.314	52.571	-0.136	509.317	13084.683	-1.800	0.000	0.000
0:00:41	1446.756	52.083	-0.135	523.852	13070.148	-1.800	0.000	0.000
0:00:42	1433.318	51.599	-0.134	538.252	13055.748	-1.800	0.000	0.000
0:00:43	1419.997	51.120	-0.133	552.519	13041.481	-1.800	0.000	0.000
0:00:44	1406.792	50.645	-0.131	566.652	13027.348	-1.800	0.000	0.000
0:00:45	1393.701	50.173	-0.130	580.655	13013.345	-1.800	0.000	0.000
0:00:46	1380.721	49.706	-0.129	594.527	12999.473	-1.800	0.000	0.000
0:00:47	1367.852	49.243	-0.128	608.270	12985.730	-1.800	0.000	0.000
0:00:48	1355.091	48.783	-0.127	621.884	12972.116	-1.800	0.000	0.000
0:00:49	1342.438	48.328	-0.126	635.372	12958.628	-1.800	0.000	0.000
0:00:50	1329.889	47.876	-0.125	648.733	12945.267	-1.800	0.000	0.000
0:00:51	1317.445	47.428	-0.124	661.970	12932.030	-1.800	0.000	0.000
0:00:52	1305.102	46.984	-0.123	675.083	12918.917	-1.800	0.000	0.000
0:00:53	1296.077	46.659	-0.018	688.077	12905.923	3.500	0.000	0.000
0:00:54	1294.254	46.593	-0.018	701.029	12892.971	3.500	0.000	0.000
0:00:55	1292.447	46.528	-0.018	713.962	12880.038	3.500	0.000	0.000
0:00:56	1290.654	46.464	-0.018	726.878	12867.122	3.500	0.000	0.000
0:00:57	1288.875	46.400	-0.018	739.776	12854.224	3.500	0.000	0.000
0:00:58	1287.111	46.336	-0.018	752.655	12841.345	3.500	0.000	0.000
0:00:59	1285.361	46.273	-0.017	765.518	12828.482	3.500	0.000	0.000
0:01:00	1283.625	46.211	-0.017	778.363	12815.637	3.500	0.000	0.000
0:01:01	1281.903	46.149	-0.017	791.190	12802.810	3.500	0.000	0.000
0:01:02	1280.196	46.087	-0.017	804.001	12789.999	3.500	0.000	0.000
0:01:03	1278.501	46.026	-0.017	816.794	12777.206	3.500	0.000	0.000
0:01:04	1276.821	45.966	-0.017	829.571	12764.429	3.500	0.000	0.000
0:01:05	1275.154	45.906	-0.017	842.331	12751.669	3.500	0.000	0.000

0:01:06	1273.500	45.846	-0.016	855.074	12738.926	3.500	0.000	0.000
0:01:07	1271.859	45.787	-0.016	867.801	12726.199	3.500	0.000	0.000
0:01:08	1270.232	45.728	-0.016	880.511	12713.489	3.500	0.000	0.000
0:01:09	1268.617	45.670	-0.016	893.206	12700.794	3.500	0.000	0.000
0:01:10	1267.016	45.613	-0.016	905.884	12688.116	3.500	0.000	0.000
0:01:11	1265.427	45.555	-0.016	918.546	12675.454	3.500	0.000	0.000
0:01:12	1263.851	45.499	-0.016	931.192	12662.808	3.500	0.000	0.000
0:01:13	1262.288	45.442	-0.016	943.823	12650.177	3.500	0.000	0.000
0:01:14	1260.737	45.387	-0.015	956.438	12637.562	3.500	0.000	0.000
0:01:15	1259.198	45.331	-0.015	969.038	12624.962	3.500	0.000	0.000
0:01:16	1257.671	45.276	-0.015	981.622	12612.378	3.500	0.000	0.000
0:01:17	1256.157	45.222	-0.015	994.191	12599.809	3.500	0.000	0.000
0:01:18	1254.655	45.168	-0.015	1006.745	12587.255	3.500	0.000	0.000
0:01:19	1253.164	45.114	-0.015	1019.284	12574.716	3.500	0.000	0.000
0:01:20	1251.685	45.061	-0.015	1031.809	12562.191	3.500	0.000	0.000
0:01:21	1250.218	45.008	-0.015	1044.318	12549.682	3.500	0.000	0.000
0:01:22	1248.763	44.955	-0.014	1056.813	12537.187	3.500	0.000	0.000
0:01:23	1247.319	44.903	-0.014	1069.293	12524.707	3.500	0.000	0.000
0:01:24	1245.887	44.852	-0.014	1081.759	12512.241	3.500	0.000	0.000
0:01:25	1244.465	44.801	-0.014	1094.211	12499.789	3.500	0.000	0.000
0:01:26	1243.055	44.750	-0.014	1106.649	12487.351	3.500	0.000	0.000
0:01:27	1241.656	44.700	-0.014	1119.072	12474.928	3.500	0.000	0.000
0:01:28	1240.268	44.650	-0.014	1131.482	12462.518	3.500	0.000	0.000
0:01:29	1238.891	44.600	-0.014	1143.878	12450.122	3.500	0.000	0.000
0:01:30	1237.525	44.551	-0.014	1156.260	12437.740	3.500	0.000	0.000
0:01:31	1236.170	44.502	-0.014	1168.628	12425.372	3.500	0.000	0.000
0:01:32	1234.825	44.454	-0.013	1180.983	12413.017	3.500	0.000	0.000
0:01:33	1233.490	44.406	-0.013	1193.325	12400.675	3.500	0.000	0.000
0:01:34	1232.167	44.358	-0.013	1205.653	12388.347	3.500	0.000	0.000
0:01:35	1230.853	44.311	-0.013	1217.968	12376.032	3.500	0.000	0.000
0:01:36	1229.550	44.264	-0.013	1230.270	12363.730	3.500	0.000	0.000
0:01:37	1228.257	44.217	-0.013	1242.559	12351.441	3.500	0.000	0.000
0:01:38	1226.974	44.171	-0.013	1254.835	12339.165	3.500	0.000	0.000
0:01:39	1225.701	44.125	-0.013	1267.099	12326.901	3.500	0.000	0.000
0:01:40	1152.050	41.474	-0.861	1279.041	12314.959	3.500	0.000	0.000
0:01:41	1066.222	38.384	-0.855	1290.132	12303.868	3.500	0.000	0.000
0:01:42	980.999	35.316	-0.849	1300.368	12293.632	3.500	0.000	0.000
0:01:43	892.101	32.116	-0.912	1309.741	12284.259	0.000	0.000	0.000
0:01:44	801.167	28.842	-0.907	1318.207	12275.793	0.000	0.000	0.000
0:01:45	710.765	25.588	-0.902	1325.766	12268.234	0.000	0.000	0.000

