arXiv:2304.01806v1 [math.OC] 4 Apr 2023

网联与自动驾驶车辆调度问题：一些模型与算法。

Evgeny R. Gafarov

俄罗斯科学院瓦·阿·特拉佩兹尼科夫控制科学研究所，普罗索尤兹纳亚街*65*号，*117997* 莫斯科，俄罗斯，电子邮箱：*axel73@mail.ru*

Frank Werner

Fakult¨at f¨ur Mathematik，Otto-von-Guericke-Universit¨at Magdeburg，PSF 4120，39016 马格德堡，德国，电子邮箱：*frank.werner@mathematik.uni-magdeburg.de*

*2023*年*4*月*4*日

摘要

本文研究联网与自动驾驶车辆系统中出现的调度问题。针对这类问题的四种变体，本文给出了数学模型和求解算法。具体而言，提出了三种多项式算法和一种分支定界算法。

关键词： 调度，优化，动态规划，联网与自动驾驶车辆，优先关系

数学学科分类：90 B 35, 90 C 27, 68 Q 25, 68 W 40

1 引言

车车通信被视为应对该领域诸多挑战的一种可能方案。许多解决方案将整个自动驾驶系统建模为多智能体系统，其中车辆相互协作，实现紧急制动和交通拥堵规避等自动驾驶功能。车辆系统正朝着完全互联、完全自动驾驶的方向发展。车载通信技术已在例如 [1, 2] 中得到研究。

这些车辆系统能够协调车辆运行，以提高交通效率并避免交通拥堵。车辆可以由驻留在网络中的集中式调度器（例如，蜂窝系统中的基站）进行协调，也可以由分布式调度器进行协调，在后一种情况下，资源由车辆自主选择。

在 [3]，中，作者提出对网联自动驾驶汽车（CAVs）的出发时间、行驶路线和纵向轨迹，以及交叉口的信号灯时序进行优化，以实现稳定交通状态，即所有车辆在进入任何交叉口之前均无需停车，且任何交叉口均不会发生排队溢出。出发时间、行驶路线和信号灯时序由中央控制器进行优化，而车辆轨迹则可由分布式路侧处理器进行优化，这些共同构成了一种分层交通管理方案。

在 [4], 中，作者研究了自动驾驶车辆在到达给定关键位置之前于双车道道路路段上的变道协调问题。文中提出了一种算法，能够在尽可能短的时间内完成单辆车辆的变道。随后，该算法被迭代应用，以处理所研究道路路段上所需的所有变道，同时确保交通安全。

在 [5], 中，研究了网联自动驾驶汽车穿越交叉口的调度问题，以优化交叉口效率。此外，还提出了一种求解算法。

在 [6], 中，研究了交通信号灯调度问题中的时相，旨在通过缩短信号控制交叉口通行车辆的等待时间来提高交通流畅性。

本文研究了与网联自动驾驶汽车相关的四类调度问题。本文其余部分安排如下：第2至第5节分别讨论其中一类问题。第2节研究道路具有两条车道且其中一条车道设有屏障的情况。第3节讨论车辆转入主干道的情况。第4节研究道路具有三条车道且中间车道设有屏障的情况。第5节讨论设有分隔车道的交叉路口。针对上述每种情况，本文均构建了相应的调度问题并给出求解算法。最后，第6节给出几点结论。

2 两条车道且一条车道上有屏障的道路

本节考虑一条有两条车道的道路，其中给定两组网联自动驾驶汽车 N_1 和 N_2 。集合 N_1 中的网联自动驾驶汽车行驶在车道 1 上，集合 N_2 中的网联自动驾驶汽车行驶在车道 2 上。两条车道的行驶方向相同。车道 2 上设有屏障，集合 N_2 中的网联自动驾驶汽车必须驶入车道 1，见图 1。

