

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

学 校	湖南大学 武汉大学 三峡大学
参赛队号	K0076
队员姓名	1.李金武 2.杨志巧 3.艾鑫

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

摘 要：

本文解决的是面向节能的单/多列车优化决策问题。对于单列车，本文建立了以总耗能最少为目标的最优控制模型；对于多列车，在单列车最优控制模型的基础上，建立多列车的发车间隔优化模型，充分利用了列车制动时再生能量。最后给出了在列车发车延误情况下的优化调整方案。

对于问题一的第一小问，建立了以总耗能最小为目标的单列车优化控制模型。对于恒定限速的情况，通过极大值原理分析得到，最优控制方案应该由最大牵引、巡航、惰行、最大制动组成。由于本问题中根据各区段限速不同，故将行程划分为不同的区间，分别制定控制方案。为了使总耗能最小，首先构建一个耗能最小的初始解，其耗时一般会超过给定时间。然后对初始解进行逐步修正，每次将一小份能量分给节省时间最多的区间，直到耗时达到给定时间为止。由此求解得到列车消耗的最少能量为 $13.18kW \cdot h$ （即 $4.74 \times 10^7 J$ ），其相应的速度距离曲线图见图 4，曲线数据见附录 1。

对于问题一的第二小问，此问仅在第一小问的基础上增加了一个站点，求解思路类似，即分别构造两站点间的初始解，然后对两个初始解的区间统一进行逐步修正。求解得到列车消耗的最少能量为 $13.18kW \cdot h$ （即 $4.74 \times 10^7 J$ ），其相应的速度距离曲线图见图 7，曲线数据见附录 2。现将两问的结果对比如下表所示：

	时间(s)	路程(m)	能量($kW \cdot h$)
$A_6 - A_7$	110	1534	6.88
$A_6 - A_8$	220	2814	13.18

对于问题二的第一小问：首先根据问题一中建立的模型解得列车从 A_1 站开往 A_4 站的最优行车方案，其消耗的能量为 $131.12kW \cdot h$ ，在每站的停站时间为 38 秒。对于本问中的 100 辆列车，每辆车都按照此方案运行。然后考虑到再生能量的利用，本文通过控制每辆车的发车间隔不同达到耗能最小的目的，建立了发车间隔优化模型，最后利用遗传算法求解得到 100 辆列车运行时所消耗的最少能量为 $5336.35kW \cdot h$ （即 $1.92 \times 10^{10} J$ ），相较于不使用再生能量的方法节省了 59% 的

能量。其中部分车辆的发车间隔如下表所示。

序号	h1	h2	h3	h4	h5
发车间隔(s)	647	647	649	648	648
序号	h6	h7	h8	h9	h10
发车间隔(s)	648	649	648	649	650

对于问题二的 240 辆列车且考虑到高峰时期的多列车的运行优化控制问题，分别讨论了高峰期和非高峰期的列车发车间隔，同样建立了发车间隔优化模型。求解得到 240 辆列车运行时所消耗的总能量为 $11296.52kW \cdot h$ （即 $4.07 \times 10^{10} J$ ），发车间隔如下表所示，速度距离曲线图见图 13。

序号	h1	h2	h3	h4	h5
发车间隔(s)	382	388	309	336	343
序号	h6	h7	h8	h9	h10
发车间隔(s)	382	341	354	343	356

对于问题三的某辆列车出现延误 10 秒钟的问题。由于不会影响其它列车的正常运行，因此仅考虑调整该列车的运行情况。在此基础上建立了以列车恢复正常运转经过的站点数最少、总延时最少以及总耗能最少为目标函数的多目标优化模型。求解得到在各个站点发生延误后调整所经过的站数以及所多消耗的能量的一部分结果如下表所示，具体结果见表 9。

延误站台	多消耗的能量($kW \cdot h$)	实际运行时间(s)	调整的站数
2	2.83	2083	1
3	2.35	2083	1
4	3.78	2079	1
5	2.28	2084	1
6	2.66	2083	1

对于问题三中随机出现延误的情况。首先根据延误的时间分为普通延误和严重延误，然后对这两种情况分别建立以总延时最少、总耗能最少为目标函数的多目标优化模型。考虑到发生延时站点、延时情况的随机性，故进行计算机模拟仿真，即随机模拟各站台发生不同类型的延误，然后针对具体情况进行调整，最后求得调整时延过程中晚点时间的平均值为 $0.55s$ ，消耗能量的平均值为 $132.33kW \cdot h$ ，额外消耗能量的平均值为 $1.2106kW \cdot h$ 。其具体各站台发生时延后的调整情况见附录 12。

关键词： 节能优化控制 遗传算法 Ponryagin 极大值原理 计算机仿真

1 问题重述

1.1 问题背景

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。

1.2 问题的相关条件

(1) 列车运行过程

列车在站间运行时会根据线路条件、自身列车特性、前方线路状况计算出一个限制速度。列车运行过程中不允许超过此限制速度。限制速度会周期性更新。在限制速度的约束下列车通常包含四种运行工况：牵引、巡航、惰行和制动。

- 牵引阶段：列车加速，发动机处于耗能状态。
- 巡航阶段：列车匀速，列车所受合力为 0，列车是需要牵引还是需要制动取决于列车当时受到的总阻力。
- 惰行阶段：列车既不牵引也不制动，列车运行状态取决于受到的列车总阻力，发动机不耗能。
- 制动阶段：列车减速，发动机不耗能。如果列车采用再生制动技术，此时可以将动能转换为电能反馈回供电系统供其他用电设备使用，例如其他正在牵引的列车或者本列车的空调等（本列车空调的耗能较小，通常忽略不计）。

如果车站间距离较短，列车一般采用“牵引-惰行-制动”的策略运行。如果站间距离较长，列车通常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。

(2) 列车动力学模型

列车在运行过程中，实际受力状态非常复杂。采用单质点模型是一种常见的简化方法。单质点模型将列车视为单质点，列车运动符合牛顿运动学定律。其受力可分为四类：重力 G 在轨道垂直方向上的分力与受到轨道的托力抵消，列车牵引力 F ，列车制动力 B 和列车运行总阻力 W ，示意图如下所示：

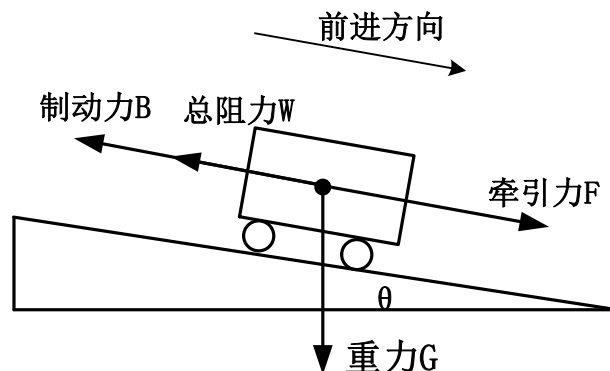


图 1：列车受力示意图

(3) 运行时间与运行能耗的关系

当列车在站间运行时，存在着多条速度距离曲线供选择。不同速度距离曲线对应不同的站间运行时间和不同的能耗。列车按照图 4 所示 4 条曲线可以走完相同的距离，但运行时间和能耗并不相同。此外，即便站间运行时间相同时，也存在多条速度距离曲线可供列车选择。

(4) 再生能量利用原理

随着制动技术的进步，目前城市轨道交通普遍采用再生制动。再生制动时，牵引电动机转变为发电机工况，将列车运行的动能转换为电能，发电机产生的制动力使列车减速，此时列车向接触网反馈电能，此部分能量即为再生制动能。如图 6 所示，列车 $i+1$ 在制动时会产生能量 E_{reg} ，如果相邻列车 i 处于加速状态，其可以利用 E_{reg} ，从而减少从变电站获得的能量，达到节能的目的。如果列车 $i+1$ 制动时，其所处供电区段内没有其他列车加速，其产生的再生能量除用于本列车空调、照明等设备外，通常被吸收电阻转化为热能消耗掉。

1.3 需要解决的问题

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”。

问题二：多列车节能运行优化控制问题

(1) 当 100 列列车以间隔 $H=\{h_1, \dots, h_{99}\}$ 从 A_1 站出发，追踪运行，依次经过 A_2 , A_3 , ……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”。

(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。

问题三：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 模型的假设

假设 1: 假设有多列车运行时每辆车在运行状态中的运行方式是一样的, 只是发车间隔不同。

假设 2: 假设对于任意一辆列车而言在每一站的停站时间是固定值。

假设 3: 假设列车每天从早上 5: 30 开始运行, 到晚上 12: 00 结束。

假设 4: 假设所有列车处于同一供电区间内, 因此如果存在一辆车正在减速, 则其余所有的正在加速的列车均可使用此再生能量。

假设 5: 假设当列车出现晚点情况时, 不再考虑使用再生能量进行节能。

假设 6: 假设列车延误后不考虑客流量的问题。

假设 7: 假设在每一种延误情况下列车晚点时间服从均匀分布。

3 符号说明

符号	符号说明
i	表示第 i 个站点
n	表示第 n 辆列车
t	表示在第 t 时刻
v_t	表示在第 t 时刻的列车运行速度
$F(t)$	表示列车在 t 时刻的运行牵引力
$B(t)$	表示列车在 t 时刻的运行制动力
$W(t)$	表示列车在 t 时刻的运行总阻力
$w_0(t)$	表示单位基本阻力
$w_1(t)$	表示单位附加阻力
E	表示列车运行消耗的总能量
h_n	表示第 n 辆列车的发车间隔
h_t	表示在 t 时刻的发车间隔
d_i^n	表示第 n 辆列车离开第 i 个站点的时间
a_i^n	表示第 n 辆列车到达第 i 个站点的时间
$d_i^{n'}$	表示第 n 辆列车出现延误后离开第 i 个站点的时间
$a_i^{n'}$	表示第 n 辆列车出现延误后到达第 i 个站点的时间
x	表示列车在每个站点的等待时间
v_t^n	表示第 n 辆列车在 t 时刻的速度
d_i^{n+}	表示第 n 辆列车离开第 i 个站点后牵引完成的时间
a_i^{n-}	表示第 n 辆列车到达第 i 个站点前制动开始的时间
t_a^i	表示在第 i 个站点和第 $i+1$ 个站点之间的牵引时间
t_b^i	表示在第 i 个站点和第 $i+1$ 个站点之间的制动时间
E_n'	表示出现延误后第 n 辆列车运行过程中消耗的总能量
$F_n(t)'$	表示列车出现延误后第 n 辆列车在 t 时刻的运行牵引力
$v_t^{n'}$	表示列车出现延误后第 n 辆列车在 t 时刻的速度

4 问题分析

4.1 问题一的分析

问题一共有两小问，第一小问是仅考虑一条列车从 A_6 站出发到 A_7 站的最节能运行的速度距离曲线，通过分析可知耗能与牵引力的大小和时间有关。而题设已经明确给出两站之间的运行时间为 110 秒，因此只需确定运动过程中各个时刻的牵引力的大小。根据题设我们知道整个运行过程分为四个阶段（牵引、巡航、惰行、制动），四个阶段列车的状态、速度、牵引力的大小均不相同，我们所研究的问题是四个阶段各自的特点考虑在总时间一定、总路程一定的条件下的速度距离曲线。可建立以耗能最小的最优化模型来求解。

第二小问在第一小问的基础上增加了一个站点，即列车从 A_6 站出发到 A_7 站然后停留 45 秒后再出发到达 A_8 站点。此时列车运行了两段路程，进行了两次加速减速的过程，故耗能总和极为两段路程的耗能之和。故此问题在第一小问的基础上增加了一个新的阶段，两个阶段的时间和仍然是一定的，同样地，可建立以总耗能最小的最优化模型来求解。

4.2 问题二的分析

问题二分为两小问：第一小问考虑的情况是 100 列列车以间隔 H_i 从 A_1 站出发依次途径 $A_2, A_3 \dots$ 到达 A_4 站，且从 A_1 站到 A_4 站的运行时间均为 2086s（包括停站时间），此时运用问题一中的模型可求得每辆车的速度距离曲线。通过假设 1 可知每辆列车在整个行驶过程中的运行方式是一样的，但是发车间隔可能不同。通过分析可以发现当一辆列车制动时会产生能量供给其它正在加速的列车使用，如果此时没有列车进行加速，那么产生的能量就会被转化为热能消耗掉，因此设定合理的发车间隔会大大的减少能量的消耗。此时以消耗的总能量最小的目标函数就转化为以任意两辆车之间的牵引时间和制动时间重叠最大的目标函数。由此可建立最优化模型进行求解。

第二小问在第一小问的基础上增加考虑了一个高峰时期，因此我们可以将时间分为两部分：高峰时期和非高峰时期。两个时期的发车间隔满足的条件不同，同样可通过控制不同的发车间隔和停站时间来实现能量被消耗的最少。

4.3 问题三的分析

问题三也分为两小问：第一小问仅考虑某辆列车在某个站点出现晚点 10 秒钟的情况，从问题二中可以知道每两辆车之间的发车间隔均大于两分钟，因此当某辆列车出现晚点 10 秒钟的情况时不会出现列车相撞的情况。此时我们仅考虑对于这辆晚点列车进行调整，使得该列车在接下来的运行过程中恢复正常运行所经过的站点尽可能少，即以最短的时间恢复正常。由此可建立以恢复正常运行耗时最短、耗能最少为目标函数的多目标优化模型进行求解。

第二小问因为列车可能出现的延误情况分为普通延误和严重延误等两种情况。如果某辆列车延误时间过长可能会导致后面列车与之相撞的情况发生，因此为了避免此问题的发生，一方面对发生延误的车辆进行加速以便其能尽快恢复运行状态；另一方面对于后面的车辆需要对其运行状态进行判断看其是否与前面的车辆发生碰撞，如果不会发生碰撞则按照原计划运行，否则增加其发车时间或者增加停站时间。由此按照晚点情况分为两部分进行讨论，分别求出所需要的最短时间和最少耗能后按照发生两种延误的概率求出最终的数学期望即可。同样可建立以列车恢复原状态时间最短，消耗能量最少的多目标优化模型进行求解。

5 数据处理

5.1 确定各路段的单位附加阻力

由题设可以知道，对于任意路段而言，其单位附加阻力由该路段的坡度和曲率共同决定，根据给出的线路参数，可以求得任意路段的单位附加阻力，其中部分结果如下表 1 所示：

表 1：各路段的单位附加阻力（单位：N/kN）

路段公里标	0-91	91-174	174-355	355-451	451-535	535-628
单位附加阻力	-2	-1.4	-2	-3	-2.4	12.678
628-695	695-865	865-1265	1265-1525	1525-1692	1692-1945	1945-2055
12.078	13.792	4.872	3.158	-8.041	-7.291	-8.041
2055-2425	2425-2625	2625-2945	2945-3091	3091-3325	3325-3724	3724-3745
3	4.254	-2	-3.84	-2.544	9.393	8.097

5.2 确定任意两个相邻站点之间的距离

根据文件中给出的数据参数，我们可以求得任意两个相邻站点之间的距离如下表 2 所示：

表 2：任意两个相邻站点之间的距离（单位：m）

站点	$A_1 - A_2$	$A_2 - A_3$	$A_3 - A_4$	$A_4 - A_5$	$A_5 - A_6$	$A_6 - A_7$
距离	1334	1286	2086	2265	2338	1354
$A_7 - A_8$	$A_8 - A_9$	$A_9 - A_{10}$	$A_{10} - A_{11}$	$A_{11} - A_{12}$	$A_{12} - A_{13}$	$A_{13} - A_{14}$
1280	1538	993	1982	2366	1275	1911

6 问题一：单列车两个站点间优化控制问题

通过查阅相关资料可知^[1]，在短距离的路程里，列车的航行仅分为 3 步：牵引—惰行—制动。通过分析可以知道，一辆列车在运行过程中的受力分为 4 大类：重力 G 在轨道垂直方向上的分力与受到轨道的托力抵消，列车牵引力 F ，列车制动力 B 和列车运行总阻力 W ，且其运动符合牛顿运动定律，因此首先需要确定各速度下对应的各种力的大小。

6.1 各速度下牵引力大小的确定

根据题设可知牵引力 F 在不同速度下存在不同的最大值 $F_{\max}(t) = f_{F(t)}(v_t)$ 。而列车实际输出牵引力（kN）根据公式 $F(t) = k_t F_{\max}(t)$ 进行计算。 k_t 为实际输出的牵引加速度 a_t 与最大加速度 a_{\max} （1 米/秒²）的百分比， $F_{\max}(t)$ 为牵引力最大值（kN）。因此

$$F(t) = k_t F_{\max}(t)$$

$$\text{其中 } F_{\max}(t) = \begin{cases} 203 & 0 \leq v_t < 51.5 \text{ km/h} \\ -0.002032v_t^3 + 0.4928v_t^2 - 42.13v_t + 1343 & 51.5 < v_t \leq 80 \text{ km/h} \end{cases}。$$

6.2 各速度下制动力大小的确定

制动力 B 与制动时列车速度的最大值有关， $B_{\max}(t) = f_{B(t)}(v_t)$ ，制动力小于

等于 $B_{\max}(t)$ 。列车实际输出制动力 $B(t)$ 可以通过公式 $B(t) = k_b B_{\max}(t)$ 来进行计算。同样地， k_b 为实际输出的制动加速度 a_t 与最大加速度 a_{\max} （1 米/秒²）的百分比， $B_{\max}(t)$ 为制动力最大值（kN）。故：

$$B(t) = k_b B_{\max}(t)$$

$$\text{其中 } B_{\max}(t) = \begin{cases} 203 & 0 \leq v_t < 77 \text{ km/h} \\ 0.1343v_t^2 - 25.07v_t + 1300 & 77 < v_t \leq 80 \text{ km/h} \end{cases}^{\circ}$$

6.3 各速度下阻力大小的确定

阻力大小分为两部分：基本阻力和附加阻力。其中基本阻力是由于机械摩擦，空气摩擦等因素作用而产生的固有阻力，而附加阻力则是由于在附加条件下（通过坡道、曲线、隧道）运行所增加的阻力。

(1) 单位基本阻力 $w_0(t)$

由于产生基本阻力的因素极为负责并且相互影响，很难用理论公式来进行计算，因此采用如下经验公式进行计算单位基本阻力：

$$w_0(t) = 2.031 + 0.0622v_t + 0.001807v_t^2$$

(2) 单位附加阻力 $w_1(t)$

附加阻力分为坡道附加阻力 $w_i(t)$ 和曲线附加阻力 $w_c(t)$ 两部分。其中坡道附加阻力 $w_i(t)$ 是列车上下坡时重力在列车运行方向上的一个分力，用公式 $w_i(t) = i_t$ 来计算，其中 i_t 为该时刻列车所在位置的线路坡度。曲线附加阻力 $w_c(t)$ 取决于轨道线路的曲率半径，用公式 $w_c(t) = c/R_t$ 计算，其中 $c = 600$ ， R_t 为列车该时刻所处位置的曲率半径。注：当曲率半径为 0 时，表示轨道是一条直线，曲线附加阻力也为 0。由此可以得到单位附加阻力 $w_1(t) = w_i(t) + w_c(t)$ 。