0:01:46	620.855	22.351	-0.897	1332.424	12261.576	0.000	0.000	0.000
0:01:47	531.398	19.130	-0.892	1338.185	12255.815	0.000	0.000	0.000
0:01:48	442.355	15.925	-0.889	1343.053	12250.947	0.000	0.000	0.000
0:01:49	353.687	12.733	-0.885	1347.033	12246.967	0.000	0.000	0.000
0:01:50	265.358	9.553	-0.882	1350.128	12243.872	0.000	0.000	0.000
0:01:51	177.329	6.384	-0.879	1352.341	12241.659	0.000	0.000	0.000
0:01:52	89.565	3.224	-0.876	1353.675	12240.325	0.000	0.000	0.000
0:01:53	0.000	0.000	-0.874	1354.000	12240.000	0.000	0.000	0.000
0:02:38	0.000	0.000	0.977	1354.000	12240.000	0.000	193865.000	0.000
0:02:39	97.675	3.516	0.976	1354.489	12239.511	0.000	193865.000	189.358
0:02:40	195.092	7.023	0.973	1355.953	12238.047	0.000	193865.000	378.216
0:02:41	292.209	10.520	0.970	1358.389	12235.611	0.000	193865.000	566.492
0:02:42	388.984	14.003	0.966	1361.796	12232.204	0.000	193865.000	754.104
0:02:43	485.374	17.473	0.962	1366.168	12227.832	0.000	193865.000	940.971
0:02:44	581.340	20.928	0.957	1371.502	12222.498	0.000	193865.000	1127.014
0:02:45	676.840	24.366	0.953	1377.793	12216.207	0.000	193865.000	1312.156
0:02:46	771.835	27.786	0.947	1385.037	12208.963	0.000	193865.000	1496.318
0:02:47	866.287	31.186	0.942	1393.228	12200.772	0.000	193865.000	1679.428
0:02:48	960.159	34.566	0.936	1402.361	12191.639	0.000	193865.000	1861.411
0:02:49	1053.412	37.923	0.929	1412.429	12181.571	0.000	193865.000	2042.198
0:02:50	1146.013	41.256	0.923	1423.427	12170.573	0.000	193865.000	2221.719
0:02:51	1237.927	44.565	0.916	1435.347	12158.653	0.000	193865.000	2399.906
0:02:52	1329.119	47.848	0.908	1448.183	12145.817	0.000	193865.000	2576.697
0:02:53	1419.558	51.104	0.901	1461.927	12132.073	0.000	193865.000	2752.027
0:02:54	1504.923	54.177	0.800	1476.559	12117.441	0.000	175748.104	2644.874
0:02:55	1580.181	56.887	0.710	1491.992	12102.008	0.000	159664.781	2522.993
0:02:56	1647.439	59.308	0.638	1508.136	12085.864	0.000	147037.917	2422.360
0:02:57	1708.254	61.497	0.580	1524.919	12069.081	0.000	136895.582	2338.525
0:02:58	1763.766	63.496	0.532	1542.283	12051.717	0.000	128586.710	2267.969
0:02:59	1806.669	65.040	-0.134	1560.174	12033.826	0.000	0.000	0.000
0:03:00	1793.292	64.559	-0.133	1578.174	12015.826	0.000	0.000	0.000
0:03:01	1779.408	64.059	-0.190	1596.040	11997.960	0.000	0.000	0.000
0:03:02	1760.463	63.377	-0.188	1613.739	11980.261	-3.000	0.000	0.000
0:03:03	1741.712	62.702	-0.187	1631.250	11962.750	-3.000	0.000	0.000
0:03:04	1723.151	62.033	-0.185	1648.574	11945.426	-3.000	0.000	0.000
0:03:05	1704.777	61.372	-0.183	1665.714	11928.286	-3.000	0.000	0.000
0:03:06	1686.585	60.717	-0.181	1682.670	11911.330	-3.000	0.000	0.000
0:03:07	1668.574	60.069	-0.179	1699.446	11894.554	-3.000	0.000	0.000
0:03:08	1650.739	59.427	-0.177	1716.042	11877.958	-3.000	0.000	0.000
0:03:09	1633.078	58.791	-0.176	1732.461	11861.539	-3.000	0.000	0.000