我们必须确定集合 N_1 和N_2 中的网联自动驾驶汽车通过屏障的顺序，以使给定的目标函数（例如总通过时间）最小化。

我们假设

- 已知网联自动驾驶汽车的最大可行速度。网联自动驾驶汽车要么以最大可行速度行驶，要么减速，以便让另一辆网联自动驾驶汽车变换车道；
- 不考虑加速度；
- 未考虑变道所需的时间，即将其视为零；
- 网联自动驾驶汽车的长度相同；

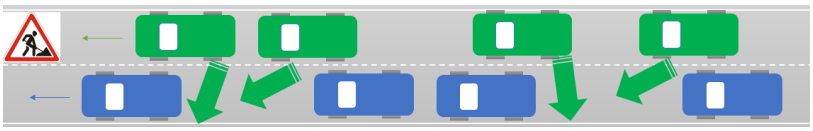


图 1：有两条车道且其中一条车道上设有屏障的道路

- 所有车辆之间的安全距离都相同。

例如，在铁路区段以及配备自主机器人运输车的物流公司自动化仓库中，也会出现同样的问题。这个简化问题可以表述为如下的单机调度问题。

给定一组 $N = N_1 \bigcup N_2$ n 作业，这些作业必须从时间 0 开始在一台单机上加工。对于每个作业 j，给定其加工时间 $p_j = p$、释放日期 $r_j > 0$、交期 d_j 和权重 w_j。加工时间 p 可根据最大可行速度和网联自动驾驶汽车的长度计算得出。数值 r_j 对应网联自动驾驶汽车 j 在道路上的位置。

网联自动驾驶汽车集合 N的排列 π 唯一确定一个调度。设 C_j（π）为调度 π中作业 j 的完工时间。可以定义优先关系，即对于集合 N_1中的作业，有 $j_1^1 \to j_2^1 \to \ldots \to j_{n_1}^1$，其中$n_1 = |N_1|$ 和 $j \to i$ 表示作业 j 的加工先于作业i的加工。因此，车道 1 上存在一条作业链。同理，可以为集合 N_2定义一条作业链。

对于使总完工时间最小化的单机调度问题，目标是找到一个最优调度 π^*，使总完工时间最小，即

$$\sum C_j = \sum_{j \in N} C_j. \tag{1}$$

这里，作业的完工时间等于汽车通过屏障的时刻。根据 Graham 等人 [7],提出的调度问题传统三字段表示法 $\alpha|\beta|\gamma$，我们将该问题记为 $1|2\ chains,\ p_j = p,\ r_j|\sum C_j$，其中 α 描述机器环境，β 给出作业特征及其他约束条件，而 γ 描述目标函数。

Let

$$T_j(\pi) = \max\{0, C_j(\pi) - d_j\}$$

b调度 π中作业 j 的延迟时间。此外，还可以考虑 e

fo以下目标函数：

$$\begin{aligned} \sum w_j C_j &= \sum w_j C_j(\pi) \text{ – total weighted completion time,} \\ \sum T_j &= \sum T_j(\pi) \text{ – total tardiness,} \\ \sum w_j T_j &= \sum w_j T_j(\pi) \text{ – total weighted tardiness.} \end{aligned}$$

已知，对于具有任意数量链的情形，问题 $1|chains,\ p_j = p,\ r_j|\sum w_j C_j$和 $1|chains,\ p_j = p,\ r_j|\sum w_j T_j$均为 NP难 [8]。这一结论已通过从 3划分问题归约得到证明。

在 [9], Baptiste 提出了多项式时间动态规划算法，用于解决问题 $1|p_j = p,\ r_j|\sum T_j$和 $1|p_j = p,\ r_j|\sum w_j U_j$。

对于问题 $1|2\ chains, p_j = p, r_j|\sum C_j$ 的一个最优调度中，作业按照数值 r_j的非递减顺序进行处理。通过反证法可以很容易地证明这一点。为了说明上文引入的概念，我们考虑以下简单示例。