综上所述，列车在 t 时刻运行的总阻力如下所示：

$$W(t) = [w_0(t) + w_1(t)] \times g \times m / 1000$$

其中 g 为重力加速度， m 为列车质量 194.295 吨。

6.4 目标函数的确定

题设给出的目标是使得耗能最少，根据题设可知耗能主要是由于列车行驶需要牵引力，导致发电机就处于耗能状态。查阅相关参考文献^[2]可知在整个运行过程中所消耗的能量 E 由牵引力来决定，如下所示：

$$E = \int_0^{t_{\max}} F(t) v_t dt$$

注： $t_{\max} = 110$ ，因为总的运行时间为 110 秒。

6.5 约束条件的确定

约束条件一：由题目所给的数据中可以知道从 A_6 站到 A_7 站的总路程为 1354m，因此整个过程中行走的总路程 $L_{\max} = 1354 \text{ m}$ ，即：

$$\int_0^{t_{\max}} v_t dt = L_{\max}$$

约束条件二：列车起始时刻和到达 A_7 站时刻的速度均为 0 且在运行的任何时刻速度都不能大于该时刻所处路段的最大速度 \bar{v}_t ，即：

$$\begin{cases} v_0 = v_{t_{\max}} = 0 \\ v_t \leq \bar{v}_t \end{cases}$$

约束条件三：由牛顿学第二定律可知，实际输出的牵引加速度乘以质量等于合外力，即：

$$ma_t = m \frac{dv_t}{dt} = F(t) - B(t) - W(t)$$

6.6 综上所述，建立的单列车两个站点之间的最优化模型为：

$$\begin{aligned} \min \quad & E = \int_0^{t_{\max}} F(t) v_t dt \\ s.t. \quad & \begin{cases} \int_0^{t_{\max}} v_t dt = L_{\max} \\ \begin{cases} v_0 = v_{t_{\max}} = 0 \\ v_t \leq \bar{v}_t \end{cases} \\ ma_t = m \frac{dv_t}{dt} = F(t) - B(t) - W(t) \\ F(t) = k_t F_{\max}(t) \\ B(t) = k_b B_{\max}(t) \\ W(t) = [w_0(t) + w_1(t)] \times g \times m / 1000 \end{cases} \end{aligned}$$

6.7 求初始解

对于上述建立的模型我们考虑采用如下的方法确定初始解：由于路程较短，让列车按照牵引—惰行—制动的方式进行航行，首先将列车按最大的加速度牵引到 50km/h，然后使列车惰行，到要到达终点时按照最大减速度进行制动，由此进行反推可以得到惰行与制动的交叉时间点。此时消耗的能量最少，原因如下：

- (1) 当一辆列车加速或者制动的越慢，那么它就需要更多地时间去到达下一站点，为了以较低的加速度或制动速率走完相同的行程，列车应该加速到更高的速度，这样才可以耗能更少。因此，以最大加速度加速和刹车一定是最节能的。
- (2) 这两种不同的巡航策略（一个获得局部加速，一个获得局部制动）的应用依赖于运行阻力。在大多数情况下，运行阻力是正的，因此需要施加相应的加速度，而只有在非常陡的下坡时运行阻力才有可能变为负值。
- (3) 在惰行的过程中，没有施加牵引力和制动力，列车只能向前滚动，此时不消耗能量。因此，较早的启动惰行可以保存更多地能量。

在求初始解的过程中，按照 Ponryagin 极大值原则^{[3][4]}将目标函数转化为使哈密顿函数取得最大值：

$$H = \frac{p_1}{v_t} \times (F(t) - B(t) - W(t)) + p_2 v_t - F(t) v_t$$

其中 $\frac{dp_1}{ds} = -\frac{\partial H}{\partial v}$ ，可以证明当哈密顿函数取得最大值时 k_t 和 k_b 满足如下所示的关系：

$$\begin{cases} k_t = 1, k_b = 0 & p_1 > v^2 \\ k_t \in [0, 1], k_b = 0 & p_1 = v^2 \\ k_t = 0, k_b \in [0, 1] & p_1 = 0 \\ k_t = 0, k_b = 0 & 0 < p_1 < v^2 \\ k_t = 0, k_b = 1 & p_1 < 0 \end{cases}$$

因此，当 k_t 和 k_b 满足相应的条件时存在有以下 5 种可能的阶段：

$$\begin{cases} k_t = 1, k_b = 0 & \text{以最大加速度加速} \\ k_t \in [0, 1], k_b = 0 & \text{巡航} \\ k_t = 0, k_b \in [0, 1] & \text{巡航} \\ k_t = 0, k_b = 0 & \text{惰行} \\ k_t = 0, k_b = 1 & \text{以最大减速度制动} \end{cases}$$

按照上述方法，利用 MATLAB 软件^[5]（源程序见附录 3）求解，在求解过程中发现可能会出现在列车进行惰行过程中会出现上坡坡度很陡的情况，此时列车可能还未到达终点就已经停止运行。因此，在此处需要加入一个判断，即当加速度小于负 0.04 米/秒²时，就给予施加一个牵引力让其可以继续前行。利用改进的算法求得全程需要的时间为 120 秒，消耗的能量为 5.523 kW·h（即 $1.99 \times 10^7 J$ ），其速度距离曲线如下图 2 所示：

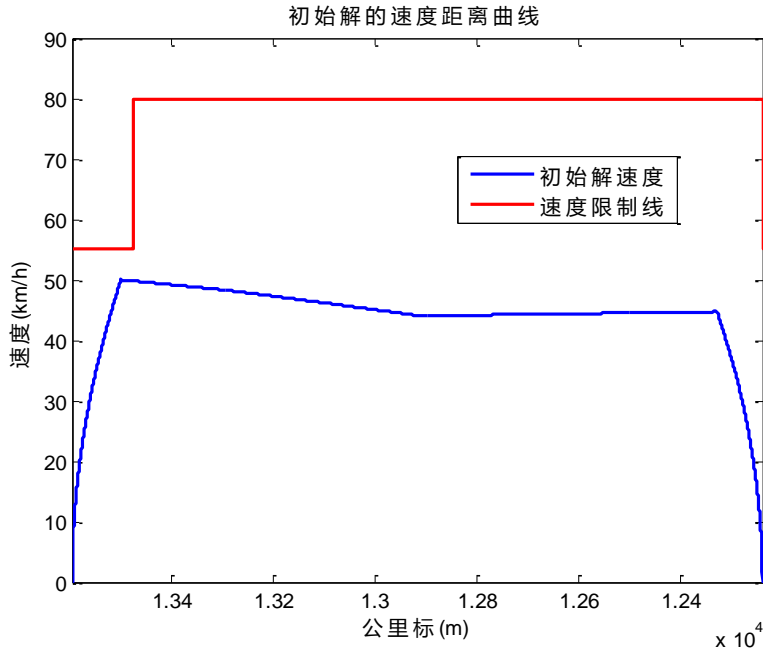


图 2：初始解的速度距离曲线图

6.7 算法简介

显然，题设要求从 A_6 站到 A_7 站所运行的时间仅为 110 秒，与所求得的初始解不一致。故需要考虑适当增加能量的供给以减少运行时间。首先根据各路段所限制的最高速度不同将全程路段分为 2 段，在每一段内增加能量供给可以得到对应的最少的运行时间，其相应的算法^[6]流程图如下图 3 所示：

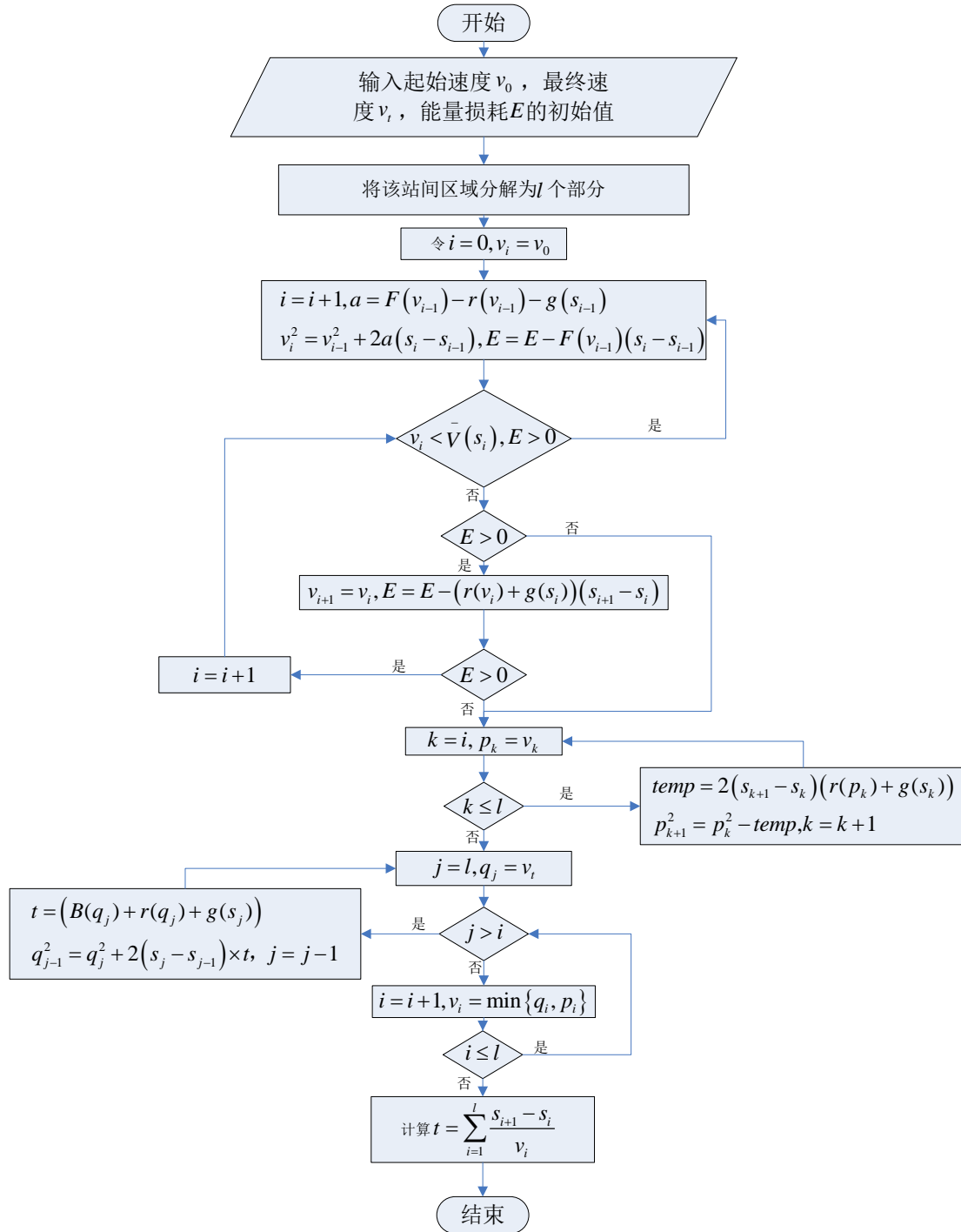


图 3: 最短时间算法流程图

最优能量分配:

根据上述算法, 我们可以知道分别将相同的能量供给到 2 个阶段时, 会得到 2 种不同的最短时间, 选择 2 个中最短的时间, 由此不断的进行下去, 每次取等量的 $\Delta E = 0.03 kW \cdot h$ 能量, 直到总的最短运行时间为 110 秒为止, 其算法^[7]如下所示:

Step1: 计算出上述三部分各自行程时间 T_i 以及总的行程时间 T ;

Step2: 给第 i 段添加单位能量 ΔE 并利用上述定理计算出此时的行程时间 T_i' , 由此可以得到两次的时间差 $\Delta T_i = T_i - T_i'$;

- Step3: 如果 $\Delta T_j = \max\{\Delta T_i\}$, 那么将 ΔE 的能量分配给第 j 部分;
- Step4: 返回新的速度序列和行程时间 T_i' ;
- Step5: 设置 $T_i = T_i'$ 并计算出新的总的行程时间 T' , 如果 $T' < 110$, 重复 Step3;
- Step6: 直到 $T' = 110$ 时返回最终的速度序列以及在每一阶段所花费的时间。

6.8 模型的求解

根据上述建立的最优化模型以及算法, 利用 MATLAB 软件(源程序见附录 4), 求得出所消耗的最少能量为 $6.88kW \cdot h$ (即 $2.48 \times 10^7 J$), 其对应的速度距离曲线图、速度时间曲线图以及时间距离曲线图如下图 4、图 5 和图 6 所示。其相应的曲线数据见附录 1。

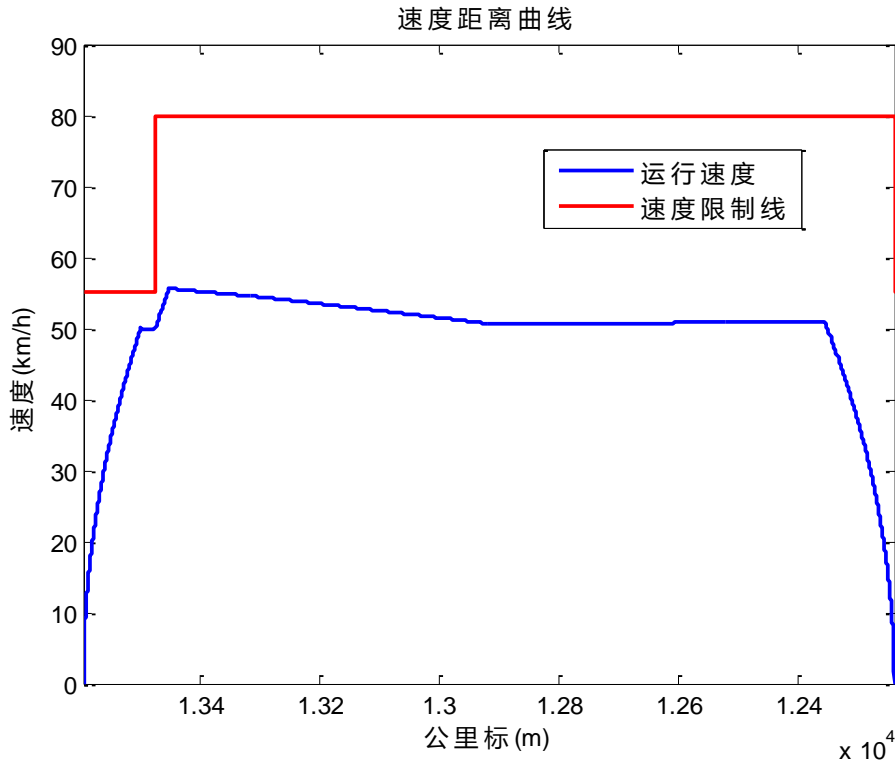


图 4: 最节能速度距离曲线图

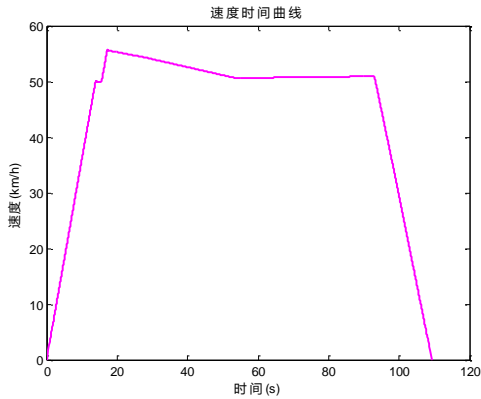


图 5: 最节能速度时间曲线图

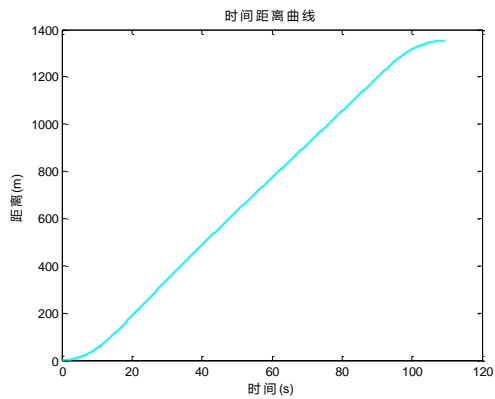


图 6: 时间距离曲线图

分析解得的结果可以知道, 当列车从 A_6 站到 A_7 站的过程中, 整个运动过程根据最高限速分为两个阶段, 每个阶段内部进行牵引—惰行—制动的过程, 其各个运动的开始时间点、结束时间点、开始速度、结束速度以及各点所处的位置分

别如下表 3 所示：

表 3：列车运动过程中的各时间节点

	第一阶段			第二阶段		
	牵引开始	牵引结束	惰行结束	开始牵引	牵引结束	开始制动
时间(s)	0	13.9	15.63	15.63	17.36	93.03
速度(km/h)	0	13.9	13.89	13.89	15.45	14.15
公里标(m)	13594	13497.45	13473.38	13473.38	13447.84	12356.13

6.9 结果分析

根据上述得到的最优结果可以知道，相较于求出的初始解而言，由于初始解只是给了其一个牵引力然后让其自由惰行直至进行制动为止，这样虽然消耗的能量少，但是花费的时间多，不能满足题目要求，因而在增加适当的能量供给后，所消耗的时间比以前减少了 10 秒，从而满足了题目要求。其初始解和最优解对比情况如下所示：

表 4：初始解和最优解对比表

	花费的时间 (s)	消耗的能量 (kW·h)
初始解	120	5.523
最优解	110	6.88

7 问题一：单列车三个站点之间的优化控制问题

第二小问在第一小问的基础上增加了一个站点，列车到达 A_7 站时会停站 45 秒，速度有一个减少为 0 然后又重新牵引的过程，因此相较于问题一而言仅仅只是增加了一些约束条件，目标函数并没有发生变化。

7.1 目标函数的建立

类似于第一小问，在整个运行过程中所消耗的能量 E 由牵引力来决定，如下所示：

$$E = \int_0^{t_{\max}} F(t) v_t dt$$

注： $t_{\max} = 265$ ，因为总的运行时间为 220 秒加上停站的 45 秒。

7.2 约束条件的确定

约束条件一：由题目所给的数据中可以知道从 A_6 站到 A_8 站的总路程为 2634m，因此整个过程中行走的总路程 $L_{\max} = 2634$ m，即：

$$\int_0^{t_{\max}} v_t dt = L_{\max}$$

约束条件二：列车起始时刻、到达 A_7 站、从 A_7 站出发和到达 A_8 站时刻的速度均为 0。假设从 A_6 到 A_7 站的时间为 t_1 ，从 A_7 站到 A_8 站的时间为 t_2 ，显然 $t_1 + t_2 = 220$ 。而且在运行的任何时刻速度都不能大于该时刻所处路段的最大速度 \bar{v}_t ，即：

$$\begin{cases} v_0 = v_{t_1} = v_{t_1+45} = v_{t_{\max}} = 0 \\ v_t \leq \bar{v}_t \end{cases}$$

约束条件三：由牛顿学第二定律可知，实际输出的牵引加速度乘以质量等于

合外力，即：

$$ma_t = m \frac{dv_t}{dt} = F(t) - B(t) - W(t)$$

7.3 综上所述，建立的单列车两个站点之间的最优化模型为：

$$\begin{aligned} \min \quad & E = \int_0^{t_{\max}} F(t) v_t dt \\ \text{s.t.} \quad & \begin{cases} \int_0^{t_{\max}} v_t dt = L_{\max} \\ \begin{cases} v_0 = v_{t_1} = v_{t_1+45} = v_{t_{\max}} = 0 \\ v_t \leq \bar{v}_t \end{cases} \\ ma_t = m \frac{dv_t}{dt} = F(t) - B(t) - W(t) \\ F(t) = k_t F_{\max}(t) \\ B(t) = k_b B_{\max}(t) \\ W(t) = [w_0(t) + w_1(t)] \times g \times m / 1000 \end{cases} \end{aligned}$$

7.4 模型的求解

根据上述建立的模型，利用 MATLAB 软件（源程序见附录 5）求解，求得列车从 A_6 站到 A_8 站所消耗的最少能量为 $13.18 \text{ kW} \cdot \text{h}$ （即 $4.74 \times 10^7 \text{ J}$ ）。其速度距离曲线图、速度时间曲线图以及时间距离曲线图分别如下图 7、图 8 和图 9 所示，其具体的曲线数据见附录 2 所示。其中时间距离曲线是针对于前一个站点而言的，即该距离相对于前一个站点运行的路程。