0:03:10	1615.587	58.161	-0.174	1748.704	11845.296	-3.000	0.000	0.000
0:03:11	1598.263	57.537	-0.172	1764.774	11829.226	-3.000	0.000	0.000
0:03:12	1581.104	56.920	-0.171	1780.670	11813.330	-3.000	0.000	0.000
0:03:13	1564.106	56.308	-0.169	1796.396	11797.604	-3.000	0.000	0.000
0:03:14	1547.267	55.702	-0.168	1811.953	11782.047	-3.000	0.000	0.000
0:03:15	1530.583	55.101	-0.166	1827.342	11766.658	-3.000	0.000	0.000
0:03:16	1514.053	54.506	-0.165	1842.565	11751.435	-3.000	0.000	0.000
0:03:17	1497.673	53.916	-0.163	1857.624	11736.376	-3.000	0.000	0.000
0:03:18	1481.441	53.332	-0.162	1872.519	11721.481	-3.000	0.000	0.000
0:03:19	1465.355	52.753	-0.160	1887.253	11706.747	-3.000	0.000	0.000
0:03:20	1449.411	52.179	-0.159	1901.827	11692.173	-3.000	0.000	0.000
0:03:21	1433.608	51.610	-0.157	1916.242	11677.758	-3.000	0.000	0.000
0:03:22	1417.943	51.046	-0.156	1930.499	11663.501	-3.000	0.000	0.000
0:03:23	1402.413	50.487	-0.155	1944.601	11649.399	-3.000	0.000	0.000
0:03:24	1387.017	49.933	-0.153	1958.548	11635.452	-3.000	0.000	0.000
0:03:25	1371.753	49.383	-0.152	1972.342	11621.658	-3.000	0.000	0.000
0:03:26	1356.617	48.838	-0.151	1985.983	11608.017	-3.000	0.000	0.000
0:03:27	1345.522	48.439	-0.052	1999.482	11594.518	-3.000	0.000	0.000
0:03:28	1340.365	48.253	-0.051	2012.912	11581.088	2.000	0.000	0.000
0:03:29	1335.251	48.069	-0.051	2026.290	11567.710	2.000	0.000	0.000
0:03:30	1330.180	47.886	-0.051	2039.617	11554.383	2.000	0.000	0.000
0:03:31	1325.151	47.705	-0.050	2052.893	11541.107	2.000	0.000	0.000
0:03:32	1320.163	47.526	-0.050	2066.120	11527.880	2.000	0.000	0.000
0:03:33	1315.216	47.348	-0.049	2079.297	11514.703	2.000	0.000	0.000
0:03:34	1310.309	47.171	-0.049	2092.424	11501.576	2.000	0.000	0.000
0:03:35	1305.443	46.996	-0.048	2105.503	11488.497	2.000	0.000	0.000
0:03:36	1300.616	46.822	-0.048	2118.533	11475.467	2.000	0.000	0.000
0:03:37	1295.829	46.650	-0.048	2131.516	11462.484	2.000	0.000	0.000
0:03:38	1291.080	46.479	-0.047	2144.450	11449.550	2.000	0.000	0.000
0:03:39	1286.370	46.309	-0.047	2157.337	11436.663	2.000	0.000	0.000
0:03:40	1281.698	46.141	-0.047	2170.178	11423.822	2.000	0.000	0.000
0:03:41	1277.063	45.974	-0.046	2182.971	11411.029	2.000	0.000	0.000
0:03:42	1272.466	45.809	-0.046	2195.719	11398.281	2.000	0.000	0.000
0:03:43	1267.905	45.645	-0.045	2208.421	11385.579	2.000	0.000	0.000
0:03:44	1263.381	45.482	-0.045	2221.077	11372.923	2.000	0.000	0.000
0:03:45	1258.893	45.320	-0.045	2233.689	11360.311	2.000	0.000	0.000
0:03:46	1254.440	45.160	-0.044	2246.255	11347.745	2.000	0.000	0.000
0:03:47	1250.023	45.001	-0.044	2258.778	11335.222	2.000	0.000	0.000
0:03:48	1245.641	44.843	-0.044	2271.256	11322.744	2.000	0.000	0.000
0:03:49	1241.293	44.687	-0.043	2283.690	11310.310	2.000	0.000	0.000

0:03:50	1236.979	44.531	-0.043	2296.082	11297.918	2.000	0.000	0.000
0:03:51	1232.699	44.377	-0.043	2308.430	11285.570	2.000	0.000	0.000
0:03:52	1228.453	44.224	-0.042	2320.736	11273.264	2.000	0.000	0.000
0:03:53	1224.240	44.073	-0.042	2332.999	11261.001	2.000	0.000	0.000
0:03:54	1220.060	43.922	-0.042	2345.221	11248.779	2.000	0.000	0.000
0:03:55	1215.912	43.773	-0.041	2357.401	11236.599	2.000	0.000	0.000
0:03:56	1211.796	43.625	-0.041	2369.539	11224.461	2.000	0.000	0.000
0:03:57	1207.712	43.478	-0.041	2381.637	11212.363	2.000	0.000	0.000
0:03:58	1203.659	43.332	-0.040	2393.694	11200.306	2.000	0.000	0.000
0:03:59	1199.638	43.187	-0.040	2405.710	11188.290	2.000	0.000	0.000
0:04:00	1195.647	43.043	-0.040	2417.686	11176.314	2.000	0.000	0.000
0:04:01	1191.687	42.901	-0.039	2429.623	11164.377	2.000	0.000	0.000
0:04:02	1187.758	42.759	-0.039	2441.520	11152.480	2.000	0.000	0.000
0:04:03	1183.858	42.619	-0.039	2453.378	11140.622	2.000	0.000	0.000
0:04:04	1179.988	42.480	-0.039	2465.197	11128.803	2.000	0.000	0.000
0:04:05	1176.147	42.341	-0.038	2476.978	11117.022	2.000	0.000	0.000
0:04:06	1172.335	42.204	-0.038	2488.721	11105.279	2.000	0.000	0.000
0:04:07	1168.552	42.068	-0.038	2500.425	11093.575	2.000	0.000	0.000
0:04:08	1164.798	41.933	-0.037	2512.092	11081.908	2.000	0.000	0.000
0:04:09	1161.071	41.799	-0.037	2523.721	11070.279	2.000	0.000	0.000
0:04:10	1157.373	41.665	-0.037	2535.313	11058.687	2.000	0.000	0.000
0:04:11	1153.702	41.533	-0.037	2546.869	11047.131	2.000	0.000	0.000
0:04:12	1148.571	41.349	-0.075	2558.384	11035.616	2.000	0.000	0.000
0:04:13	1078.856	38.839	-0.925	2569.605	11024.395	0.000	0.000	0.000
0:04:14	986.710	35.522	-0.918	2579.933	11014.067	0.000	0.000	0.000
0:04:15	895.181	32.227	-0.912	2589.342	11004.658	0.000	0.000	0.000
0:04:16	804.228	28.952	-0.907	2597.838	10996.162	0.000	0.000	0.000
0:04:17	713.809	25.697	-0.902	2605.428	10988.572	0.000	0.000	0.000
0:04:18	623.883	22.460	-0.897	2612.116	10981.884	0.000	0.000	0.000
0:04:19	534.412	19.239	-0.893	2617.907	10976.093	0.000	0.000	0.000
0:04:20	445.355	16.033	-0.889	2622.806	10971.194	0.000	0.000	0.000
0:04:21	356.675	12.840	-0.885	2626.815	10967.185	0.000	0.000	0.000
0:04:22	268.335	9.660	-0.882	2629.940	10964.060	0.000	0.000	0.000
0:04:23	180.297	6.491	-0.879	2632.183	10961.817	0.000	0.000	0.000
0:04:24	92.525	3.331	-0.877	2633.547	10960.453	0.000	0.000	0.000
0:04:25	0.000	0.000	-0.873	2634.000	10960.000	0.000	0.000	0.000