示例。设 $N_1 = \{1, 2\}$, $N_2 = \{3, 4\}$。此外，给定数值 $p = 2$, $r_1 = 0$, r_2 =3, $r_3 = 1$, $r_4 = 4$ 和 $d_1 = d_2 = 0$, $d_3 = 3$, $d_4 = 6$ 。对于选定的作业序列 π = (1, 3, 2, 4)，我们得到 $S_1(\pi) = 0$, $S_3(\pi) = 2$, $S_2(\pi) = 4$, $S_4(\pi) = 6$,$C_1(\pi) = 2$, $C_3(\pi) = 4$, $C_2(\pi) = 6$, $C_4(\pi) = 8$。 $\sum_{j=1}^{4} C_j(\pi) = 20$ 和 $\sum_{j=1}^{4} T_j(\pi) = 1 + 2 = 3$。对于作业序列 π' = (3, 4, 1, 2)，我们得到 $\sum_{j=1}^{4} T_j(\pi') = 0$。

注意，存在一个集合 Θ，其中包含所有作业可能的完工时间，且 $|\Theta| \leq n^2$，因为：

- 不失一般性，我们只考虑主动调度，即不存在某个作业可以在不损害可行性的情况下更早加工；
- 不同的值 r_j 不超过 n 个；
- 所有加工时间均等于 p ，因此，对任意作业 $j \in N$，其完工时间等于 $r_i + lp$, $i \in N$, $l \leq n$。

定理1. 问题 $1|2\ chains, p_j = p, r_j|f$, $f \in \{\sum w_j C_j, \sum w_j T_j\}$ 可以通过动态规划在 $O(n^5)$ 时间内求解。

证明概要如下。在动态规划（DP）中，我们逐个考虑作业 i_1、i_2、.....、$i_{n_2} \in N_2$，其中 $i_1 \to i_2 \to \ldots \to i_{n_2}$。因此，在动态规划的每个阶段 k，我们考虑一个作业 i_k、 k = 1、2、.....、n_2。此外，在每个阶段，我们都会考虑上一阶段存储的所有状态 (f, C_{max}, pos)。另外，对于每个状态，我们都存储其对应的最佳部分解（作业序列）。上述三元组的含义如下：其中， $pos \in \{0, 1, 2, \ldots, n_1\}$ 描述作业的位置，表示作业 i_{k-1} 在作业 $j_{pos} \in N_1$与 $j_{pos+1} \in N_1$之间加工， $0 < pos < n_1$；而 $C_{max} = C_{i_{k-1}} \in \Theta$ 表示对应部分解中作业 i_{k-1} 的完工时间。最后， f 是与该部分解对应的目标函数值。对于每个作业 i_k 和状态 (f, C_{max}, pos)，我们计算新的状态 (f', C'_{max}, pos')，其中 $pos' \geq pos$； C'_{max} 是新部分解中作业 i_k 的完工时间，在该解中，作业 i_k 安排在作业 $j_{pos'} \in N_1$之后。如果在任一阶段存在两个状态 (f', C'_{max}, pos') 和 $(f'', C''_{max} pos')$，满足 $f' \leq f''$ 和 $C'_{max} \leq C''_{max}$，我们只保留状态 (f', C'_{max}, pos')。最后一个阶段结束后，我们需要从生成的所有状态中选出找到的最佳完整解。

下面给出了动态规划算法的伪代码。

动态规划算法

1. $StatesSet = \{(0, 0, 0)\}$；2. 对每个 $i_k \in N_2$ 执行 2.1 $NewStatesSet = \{\}$；2.2 对每个 $(f, C_{max}, pos) \in StatesSet$ 执行 2.2.1 令 $PositionsList = \{pos, pos + 1, \ldots, n_1\}$；2.2.2 对每个 $pos' \in PositionsList$ 执行 2.2.2.1 根据与状态 (f, C_{max}, pos) 对应的部分解，计算在作业 i_k 于 $j_{pos'}$之后处理时所得部分解的 f' ； 2.2.2.2 将 (f', C'_{max}, pos') 添加到 $NewStatesSet$。如果在 $NewStatesSet$中存在状态 (f'', C''_{max}, pos')，满足 $f' \leq f''$和 $C'_{max} \leq C''_{max}$，则从 $NewStatesSet$中排除状态 (f'', C''_{max}, pos')；2.3 $StatesSet := NewStatesSet$；