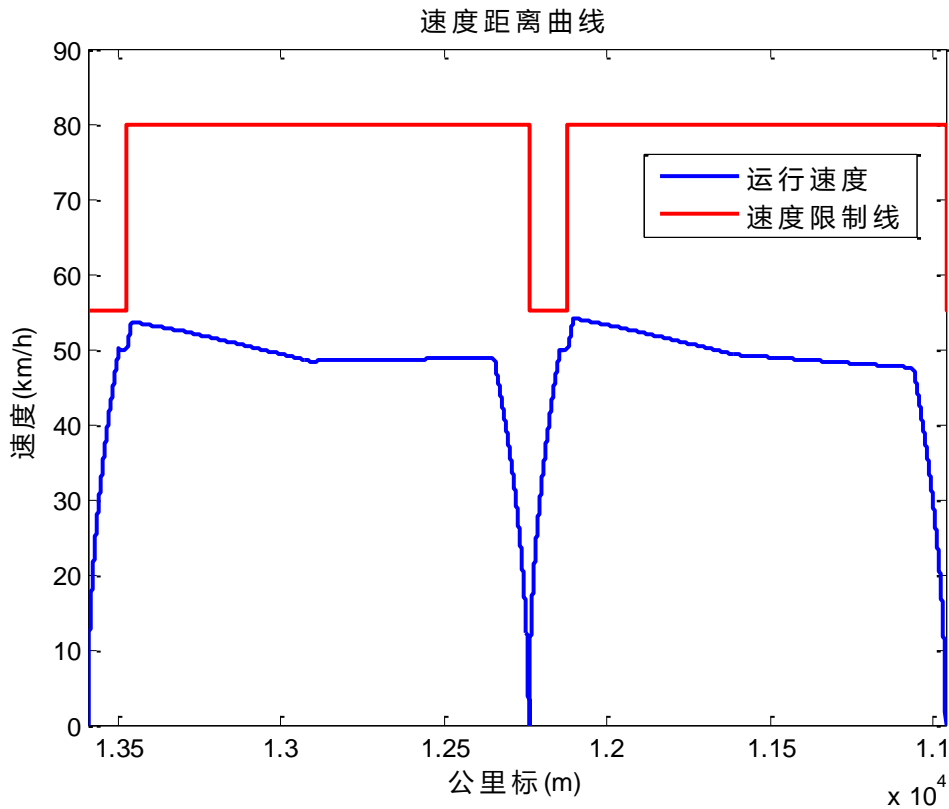


图 7：速度距离曲线图

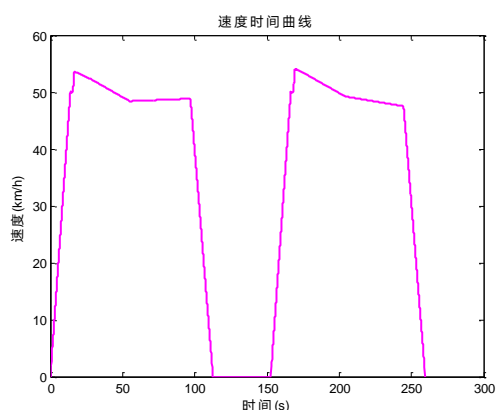


图 8: 速度时间曲线

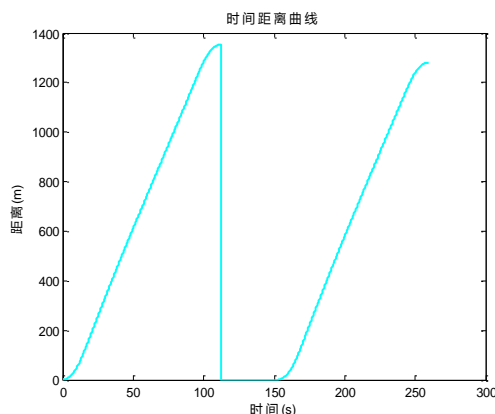


图 9: 时间距离曲线

同样的，类似于第一小问将整个运动过程按照最高限速将整个运动过程分为四个阶段，可以求得四个运动阶段各关键时间点、速度分别如下表所示：

表 5: 运动过程中各关键时间点以及速度

	第一阶段			第二阶段		
	牵引开始	牵引结束	惰行结束	牵引开始	牵引结束	开始制动
时间 (s)	0	13.98	15.59	15.59	16.82	96.27
速度 (km/h)	0	50.02	49.86	49.86	53.64	48.84
公里标 (m)	13594	13496.23	13473.88	13473.88	13456.11	12356.38
	第三阶段			第四阶段		
	牵引开始	牵引结束	惰行结束	牵引开始	牵引结束	开始制动
时间 (s)	152.6	166.5	168.2	168.2	169.4	243.1
速度 (km/h)	0	49.99	49.82	49.82	54.11	47.61
公里标 (m)	12240	12143.61	12120	12120	12102.60	11079.03

7.5 结果分析

此小问是在第一小问的基础上增加考虑了一个站点，列车从 A_6 站到 A_8 站，途径 A_7 站，路程由原来的 1534m 增加到了 2814m，总的运行时间变为原来的两倍，但能量仅增加了 $6.3kW \cdot h$ 。通过对比以下结果可知，求出的结果较为合理。

表 6: 对比结果表

	时间	路程	能量
$A_6 - A_7$	110	1534m	$6.88kW \cdot h$
$A_6 - A_8$	220	2814m	$13.18kW \cdot h$

8 问题二：多列车节能运行优化控制问题

根据上述问题分析可以知道，解决此问题分为两部分：首先考虑单个列车从 A_1 站开往 A_4 的最节能运行的速度距离曲线；然后再考虑多列车之间再生能量的相互利用原理使得总的能量达到最小。

8.1 单列车多站点间的运行优化控制问题

利用问题一的第二小问中的模型，增加各站点的停留时间，此时各站的停留时间是不固定的，但是总运行时间不变，均为 2086s（包括停站时间）。因此可

建立类似于问题一的模型。

8.1.1 目标函数的确定

同样地，在整个运行过程中所消耗的能量 E 由牵引力来决定，如下所示：

$$E = \int_0^{t_{\max}} F(t) v_t dt$$

其中 $t_{\max} = 2086$ 。

8.1.2 约束条件的确定

约束条件一：由题目所给的数据中可以知道从 A_1 站到 A_{14} 站的总路程为 22728m，因此整个过程中行走的总路程 $L_{\max} = 22728$ m，即：

$$\int_0^{t_{\max}} v_t dt = L_{\max}$$

约束条件二：列车到达任一站点、从任一出发点的速度均为 0。总的行驶时间加上停留时间之和为 2086s，且下一站的到达时间为前面所经所有站点花费的时间和停留的时间之和，即：

$$\begin{cases} \sum_{i=1}^{14} t_i + 13x = 2086 \\ a_{i+1} = \sum_{k=2}^{i+1} t_k + (i-1)x \end{cases}$$

约束条件三：列车起始时刻、到达任意一站的时刻以及从任意一站出发时刻的速度均为 0，而且在运行的任何时刻速度都不能大于该时刻所处路段的最大速度 \bar{v}_t ，即：

$$\begin{cases} v_0 = v_{t_{\max}} = 0 \\ v_t \leq \bar{v}_t \end{cases}$$

约束条件四：由牛顿学第二定律可知，实际输出的牵引加速度乘以质量等于合外力，即：

$$ma_t = m \frac{dv_t}{dt} = F(t) - B(t) - W(t)$$

8.1.3 综上所述，建立的单列车度站点之间的最优化模型为：

$$\begin{aligned} \min \quad & E = \int_0^{t_{\max}} F(t) v_t dt \\ \text{s.t.} \quad & \begin{cases} \int_0^{t_{\max}} v_t dt = L_{\max} \\ \begin{cases} v_0 = v_{t_{\max}} = 0 \\ v_t \leq \bar{v}_t \end{cases} \\ ma_t = m \frac{dv_t}{dt} = F(t) - B(t) - W(t) \\ \begin{cases} \sum_{i=1}^{14} t_i + 13x = 2086 \\ a_{i+1} = \sum_{k=2}^{i+1} t_k + (i-1)x \end{cases} \end{cases} \end{aligned}$$

8.1.4 模型的求解

根据上述建立的模型,利用 MATLAB 软件求解,求得一辆列车从 A_1 站开往 A_{14} 站所消耗的最少能量为 $131.12kW \cdot h$ (即 $4.72 \times 10^8 J$), 在每个站点的停留时间为 38 秒。

8.2 多列车多站点的运行优化控制问题

运用上述模型求出了列车在运动状态中的最节能的速度距离曲线, 由于每辆的运行状态都是相同的, 因此此时要想达到更加节能的效果只能通过控制发车间隔来实现。

8.2.1 再生能量的利用原理

根据题设可知, 列车在制动时会产生能量 E_{reg} , 如果有其余车辆处于加速状态, 其可以利用能量 E_{reg} 。产生的再生能量为:

$$E_{reg} = (E_{mech} - E_f) \cdot 95\%。$$

其中 $E_{mech} = \frac{1}{2}m(v_i^2 - v_0^2)$, $E_f = W \cdot s$ (s 为运动的距离)。

那么产生的再生能量中被利用的部分 E_{used} 为:

$$E_{used} = E_{reg} \cdot \frac{t_{overlap}}{t_{brake}}$$

其中 $t_{overlap}$ 是列车的制动时间与其他加速时间的重叠时间, t_{brake} 为列车的制动时间。

由此便可知道对于正在加速的列车而言, 如果使用过再生能量后那么其消耗的发动机供应的能量 E' 仅为:

$$E' = E - E_{used} = E - E_{reg} \cdot \frac{t_{overlap}}{t_{brake}}$$

8.2.2 目标函数的确定

从问题分析中可以知道要想使得总能量最小, 应该尽可能多的让当某车在制动时其余的车在加速, 这样就应该使所有列车站的总重叠的时间^[8]达到最大。现分析可能出现的重叠时间情况如下图所示:

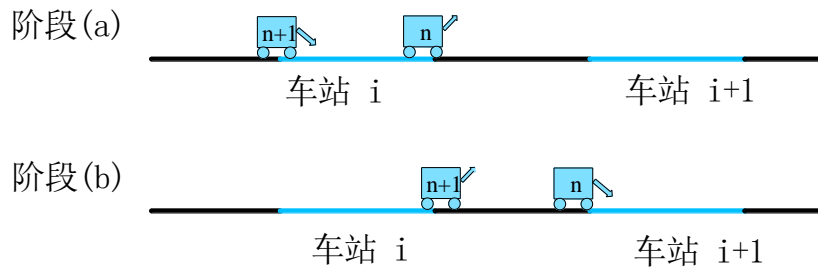


图 10: 列车出现使用再生能量的情况

分析上图不难看出, 可以节省能量的情况有 2 种: 第一种是在第 $n+1$ 辆车刚准备减速进入第 i 个站点时第 n 辆车刚好要加速驶离第 i 个站点, 此时第 n 辆车可以利用第 $n+1$ 辆车产生的能量; 第二种情况是第 n 辆车刚准备减速驶入第 $i+1$ 个站点时第 $n+1$ 辆车刚准备加速驶离第 i 个站点, 此时第 $n+1$ 辆车可以利用第 n 辆车产生的能量。由于所有列车处于同一供电区段, 所以所有列车只要满足如上

所示的关系均可节省能量。

根据上述分析不难发现：当第 n 辆列车从第 i 个站点驶向第 $i+1$ 个站点时，可以将该过程分为 3 部分，牵引时间段 $[d_i^n, d_i^{n+}]$ ，惰行时间段 $[d_i^{n+}, a_{i+1}^{n-}]$ 和制动时间段 $[a_{i+1}^{n-}, a_{i+1}^n]$ 。当第 $n+1$ 辆列车从第 $i-1$ 个站点驶向第 i 个站点的时间段为 $[d_{i-1}^{n+1}, a_i^{n+1}]$ ，从时间点 d_i^{n+1} 到时间点 a_{i+1}^{n+1} 表示第 $n+1$ 辆列车从第 i 个站点驶向第 $i+1$ 个站点。按照上述分析我们可以将两车出现时间重叠的情况绘制成如下所示的图 11 所示，其中红色部分表示列车正在进行牵引，绿色表示正在进行制动，灰色打点部分代表牵引和制动重叠的时间部分。

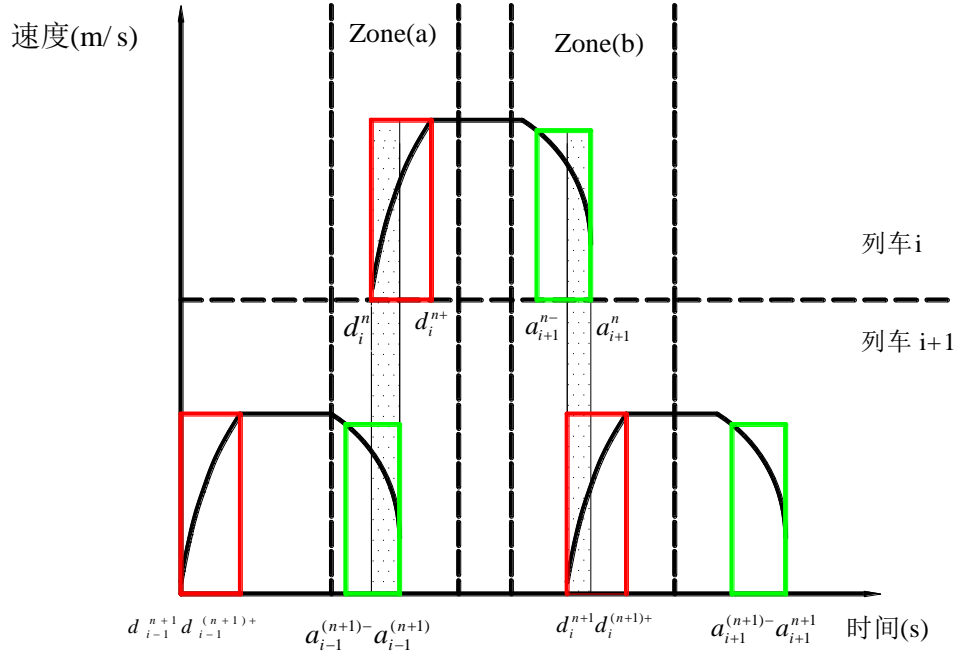


图 11：重叠时间图

分析上图可以得到第 n 辆列车和第 $n+1$ 辆列车在第 i 个站点处两种出现重叠情况的时间如下所示：

$$F_1(a, d, n, i) = \begin{cases} 0 & d_i^{n+} \leq a_i^{n+1} \\ \min\{t_a^i, d_i^{n+} - a_i^{(n+1)-}\} & a_i^{(n+1)-} \leq d_i^n \leq a_i^{n+1} \\ \min\{t_b^{i-1}, a_i^{n+1} - d_i^n\} & d_i^n \leq a_i^{(n+1)-} \leq a_i^{n+1} \leq d_i^{n+} \\ 0 & a_{i+1}^n < d_i^{n+1} \end{cases}$$

$$F_2(a, d, n, i) = \begin{cases} 0 & d_i^{(n+1)+} \leq a_i^{n-} \\ \min\{t_a^i, d_i^{n+1} - a_{i+1}^{n-}\} & a_i^{n-} \leq d_i^{(n+1)+} \leq a_{i+1}^n \\ \min\{t_b^{i-1}, a_{i+1}^n - d_i^{n+1}\} & d_{i+1}^n \leq a_{i+1}^n \leq d_i^{(n+1)+} \\ 0 & a_{i+1}^n < d_i^{n+1} \end{cases}$$

由此可以得到总的重叠时间为：

$$F(a, d) = \sum_{n=1}^{100} \sum_{i=1}^{14} [F_1(a, d, n, i) + F_2(a, d, n, i)]$$

8.2.3 约束条件的确定

约束条件一：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。即：

$$\sum_{n=1}^{100} h_n = 63900$$

约束条件二：列车进行追踪时，后车的速度不能超过限制速度：

$$v_t^{n+1} \leq \min \left\{ \bar{v}_t, \sqrt{2L_t B_e} \right\}$$

其中 L_t 是列车当前时刻前后车之间的距离， B_e 是列车制动的最大减速度。

约束条件三：中间各列车在各个车站停站最少 $D_{\min} = 30$ 秒，最多 $D_{\max} = 45$ 秒。即：

$$30 \leq x \leq 45$$

约束条件四：每辆列车的发车间隔的变化范围是 $H_{\min} = 120$ 秒至 $H_{\max} = 660$ 秒，即：

$$120 \leq h_n \leq 660$$

约束条件五：对于任意一辆列车而言，在任意一站的出站时间和进站的时间之差为一个固定的停车时间 x ，即：

$$d_i^n - a_i^n = x$$

8.2.4 综上所述，建立的模型为：

$$\begin{aligned} \max F(a, d) &= \sum_{n=1}^{100} \sum_{i=1}^{14} [F_1(a, d, n, i) + F_2(a, d, n, i)] \\ s.t. \left\{ \begin{array}{l} F_1(a, d, n, i) = \begin{cases} 0 & d_i^{n+} \leq a_i^{n+1} \\ \min \{t_a^i, d_i^{n+} - a_i^{(n+1)-}\} & a_i^{(n+1)-} \leq d_i^n \leq a_i^{n+1} \\ \min \{t_b^{i-1}, a_i^{n+1} - d_i^n\} & d_i^n \leq a_i^{(n+1)-} \leq a_i^{n+1} \leq d_i^{n+} \\ 0 & a_i^n < d_i^{n+1} \end{cases} \\ v_t^{n+1} \leq \min \left\{ \bar{v}_t, \sqrt{2L_t B_e} \right\} \\ 30 \leq x \leq 45 \\ \sum_{n=1}^{100} h_n = 63900 \\ 120 \leq h_n \leq 660 \\ d_i^n - a_i^n = x \\ F_2(a, d, n, i) = \begin{cases} 0 & d_i^{(n+1)+} \leq a_i^{n-} \\ \min \{t_a^i, d_i^{n+1} - a_{i+1}^{n-}\} & a_i^{n-} \leq d_i^{(n+1)+} \leq a_{i+1}^n \\ \min \{t_b^{i-1}, a_{i+1}^n - d_i^{n+1}\} & d_{i+1}^n \leq a_{i+1}^n \leq d_i^{(n+1)+} \\ 0 & a_{i+1}^n < d_i^{n+1} \end{cases} \end{array} \right. \end{aligned}$$

8.2.5 遗传算法

根据上述建立的模型，在此处考虑使用遗传算法^{[9][10]}进行求解。遗传算法（Genetic Algorithm）是一类借鉴生物界的进化规律（适者生存，优胜劣汰遗传机制）演化而来的随机化搜索方法，它是由美国的 J.Holland 教授 1975 年首先提出。由于其广泛的通用性，鲁棒性强，高效率，和实用性，GA 已广泛地被用于解决整数规划问题和对交通系统的研究。在本文中，我们采用遗传算法来解决提出的模型。

遗传算法首先随机产生带有染色体的种群。在种群中每个染色体通过适应度函数的计算会有一个生存概率。遗传算法根据这个概率选择染色体，再通过交叉变异等操作来产生子代。该算法不断地产生新的子代直到满足收敛要求。为简单起见，本文采用二进制编码方式。一个染色体 $C = (c_1, c_2, \dots, c_N)$ 表示一个解 (h, x_1, \dots, x_{N-1}) ，其中 c_i 表示二进制编码中的第 i 决策变量。以上过程可以总结为以下算法：

(1)初始化：随机产生 pop_size 个可行的染色体。如果一个染色体对应的解满足约束条件，它就是可行的。初始化算法的方法如下。

Step1. 令 $i = 1$ 。

Step2. 随机产生 N 个二进制向量 c_1, c_2, \dots, c_N ，使得 $C_i = (c_1, c_2, \dots, c_N)$ 是一个可行染色体。

Step3. 如果 $i = pop_size$ ，停止；否则 $i = i + 1$ 并进行 step2。

(2)计算适应度：适应度函数用来评估每个染色体适应能力。本文使用目标函数作为适应度函数，即：

$$Eval(C) = F(h, x)$$

最大的适应度意味着最好的个体。

(3)选择过程：染色体的选择是通过旋转轮盘完成的，这是一种根据适应度比例的选择方法。每次选择一个染色体作为新的子群。重复这个过程 pop_size 次，可以得到新一代。不失一般性，假定染色体已经根据该评价函数值排序。令