附录3 求解过程中主要程序

附录 3-1 第一题主要程序

```
%第一题第一问
clear
clc
close all
% A6 站到 A7 站
x0 = 13594;
xz = 12240;
detaT = 0.01;
% 牵引阶段
[Eq1,Fq1,Pq1,v1,a1,s1]=qianYin(0,0,13594,0.96,20.38,detaT);
% 巡航阶段
[Eq2,Fq2,Pq2,v2,a2,s2]=xunHang(v1(end),s1(end),0,detaT);
% 惰行阶段
[v3,a3,s3] = duoXing(v2(end),0,s2(end),75.25,detaT);
% 制动阶段
[v4,a4,s4]=zhiDong(v3(end),a3(end),s3(end),1,14.36,detaT);
Eq = Eq1+Eq2;
s = [s1,s2(2:end),s3(2:end),s4(2:end)]; % 地标
v = [v1,v2(2:end),v3(2:end),v4(2:end)]; % 速度
a = [a1,a2(2:end),a3(2:end),a4(2:end)]; %加速度
Pq = [Pq1,Pq2(2:end)];
Fq = [Fq1,Fq2(2:end)];
s = s(1:1/detaT:end);
v = v(1:1/detaT:end);
a = a(1:1/detaT:end);
Fq = Fq(1:1/detaT:end);
Pq = Pq(1:1/detaT:end);
xS = load('限速.txt');
poDu = load('线路坡度.txt');
vx = []; % 限速大小
po = []; %当前坡度
for i = 1:length(s)
    for j = 1:length(xS)
        if s(i)>=xS(j,1)&& s(i)<xS(j,3)
            vx(i) = xS(j,2);
            break;
        end
    end
    for k = 1:length(poDu)
        if s(i)>=poDu(k,1)&& s(i)<poDu(k,3)
            po(i) = poDu(k,2);
            break;
        end
    end
end
```

```

        end
end
figure;
plot(s,v,s,vx)
ans1 = [s;v;vx;a;po];
ans2 = [Fq;Pq];
%第一题第二问
clear
clc
close all
xS = load('限速.txt');
poDu = load('线路坡度.txt');
% A6 到 A7 停 45s 再到 A8，共 220s
x6 = 13594; x7 = 12240; x8 = 10960;
detaT = 0.01;
%% A6 到 A7    A6 = 13594; A7 = 12240;    113s
% 牵引阶段
[Eq1,Fq1,Pq1,v1,a1,s1]=qianYin(0,0,13594,0.96,19.48,detaT);
% 巡航阶段
[Eq2,Fq2,Pq2,v2,a2,s2]=xunHang(v1(end),s1(end),0,detaT);
% 惰行阶段
[v3,a3,s3] = duoXing(v2(end),0,s2(end),79.67,detaT);
% 制动阶段
[v4,a4,s4]=zhiDong(v3(end),a3(end),s3(end),1,13.85,detaT);
Eq = Eq1+Eq2;
s = [s1,s2(2:end),s3(2:end),s4(2:end)]; % 地标
v = [v1,v2(2:end),v3(2:end),v4(2:end)]; % 速度
a = [a1,a2(2:end),a3(2:end),a4(2:end)]; %加速度
Pq = [Pq1,Pq2(2:end)];
Fq = [Fq1,Fq2(2:end)];
s = s(1:1/detaT:end);
v = v(1:1/detaT:end);
a = a(1:1/detaT:end);
Fq = Fq(1:1/detaT:end);
Pq = Pq(1:1/detaT:end);
xS = load('限速.txt');
poDu = load('线路坡度.txt');
vx = []; % 限速大小
po = []; %当前坡度
for i = 1:length(s)
    for j = 1:length(xS)
        if s(i)>=xS(j,1)&& s(i)<xS(j,3)
            vx(i) = xS(j,2);
            break;
        end
    end
end