3. 从生成的所有状态中选出找到的最佳完整解。

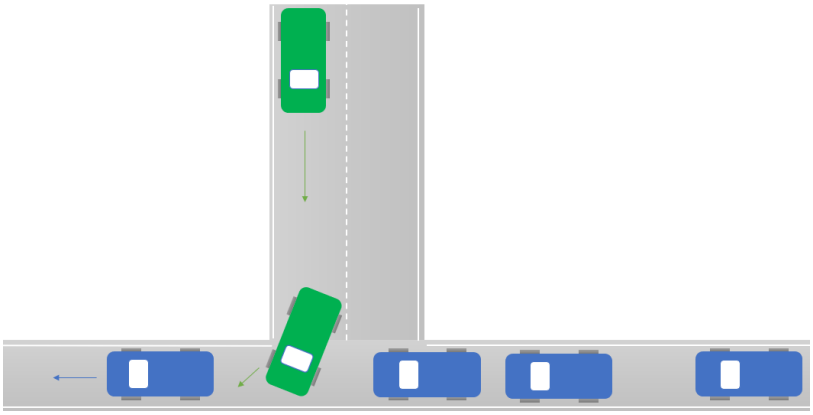


图 2：转入主干道

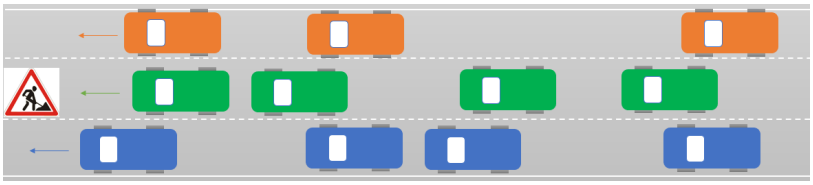


图 3：一条有三条车道且中间车道设有屏障的道路

3 转入主干道

有一组 N_1 网联自动驾驶汽车沿主干道行驶，另一组 N_2 网联自动驾驶汽车从支路转入主干道，见图 2。与问题 $1|2\,chains,\ p_j = p|\gamma$ 相比，我们现在有 $p_j = p^1,\ j \in N_1$ 和 $p_j = p^2,\ j \in N_2$。我们用 $1|2\,chains,\ p_j \in p^1,\ p^2,\ r_j|\gamma$ 表示这个问题。这个问题可以用相同的动态规划求解。

4 一条有三条车道且中间车道设有屏障的道路

除问题 $1|2\,chains,\ p_j = p|\gamma$ 外，还有一条额外的车道 3 和一个作业子集 N_3，见图 3。集合 N_1 中的作业应在机器 1 上处理，集合 N_3 中的作业应在机器 3 上处理。集合 N_2 中的作业可以在这两台机器中的任意一台上处理。集合 N_3 中作业之间的优先关系可以定义为一条作业链。

我们将此问题记为 $P2|dedicated,\ 3\,chains,\ p_j = p,\ r_j|\gamma$。此问题可以通过修改后的动态规划求解，其中我们考虑集合 N_1 中作业之间以及集合 N_3 中作业之间的位置 pos。

5 带分隔线的交叉路口

在本节中，我们考虑一个带有分隔线的交叉路口，以及四组网联自动驾驶汽车 N_1，N_2，N_3，N_4。它们共用交叉路口的四个区域，分别记为 M_1，M_2，M_3，M_4，见图 4。我们需要确定通过这些区域的最优序列。