$p_0 = 0$ ， $p_i = \sum_{j=1}^i Eval(C_j)$ 。然后按照下面的算法进行选择。

Step1. 令 $j = 1$ 。

Step2. 随机产生一个数， $r \in (0, p_{pop_size}]$ 。

Step3. 选择满足 $r \in (p_{i-1}, p_i]$ 的染色体 C_i 。

Step4. 如果 $j \geq P_{pop_size}$ ，停止；否则，令 $j = j + 1$ ，并进行 step2。

(4)交叉操作：假设交叉操作的概率是 P_c 。随机选择两个父母染色体 C_i 和 C_j ，并随机产生一个实数 $r \in [0, 1]$ 。如果 $r < P_c$ ， C_i 和 C_j 通过交叉操作产生两个新的染色体 X 和 Y 。

(5)变异操作：假设交叉操作的概率是 P_m 。随机选择一个染色体 C_i 作为变异的父本并随机产生实数 $s \in [0, 1]$ 。如果 $s < P_m$ ，随机选择该染色体的一个基因上的一个位置。如果选择位是 0（或 1），用 1（或 0）来代替它。

其算法流程图如下所示：

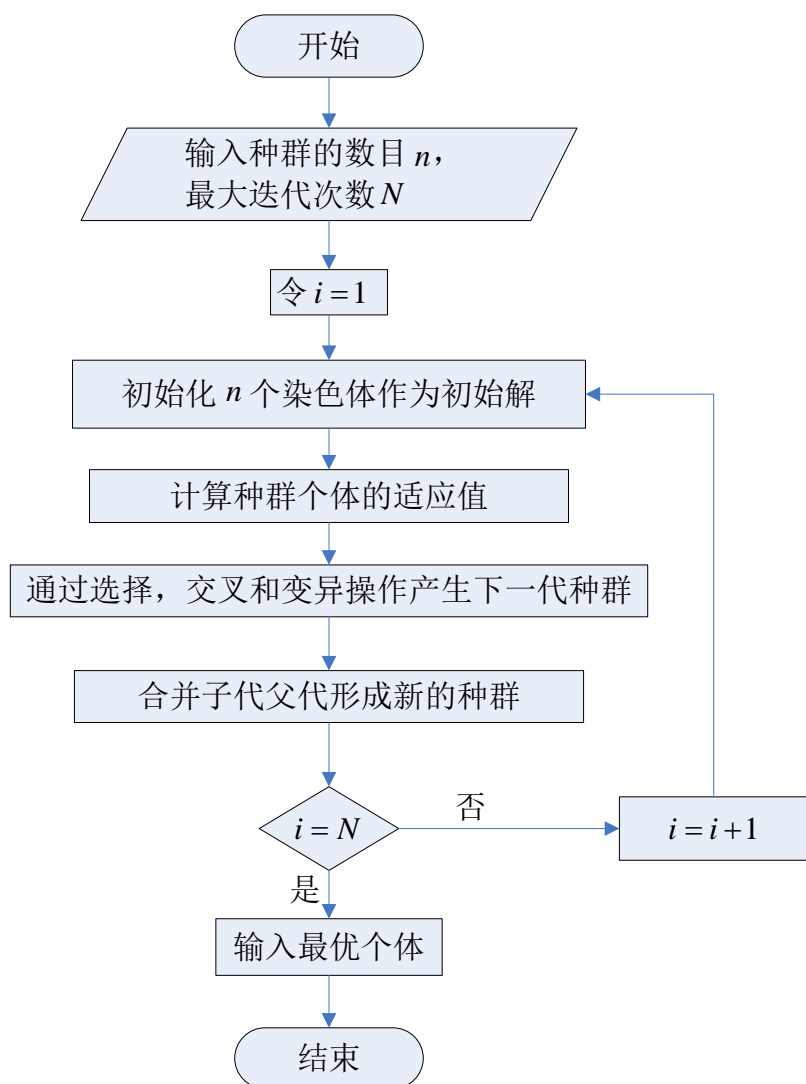


图12: 遗传算法流程图

8.2.6 模型的求解

根据上述建立的模型, 采用遗传算法, 利用 MATLAB 软件 (源程序见附录 6), 再根据再生能量利用原理可知 100 辆车从 A_1 站开往 A_{14} 站所消耗的最少总能量为 $5336.35kW \cdot h$ (即 $1.92 \times 10^{10} J$)。其部分车辆的发车间隔如下表所示 (具体结果见附录 7):

表 7: 部分车辆发车间隔

序号	h1	h2	h3	h4	h5	h6	h7	h8	h9	h10	h11
发车间隔	647	647	649	648	648	648	649	648	649	650	650
序号	h12	h13	h14	h15	h16	h17	h18	h19	h20	h21	h22
发车间隔	648	646	648	650	647	648	649	394	647	648	647
序号	h78	h79	h80	h81	h82	h83	h84	h85	h86	h87	h88
发车间隔	644	648	649	649	650	647	649	649	648	649	650
序号	h89	h90	h91	h92	h93	h94	h95	h96	h97	h98	h99
发车间隔	647	650	649	648	648	648	648	649	656	649	650

8.2.7 结果分析

根据上述求得的结果，当单个列车从起始站运行到终点站时所消耗的能量为 $131.12kW \cdot h$ ，理论上 100 辆列车运行应该消耗 $13112kW \cdot h$ 的能量，但是经过调整发车间隔以后所消耗的总能量仅为 $5336.35kW \cdot h$ ，节约了 $7745.65kW \cdot h$ 。与原来相比减少了 59% 的能量消耗，由此证明该模型大大的实现了节约能量的目的，由此证明了模型的合理性。

9 问题二：考虑高峰期的多列车多站点的运行优化控制问题

同样地，类似于第一小问，假设 240 辆车在运行过程中的运动状态是一致的，运用问题一中的模型可求解出和第一小问相同的每辆车的速度距离曲线。求解出来以后再考虑发车间隔和停车时间的问题。

9.1 目标函数的确定

此小问和第一小问的思路相同，但是要增加考虑一个限制条件：高峰期的发车间隔问题。根据假设 9 可知列车从早上 5:30 开始发车，而早高峰为 7200 秒至 12600 秒，晚高峰为 43200 至 50400 秒，因此高峰期在早上 7:30 至 9:00，下午 17:30 至 19:30。列车每天要发车 240 列，要在 24:00 之前完成全部发车工作。在此问中对于列车的发车情况分以下两种情况进行讨论：

9.1.1 高峰时期

高峰时期由于人们对列车的需求较多，因此考虑较小的发车间隔。类似于第一小问，当第 n 辆列车从第 i 个站点驶向第 $i+1$ 个站点时，可以将该过程分为 3 部分：牵引时间段 $[d_i^n, d_i^{n+}]$ ，惰行时间段 $[d_i^{n+}, a_{i+1}^{n-}]$ 和制动时间段 $[a_{i+1}^{n-}, a_{i+1}^n]$ 。当第 $n+1$ 辆列车从第 $i-1$ 个站点驶向第 i 个站点的时间段为 $[d_{i-1}^{n+1}, a_i^{n+1}]$ ，从时间点 d_i^{n+1} 到时间点 a_{i+1}^{n+1} 表示第 $n+1$ 辆列车从第 i 个站点驶向第 $i+1$ 个站点。同样地，第 $n+1$ 辆列车在第 i 个站点的停留时间为 $w_i^{n+1} = d_i^{n+1} - a_i^{n+1}$ 。

$$F_1(a, d, n, i) = \begin{cases} 0 & d_i^{n+} \leq a_i^{n+1} \\ \min\{t_a^i, d_i^{n+} - a_i^{(n+1)-}\} & a_i^{(n+1)-} \leq d_i^n \leq a_i^{n+1} \\ \min\{t_b^{i-1}, a_i^{n+1} - d_i^n\} & d_i^n \leq a_i^{(n+1)-} \leq a_i^{n+1} \leq d_i^{n+} \\ 0 & a_{i+1}^n < d_i^{n+1} \end{cases}$$

$$F_2(a, d, n, i) = \begin{cases} 0 & d_i^{(n+1)+} \leq a_i^{n-} \\ \min\{t_a^i, d_i^{n+1} - a_{i+1}^{n-}\} & a_i^{n-} \leq d_i^{(n+1)+} \leq a_{i+1}^n \\ \min\{t_b^{i-1}, a_{i+1}^n - d_i^{n+1}\} & d_{i+1}^n \leq a_{i+1}^n \leq d_i^{(n+1)+} \\ 0 & a_{i+1}^n < d_i^{n+1} \end{cases}$$

由此可以得到总的重叠时间为：

$$F_B(a, d) = \sum_{n=1}^{240} \sum_{i=1}^{14} [F_1(a, d, n, i) + F_2(a, d, n, i)]$$

9.1.2 非高峰时期

在这种情况下，乘客们对列车的需求比较小，因此会导致较大的发车间隔。这时候也会出现两种可能的重叠时间。第一种是在第 n 辆车刚准备加速离开第 $i+1$ 个站点时第 $n+1$ 辆车刚好要减速驶入第 i 个站点，此时第 n 辆车可以利用第 $n+1$ 辆车产生的能量；经过一段时间的运行后第 n 辆车刚准备减速驶入第 $i+2$ 个站点时第 $n+1$ 辆车刚准备加速驶离第 i 个站点，此时第 $n+1$ 辆车可以利用第 n 辆车产生的能量。如下图所示：

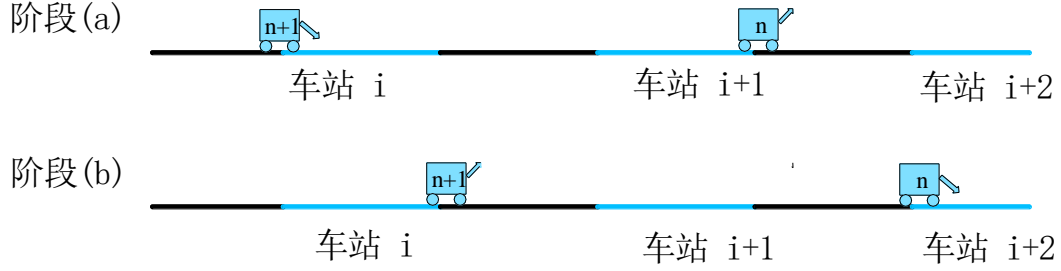


图 14：非高峰期可利用再生能量情况图

上述情况是可以通过控制发车间隔来实现的。同样地，类似于高峰时期，我们可以求出两种情况下的重叠时间分别为：

$$F_3(a, d, n, i) = \begin{cases} 0 & d_{i+1}^{n+} \leq a_i^{(n+1)-} \\ \min\{t_a^i, d_{i+1}^{n+} - a_i^{(n+1)-}\} & a_i^{(n+1)-} \leq d_{i+1}^{n+} \leq a_i^{n+} \\ \min\{t_b^{i-1}, a_i^{n+1} - d_i^n\} & d_{i+1}^n \leq a_i^{n+1} < d_{i+1}^{n+} \\ 0 & a_i^{n+1} < d_{i+1}^n \end{cases}$$

$$F_4(a, d, n, i) = \begin{cases} 0 & d_i^{(n+1)+} \leq a_{i+2}^{n-} \\ \min\{t_a^i, d_i^{(n+1)+} - a_{i+2}^{n-}\} & a_{i+2}^{n-} \leq d_i^{(n+1)+} \leq a_{i+2}^n \\ \min\{t_b^{i-1}, a_{i+2}^n - d_i^{n+1}\} & d_i^{n+1} \leq a_{i+2}^n \leq d_i^{(n+1)+} \\ 0 & a_{i+2}^n < d_i^{n+1} \end{cases}$$

也可得到总的重叠时间为：

$$F_L(a, d) = \sum_{n=1}^{240} \sum_{i=1}^{14} [F_3(a, d, n, i) + F_4(a, d, n, i)]$$

综上所述，两个时间段的重叠时间之和即为在整个运行过程中的总的重叠时间，即为：

$$F(a, d) = F_B(a, d) + F_L(a, d)$$

9.2 约束条件的确定

约束条件一： 由于最后一列列车的发车时间以及到达 A_{14} 的时间必须在当天 24:00 以前，从早上 5:30 到晚上 24:00 一共有 66600 秒。即：

$$\begin{cases} \sum_{n=1}^{240} h_n \leq 66600 \\ a_{240}^{14} \leq 66600 \end{cases}$$

约束条件二： 列车进行追踪时，后车的速度不能超过限制速度：

$$v_t^{n+1} \leq \min \left\{ \bar{v}_t, \sqrt{2L_t B_e} \right\}$$

其中 L_t 是列车当前时刻前后车之间的距离， B_e 是列车制动的最大减速度。

约束条件三：中间各列车在各个车站停站最少 $D_{\min} = 30$ 秒，最多 $D_{\max} = 45$ 秒。

即：

$$30 \leq x_i^n \leq 45$$

约束条件四：发车间隔分为两部分：高峰时间的列车发车间隔的变化范围是 $H_{\min} = 120$ 秒至 $H_{\max} = 150$ 秒，其余时间发车间隔不少于 $H_{\min} = 300$ 秒，即：

$$\begin{cases} 120 \leq h_t \leq 150 & 7200 \leq t \leq 12600 \cup 43200 \leq t \leq 50400 \\ h_t \geq 300 & \text{其它时间} \end{cases}$$

约束条件五：对于任意一辆列车而言，在任意一站的出站时间和进站的时间之差为一个固定的停车时间 x ，即：

$$d_i^n - a_i^n = x$$

9.3 综上所述，建立的模型为：

$$\begin{aligned} \max F(a, d) &= F_B(a, d) + F_L(a, d) \\ s.t. \quad &\begin{cases} F_B(a, d) = \sum_{n=1}^{240} \sum_{i=1}^{14} [F_1(a, d, n, i) + F_2(a, d, n, i)] \\ F_L(a, d) = \sum_{n=1}^{240} \sum_{i=1}^{14} [F_3(a, d, n, i) + F_4(a, d, n, i)] \\ \sum_{n=1}^{240} h_n \leq 66600 \\ a_{240}^{14} \leq 66600 \\ v_t^{n+1} \leq \min \left\{ \bar{v}_t, \sqrt{2L_t B_e} \right\} \\ 30 \leq x_i^n \leq 45 \\ \begin{cases} 120 \leq h_t \leq 150 & 7200 \leq t \leq 12600 \cup 43200 \leq t \leq 50400 \\ h_t \geq 300 & \text{其它时间} \end{cases} \end{cases} \end{aligned}$$

9.4 模型的求解

根据上述建立的模型，利用 MATLAB 软件求解（源程序见附录 8），求得考虑高峰期后 240 辆列车消耗的总能量为 $11296.52kW \cdot h$ （即 $4.07 \times 10^{10} J$ ），部分列车的发车时间间隔如下表所示（具体结果见附录 9）：

表 8：考虑高峰期时部分列车发车间隔

序号	h1	h2	h3	h4	h5	h6	h7	h8	h9	h10	h11
发车间隔	382	388	309	336	343	382	341	354	343	356	320
序号	h12	h13	h14	h15	h16	h17	h18	h19	h20	h21	h22
发车间隔	322	322	357	348	352	348	381	303	355	353	120
序号	h101	h102	h103	h104	h105	h106	h107	h108	h109	h110	h111
发车间隔	314	366	322	359	336	324	369	321	362	314	348
序号	h112	h113	h114	h115	h116	h117	h118	h119	h120	h121	h122
发车间隔	341	376	357	378	388	312	388	377	317	374	347

其相应的列车全程的速度距离曲线图如下图所示：

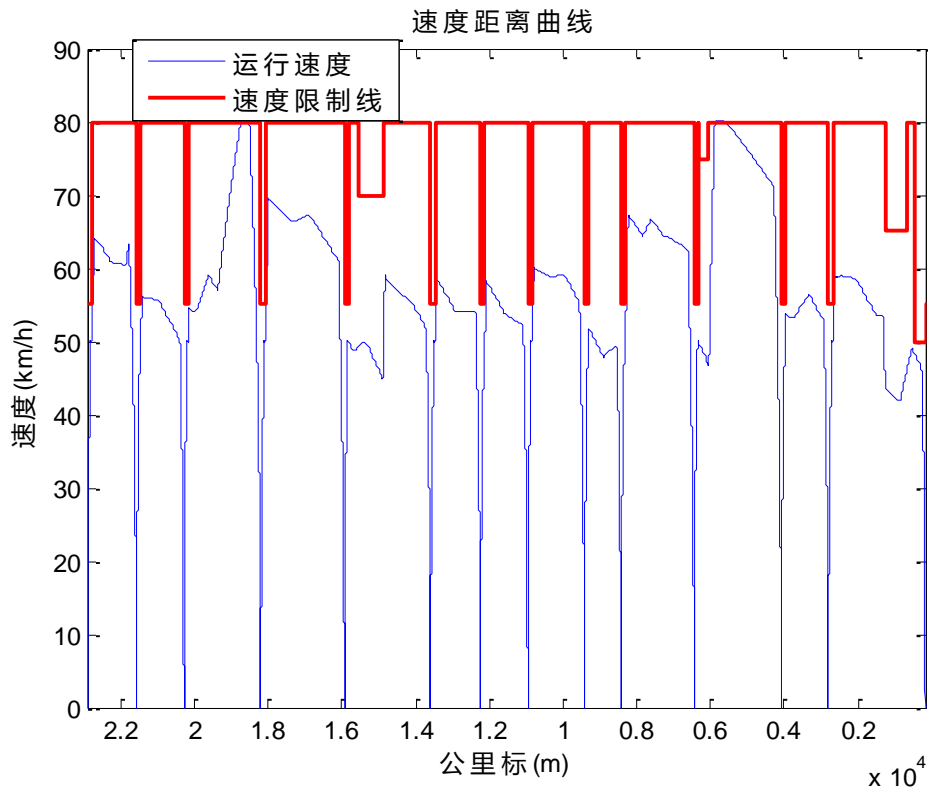


图 13：速度距离曲线图

9.5 结果分析

将上述结果与第一小问的结果进行对比，由于在一天时间内需要完成 240 辆列车的发车，因此列车的发车间隔相比以前有所减少，由以前 645 秒的平均发车间隔减少为现在的 267 秒的平均发车间隔，减少了 58%的发车间隔时间，由此来缓解高峰期乘客对于列车的需求。

10 问题三：单列车延误 10 秒钟后的运行优化控制问题

从问题分析中可以看出当某辆列车出现晚点 10 秒后不会与其它列车出现相撞的情况，因此仅对该列车的运行情况进行调整即可。列车的晚点示意图如下图 14 所示，分析可知由于晚点时间不长，不需要调整后面车辆的发车计划，此时调整需要满足三个条件：一是在接下来的运行过程中应该使所有列车的到站时间与原计划到站时间之差尽可能少，二是使得在达到减少 10 秒钟的目的时列车经过的站点尽可能少，三是在整个运行过程中消耗的能量最少。