```



```

        end
    end
    for k = 1:length(poDu)
        if s(i)>=poDu(k,1)&&s(i)<poDu(k,3)
            po(i) = poDu(k,2);
            break;
        end
    end
end
figure;
plot(s,v,s,vx)
%% A7 到 A8  A7 = 12240,A8 = 10960; 107s
% 牵引阶段
[Eq5,Fq5,Pq5,v5,a5,s5]=qianYin(0,0,12240,0.955,20.87,detaT);
% 巡航阶段
[Eq6,Fq6,Pq6,v6,a6,s6]=xunHang(v5(end),s5(end),0,detaT);
% 惰行阶段
[v7,a7,s7] = duoXing(v6(end),0,s6(end),73.41,detaT);
s7(end)
% 制动阶段
[v8,a8,s8]=zhiDong(v7(end),a7(end),s7(end),1,12.72,detaT);
s8(end)
v8(end)
Eqq = Eq5+Eq6;
ss = [s5,s6(2:end),s7(2:end),s8(2:end)]; % 地标
vv = [v5,v6(2:end),v7(2:end),v8(2:end)]; % 速度
aa = [a5,a6(2:end),a7(2:end),a8(2:end)]; %加速度
Pqq = [Pq5,Pq6(2:end)];
Fqq = [Fq5,Fq6(2:end)];
ss = ss(1:1/detaT:end);
vv = vv(1:1/detaT:end);
aa = aa(1:1/detaT:end);
Fqq = Fqq(1:1/detaT:end);
Pqq = Pqq(1:1/detaT:end);
vxx = []; % 限速大小
poo = []; %当前坡度
for i = 1:length(ss)
    for j = 1:length(xS)
        if ss(i)>=xS(j,1)&&ss(i)<xS(j,3)
            vxx(i) = xS(j,2);
            break;
        end
    end
end
end
for k = 1:length(poDu)

```

```

        if ss(i)>=poDu(k,1)&&ss(i)<poDu(k,3)
            poo(i) = poDu(k,2);
            break;
        end
    end
end
figure;
plot(ss,vv,ss,vxx)
S = [s,ss];
V = [v,vv];
Vx = [vx,vxx];
A = [a,aa];
Po = [po,poo];
figure;
plot(S,V,S,Vx)
%%%%%%%%%%%%各运行阶段程序%%%%%%%%
%牵引阶段
function [Eq,Fq,Pq,v,a,s]=qianYin(v0,a0,s0,miu,t,detaT)
% 牵引阶段
% s0 初始地标
I = load('线路坡度.txt');
I = I.*[ones(length(I),1),-1*ones(length(I),1),ones(length(I),1)];
Q = load('线路曲率.txt');
% detaT = 1;
v = []; a = []; s = []; Fq=[]; Pq = [];
v(1) = v0; a(1) = a0; s(1) = s0;
    if v0<(51.5*(5/18))&& v0>=0
        Fq(1) = 203*1000;
    elseif v0>=(51.5*(5/18))&&v0<=(80*(5/18))
        Fq(1) = -0.002032*(v0*18/5)^3+0.4928*(v0*18/5)^2-42.13*(v0*18/5)+1343;
        Fq(1) = Fq(1)*1000;
    end
Fq(1) = miu*Fq(1);
Pq(1) = Fq(1)*v(1);
Eq = 0; % 牵引力做功
for i = 2:(t/detaT+1)
    %牵引力
    if v(i-1)<(51.5*(5/18))&& v(i-1)>=0
        Fq(i) = 203*1000;
    elseif v(i-1)>=(51.5*(5/18))&&v(i-1)<=(80*(5/18))
        Fq(i) = -0.002032*(v(i-1)*18/5)^3+0.4928*(v(i-1)*18/5)^2-42.13*(v(i-1)*18/5)+1343;
        Fq(i) = Fq(i)*1000;
    else
        i*detaT
    end
end

```

```

        disp('错误！速度过大');
        break;
    end
    Fq(i) = miu*Fq(i);
    % 阻力
    % 基本阻力
    Fzu0 = 2.031+0.0622*(v(i-1)*18/5)+0.001807*(v(i-1)*18/5)^2;
    % 附加阻力
    for j = 1:length(I)
        if s(i-1)>=I(j,1)&&s(i-1)<I(j,3)
            podu = I(j,2);
        end
    end
    Fzui = podu;
    for j = 1:length(Q)
        if s(i-1)>=Q(j,1)&&s(i-1)<Q(j,3)
            R = Q(j,2);
        end
    end
    if R == 0
        Fzuc = 0;
    else
        Fzuc = 600/R;
    end
    Fzu1 = Fzui+Fzuc;
    Fzu = (Fzu0+Fzu1)*9.8*194.295;
    %重力水平分量
    Fg = 194295*9.8*podu/1000;
    % 加速度
    Fh = Fq(i)-Fg-Fzu;
    a(i) = Fh/194295;
    % 速度
    v(i) = v(i-1)+a(i)*detaT;
    % 牵引功率
    Pq(i) = Fq(i)*v(i);
    % 地标,地标是倒着走
    sk = v(i-1)*detaT + 0.5*a(i)*detaT^2;
    s(i) = s(i-1)-sk; %上一时刻地标减去这个 detaT 时间段内位移
    % k 时刻前牵引力做的功
    Eq = Eq + Fq(i-1)*sk;
end
end

%巡航阶段

```

```

function [Eq,Fq,Pq,v,a,s]=xunHang(v0,s0,t,detaT)
% 巡航阶段 列车匀速运动
% s0 初始地标
if t==0
    Eq = 0;
    Fq = 0;
    Pq = 0;
    v = v0;
    s = s0;
    a = 0;
else
    I = load('线路坡度.txt');
    I = I.*[ones(length(I),1),-1*ones(length(I),1),ones(length(I),1)];
    Q = load('线路曲率.txt');
% detaT = 1;
    v = [];a = [];s = [];Pq = [];
    v(1) = v0;a(1) = 0;s(1) = s0;
    % if v0<(51.5*(5/18))&& v0>=0
    %     Fq(1) = 203*1000;
    % elseif v0>=(51.5*(5/18))&&v0<=(80*(5/18))
    %     Fq(1) = -0.002032*(v0*18/5)^3+0.4928*(v0*18/5)^2-42.13*(v0*18/5)+1343;
    %     Fq(1) = Fq(1)*1000;
    % end
    % Pq(1) = Fq(1)*detaT;
    Eq = 0; % 牵引力做功
    for i = 2:(t/detaT+1)
        v(i) = v(i-1); % 匀速运动
        a(i) = 0;
        % 阻力
        % 基本阻力
        Fzu0 = 2.031+0.0622*(v(i-1)*18/5)+0.001807*(v(i-1)*18/5)^2;
        % 附加阻力
        for j = 1:length(I)
            if s(i-1)>=I(j,1)&&s(i-1)<I(j,3)
                podu = I(j,2);
            end
        end
        Fzui = podu;
        for j = 1:length(Q)
            if s(i-1)>=Q(j,1)&&s(i-1)<Q(j,3)
                R = Q(j,2);
            end
        end
        end
        if R == 0