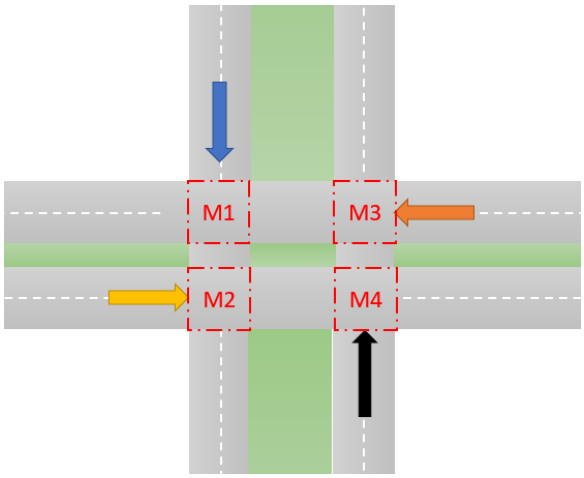


图 4：带分隔线的交叉路口

我们可以将其表述为一个包含四台机器的作业车间调度问题。共有四组作业 N_1，N_2，N_3，N_4，以及分别对应四个区域的四台机器 M_1，M_2，M_3，M_4。每个作业 j 由两个工序组成。对于每个作业 $j \in N_1$，其第一个工序必须在机器 M_1 上加工，第二个工序必须在机器 M_2上加工。对于每个作业 $j \in N_2$，其第一个工序必须在机器 M_2 上加工，第二个工序必须在机器 M_4上加工。对于每个作业 $j \in N_3$，其第一个工序必须在机器 M_3 上加工，第二个工序必须在机器 M_1上加工。对于每个作业 $j \in N_4$，其第一个工序必须在机器 M_4 上加工，第二个工序必须在机器 M_3上加工。各工序的加工时间均为 p。优先关系可以表示为作业链。

如果分隔线的长度等于 0，那么作业j 的第二道工序应紧接在第一道工序之后处理。否则，对于每个集合N_1, N_2, N_3, N_4，对应的机器都有四个容量有限的缓冲区，即 b_1, b_2，b_3, b_4 个作业。对于集合 N_1，任意时刻最多可以有 b_1 个作业已完成第一道工序但尚未开始第二道工序。我们将这些问题记为 $J4|4\ chains, p_j = p, r_j|f$，

$f \in \{C_{max}, \sum w_j C_j, \sum w_j T_j\}$，其中 C_{max} 为最大完工时间。

问题 $J4|4\ chains, p_j = p, r_j|f$，$f \in \{C_{max}, \sum w_j C_j, \sum w_j T_j\}$ 可以通过分支定界（B&B）算法求解。按照以下分支规则构建搜索（根）树。对于树中的任意节点，我们考虑以下 8 个可能的分支：

- 在机器 M_1上，为作业 $j \in N_1$ 安排第一个尚未调度且可执行的工序，并尽可能早地开始。如果没有这样的工序，则跳过此分支。

- 在机器 M_1上，为作业 $j \in N_3$ 安排第一个尚未调度且可执行的工序，并尽可能早地开始。
- 在机器 M_2上，为作业 $j \in N_1$ 安排第一个尚未调度且可执行的工序，并尽可能早地开始。
- 在机器 M_2上，为作业 $j \in N_2$ 安排第一个尚未调度且可执行的工序，并尽可能早地开始。
- 在机器 M_3上，尽可能早地为作业 $j \in N_3$ 调度第一个尚未调度且可执行的工序。

• 在机器 M_3上，尽可能早地为作业 $j \in N_4$ 调度第一个尚未调度且可执行的工序。

• 在机器 M_4上，以最早可能的开始时间为作业 $j \in N_2$ 安排第一个尚未调度且可执行的工序。

• 在机器 M_4上，以最早可能的开始时间为作业 $j \in N_4$ 安排第一个尚未调度且可执行的工序。

因此，每个节点最多有 $2^3 = 8$ 个分支需要考虑。由于共有 $2n$ 道工序，其中 $n = |N_1 \bigcup N_2 \bigcup N_3 \bigcup N_4|$，搜索树最多有 $2n$ 层。因此，需要考虑的节点不超过 $(2^3)^{2n} = 2^{6n}$ 个。如果某些值 b_1, b_2, b_3, b_4 等于 0，则节点数会更少。例如，如果它们都等于 0，那么我们只有 2^{3n} 个节点。