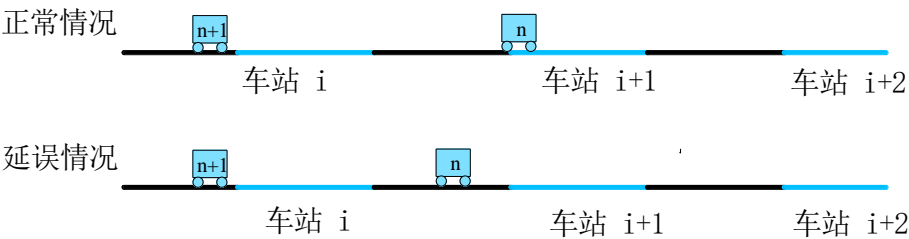


图 14：列车晚点示意图

10.1 确定目标函数

目标函数一：由上述分析可知，如果第 n 辆列车在第 j 个站点延误 10 秒发车，则从第 j 个站点开始以后各站点的到站时间、离站时间均会发生变化。直到第 N 个站点列车才恢复正常运行，由于题设要求列车尽快恢复正点运行，因此途径的站点 y 应该尽可能少，即：

$$\min y = N - j$$

目标函数二：经过调整后列车的到站时间与原时间相比总延时最少，即到达最后站点的时间与原来的时间差应该尽可能少：

$$\min \sum_{i=j}^{14} (a_{i+1}^n - a_{i+1}^n)$$

目标函数三：对出现延迟的列车在调整列车的出行计划时耗能应该尽可能少，即：

$$\min E = \int_0^{t_{\max}} F(t) v_t dt$$

10.2 确定约束条件

约束条件一：在列车恢复正常运行期间，停站时间应该保持不变，即：

$$d_{i+1}^n - a_{i+1}^n = x (i = j, j+1 \dots 14)$$

约束条件二：列车出现晚点后在追赶过程中该车的最大速度不能超过限制速度：

$$v_t^n \leq \min \{ \bar{v}_t, \sqrt{2L_t B_e} \}$$

其中 L_t 是列车当前时刻前后车之间的距离， B_e 是列车制动的最大减速度。

约束条件三：对于任意列车到站时间不能早于原计划时间，即：

$$a_{i+1}^n \geq a_{i+1}^n$$

约束条件四：对于晚点的列车而言，在第 N 个站点恢复正常运行，因此在第 $j+1$ 站点到第 N 个站点之间列车总的行驶时间与原行驶时间相比少了 10 秒，即：

$$\sum_{i=j+1}^n (a_{i+1}^n - d_i^n) = \sum_{i=j+1}^n (a_{i+1}^n - d_i^n) - 10$$

10.3 综上所述，建立的多目标优化模型为：

$$\begin{cases} \min y = N - j \\ \min \sum_{i=j}^{14} (a_{i+1}^n - a_{i+1}^n) \\ \min E = \int_0^{t_{\max}} F(t) v_t dt \\ s.t. \begin{cases} d_{i+1}^n - a_{i+1}^n = x (i = j, j+1 \dots 14) \\ v_t^n \leq \min \{ \bar{v}_t, \sqrt{2L_t B_e} \} \\ a_{i+1}^n \geq a_{i+1}^n \\ \sum_{i=j+1}^n (a_{i+1}^n - d_i^n) = \sum_{i=j+1}^n (a_{i+1}^n - d_i^n) - 10 \end{cases} \end{cases}$$

10.4 模型的求解

根据上述建立的模型，首先采用乘除法将多目标转化为单目标，然后利用 MATLAB 软件求解（源程序见附录 10），求解得到当列车在不同站延误时，进行调整时需要多消耗的能量以及最快恢复正常需要的调整站数，具体数据见表 8 所示：

表 9：在不同站延误时进行恢复调整的数据（能量单位： $kW \cdot h$ ）

延误站台	实际消耗能量	多消耗的能量	实际运行时间(s)	调整的站数
2	133.95	2.83	2083	1
3	133.47	2.35	2083	1
4	134.90	3.78	2079	1
5	133.40	2.28	2084	1
6	133.78	2.66	2083	1
7	134.29	3.17	2082	1
8	133.92	2.80	2083	1
9	134.11	2.99	2083	1
10	135.69	4.57	2078	1
11	135.75	4.63	2082	2
12	134.33	3.22	2082	1
13	133.35	2.23	2083	1

其中在 8 站台延误后恢复期间耗能最少的列车运行曲线如下图所示：

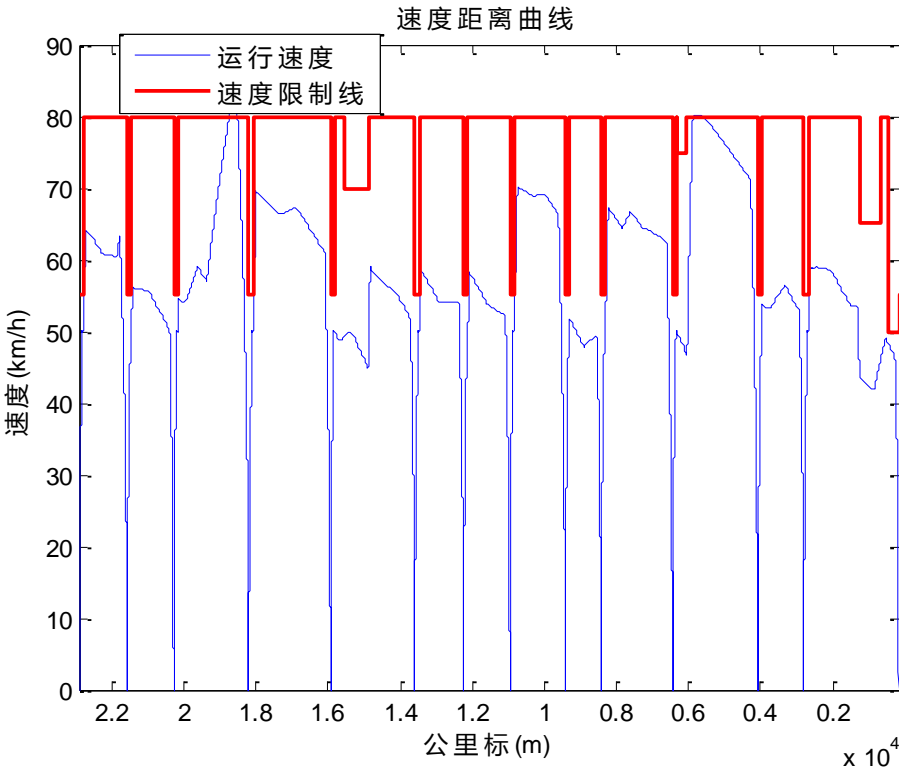


图 15：列车在 8 站台出现延误后的耗能最少的列车运行曲线

10.5 结果分析

当列车发生短时间 10s 的延误时，可以对列车进行很及时的调整。为了恢复正常，列车需要消耗一些额外能量，这使得该列车的实际消耗能量比正常的多。从表 8 中可以看出，列车在任意站发生延误时，只要进行适当的调整都可以在规定的时间内到达，而且是大部分都只需要一站的调整就可以恢复正常。

11 问题三：列车随机出现延误后的运行优化控制问题

从问题分析中可以看出，列车出现延误的情况是随机的，分为以下三种：普通延误、严重延误和无延误。出现普通延误的情况类似于第一小问来进行解决，而出现严重延误时需要考虑后续车辆的运行情况，无延误则无需考虑。对于随机出现延误的情况而言，我们所采用的方案是使该辆列车尽量在下一站能实现恢复正常，如果不能则在该站的下一站实现，依次进行下去直至解决此时延为止。

11.1 列车出现普通延误后的运行优化控制问题

列车出现普通延误的情况与上述延误 10 秒的处理情况相同，只是此时的具体延误时间 L 是服从均匀分布的，即出现任何一种延误时间的可能性是相等的。由此可建立如下所示的最优化模型：

$$\begin{cases} \min y = N - j \\ \min \sum_{i=j}^{14} (a_{i+1}^n - a_i^n) \\ \min E_1 = \int_0^{t_{\max}} F(t) v_t dt \end{cases}$$

$$s.t. \begin{cases} d_{i+1}^n - a_{i+1}^n = x(i = j, j+1 \dots 14) \\ v_t^n \leq \min \{ \bar{v}_t, \sqrt{2L_t B_e} \} \\ a_{i+1}^n \geq a_i^n \\ \sum_{i=j+1}^n (a_{i+1}^n - d_i^n) = \sum_{i=j+1}^n (a_{i+1}^n - d_i^n) - L \end{cases}$$

11.2 列车出现严重延误后的运行优化控制问题

由于列车出现严重延误后可能会影响后续车辆的运行，因此不能仅仅对该车辆进行调整，后面车辆的发车时间以及运行速度都可能发生改变，由此需要重新制定运行计划。

11.2.1 确定目标函数

目标函数一：如果第 n 辆列车在第 j 个站点发生严重延误，则从第 n 辆列车开始以及之后的每辆车都需要调整，但经过调整后列车的到站时间与原时间相比总延时应达到最少，即到达最后站点的时间与原来的时间差应该尽可能少：

$$\min \sum_{n=1}^{240} \sum_{i=j}^{14} (a_{i+1}^n - a_i^n)$$

目标函数二：对出现延迟的列车在调整列车的出行计划时耗能应该尽可能少，即：

$$\min E_2 = \sum_{n=1}^{240} E_n' = \sum_{n=1}^{240} \int_0^{t_{\max}} F_n(t) v_t^n dt$$

11.2.2 确定约束条件

约束条件一：在列车恢复正常运行期间，停站时间与原时间相比时间差不超过 10 秒，即：

$$|d_{i+1}^n - a_{i+1}^n| \leq |x-10| (i = j, j+1 \dots 14)$$

约束条件二：列车出现晚点后在追赶过程中该车的最大速度不能超过限制速度：

$$v_t^n \leq \min \{ \bar{v}_t, \sqrt{2L_t B_e} \}$$

其中 L_t 是列车当前时刻前后车之间的距离， B_e 是列车制动的最大减速度。

约束条件三：对于任意列车到站时间、离站时间与原计划时间相比时间差不超过 10 秒，即：

$$\begin{cases} |a_i^n - a_i^n| \leq 10 \\ |d_i^n - d_i^n| \leq 10 \end{cases}$$

约束条件四：对于晚点的列车而言，如果列车能够在第 N 个站点恢复正常运行，那么在第 $j+1$ 站点到第 N 个站点之间列车总的行驶时间与原行驶时间相比少了随机的延误时间 L 秒，即：

$$\sum_{i=j+1}^n (a_{i+1}^n - d_i^n) = \sum_{i=j+1}^n (a_{i+1}^n - d_i^n) - L$$

约束条件五：在调整列车运行计划期间，为了保证列车运行的安全性，后一辆列车进站必须在前一辆列车出站之后，即：

$$a_i^n > d_i^{n-1}$$

11.2.3 综上所述，发生严重延误时建立的多目标优化模型为：

$$\begin{cases} \min \sum_{n=1}^{240} \sum_{i=j}^{14} (a_{i+1}^n - a_{i+1}^n) \\ \min E_2 = \sum_{n=1}^{240} E_n' = \sum_{n=1}^{240} \int_0^{t_{\max}} F_n(t) v_t^n dt \\ s.t. \begin{cases} |d_{i+1}^n - a_{i+1}^n| \leq |x-10| (i = j, j+1 \dots 14) \\ v_t^n \leq \min \{ \bar{v}_t, \sqrt{2L_t B_e} \} \\ |a_i^n - a_i^n| \leq 10 \\ |d_i^n - d_i^n| \leq 10 \\ \sum_{i=j+1}^n (a_{i+1}^n - d_i^n) = \sum_{i=j+1}^n (a_{i+1}^n - d_i^n) - L \\ a_i^n > d_i^{n-1} \end{cases} \end{cases}$$

11.4 模型的求解

由于题目中没有给出列车晚点的具体情况，在求解时可以根据题目中给出的统计数据来模拟列车延误的情况。每次以一列车为研究对象，假设该列车在任何站都可能出现到站延误，根据延误的时间对列车的后续运行进行调整。计算该列车到达终点站后运行的实际时间和消耗的能量。重复这样的过程 100 次，分别计算考虑延误后的列车运行时间的期望和消耗的能量。其算法流程图如下图所示：

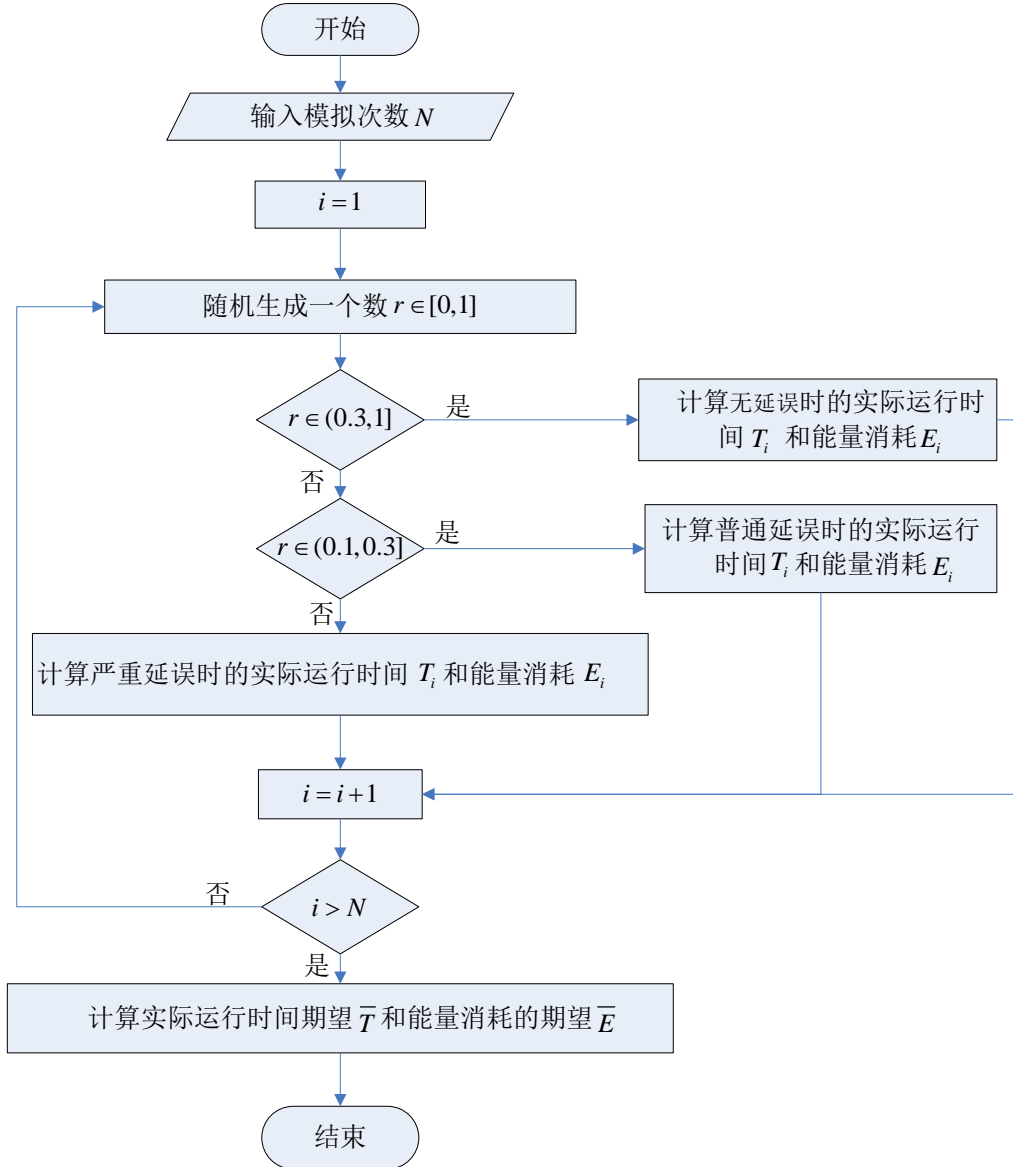


图 16: 仿真算法流程图

按照上述算法对正在运行的列车随机选择延误站点以及延误类型（其中 0 表示无延时，1 表示普通延时，2 表示严重延时），然后针对每一种出现的延误情况利用 **MATLAB** 软件（源程序见附录 11）分别计算出总的晚点时间以及整个运行过程中所消耗的能量分别如下表所示（全部结果见附录 12）：

表 10：列车随机出现晚点后所消耗的能量（单位： $kW \cdot h$ ）

序号	站台	延误类型	延时时间	总晚点时间	消耗能量	多消耗的能量
1	4	0	0	0	131.12	0.00
2	4	1	3	0	132.93	1.81
3	5	2	33	0	144.40	13.28
4	3	0	0	0	131.12	0.00
5	5	2	22	0	137.64	6.52
6	3	0	0	0	131.12	0.00
7	11	1	7	0	136.73	5.62
8	3	0	0	0	131.12	0.00
9	12	0	0	0	131.12	0.00
10	6	0	0	0	131.12	0.00
11	2	0	0	0	131.12	0.00
12	5	0	0	0	131.12	0.00
13	10	1	9	0	133.35	2.23
14	10	1	3	0	133.35	2.23
15	7	0	0	0	131.12	0.00
16	9	0	0	0	131.12	0.00
17	11	1	7	0	137.18	6.07
18	2	2	17	0	136.91	5.79
19	5	1	8	0	133.26	2.14
20	2	0	0	0	131.12	0.00

11.4 结果分析

分析上述结果可以知道对于列车随机出现延误的情况后，我们采取了相应的调整方案后使得最终晚点时间的平均值为 $0.55s$ ，所消耗能量的平均值为 $132.33kW \cdot h$ 以及额外消耗能量的平均值为 $1.2106kW \cdot h$ 。可以发现，所晚点的时间仅为 $0.55s$ 并不会对列车的出行造成过大的影响且其额外消耗的能量也仅占所耗总能量的 0.9% ，由此证明了调整方案的合理性。

13 模型的评价、改进及推广

13.1 模型的评价

优点：(1)在问题一中考虑将整个行程按照最高限速分为几部分，然后分别讨论每部分的牵引、惰行以及制动过程，考虑的较为全面、合理。

(2)在问题一的求解过程中，考虑到了在惰行过程中会出现上坡时坡度过陡的情况，此时列车可能还没到达终点时就已经停止，因此增加考虑了当加速度小于负 0.04 米/秒²时给其一个牵引力使其到达终点。

(3)在对列车的延时进行优化，采用随机模型方法对列车延误的影响和调节进行模型，检验了模型的可靠性。

缺点：(1)在问题二中为了简化模型，假设所有列车的运行状态、在各个站点的停留时间都相等，而在实际生活中会因为乘客的多少而导致各个站点的停留时间不同。

(2)在建模的过程中，问题中只有一个供电区间，而列车在实际运行时应该

会出现多个供电区间。

13.2 模型的改进

- (1)在多列车运行优化控制的问题中，我们假设每辆列车在所有站点的停车时间是一个常数，这样起到了简化模型的效果，而在实际生活中可能会出现不同的停车时间，下一步应该充分考虑这种情况。
- (2)在解决列车晚点的情况时应该考虑到多个列车同时晚点的情况，此时的调度方案会发生变化。

13.3 模型的推广

本文建立的调度模型是针对于地铁运行的，如边改若干参数则可运用到轻轨以及火车的运行。在问题二中，本文运用的遗传算法，不仅可以运用到列车优化问题上，也可运用于非线性的断裂问题、非线性力学问题、电力系统计算、现代光学和经济与非线性规划问题等。也可以用于各种多目标优化的问题，如农作物庄家的抗性研究，背包问题，旅行商问题。

14 参考文献

- [1]Howlett P G, Pudney P J. Energy-efficient train control[M]. Springer Science & Business Media, 2012.
- [2] Su S, Li X, Tang T, et al. A subway train timetable optimization approach based on energy-efficient operation strategy[J]. Intelligent Transportation Systems, IEEE Transactions on, 2013, 14(2): 883-893.
- [3]Ross I M. A primer on pontryagin's principle in optimal control[M]. Collegiate Publ., 2009.
- [4]Geering H P. Optimal control with engineering applications[M]. Berlin: Springer, 2007.
- [5]张志涌，精通 Matlab6.5 版[M]，北京：北京航空航天大学出版社，2003.
- [6] Su S, Tang T, Li X, et al. Optimization of multitrain operations in a subway system[J].Intelligent Transportation Systems, IEEE Transactions on, 2014, 15(2): 673-684.
- [7]A.Thomas, Railway Timetable and Traffic.Hamburg, Germany: Eurailpress, 2008.
- [8] Yang X, Li X, Gao Z, et al. A cooperative scheduling model for timetable optimization in subway systems[J]. Intelligent Transportation Systems, IEEE Transactions on, 2013, 14(1): 438-447.
- [9]张强. 基于遗传算法的列车自动驾驶系统研究与实现[D].北京交通大学,2008.
- [10]卢启衡,冯晓云,王青元. 基于遗传算法的追踪列车节能优化[J]. 西南交通大学学报,2012,02:265-270.