```

```

        Fzuc = 0;
    else
        Fzuc = 600/R;
    end
    Fzu1 = Fzui+Fzuc;
    Fzu = (Fzu0+Fzu1)*9.8*194.295;
    %重力水平分量
    Fg = 194295*9.8*podu/1000;
    % 牵引力
    Fq(i)= Fg+Fzu;
    % 牵引功率
    Pq(i) = Fq(i)*v(i);
    % 地标,地标是倒着走
    sk = v(i-1)*detaT ;
    s(i) = s(i-1)-sk; %上一时刻地标减去这个 detaT 时间段内位移
    % k 时刻前牵引力做的功
    Eq = Eq + Fq(i-1)*sk;
end
end
end

```

%惰行阶段

```

function [v,a,s]=duoXing(v0,a0,s0,t,detaT)
% 惰行阶段 无牵引力
% s0 初始地标
I = load('线路坡度.txt');
I = I.*[ones(length(I),1),-1*ones(length(I),1),ones(length(I),1)];
Q = load('线路曲率.txt');
% detaT = 1;
v = []; a = []; s = [];
v(1) = v0; a(1) = a0; s(1) = s0;
for i = 2:(t/detaT+1)
    % 阻力
    % 基本阻力
    Fzu0 = 2.031+0.0622*(v(i-1)*18/5)+0.001807*(v(i-1)*18/5)^2;
    % 附加阻力
    for j = 1:length(I)
        if s(i-1)>=I(j,1)&& s(i-1)<I(j,3)
            podu = I(j,2);
        end
    end
    Fzui = podu;
    for j = 1:length(Q)
        if s(i-1)>=Q(j,1)&& s(i-1)<Q(j,3)

```

```

        R = Q(j,2);
    end
end
if R == 0
    Fzuc = 0;
else
    Fzuc = 600/R;
end
Fzu1 = Fzui+Fzuc;
Fzu = (Fzu0+Fzu1)*9.8*194.295;
%重力水平分量
Fg = 194295*9.8*podu/1000;
% 加速度
Fh = -Fg-Fzu;
a(i) = Fh/194295;
% 速度
v(i) = v(i-1)+a(i)*detaT;
% 地标,地标是倒着走
sk = v(i-1)*detaT + 0.5*a(i)*detaT^2;
s(i) = s(i-1)-sk;    %上一时刻地标减去这个 detaT 时间段内位移
end
end

%制动阶段
function [v,a,s]=zhiDong(v0,a0,s0,miu,t,detaT)
% 第一题 制动阶段 无牵引力, 有制动力
% s0 初始地标
I = load('线路坡度.txt');
I = I.*[ones(length(I),1),-1*ones(length(I),1),ones(length(I),1)];
Q = load('线路曲率.txt');
v = []; a = []; s = []; Fb = [];
v(1) = v0; a(1) = a0; s(1) = s0; Fb(1) = 166*1000;
for i = 2:(t/detaT+1)
    % 阻力
    % 基本阻力
    Fzu0 = 2.031+0.0622*(v(i-1)*18/5)+0.001807*(v(i-1)*18/5)^2;
    % 附加阻力
    for j = 1:length(I)
        if s(i-1)>=I(j,1)&& s(i-1)<I(j,3)
            podu = I(j,2);
        end
    end
    Fzui = podu;
    for j = 1:length(Q)

```

```

        if s(i-1)>=Q(j,1)&& s(i-1)<Q(j,3)
            R = Q(j,2);
        end
    end
    if R == 0
        Fzuc = 0;
    else
        Fzuc = 600/R;
    end
    Fzu1 = Fzui+Fzuc;
    Fzu = (Fzu0+Fzu1)*9.8*194.295;
    %重力水平分量
    Fg = 194295*9.8*rodu/1000;
    % 制动力
    if v(i-1)<(77*(5/18))&& v(i-1)>=0
        Fb(i) = 166*1000;
    elseif v(i-1)>=(77*(5/18))&& v(i-1)<=(80*(5/18))
        Fb(i) = -0.1343*(v(i-1)*18/5)^2-25.07*(v(i-1)*18/5)+1300;
        Fb(i) = Fb(i)*1000;
    elseif v(i-1)>(80*(5/18))
        i*detaT
        disp('错误！速度过大');
        break;
    elseif v(i-1)<0
        i*detaT
        disp('已减小过 0! ');
        break;
    end
    Fb(i) = miu*Fb(i);
    % 加速度
    Fh = -Fg-Fzu-Fb(i);
    a(i) = Fh/194295;
    % 速度
    v(i) = v(i-1)+a(i)*detaT;
    % 地标,地标是倒着走
    sk = v(i-1)*detaT + 0.5*a(i)*detaT^2;
    s(i) = s(i-1)-sk; %上一时刻地标减去这个 detaT 时间段内位移
end
end

% 布谷鸟寻优算法
clc
clear
close all