此外，我们可以使用以下简单的下界和上界来处理该问题$J4|4\ chains, p_j = p, r_j|C_{max}$。

上界。为构造一个可行解，我们使用列表调度算法。在该算法中，我们按照相应作业释放日期的非递减顺序，逐一考虑尚未调度的工序。根据当前部分调度，在最早可能的开始时间安排下一个尚未调度的工序。对作业集合进行排序需要 $O(n\log\ n)$ 次操作。此外，构造一个可行解还需要 $O(n)$ 次操作。

下界。考虑一组尚未调度的工序 N'。对于其中每一道工序，我们在不考虑其他尚未调度工序的情况下，根据当前部分调度计算其最早可能的开始时间。这样，我们得到一个调度 π，但它可能不可行。设 $C_{M_1}(\pi)$ 为机器 M_1上各工序的最大完工时间（即分配给该机器的工序的最大完工时间），$IT_{M_1}(\pi)$ 为机器 M_1 在集合 N'中的工序之间的空闲时间，$OT_{M_1}(\pi)$ 为总重叠时间，即同一时刻处理多个工序的时间。此外，设

$$C'_{M_1}(\pi) = C_{M_1}(\pi) + \max\{0, OT_{M_1}(\pi) - IT_{M_1}(\pi)\}.$$

Then

$$LB1 = \max\{C'_{M_1}(\pi), C'_{M_2}(\pi), C'_{M_3}(\pi), C'_{M_4}(\pi)\}$$

这是一个下界。很容易验证，计算该下界需要进行 $O(n)$ 次工序。

如果使用上界和下界，那么分支定界算法需要进行 $O(n2^{6n})$ 次工序。

6 结语

本文给出了网联自动驾驶汽车调度问题的四种模型。其中三种模型可以在多项式时间内通过动态规划算法求解。对于第四个问题，本文提出了一种分支定界算法。

未来可以考虑以下问题：

• 这些问题 $J4|4\ chains,\ p_j = p,\ r_j|f,\ f \in \{C_{max}, \sum w_j C_j, \sum w_j T_j\}$ 是 NP-难问题，还是可以在多项式时间内求解?

• 对于网联自动驾驶汽车，作业加工时间相同且作业链数量固定的调度问题是否是NP难问题?

参考文献

[1] Bazzal, M.; Krawczyk, L.; Govindarajan, R. P.; Wolff, C. 《基于实时演算的车车通信系统时序分析：案例研究》，2020年，1-8。[2] S.ahin, T.; Khalili, R.; Boban, M.; Wolisz, A. 《覆盖范围外车车通信的强化学习调度器》，2018年IEEE车辆网络会议（VNC），台北，台湾，2018年，第1-8页。[3] Qian, G.; Guo, M.; Zhang, L.; Wang, Y.; Hu, S.; Wang, D.: 《全互联自动化网络中的交通调度与控制》，《交通研究C辑：新兴技术》，第126卷，2021年，103011。[4] Atagoziev, M.; Schmidt, E.G.; Schmidt, K.W.《网联与自动驾驶车辆的变道调度》，《交通研究C辑：新兴技术》，第147卷，2023年，103985。[5] Ma, M.《预约式交叉口网联与自动驾驶车辆的最优调度》，电子学位论文，路易斯维尔大学，2022年。[6] Bani Younes, M.; Boukerche, A.《基于车载自组网的智能交通信号灯调度算法》，第39届IEEE局部计算机网络年会研讨会，埃德蒙顿，艾伯塔省，加拿大，2014年，第637-642页。[7] Graham, R.L.; Lawler, E.L.; Lenstra, J.K.; Rinnooy Kan, A.H.G.《确定性机器调度中的优化与近似：综述》。《离散数学年刊》**1979年**，*5*，287–326。[8] Lenstra, J.K.; Rinnooy Kan, A.H.G.《单机链式作业调度的复杂性结果》。《欧洲运筹学杂志》，4(4)，1980年，270-275。[9] Baptiste, P.《同型并行机上的等长作业调度》。《离散应用数学》，103（1-3），2000年，21-32。