15 附录

附录 1:

时刻	实际速度	实际速度	计算加速度	计算距离	计算公里标	当前坡度	计算牵引力	计算牵引功率
----	------	------	-------	------	-------	------	-------	--------

0:00:00	0.00	0.00	1.00	0	0	0	198162.21	0.00
0:00:01	100.00	3.60	1.00	1	1	0	198284.10	198284.10
0:00:02	200.00	7.20	1.00	2	2	0	198412.84	396825.69
0:00:03	300.00	10.80	1.00	5	5	0	198548.48	595645.44
0:00:04	400.00	14.40	1.00	8	8	0	198691.00	794763.99
0:00:05	500.00	18.00	1.00	13	13	0	198840.40	994201.99
0:00:06	600.00	21.60	1.00	18	18	0	198996.68	1193980.08
0:00:07	700.00	25.20	1.00	25	25	0	199159.84	1394118.91
0:00:08	800.00	28.80	1.00	32	32	0	199329.89	1594639.11
0:00:09	900.00	32.40	1.00	41	41	0	199506.82	1795561.34
0:00:10	1000.00	36.00	1.00	50	50	0	199690.62	1996906.23
0:00:11	1100.00	39.60	1.00	61	61	0	199881.31	2198694.43
0:00:12	1200.00	43.20	1.00	72	72	0	200078.88	2400946.59
0:00:13	1300.00	46.80	1.00	85	85	0	200283.33	2603683.34
0:00:14	1389.27	50.01	-0.03	98	98	0	0.00	0.00
0:00:15	1386.10	49.90	-0.03	112	112	0	0.00	0.00
0:00:16	1425.72	51.33	1.00	126	126	0	200550.13	2859280.63
0:00:17	1525.72	54.93	1.00	141	141	0	200770.12	3063187.11
0:00:18	1542.46	55.53	-0.03	156	156	0	0.00	0.00
0:00:19	1539.11	55.41	-0.03	171	171	0	0.00	0.00
0:00:20	1535.77	55.29	-0.03	187	187	0	0.00	0.00
0:00:21	1532.43	55.17	-0.03	202	202	0	0.00	0.00
0:00:22	1529.09	55.05	-0.03	217	217	0	0.00	0.00
0:00:23	1525.75	54.93	-0.03	233	233	0	0.00	0.00
0:00:24	1522.42	54.81	-0.03	248	248	0	0.00	0.00
0:00:25	1519.09	54.69	-0.03	263	263	0	0.00	0.00
0:00:26	1515.77	54.57	-0.03	278	278	0	0.00	0.00
0:00:27	1512.45	54.45	-0.03	293	293	0	0.00	0.00
0:00:28	1508.93	54.32	-0.04	309	309	1.8	2093.26	31585.85
0:00:29	1504.93	54.18	-0.04	324	324	1.8	2084.38	31368.40
0:00:30	1500.93	54.03	-0.04	339	339	1.8	2075.50	31151.83
0:00:31	1496.93	53.89	-0.04	354	354	1.8	2066.64	30936.13
0:00:32	1492.93	53.75	-0.04	369	369	1.8	2057.79	30721.31

0:00:33	1488.93	53.60	-0.04	384	384	1.8	2048.95	30507.36
0:00:34	1484.93	53.46	-0.04	398	398	1.8	2040.12	30294.28
0:00:35	1480.93	53.31	-0.04	413	413	1.8	2031.30	30082.07
0:00:36	1476.93	53.17	-0.04	428	428	1.8	2022.49	29870.72
0:00:37	1472.93	53.03	-0.04	443	443	1.8	2013.69	29660.25
0:00:38	1468.93	52.88	-0.04	457	457	1.8	2004.91	29450.64
0:00:39	1464.93	52.74	-0.04	472	472	1.8	1996.13	29241.89
0:00:40	1460.93	52.59	-0.04	487	487	1.8	1987.37	29034.01
0:00:41	1456.93	52.45	-0.04	501	501	1.8	1978.61	28826.98
0:00:42	1452.93	52.31	-0.04	516	516	1.8	1969.87	28620.82
0:00:43	1448.93	52.16	-0.04	530	530	1.8	1961.14	28415.52
0:00:44	1444.93	52.02	-0.04	545	545	1.8	1952.42	28211.07
0:00:45	1440.93	51.87	-0.04	559	559	1.8	1943.71	28007.49
0:00:46	1436.93	51.73	-0.04	574	574	1.8	1935.01	27804.75
0:00:47	1432.93	51.59	-0.04	588	588	1.8	1926.33	27602.87
0:00:48	1428.93	51.44	-0.04	602	602	1.8	1917.65	27401.84
0:00:49	1424.93	51.30	-0.04	617	617	1.8	1908.98	27201.67
0:00:50	1420.93	51.15	-0.04	631	631	1.8	1900.33	27002.34
0:00:51	1416.93	51.01	-0.04	645	645	1.8	1891.69	26803.86
0:00:52	1412.93	50.87	-0.04	659	659	1.8	1883.06	26606.23
0:00:53	1408.93	50.72	-0.04	673	673	1.8	1874.43	26409.44
0:00:54	1405.91	50.61	0.00	687	687	-3.5	0.00	0.00
0:00:55	1406.15	50.62	0.00	701	701	-3.5	0.00	0.00
0:00:56	1406.38	50.63	0.00	716	716	-3.5	0.00	0.00
0:00:57	1406.61	50.64	0.00	730	730	-3.5	0.00	0.00
0:00:58	1406.84	50.65	0.00	744	744	-3.5	0.00	0.00
0:00:59	1407.07	50.65	0.00	758	758	-3.5	0.00	0.00
0:01:00	1407.30	50.66	0.00	772	772	-3.5	0.00	0.00
0:01:01	1407.53	50.67	0.00	786	786	-3.5	0.00	0.00
0:01:02	1407.77	50.68	0.00	800	800	-3.5	0.00	0.00
0:01:03	1408.00	50.69	0.00	814	814	-3.5	0.00	0.00
0:01:04	1408.23	50.70	0.00	828	828	-3.5	0.00	0.00
0:01:05	1408.46	50.70	0.00	842	842	-3.5	0.00	0.00
0:01:06	1408.69	50.71	0.00	856	856	-3.5	0.00	0.00
0:01:07	1408.91	50.72	0.00	870	870	-3.5	0.00	0.00
0:01:08	1409.14	50.73	0.00	884	884	-3.5	0.00	0.00
0:01:09	1409.37	50.74	0.00	899	899	-3.5	0.00	0.00
0:01:10	1409.60	50.75	0.00	913	913	-3.5	0.00	0.00
0:01:11	1409.83	50.75	0.00	927	927	-3.5	0.00	0.00
0:01:12	1410.06	50.76	0.00	941	941	-3.5	0.00	0.00
0:01:13	1410.29	50.77	0.00	955	955	-3.5	0.00	0.00
0:01:14	1410.51	50.78	0.00	969	969	-3.5	0.00	0.00
0:01:15	1410.74	50.79	0.00	983	983	-3.5	0.00	0.00

0:01:16	1410.97	50.79	0.00	997	997	-3.5	0.00	0.00
0:01:17	1411.20	50.80	0.00	1011	1011	-3.5	0.00	0.00
0:01:18	1411.42	50.81	0.00	1025	1025	-3.5	0.00	0.00
0:01:19	1411.65	50.82	0.00	1040	1040	-3.5	0.00	0.00
0:01:20	1411.87	50.83	0.00	1054	1054	-3.5	0.00	0.00
0:01:21	1412.10	50.84	0.00	1068	1068	-3.5	0.00	0.00
0:01:22	1412.33	50.84	0.00	1082	1082	-3.5	0.00	0.00
0:01:23	1412.55	50.85	0.00	1096	1096	-3.5	0.00	0.00
0:01:24	1412.78	50.86	0.00	1110	1110	-3.5	0.00	0.00
0:01:25	1413.00	50.87	0.00	1124	1124	-3.5	0.00	0.00
0:01:26	1413.23	50.88	0.00	1138	1138	-3.5	0.00	0.00
0:01:27	1413.45	50.88	0.00	1153	1153	-3.5	0.00	0.00
0:01:28	1413.67	50.89	0.00	1167	1167	-3.5	0.00	0.00
0:01:29	1413.90	50.90	0.00	1181	1181	-3.5	0.00	0.00
0:01:30	1414.12	50.91	0.00	1195	1195	-3.5	0.00	0.00
0:01:31	1414.35	50.92	0.00	1209	1209	-3.5	0.00	0.00
0:01:32	1414.57	50.92	0.00	1223	1223	-3.5	0.00	0.00
0:01:33	1414.79	50.93	0.00	1237	1237	-3.5	0.00	0.00
0:01:34	1332.66	47.98	-0.85	1251	1251	-3.5	0.00	0.00
0:01:35	1247.58	44.91	-0.85	1264	1264	-3.5	0.00	0.00
0:01:36	1162.60	41.85	-0.85	1276	1276	-3.5	0.00	0.00
0:01:37	1077.69	38.80	-0.85	1287	1287	-3.5	0.00	0.00
0:01:38	992.88	35.74	-0.85	1298	1298	-3.5	0.00	0.00
0:01:39	906.91	32.65	-0.88	1307	1307	0	0.00	0.00
0:01:40	818.83	29.48	-0.88	1316	1316	0	0.00	0.00
0:01:41	730.82	26.31	-0.88	1324	1324	0	0.00	0.00
0:01:42	642.89	23.14	-0.88	1330	1330	0	0.00	0.00
0:01:43	555.04	19.98	-0.88	1336	1336	0	0.00	0.00
0:01:44	467.25	16.82	-0.88	1342	1342	0	0.00	0.00
0:01:45	379.54	13.66	-0.88	1346	1346	0	0.00	0.00
0:01:46	291.88	10.51	-0.88	1349	1349	0	0.00	0.00
0:01:47	204.30	7.35	-0.88	1352	1352	0	0.00	0.00
0:01:48	116.77	4.20	-0.87	1353	1353	0	0.00	0.00
0:01:49	29.30	1.05	-0.55	1354	1354	0	0.00	0.00
0:01:50	0.00	0.00	0.00	1354	1354	0	0.00	0.00

附录 2:

时刻	实际速度	实际速度	计算加速度	计算距离	计算公里标	当前坡度	计算牵引力	计算牵引功率
0:00:00	0	0	1.00	0	0	0	198162	0
0:00:01	100	4	1.00	1	1	0	198284	198284
0:00:02	200	7	1.00	2	2	0	198413	396826
0:00:03	300	11	1.00	5	5	0	198548	595645
0:00:04	400	14	1.00	8	8	0	198691	794764

0:00:05	500	18	1.00	13	13	0	198840	994202
0:00:06	600	22	1.00	18	18	0	198997	1193980
0:00:07	700	25	1.00	25	25	0	199160	1394119
0:00:08	800	29	1.00	32	32	0	199330	1594639
0:00:09	900	32	1.00	41	41	0	199507	1795561
0:00:10	1000	36	1.00	50	50	0	199691	1996906
0:00:11	1100	40	1.00	61	61	0	199881	2198694
0:00:12	1200	43	1.00	72	72	0	200079	2400947
0:00:13	1300	47	1.00	85	85	0	200283	2603683
0:00:14	1389	50	-0.03	98	98	0	0	0
0:00:15	1386	50	-0.03	112	112	0	0	0
0:00:16	1426	51	1.00	126	126	0	200550	2859281
0:00:17	1490	54	-0.03	141	141	0	0	0
0:00:18	1486	54	-0.03	155	155	0	0	0
0:00:19	1483	53	-0.03	170	170	0	0	0
0:00:20	1480	53	-0.03	185	185	0	0	0
0:00:21	1476	53	-0.03	200	200	0	0	0
0:00:22	1473	53	-0.03	215	215	0	0	0
0:00:23	1470	53	-0.03	229	229	0	0	0
0:00:24	1467	53	-0.03	244	244	0	0	0
0:00:25	1463	53	-0.03	259	259	0	0	0
0:00:26	1460	53	-0.03	273	273	0	0	0
0:00:27	1457	52	-0.03	288	288	0	0	0
0:00:28	1454	52	-0.03	302	302	0	0	0
0:00:29	1450	52	-0.04	317	317	1.8	1963	28452
0:00:30	1446	52	-0.04	331	331	1.8	1954	28248
0:00:31	1442	52	-0.04	346	346	1.8	1945	28044
0:00:32	1438	52	-0.04	360	360	1.8	1937	27841
0:00:33	1434	52	-0.04	375	375	1.8	1928	27639
0:00:34	1430	51	-0.04	389	389	1.8	1919	27438
0:00:35	1426	51	-0.04	403	403	1.8	1911	27237
0:00:36	1422	51	-0.04	417	417	1.8	1902	27038
0:00:37	1418	51	-0.04	432	432	1.8	1893	26839
0:00:38	1414	51	-0.04	446	446	1.8	1885	26641
0:00:39	1410	51	-0.04	460	460	1.8	1876	26445
0:00:40	1406	51	-0.04	474	474	1.8	1867	26248
0:00:41	1402	50	-0.04	488	488	1.8	1859	26053
0:00:42	1398	50	-0.04	502	502	1.8	1850	25859
0:00:43	1394	50	-0.04	516	516	1.8	1842	25665
0:00:44	1390	50	-0.04	530	530	1.8	1833	25472
0:00:45	1386	50	-0.04	544	544	1.8	1824	25281
0:00:46	1382	50	-0.04	558	558	1.8	1816	25090
0:00:47	1378	50	-0.04	571	571	1.8	1807	24899

0:00:48	1374	49	-0.04	585	585	1.8	1799	24710
0:00:49	1370	49	-0.04	599	599	1.8	1790	24521
0:00:50	1366	49	-0.04	613	613	1.8	1782	24334
0:00:51	1362	49	-0.04	626	626	1.8	1773	24147
0:00:52	1358	49	-0.04	640	640	1.8	1765	23961
0:00:53	1354	49	-0.04	653	653	1.8	1756	23775
0:00:54	1350	49	-0.04	667	667	1.8	1748	23591
0:00:55	1346	48	-0.04	680	680	1.8	1740	23407
0:00:56	1345	48	0.00	694	694	-3.5	0	0
0:00:57	1345	48	0.00	707	707	-3.5	0	0
0:00:58	1345	48	0.00	721	721	-3.5	0	0
0:00:59	1346	48	0.00	734	734	-3.5	0	0
0:01:00	1346	48	0.00	748	748	-3.5	0	0
0:01:01	1346	48	0.00	761	761	-3.5	0	0
0:01:02	1347	48	0.00	774	774	-3.5	0	0
0:01:03	1347	48	0.00	788	788	-3.5	0	0
0:01:04	1347	48	0.00	801	801	-3.5	0	0
0:01:05	1347	49	0.00	815	815	-3.5	0	0
0:01:06	1348	49	0.00	828	828	-3.5	0	0
0:01:07	1348	49	0.00	842	842	-3.5	0	0
0:01:08	1348	49	0.00	855	855	-3.5	0	0
0:01:09	1349	49	0.00	869	869	-3.5	0	0
0:01:10	1349	49	0.00	882	882	-3.5	0	0
0:01:11	1349	49	0.00	896	896	-3.5	0	0
0:01:12	1349	49	0.00	909	909	-3.5	0	0
0:01:13	1350	49	0.00	923	923	-3.5	0	0
0:01:14	1350	49	0.00	936	936	-3.5	0	0
0:01:15	1350	49	0.00	950	950	-3.5	0	0
0:01:16	1351	49	0.00	963	963	-3.5	0	0
0:01:17	1351	49	0.00	977	977	-3.5	0	0
0:01:18	1351	49	0.00	990	990	-3.5	0	0
0:01:19	1352	49	0.00	1004	1004	-3.5	0	0
0:01:20	1352	49	0.00	1017	1017	-3.5	0	0
0:01:21	1352	49	0.00	1031	1031	-3.5	0	0
0:01:22	1352	49	0.00	1044	1044	-3.5	0	0
0:01:23	1353	49	0.00	1058	1058	-3.5	0	0
0:01:24	1353	49	0.00	1071	1071	-3.5	0	0
0:01:25	1353	49	0.00	1085	1085	-3.5	0	0
0:01:26	1354	49	0.00	1098	1098	-3.5	0	0
0:01:27	1354	49	0.00	1112	1112	-3.5	0	0
0:01:28	1354	49	0.00	1126	1126	-3.5	0	0
0:01:29	1354	49	0.00	1139	1139	-3.5	0	0
0:01:30	1355	49	0.00	1153	1153	-3.5	0	0

0:01:31	1355	49	0.00	1166	1166	-3.5	0	0
0:01:32	1355	49	0.00	1180	1180	-3.5	0	0
0:01:33	1356	49	0.00	1193	1193	-3.5	0	0
0:01:34	1356	49	0.00	1207	1207	-3.5	0	0
0:01:35	1356	49	0.00	1220	1220	-3.5	0	0
0:01:36	1356	49	0.00	1234	1234	-3.5	0	0
0:01:37	1356	49	-0.85	1248	1248	-3.5	0	0
0:01:38	1271	46	-0.85	1261	1261	-3.5	0	0
0:01:39	1186	43	-0.85	1273	1273	-3.5	0	0
0:01:40	1101	40	-0.85	1284	1284	-3.5	0	0
0:01:41	1016	37	-0.85	1295	1295	-3.5	0	0
0:01:42	931	34	-0.88	1305	1305	0	0	0
0:01:43	843	30	-0.88	1314	1314	0	0	0
0:01:44	755	27	-0.88	1322	1322	0	0	0
0:01:45	667	24	-0.88	1329	1329	0	0	0
0:01:46	579	21	-0.88	1335	1335	0	0	0
0:01:47	491	18	-0.88	1340	1340	0	0	0
0:01:48	403	15	-0.88	1345	1345	0	0	0
0:01:49	316	11	-0.88	1348	1348	0	0	0
0:01:50	228	8	-0.88	1351	1351	0	0	0
0:01:51	140	5	-0.88	1353	1353	0	0	0
0:01:52	53	2	-0.87	1354	1354	0	0	0
0:01:53	0	0	0.00	0	1354	0	0	0
0:01:54	0	0	0.00	0	1354	0	0	0
0:01:55	0	0	0.00	0	1354	0	0	0
0:01:56	0	0	0.00	0	1354	0	0	0
0:01:57	0	0	0.00	0	1354	0	0	0
0:01:58	0	0	0.00	0	1354	0	0	0
0:01:59	0	0	0.00	0	1354	0	0	0
0:02:00	0	0	0.00	0	1354	0	0	0
0:02:01	0	0	0.00	0	1354	0	0	0
0:02:02	0	0	0.00	0	1354	0	0	0
0:02:03	0	0	0.00	0	1354	0	0	0
0:02:04	0	0	0.00	0	1354	0	0	0
0:02:05	0	0	0.00	0	1354	0	0	0
0:02:06	0	0	0.00	0	1354	0	0	0
0:02:07	0	0	0.00	0	1354	0	0	0
0:02:08	0	0	0.00	0	1354	0	0	0
0:02:09	0	0	0.00	0	1354	0	0	0
0:02:10	0	0	0.00	0	1354	0	0	0
0:02:11	0	0	0.00	0	1354	0	0	0
0:02:12	0	0	0.00	0	1354	0	0	0
0:02:13	0	0	0.00	0	1354	0	0	0