```

```

% Number of nests (or different solutions)
n=50;
% Discovery rate of alien eggs/solutions
pa=0.25;
%% Change this if you want to get better results
% Tolerance
%% Simple bounds of the search domain
% Lower bounds
nd=541;
Lb=zeros(1,nd);
% Upper bounds
Ub=99*ones(1,nd);
% Random initial solutions
for i=1:n
    nest(i,:)=Lb+(Ub-Lb).*rand(size(Lb));
end
% Get the current best
fitness=10^10*ones(n,1);
[fmin,bestnest,nest,fitness]=get_best_nest(nest,nest,fitness);
N_iter=0;
%% Starting iterations
for i=1:100
    % Generate new solutions (but keep the current best)
    new_nest=get_cuckoos(nest,bestnest,Lb,Ub);
    [fnew,best,nest,fitness]=get_best_nest(nest,new_nest,fitness);
    % Update the counter
    % Discovery and randomization
    new_nest=empty_nests(nest,Lb,Ub,pa) ;
    % Evaluate this set of solutions
    [fnew,best,nest,fitness]=get_best_nest(nest,new_nest,fitness);
    % Update the counter again
    N_iter=N_iter+1;
    % Find the best objective so far
    if fnew<fmin,
        fmin=fnew;
        bestnest=best;
    end
end %% End of iterations
%% Post-optimization processing
%% Display all the nests
disp(strcat('Total number of iterations=',num2str(N_iter)));

```


附录 3-2 第二题主要程序

```
%A1 到 A2 站运行情况
% A1 到 A2 , A1 = 22903, A2 = 21569
clear
clc
close all
% A1 站到 A2 站,A1 = 22903, A2 = 21569
x0 = 22903;
detaT = 0.01;
% 牵引阶段
[Eq1,Fq1,Pq1,v1,a1,s1]=qianYin(0,0,x0,0.97,28,detaT);
% 巡航阶段
[Eq2,Fq2,Pq2,v2,a2,s2]=xunHang(v1(end),s1(end),0,detaT);
% 惰行阶段
[v3,a3,s3] = duoXing(v2(end),0,s2(end),65.2,detaT);
v3(end)
% 制动阶段
[v4,a4,s4]=zhiDong(v3(end),a3(end),s3(end),1,18.3,detaT);
s4(end)
v4(end)
Eq = Eq1+Eq2;
s = [s1,s2(2:end),s3(2:end),s4(2:end)]; % 地标
v = [v1,v2(2:end),v3(2:end),v4(2:end)]; % 速度
a = [a1,a2(2:end),a3(2:end),a4(2:end)]; %加速度
Pq = [Pq1,Pq2(2:end)];
Fq = [Fq1,Fq2(2:end)];
s = s(1:1/detaT:end);
v = v(1:1/detaT:end);
a = a(1:1/detaT:end);
Fq = Fq(1:1/detaT:end);
Pq = Pq(1:1/detaT:end);
xS = load('限速.txt');
poDu = load('线路坡度.txt');
vx = []; % 限速大小
po = []; %当前坡度
for i = 1:length(s)
    for j = 1:length(xS)
        if s(i)>=xS(j,1)&&s(i)<xS(j,3)
            vx(i) = xS(j,2);
            break;
        end
    end
end
for k = 1:length(poDu)
    if s(i)>=poDu(k,1)&&s(i)<poDu(k,3)
```

```

        po(i) = poDu(k,2);
        break;
    end
end
end
figure;
plot(s1,v1,'r',s2,v2,'b',s3,v3,'g',s4,v4,'y')
hold on
plot(s,vx,'b')
plot(s,-1.*po,'b-')
hold off

%计算不同发车间隔时间下，前车制动阶段与后车牵引阶段的重叠时间
clear
clc
close all
Q = load('牵引时刻 1.txt');
Z = load('制动时刻 1.txt');
Hmin = 120;
Hmax = 660;
T = [];
for h = Hmin:Hmax
    zhi = Z+h;
    t = 0;
    for i = 1:length(zhi)
        zmin = zhi(i,1);
        zmax = zhi(i,2);
        for j = 1:length(Q)
            qmin = Q(j,1);
            qmax = Q(j,2);
            if zmin>=qmin&&zmax<=qmax
                t = t+zmax-zmin;
            elseif zmin>=qmin&&zmin<qmax&&zmax>qmax
                t = t+qmax-zmin;
            elseif zmax>=qmin&&zmax<qmax&&zmin<qmin
                t = t+zmax-qmin;
            elseif zmin<=qmin&&zmax>=qmax
                t = t+qmax-qmin;
            end
        end
    end
    T = [T;h,t];
end
figure;

```

```

plot(T(:,1),T(:,2))

%对 A5 到 A6 站运行情况进行调整
clear
clc
close all
Sv = load('速度距离.txt');
s = Sv(:,1);
v = Sv(:,2);
vx = Sv(:,3);
%%
h = 150; % 发车间隔
Be = 1;
vli = [];
for i = 1:(length(s)-h)
    L = -s(i+h)+s(i);
    vli(i) = (2*L*Be)^0.5;
end
vli = (vli./3.6)';
ss = []; vlim = [];
for i = 666:860
    ss = [ss,s(i)];
    vlim = [vlim,vli(i)];
end
ss(end)
s = [];v = []; a = [];
vlim = vlim+ones(size(vlim));
% A5 站到 A6 站, A5 = 15932, A6 = 13594, 180.24s~201.22s
x0 = 15932;
detaT = 0.01;
% 牵引阶段
[Eq1,Fq1,Pq1,v1,a1,s1]=qianYin(0,0,x0,0.97,20.75,detaT);
% 巡航阶段
[Eq2,Fq2,Pq2,v2,a2,s2]=xunHang(v1(end),s1(end),0,detaT);
% 惰行阶段
[v3,a3,s3] = duoXing(v2(end),0,s2(end),192.55,detaT);
% 制动阶段
[v4,a4,s4]=zhiDong(v3(end),a3(end),s3(end),1,6.3,detaT);
Eq = Eq1+Eq2;
s = [s1,s2(2:end),s3(2:end),s4(2:end)]; % 地标
v = [v1,v2(2:end),v3(2:end),v4(2:end)]; % 速度
a = [a1,a2(2:end),a3(2:end),a4(2:end)]; %加速度
Pq = [Pq1,Pq2(2:end)];
Fq = [Fq1,Fq2(2:end)];