0:02:14	0	0	0.00	0	1354	0	0	0
0:02:15	0	0	0.00	0	1354	0	0	0
0:02:16	0	0	0.00	0	1354	0	0	0
0:02:17	0	0	0.00	0	1354	0	0	0
0:02:18	0	0	0.00	0	1354	0	0	0
0:02:19	0	0	0.00	0	1354	0	0	0
0:02:20	0	0	0.00	0	1354	0	0	0
0:02:21	0	0	0.00	0	1354	0	0	0
0:02:22	0	0	0.00	0	1354	0	0	0
0:02:23	0	0	0.00	0	1354	0	0	0
0:02:24	0	0	0.00	0	1354	0	0	0
0:02:25	0	0	0.00	0	1354	0	0	0
0:02:26	0	0	0.00	0	1354	0	0	0
0:02:27	0	0	0.00	0	1354	0	0	0
0:02:28	0	0	0.00	0	1354	0	0	0
0:02:29	0	0	0.00	0	1354	0	0	0
0:02:30	0	0	0.00	0	1354	0	0	0
0:02:31	0	0	0.00	0	1354	0	0	0
0:02:32	0	0	0.00	0	1354	0	0	0
0:02:33	38	1	1.00	0	1354	0	198208	76245
0:02:34	138	5	1.00	1	1355	0	198333	274626
0:02:35	238	9	1.00	3	1357	0	198464	473272
0:02:36	338	12	1.00	6	1360	0	198602	672204
0:02:37	438	16	1.00	10	1364	0	198748	871443
0:02:38	538	19	1.00	14	1368	0	198900	1071009
0:02:39	638	23	1.00	20	1374	0	199059	1270924
0:02:40	738	27	1.00	27	1381	0	199224	1471207
0:02:41	838	30	1.00	35	1389	0	199397	1671879
0:02:42	938	34	1.00	44	1398	0	199577	1872962
0:02:43	1038	37	1.00	54	1408	0	199763	2074475
0:02:44	1138	41	1.00	65	1419	0	199956	2276439
0:02:45	1238	45	1.00	77	1431	0	200157	2478875
0:02:46	1338	48	1.00	90	1444	0	200364	2681804
0:02:47	1388	50	-0.03	103	1457	0	0	0
0:02:48	1385	50	-0.03	117	1471	0	0	0
0:02:49	1464	53	1.00	131	1485	0	200634	2937592
0:02:50	1501	54	-0.03	146	1500	0	0	0
0:02:51	1498	54	-0.03	161	1515	0	0	0
0:02:52	1494	54	-0.03	176	1530	0	0	0
0:02:53	1491	54	-0.03	191	1545	0	0	0
0:02:54	1488	54	-0.03	206	1560	0	0	0
0:02:55	1484	53	-0.03	221	1575	0	0	0
0:02:56	1481	53	-0.03	236	1590	0	0	0

0:02:57	1477	53	-0.04	251	1605	3	4308	63652
0:02:58	1473	53	-0.04	265	1619	3	4300	63350
0:02:59	1469	53	-0.04	280	1634	3	4291	63049
0:03:00	1465	53	-0.04	295	1649	3	4282	62749
0:03:01	1461	53	-0.04	309	1663	3	4273	62450
0:03:02	1457	52	-0.04	324	1678	3	4265	62151
0:03:03	1453	52	-0.04	339	1693	3	4256	61853
0:03:04	1449	52	-0.04	353	1707	3	4247	61557
0:03:05	1445	52	-0.04	368	1722	3	4238	61261
0:03:06	1441	52	-0.04	382	1736	3	4230	60966
0:03:07	1437	52	-0.04	396	1750	3	4221	60671
0:03:08	1433	52	-0.04	411	1765	3	4212	60378
0:03:09	1429	51	-0.04	425	1779	3	4204	60086
0:03:10	1425	51	-0.04	439	1793	3	4195	59794
0:03:11	1421	51	-0.04	454	1808	3	4186	59503
0:03:12	1417	51	-0.04	468	1822	3	4178	59213
0:03:13	1413	51	-0.04	482	1836	3	4169	58924
0:03:14	1409	51	-0.04	496	1850	3	4160	58636
0:03:15	1405	51	-0.04	510	1864	3	4152	58348
0:03:16	1401	50	-0.04	524	1878	3	4143	58062
0:03:17	1397	50	-0.04	538	1892	3	4135	57776
0:03:18	1393	50	-0.04	552	1906	3	4126	57491
0:03:19	1389	50	-0.04	566	1920	3	4117	57207
0:03:20	1385	50	-0.04	580	1934	3	4109	56924
0:03:21	1381	50	-0.04	594	1948	3	4100	56641
0:03:22	1377	50	-0.04	607	1961	3	4092	56360
0:03:23	1373	49	-0.04	621	1975	3	4083	56079
0:03:24	1369	49	-0.04	635	1989	3	4075	55799
0:03:25	1367	49	-0.01	649	2003	-2	0	0
0:03:26	1366	49	-0.01	662	2016	-2	0	0
0:03:27	1365	49	-0.01	676	2030	-2	0	0
0:03:28	1364	49	-0.01	690	2044	-2	0	0
0:03:29	1362	49	-0.01	703	2057	-2	0	0
0:03:30	1361	49	-0.01	717	2071	-2	0	0
0:03:31	1360	49	-0.01	730	2084	-2	0	0
0:03:32	1359	49	-0.01	744	2098	-2	0	0
0:03:33	1358	49	-0.01	758	2112	-2	0	0
0:03:34	1356	49	-0.01	771	2125	-2	0	0
0:03:35	1355	49	-0.01	785	2139	-2	0	0
0:03:36	1354	49	-0.01	798	2152	-2	0	0
0:03:37	1353	49	-0.01	812	2166	-2	0	0
0:03:38	1352	49	-0.01	825	2179	-2	0	0
0:03:39	1351	49	-0.01	839	2193	-2	0	0

0:03:40	1349	49	-0.01	852	2206	-2	0	0
0:03:41	1348	49	-0.01	866	2220	-2	0	0
0:03:42	1347	48	-0.01	879	2233	-2	0	0
0:03:43	1346	48	-0.01	893	2247	-2	0	0
0:03:44	1345	48	-0.01	906	2260	-2	0	0
0:03:45	1344	48	-0.01	920	2274	-2	0	0
0:03:46	1342	48	-0.01	933	2287	-2	0	0
0:03:47	1341	48	-0.01	947	2301	-2	0	0
0:03:48	1340	48	-0.01	960	2314	-2	0	0
0:03:49	1339	48	-0.01	973	2327	-2	0	0
0:03:50	1338	48	-0.01	987	2341	-2	0	0
0:03:51	1337	48	-0.01	1000	2354	-2	0	0
0:03:52	1335	48	-0.01	1013	2367	-2	0	0
0:03:53	1334	48	-0.01	1027	2381	-2	0	0
0:03:54	1333	48	-0.01	1040	2394	-2	0	0
0:03:55	1332	48	-0.01	1053	2407	-2	0	0
0:03:56	1331	48	-0.01	1067	2421	-2	0	0
0:03:57	1330	48	-0.01	1080	2434	-2	0	0
0:03:58	1328	48	-0.01	1093	2447	-2	0	0
0:03:59	1327	48	-0.01	1107	2461	-2	0	0
0:04:00	1326	48	-0.01	1120	2474	-2	0	0
0:04:01	1325	48	-0.01	1133	2487	-2	0	0
0:04:02	1324	48	-0.01	1146	2500	-2	0	0
0:04:03	1323	48	-0.01	1160	2514	-2	0	0
0:04:04	1322	48	-0.01	1173	2527	-2	0	0
0:04:05	1285	46	-0.87	1186	2540	-2	0	0
0:04:06	1198	43	-0.86	1198	2552	-2	0	0
0:04:07	1110	40	-0.88	1210	2564	0	0	0
0:04:08	1022	37	-0.88	1221	2575	0	0	0
0:04:09	934	34	-0.88	1230	2584	0	0	0
0:04:10	846	30	-0.88	1239	2593	0	0	0
0:04:11	758	27	-0.88	1247	2601	0	0	0
0:04:12	670	24	-0.88	1254	2608	0	0	0
0:04:13	582	21	-0.88	1261	2615	0	0	0
0:04:14	494	18	-0.88	1266	2620	0	0	0
0:04:15	406	15	-0.88	1271	2625	0	0	0
0:04:16	319	11	-0.88	1274	2628	0	0	0
0:04:17	231	8	-0.88	1277	2631	0	0	0
0:04:18	144	5	-0.88	1279	2633	0	0	0
0:04:19	56	2	-0.87	1280	2634	0	0	0
0:04:20	0	0	0.00	1280	2634	0	0	0

附录 3:

primarySolutionFun.m

```
function [ S,V,E,T,F,totalE,totalT,section ] =...
    primarySolutionFun( s0,s1,speedLimit,gradient,...
        curvature, brakingCurveS,brakingCurveV,curveTerminal )
index1 = find(speedLimit(:,3)>=s0,1);
index2 = find(speedLimit(:,3)>s1,1);
section.EndS = [];
section.SpeedLimit = [];

brakingTerminal = [];
if index1 == index2
    section.EndS = [s0,s1];
    section.SpeedLimit = speedLimit(index1,2);
else
    for i = index1:-1:index2
        if i == index1
            section.EndS = [section.EndS;s0,speedLimit(i,1)];
            section.SpeedLimit = [section.SpeedLimit;speedLimit(i,2)];
        elseif i == index2
            section.EndS = [section.EndS;speedLimit(i,3),s1];
            section.SpeedLimit = [section.SpeedLimit;speedLimit(i,2)];
        else
            section.EndS = [section.EndS;speedLimit(i,[3 1])];
            section.SpeedLimit = [section.SpeedLimit;speedLimit(i,2)];
        end
    end
end
if (section.EndS(1,1) - section.EndS(1,2)) <= 3
    section.EndS(2,1) = section.EndS(1,1);
    section.EndS(1,:) = [];
    section.SpeedLimit(1) = [];
end
if (section.EndS(end,1) - section.EndS(end,2)) <= 3
    section.EndS(end-1,2) = section.EndS(end,2);
    section.EndS(end,:) = [];
    section.SpeedLimit(end) = [];
end
sectionNum = size(section.EndS,1);
section.EndV = zeros(sectionNum,2);
section.E = zeros(sectionNum,1);
section.usedT = zeros(sectionNum,1);
section.S = cell(sectionNum,1);
section.V = cell(sectionNum,1);
section.T = cell(sectionNum,1);
section.F = cell(sectionNum,1);
M = 194295; % 列车质量 kg
highSpeed = 50/3.6; % 最初加速到的最高速度
L = 10000;
S = linspace(s0,s1,L);
Ssize = size(S);
V = zeros(Ssize);
E = zeros(Ssize);
F = zeros(Ssize);
```

```

% 终点制动曲线
[ endBrakingCurveV, endBrakingCurveS ] = brakingCurveFun( s0, s1, 0, gradient,
curvature );

E(1) = 0; % 消耗能量初始化
V(1) = 0; % 初始速度为0
% 牵引阶段
i = 2;
while (i < length(V) && V(i-1) < highSpeed)
    [ Fmax ] = maxTractionFun( V(i - 1) );
    [W] = totalResistanceFun(V(i - 1), S(i-1), gradient, curvature);
    capacityMaxA = (Fmax - W) / M; % 能够达到的最大加速度
    if capacityMaxA > 1
        a = 1; % 实际加速度，因为题目限制最大加速度不能超过1
        totalF = M * a; % 合力
        F(i - 1) = totalF + W; % 实际牵引力
    else
        a = capacityMaxA;
        F(i - 1) = Fmax;
    end
    V(i) = sqrt((V(i-1))^2 + 2 * a * (S(i-1) - S(i)));
    if checkMeetEndBrakingCurve(S(i), V(i), endBrakingCurveS, endBrakingCurveV)
        V(i-1:end) =
interp1(endBrakingCurveS, endBrakingCurveV, S(i-1:end), 'pchip');
        brakingTerminal = [brakingTerminal; S(i-1), S(end)]; % 记录制动区间
        E(i:end) = E(i-1);
        i = length(V);
    elseif
checkMeetBrakingCurve(S(i), V(i), brakingCurveS, brakingCurveV, curveTerminal)
        index = find(curveTerminal(:,1) > S(i), 1);
        tempEnd = curveTerminal(index, 2);
        index2 = find(S < tempEnd, 1);
        V(i-1:index2) =
interp1(brakingCurveS, brakingCurveV, S(i-1:index2), 'pchip');
        brakingTerminal = [brakingTerminal; S(i-1), S(index2)]; % 记录制动区间
        E(i:index2) = E(i-1);
        i = index2 + 1;
        break
    else
        E(i) = E(i-1) + F(i - 1) * (S(i-1) - S(i));
        i = i + 1;
    end
end
while (i <= length(V))
    [W] = totalResistanceFun(V(i - 1), S(i-1), gradient, curvature);
    a = -W / M; % 惰行加速度
    if a < -0.04
        a = - 0.04;
        realF = M * a;
        F(i - 1) = realF + W;
    end
    V(i) = sqrt((V(i-1))^2 + 2 * a * (S(i-1) - S(i)));
    E(i) = E(i-1) + F(i - 1) * (S(i-1) - S(i));
end

```

```

        if checkMeetEndBrakingCurve(S(i), V(i), endBrakingCurveS, endBrakingCurveV)
            V(i-1:end) =
interp1(endBrakingCurveS, endBrakingCurveV, S(i-1:end), 'pchip');
            brakingTerminal = [brakingTerminal; S(i-1), S(end)]; % 记录制动区间
            E(i:end) = E(i-1);
            i = length(V)+1;
        elseif
checkMeetBrakingCurve(S(i), V(i), brakingCurveS, brakingCurveV, curveTerminal)
            index = find(curveTerminal(:,1)>S(i),1);
            tempEnd = curveTerminal(index,2);
            index2 = find(S<tempEnd,1);
            V(i-1:index2) =
interp1(brakingCurveS, brakingCurveV, S(i-1:index2), 'pchip');
            brakingTerminal = [brakingTerminal; S(i-1), S(index2)]; % 记录制动区间
            E(i:index2) = E(i-1);
            i = index2 + 1;
        else
            i = i + 1;
        end
    end
    diffS = abs(diff(S));
    meanV = mean([V(1:end-1); V(2:end)]);
    T = cumsum([0, diffS ./ meanV]);
    totalT = T(end);
    totalE = E(end);
    for i = 1:sectionNum
        tempE = interp1(S, E, section.EndS(i,:), 'pchip');
        %tempT = interp1(S, T, section.EndS(i,:), 'pchip');
        tempV = interp1(S, V, section.EndS(i,:), 'pchip');
        section.E(i) = diff(tempE);
        %section.usedT(i) = diff(tempT);
        section.EndV(i,:) = tempV;
    end
    for i = 1:sectionNum
        if i == sectionNum
            tempIndex = S<=section.EndS(i,1)&S>=section.EndS(i,2);
        else
            tempIndex = S<=section.EndS(i,1)&S>section.EndS(i,2);
        end
        section.S{i} = S(tempIndex);
        tempT = T(tempIndex);
        section.T{i} = tempT - tempT(1);
        section.usedT(i) = section.T{i}(end);
        section.V{i} = V(tempIndex);
        section.F{i} = F(tempIndex);
    end
    % 求解区间制动区间
    section.braking = cell(sectionNum,1);
    for i = 1:sectionNum
        section.braking{i} = sectionBrakingFun(brakingTerminal, section.EndS(i,:));
    end
end
function sectionBraking=sectionBrakingFun(brakingTerminal, sectionEndS)

```

```

index1 = find(brakingTerminal(:,2)<sectionEndS(1),1);
index2 = find(brakingTerminal(:,2)<sectionEndS(2),1);
if isempty(index1)
    if isempty(index2)
        sectionBraking = brakingTerminal;
    else
        sectionBraking = brakingTerminal(1:index2,:);
        sectionBraking(end,2) = sectionEndS(2);
    end
else
    if isempty(index2)
        sectionBraking = brakingTerminal(index1:end,:);
        sectionBraking(1,1) = sectionEndS(1);
    elseif index1 == index2 && sectionEndS(2)>brakingTerminal(index1,1)
        sectionBraking = [];
    else
        sectionBraking = brakingTerminal(index1:index2,:);
        sectionBraking(1,1) = sectionEndS(1);
        sectionBraking(end,2) = sectionEndS(2);
    end
end
end
end

```

附录 4:

optimalSectionAlgo.m

```

function [ S,V,T,F,totalT,E,brakingTerminal ] =
optimalSectionAlgo( s0,s1,v0,vt,E,speedLimit,gradient,curvature )
M = 194295; % 列车质量 kg
L = 10000;
S = linspace(s0,s1,L);
V = zeros(size(S));
F = zeros(size(S));
brakingTerminal = [];
% 终点制动曲线
[ endBrakingCurveV,endBrakingCurveS ] = brakingCurveFun( s0, s1, vt*3.6, gradient,
curvature );
% 设置初始速度
V(1) = v0;
% 牵引阶段
i = 2;
while (i < length(V) && V(i-1) < speedLimit && E > 0)
    [ Fmax ] = maxTractionFun( V(i-1) );
    [W] = totalResistanceFun(V(i-1), S(i-1), gradient, curvature);
    capacityMaxA = (Fmax - W) / M; % 能够达到的最大加速度
    if capacityMaxA > 1
        a = 1; % 实际加速度, 因为题目限制最大加速度不能超过1
        totalF = M * a; % 合力
        F(i-1) = totalF + W; % 实际牵引力
    else
        a = capacityMaxA;
        F(i-1) = Fmax;
    end
    V(i) = sqrt((V(i-1))^2 + 2 * a * (S(i-1) - S(i)));
    if checkMeetEndBrakingCurve(S(i),V(i),endBrakingCurveS,endBrakingCurveV)

```

```

        V(i-1:end) =
interp1(endBrakingCurveS,endBrakingCurveV,S(i-1:end),'pchip');
        brakingTerminal = [brakingTerminal;S(i-1),S(end)];
        i = length(V);
    else
        E = E - F(i - 1) * (S(i-1) - S(i));
        i = i + 1;
    end
end
% 巡航阶段
while (i < length(V) && E > 0)
    V(i) = V(i - 1);
    if checkMeetEndBrakingCurve(S(i),V(i),endBrakingCurveS,endBrakingCurveV)
        V(i-1:end) =
interp1(endBrakingCurveS,endBrakingCurveV,S(i-1:end),'pchip');
        brakingTerminal = [brakingTerminal;S(i-1),S(end)];
        i = length(V);
    else
        [W] = totalResistanceFun(V(i - 1), S(i-1), gradient, curvature);
        E = E - W * (S(i-1) - S(i));
        i = i + 1;
    end
end
% 惰行阶段
while (i <= length(V))
    [W] = totalResistanceFun(V(i - 1), S(i-1), gradient, curvature);
    a = -W / M; % 惰行加速度
    if a < -0.04
        a = -0.04;
        realF = M * a;
        F(i - 1) = realF + W;
    end
    V(i) = sqrt((V(i-1))^2 + 2 * a * (S(i-1) - S(i)));
    E = E - F(i - 1) * (S(i-1) - S(i));
    if checkMeetEndBrakingCurve(S(i),V(i),endBrakingCurveS,endBrakingCurveV)
        V(i-1:end) =
interp1(endBrakingCurveS,endBrakingCurveV,S(i-1:end),'pchip');
        brakingTerminal = [brakingTerminal;S(i-1),S(end)];
        i = length(V)+1;
    else
        i = i + 1;
    end
end
diffS = abs(diff(S));
meanV = mean([V(1:end-1);V(2:end)]);
T = cumsum([0,diffS ./ meanV]);
totalT = T(end);
end

```

附录 5:

optimalStationAlgo.m

```

function [S,V,T,F,calS,calDist,Acce,interSta,totalT,totaleE,brakingTerminal,...
successFlag]=optimalStationAlgo( As,At,dwellTime,targetT,speedLimit,gradient,...
curvature,brakingCurveS,brakingCurveV,curveTerminal,stationP,deltaE)