```

```

s = s(1:1/detaT:end);
v = v(1:1/detaT:end);
a = a(1:1/detaT:end);
Fq = Fq(1:1/detaT:end);
Pq = Pq(1:1/detaT:end);
xS = load('限速.txt');
poDu = load('线路坡度.txt');
vx = []; % 限速大小
po = []; % 当前坡度
for i = 1:length(s)
    for j = 1:length(xS)
        if s(i)>=xS(j,1)&&s(i)<xS(j,3)
            vx(i) = xS(j,2);
            break;
        end
    end
    for k = 1:length(poDu)
        if s(i)>=poDu(k,1)&&s(i)<poDu(k,3)
            po(i) = poDu(k,2);
            break;
        end
    end
end
figure;
plot(s1,v1,'r',s2,v2,'b',s3,v3,'g',s4,v4,'y')
hold on
plot(s,vx,'b')
plot(s,-1.*po,'b-.')
plot(ss,vlim)
hold off

```

附录 3-3 第三题主要程序

```

%第三题第一问
% A2 到 A3 , A2 = 21569, A3 = 20283
clear
clc
close all
% A2 站到 A3 站,A2 = 21569, A3 = 20283, 98.5s 现延误 10s, 追赶至 88.5s
x0 = 21569;
detaT = 0.01;
% 牵引阶段
[Eq1,Fq1,Pq1,v1,a1,s1]=qianYin(0,0,x0,0.97,27.99,detaT);
% 巡航阶段

```

```

[Eq2,Fq2,Pq2,v2,a2,s2]=xunHang(v1(end),s1(end),0,detaT);
% 惰行阶段
[v3,a3,s3] = duoXing(v2(end),0,s2(end),45.31,detaT);
% 制动阶段
[v4,a4,s4]=zhiDong(v3(end),a3(end),s3(end),1,15.2,detaT);
Eq = Eq1+Eq2;
s = [s1,s2(2:end),s3(2:end),s4(2:end)]; % 地标
v = [v1,v2(2:end),v3(2:end),v4(2:end)]; % 速度
a = [a1,a2(2:end),a3(2:end),a4(2:end)]; %加速度
Pq = [Pq1,Pq2(2:end)];
Fq = [Fq1,Fq2(2:end)];
s = s(1:1/detaT:end);
v = v(1:1/detaT:end);
a = a(1:1/detaT:end);
Fq = Fq(1:1/detaT:end);
Pq = Pq(1:1/detaT:end);
xS = load('限速.txt');
poDu = load('线路坡度.txt');
vx = []; % 限速大小
po = []; %当前坡度
for i = 1:length(s)
    for j = 1:length(xS)
        if s(i)>=xS(j,1)&&s(i)<xS(j,3)
            vx(i) = xS(j,2);
            break;
        end
    end
    for k = 1:length(poDu)
        if s(i)>=poDu(k,1)&&s(i)<poDu(k,3)
            po(i) = poDu(k,2);
            break;
        end
    end
end
end

figure;
plot(s1,v1,'r',s2,v2,'b',s3,v3,'g',s4,v4,'y')
hold on
plot(s,vx,'b')
plot(s,-1.*po,'b-')
hold off
%第三题第二问
% A2 到 A3 , A2 = 21569, A3 = 20283
clear

```

```

clc
close all
% A2 站到 A3 站,A2 = 21569, A3 = 20283, 98.5s 现延误 7.52s, 追赶至 90.98s
x0 = 21569;
detaT = 0.01;
% 牵引阶段
[Eq1,Fq1,Pq1,v1,a1,s1]=qianYin(0,0,x0,0.97,25.97,detaT);
% 巡航阶段
[Eq2,Fq2,Pq2,v2,a2,s2]=xunHang(v1(end),s1(end),0,detaT);
% 惰行阶段
[v3,a3,s3] = duoXing(v2(end),0,s2(end),50.81,detaT);
v3(end)
% 制动阶段
[v4,a4,s4]=zhiDong(v3(end),a3(end),s3(end),1,14.2,detaT);
s4(end)
v4(end)
Eq = Eq1+Eq2;
s = [s1,s2(2:end),s3(2:end),s4(2:end)]; % 地标
v = [v1,v2(2:end),v3(2:end),v4(2:end)]; % 速度
a = [a1,a2(2:end),a3(2:end),a4(2:end)]; %加速度
Pq = [Pq1,Pq2(2:end)];
Fq = [Fq1,Fq2(2:end)];
s = s(1:1/detaT:end);
v = v(1:1/detaT:end);
a = a(1:1/detaT:end);
Fq = Fq(1:1/detaT:end);
Pq = Pq(1:1/detaT:end);
xS = load('限速.txt');
poDu = load('线路坡度.txt');
vx = []; % 限速大小
po = []; %当前坡度
for i = 1:length(s)
    for j = 1:length(xS)
        if s(i)>=xS(j,1)&& s(i)<xS(j,3)
            vx(i) = xS(j,2);
            break;
        end
    end
    for k = 1:length(poDu)
        if s(i)>=poDu(k,1)&& s(i)<poDu(k,3)
            po(i) = poDu(k,2);
            break;
        end
    end
end
end

```

```
end
figure;
plot(s1,v1,'r',s2,v2,'b',s3,v3,'g',s4,v4,'y')
hold on
plot(s,vx,'b')
plot(s,-1.*po,'b-')
hold off
```