```

```

A = As:At; % 起始车站到终点车站的向量
N = length(A) - 1; % 站间区间的个数
% 区间数据初始化
section.S = cell(0); % 区间的公里标向量
section.V = cell(0); % 区间的速度向量
section.T = cell(0); % 区间的时间向量
section.F = cell(0); % 区间的牵引力向量
section.braking = cell(0); % 区间制动区间
section.EndS = []; % 区间的端点公里标
section.EndV = []; % 区间的端点速度
section.SpeedLimit = []; % 区间的速度限制
section.E = []; % 区间的能量消耗
section.UsedT = []; % 区间的花费时间
interStaSectionNum = zeros(N,1); % 站间区间的子区间数目
% 计算两个站间的初始解, 将所有站间初始解的区间信息保存到section中
for i = 1:N
    [~,~,~,~,...
     ~,~,~,tempSection] = ...
    primarySolutionFun( stationP(A(i)), stationP(A(i+1)), speedLimit, gradient, ...
        curvature, brakingCurveS, brakingCurveV, curveTerminal );
    interStaSectionNum(i) = length(tempSection.S); % 保存站间的子区间数目
    section.S = [section.S;tempSection.S];
    section.V = [section.V;tempSection.V];
    section.T = [section.T;tempSection.T];
    section.F = [section.F;tempSection.F];
    section.braking = [section.braking;tempSection.braking];
    section.EndS = [section.EndS;tempSection.EndS];
    section.EndV = [section.EndV;tempSection.EndV];
    section.UsedT = [section.UsedT;tempSection.UsedT];
    section.SpeedLimit = [section.SpeedLimit;tempSection.SpeedLimit/3.6]; % 将
    km/h转换为m/s
    section.E = [section.E;tempSection.E];
end
% 对所有的区间进行优化
sectionNum = length(section.S); % 总的区间数目
totalT = sum(section.UsedT); % 路程消耗的总时间
firstFlag = 1; % 判断是否是第一次循环
saveTimePerE = zeros(sectionNum,1); % 区间的单位能量节省时间向量
previousT = 0;
successFlag = 1;
while totalT>targetT
    % 如果是第一次循环, 对所有的区间计算单位能量节省时间
    if firstFlag == 1
        for i = 1:sectionNum
            [~,~,~,~,tempSectionTotalT,tempSectionLeftE] =
            optimalSectionAlgo( section.EndS(i,1),...

            section.EndS(i,2), section.EndV(i,1), section.EndV(i,2), section.E(i)+deltaE, section.S
            peedLimit(i), gradient, curvature );
            saveTimePerE(i) = (section.UsedT(i) - tempSectionTotalT) / (deltaE -
            tempSectionLeftE);
        end
    else % 如果不是第一次循环, 仅计算更新过的区间的单位能量节省时间

```



```

        [ ~,~,~,~,tempSectionTotalT,tempSectionLeftE ] =
        optimalSectionAlgo( section.EndS(i,1),...

section.EndS(i,2),section.EndV(i,1),section.EndV(i,2),section.E(i)+deltaE,section.S
peedLimit(i),gradient,curvature );
        saveTimePerE(i) = (section.UsedT(i) - tempSectionTotalT) / (deltaE -
tempSectionLeftE);
        end
        firstFlag = 0; % 第一次循环后, firstFlag为0
        [~,i]=max(saveTimePerE); % 选取单位能量节省时间最多的区间
        % 对选中的区间进行优化
[ section.S{i},section.V{i},section.T{i},section.F{i},section.UsedT(i),tempSectionL
eftE,section.braking{i} ] = optimalSectionAlgo( section.EndS(i,1),...

section.EndS(i,2),section.EndV(i,1),section.EndV(i,2),section.E(i)+deltaE,section.S
peedLimit(i),gradient,curvature );
        section.E(i) = section.E(i) + deltaE - tempSectionLeftE;
        % 重新计算路程总时间
        totalT = sum(section.UsedT);
        if abs(totalT-previousT)<0.001
                successFlag = 0;
                break
        end
        previousT = totalT;
        disp(totalT)
    end
    interSta =section2interSta(section,interStaSectionNum);
    [S,V,T,F,calS,calDist,Acce,totalT,totalE,brakingTerminal] =
interSta2all(interSta,dwellTime,A,stationP);
    end

```

附录 6

problem2.m

% 第二问脚本

```

clear
addpath ../problem1
readData
load brakingCurve.mat
As = 1;
At = 14;
targetTime = 2086 - 75/2*12; % 1636
dwellTime = ones(1,12)*75/2;
tic
deltaE = 0.03 * 1000 * 3600; % deltaE 为 0.1 千瓦时
[S,V,T,F,calS,calDist,Acce,interSta,totalT,totalE,brakingTerminal]=...
    optimalStationAlgo( As,At,dwellTime,targetTime,speedLimit,gradient,...
    curvature,brakingCurveS,brakingCurveV,curveTerminal,stationP,deltaE);
toc
tempIndex = find(diff(T)==0);
if ~isempty(tempIndex)
    S(tempIndex) = [];
    V(tempIndex) = [];
    T(tempIndex) = [];
    F(tempIndex) = [];

```

```

        calS(tempIndex) = [];
        calDist(tempIndex) = [];
        Acce(tempIndex) = [];
    end
    Time = 0:floor(T(end));
    N = length(Time);
    resultTable = zeros(N+1,9);
    resultTable(1:N,1) = Time; % 时刻
    secondV = interp1(T,V,Time);
    resultTable(1:N,2) = secondV * 100; % 实际速度(cm/s)
    resultTable(1:N,3) = secondV * 3.6; % 实际速度(km/h)
    resultTable(1:N,4) = interp1(T,Acce,Time); % 计算加速度(m/s^2)
    resultTable(1:N,5) = interp1(T,calDist,Time); % 计算距离(m)
    resultTable(1:N,6) = interp1(T,calS,Time); % 计算公里标(m)
    secondS = interp1(T,S,Time); % 实际公里标(m)
    for i = 1:N-1
        [~,~,resultTable(i,7)] =
groundConditionFun( secondS(i),gradient,curvature );
    end
    secondF = interp1(T,F,Time); % 计算牵引力(N)
    resultTable(1:N,8) = secondF;
    resultTable(1:N,9) = secondF .* secondV;
    temp = (stationP(As) - stationP(At)) - resultTable(end-1,6);
    resultTable(end,:) =
[targetTime,0,0,0,resultTable(end-1,5)+temp,resultTable(end-1,6)+temp,resultTable(e
nd,7),0,0];
    Timestr = cell(N+1,1);
    for i = 1:N+1
        Timestr{i} = second2Time( i - 1 );
    end、

```

附录 7:

序号	h1	h2	h3	h4	h5	h6	h7	h8	h9	h10	h11
发车间隔	647	647	649	648	648	648	649	648	649	650	650
序号	h12	h13	h14	h15	h16	h17	h18	h19	h20	h21	h22
发车间隔	648	646	648	650	647	648	649	394	647	648	647
序号	h23	h24	h25	h26	h27	h28	h29	h30	h31	h32	h33
发车间隔	651	605	651	647	648	648	650	648	649	646	648
序号	h34	h35	h36	h37	h38	h39	h40	h41	h42	h43	h44
发车间隔	648	647	649	649	649	647	650	649	648	649	648
序号	h45	h46	h47	h48	h49	h50	h51	h52	h53	h54	h55
发车间隔	648	648	648	648	647	648	647	649	649	649	648
序号	h56	h57	h58	h59	h60	h61	h62	h63	h64	h65	h66
发车间隔	648	650	648	648	650	648	648	653	648	647	647
序号	h67	h68	h69	h70	h71	h72	h73	h74	h75	h76	h77
发车间隔	650	649	647	648	648	648	647	646	647	649	649
序号	h78	h79	h80	h81	h82	h83	h84	h85	h86	h87	h88
发车间隔	644	648	649	649	650	647	649	649	648	649	650
序号	h89	h90	h91	h92	h93	h94	h95	h96	h97	h98	h99

发车间隔	647	650	649	648	648	648	648	649	656	649	650
------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

附录 8:

P2subProblem1.m

```

% 第二问第二小问脚本
% 计算制动与牵引数据
[ brakingTimeSection, brakingSectionEnergy, tractionSection ] =
brakingTractionFun( brakingTerminal, S, T, F, brakingCurveS, brakingCurveEreg );
H = 2*60 + (11-2)*60*rand(1, 99);
tic
[ objValue ] =
objFun( H, brakingTimeSection, brakingSectionEnergy, tractionSection );
toc
gaObjFun =
@(x) objFun( x, brakingTimeSection, brakingSectionEnergy, tractionSection );
Aeq = ones(1, 99);
beq = 63900;
LB = 2*60*ones(1, 99);
UB = 11*60*ones(1, 99);
PopInitRange = [2*60; 11*60];
PopulationSize = 20;
nvars = 99;
[x, fval, exitflag, output, population, score] =
subProblem1GA(gaObjFun, nvars, Aeq, beq, LB, UB, PopInitRange, PopulationSize);

```

附录 9:

序号	h1	h2	h3	h4	h5	h6	h7	h8	h9	h10	h11
发车间隔	382	388	309	336	343	382	341	354	343	356	320
序号	h12	h13	h14	h15	h16	h17	h18	h19	h20	h21	h22
发车间隔	322	322	357	348	352	348	381	303	355	353	120
序号	h23	h24	h25	h26	h27	h28	h29	h30	h31	h32	h33
发车间隔	141	149	122	127	129	136	135	146	122	123	138
序号	h34	h35	h36	h37	h38	h39	h40	h41	h42	h43	h44
发车间隔	147	149	133	145	139	125	124	121	131	129	148
序号	h45	h46	h47	h48	h49	h50	h51	h52	h53	h54	h55
发车间隔	141	131	129	122	148	150	150	147	136	132	141
序号	h56	h57	h58	h59	h60	h61	h62	h63	h64	h65	h66
发车间隔	121	134	145	144	134	133	317	335	313	321	352
序号	h67	h68	h69	h70	h71	h72	h73	h74	h75	h76	h77
发车间隔	342	312	390	318	388	369	303	328	318	368	384
序号	h78	h79	h80	h81	h82	h83	h84	h85	h86	h87	h88
发车间隔	322	383	326	302	342	348	377	343	302	358	354
序号	h89	h90	h91	h92	h93	h94	h95	h96	h97	h98	h99
发车间隔	373	319	337	364	354	308	326	322	310	303	307
序号	h100	h101	h102	h103	h104	h105	h106	h107	h108	h109	h110
发车间隔	387	314	366	322	359	336	324	369	321	362	314
序号	h111	h112	h113	h114	h115	h116	h117	h118	h119	h120	h121
发车间隔	348	341	376	357	378	388	312	388	377	317	374

序号	h122	h123	h124	h125	h126	h127	h128	h129	h130	h131	h132
发车间隔	347	345	325	369	301	334	324	380	360	367	360
序号	h133	h134	h135	h136	h137	h138	h139	h140	h141	h142	h143
发车间隔	352	366	316	380	346	315	369	326	335	356	353
序号	h144	h145	h146	h147	h148	h149	h150	h151	h152	h153	h154
发车间隔	329	382	321	361	315	389	311	141	124	128	128
序号	h155	h156	h157	h158	h159	h160	h161	h162	h163	h164	h165
发车间隔	125	146	134	121	141	128	136	130	135	146	121
序号	h166	h167	h168	h169	h170	h171	h172	h173	h174	h175	h176
发车间隔	135	149	133	132	132	140	128	131	129	135	136
序号	h177	h178	h179	h180	h181	h182	h183	h184	h185	h186	h187
发车间隔	141	122	147	131	146	125	141	130	125	146	122
序号	h188	h189	h190	h191	h192	h193	h194	h195	h196	h197	h198
发车间隔	141	145	132	136	133	137	147	129	136	123	143
序号	h199	h200	h201	h202	h203	h204	h205	h206	h207	h208	h209
发车间隔	149	146	130	144	147	369	336	334	326	305	385
序号	h210	h211	h212	h213	h214	h215	h216	h217	h218	h219	h220
发车间隔	390	363	301	355	371	356	367	307	368	320	378
序号	h221	h222	h223	h224	h225	h226	h227	h228	h229	h230	h231
发车间隔	346	376	364	344	386	384	333	343	369	425	425
序号	h232	h233	h234	h235	h236	h237	h238	h239			
发车间隔	425	425	425	425	425	425	425	425			

附录 10:

simulation.m

```
function [ result ] =
simulation( n, interSta, speedLimit, gradient, curvature, brakingCurveS, brakingCurveV, curveTerminal, stationP )
%simulation 仿真函数
% 此处显示详细说明
delayTime = zeros(n,1);
lateTime = zeros(n,1);
delaytotalE = zeros(n,1);
delayType = randsrc(n,1,[0,1,2;0.7,0.2,0.1]);
fileID = fopen('simResult.txt','a');
for i = 1:length(delayType)
    if delayType(i)==0
        delayTime(i) = 0;
    elseif delayType(i)==1
        delayTime(i) = rand(1)*10;
    else
        delayTime(i) = 10 + rand(1)*(60-10);
    end
end
delaySta = randi([2,12],n,1);
for i = 1:n
    if delayTime(i) == 0
```

```

        lateTime(i) = 0;
        delaytotalE(i) = 0;
    else
        [ ~,~,~,~,~,~,~,~,delaytotalE(i),~,lateTime(i) ] =...

delayFun( delaySta(i),delayTime(i),interSta,speedLimit,gradient,curvature,...
        brakingCurveS,brakingCurveV,curveTerminal,stationP );

    end

fprintf(fileID, '%f\t%d\t%f\t%f\t%f\n', delaySta(i), delayType(i), delayTime(i), lateTim
e(i), delaytotalE(i))
    end
    result = [delaySta, delayType, delayTime, lateTime, delaytotalE];
    end

```

附录 11:

delayFun.m

```

function [ S,V,T,F,calS,calDist,Acce,totalT,totalE,brakingTerminal,lateTime ]
=...
    delayFun( delaySta,delayTime,interSta,speedLimit,gradient,curvature,...
        brakingCurveS,brakingCurveV,curveTerminal,stationP )
%delayFun 延迟调整函数
A = 1:14;
dwellTime = ones(1,12)*75/2;
deltaE = 0.001 * 1000 * 3600; % deltaE 为 0.1 千瓦时

lateTime = 0;
mayDelay = delaySta:(14-1);
delayTimeVec = zeros(size(mayDelay));
for i = 1:length(mayDelay)
    aveDelay = delayTime / i;
    delayTimeVec(1:i) = aveDelay;
    successFlag = ones(1,length(delayTimeVec));
    tempInterSta = cell(1,length(delayTimeVec));
    for k = 1:length(delayTimeVec)
        if delayTimeVec(k)~=0
            targetT = interSta{mayDelay(k)}.UsedT - delayTimeVec(k);

[~,~,~,~,~,~,~,tempInterSta{k},~,~,~,successFlag(k)]=optimalStationAlgo( mayDelay(k)
),mayDelay(k)+1,dwellTime,targetT,speedLimit,gradient,...

curvature,brakingCurveS,brakingCurveV,curveTerminal,stationP,deltaE);
        end
    end
    if all(successFlag)
        for j = 1:i
            interSta{mayDelay(j)} = tempInterSta{j}{1};
            dwellTime(mayDelay(j)-1) = dwellTime(mayDelay(j)-1) +
delayTimeVec(j);
            lateTime = delayTime - delayTime/i;
        end
        break
    end
end

```

```

end
[S, V, T, F, calS, calDist, Acce, totalT, totalE, brakingTerminal] =
interSta2all(interSta, dwellTime, A, stationP);
End

```

附录 12:

序号	站台	晚点类型	延时时间	总晚点时间	消耗能量	多消耗的能量
1	4	0	0	0	131.12	0.00
2	4	1	3	0	132.93	1.81
3	5	2	33	0	144.40	13.28
4	3	0	0	0	131.12	0.00
5	5	2	22	0	137.64	6.52
6	3	0	0	0	131.12	0.00
7	11	1	7	0	136.73	5.62
8	3	0	0	0	131.12	0.00
9	12	0	0	0	131.12	0.00
10	6	0	0	0	131.12	0.00
11	2	0	0	0	131.12	0.00
12	5	0	0	0	131.12	0.00
13	10	1	9	0	133.35	2.23
14	10	1	3	0	133.35	2.23
15	7	0	0	0	131.12	0.00
16	9	0	0	0	131.12	0.00
17	11	1	7	0	137.18	6.07
18	2	2	17	0	136.91	5.79
19	5	1	8	0	133.26	2.14
20	2	0	0	0	131.12	0.00
21	4	0	0	0	131.12	0.00
22	9	1	1	0	131.44	0.32
23	9	1	6	0	132.69	1.57
24	11	0	0	0	131.12	0.00
25	8	0	0	0	131.12	0.00
26	2	0	0	0	131.12	0.00
27	12	0	0	0	131.12	0.00
28	10	0	0	0	131.12	0.00
29	5	0	0	0	131.12	0.00
30	7	0	0	0	131.12	0.00
31	12	1	4	0	131.91	0.79
32	9	1	8	0	133.38	2.26
33	11	2	35	0	131.12	0.00
34	6	0	0	0	131.12	0.00
35	7	0	0	0	131.12	0.00
36	8	0	0	0	131.12	0.00

37	4	0	0	0	131.12	0.00
38	10	0	0	0	131.12	0.00
39	7	1	5	0	132.57	1.45
40	9	0	0	0	131.12	0.00
41	5	0	0	0	131.12	0.00
42	3	0	0	0	131.12	0.00
43	7	1	9	0	133.70	2.58
44	5	0	0	0	131.12	0.00
45	10	0	0	0	131.12	0.00
46	10	0	0	0	131.12	0.00
47	9	0	0	0	131.12	0.00
48	3	0	0	0	131.12	0.00
49	2	2	28	14	138.91	7.79
50	8	0	0	0	131.12	0.00
51	5	0	0	0	131.12	0.00
52	12	0	0	0	131.12	0.00
53	12	0	0	0	131.12	0.00
54	5	0	0	0	131.12	0.00
55	10	0	0	0	131.12	0.00
56	11	2	32	0	131.12	0.00
57	8	1	10	0	133.92	2.80
58	11	0	0	0	131.12	0.00
59	12	1	0	0	131.91	0.79
60	8	0	0	0	131.12	0.00
61	10	0	0	0	131.12	0.00
62	8	2	59	0	131.12	0.00
63	2	2	19	0	138.86	7.74
64	6	1	7	0	132.89	1.77
65	9	0	0	0	131.12	0.00
66	7	0	0	0	131.12	0.00
67	6	2	39	26	143.26	12.14
68	12	0	0	0	131.12	0.00
69	11	1	7	0	135.97	4.86
70	11	0	0	0	131.12	0.00
71	6	1	4	0	132.06	0.94
72	8	0	0	0	131.12	0.00
73	11	1	6	0	135.41	4.29
74	12	1	8	0	133.53	2.42
75	9	0	0	0	131.12	0.00
76	4	0	0	0	131.12	0.00
77	9	2	11	0	134.37	3.25
78	2	0	0	0	131.12	0.00
79	6	0	0	0	131.12	0.00

80	9	0	0	0	131.12	0.00
81	12	0	0	0	131.12	0.00
82	10	1	1	0	133.35	2.23
83	7	0	0	0	131.12	0.00
84	10	0	0	0	131.12	0.00
85	6	0	0	0	131.12	0.00
86	12	2	59	0	131.12	0.00
87	12	0	0	0	131.12	0.00
88	11	1	7	0	135.70	4.58
89	6	1	2	0	131.80	0.68
90	7	2	30	15	140.90	9.78
91	4	0	0	0	131.12	0.00
92	10	0	0	0	131.12	0.00
93	11	0	0	0	131.12	0.00
94	12	0	0	0	131.12	0.00
95	8	0	0	0	131.12	0.00
96	8	1	1	0	131.45	0.34
97	3	0	0	0	131.12	0.00
98	11	0	0	0	131.12	0.00
99	6	0	0	0	131.12	0.00
100	4	0	0	0	131.12	0.00