

基于图模型的自动驾驶推理任务调度

王娟娟^{1,2} 乔颖¹ 王宏安¹

¹(中国科学院软件研究所 北京 100190)

²(中国科学院大学 北京 100049)

(wjuanj89@126.com)

Graph-Based Auto-Driving Reasoning Task Scheduling

Wang Juanjuan^{1,2}, Qiao Ying¹, and Wang Hongan¹

¹(Institute of Software, Chinese Academy of Sciences, Beijing 100190)

²(University of Chinese Academy of Sciences, Beijing 100049)

Abstract With the increase of vehicle mounted sensors, the rapid change of urban landmarks and traffic facilities as well as the complex traffic conditions of vehicles and pedestrians, the demand for real-time auto-driving response capability is continuously becoming urgent. How to provide safety guarantee for auto-driving systems by handling the continuing events from sensors and accomplishing the reasoning process via scheduling strategies is worth studying. In this paper, a hard real-time scheduling method of reasoning tasks for automatic driving system is proposed, including a task model based on parallel directed acyclic graphs with hard deadlines, a scheduling algorithm and admission control algorithm to ensure the reasoning operations and reactions within their hard real-time constraints. The experimental results show that our proposed method can effectively increase the success ratio of auto-driving reasoning tasks by average 9.62% and 7.31% compared with the direct scheduling algorithm and model transformation scheduling algorithm; and has also higher admission control capability by average 7.15% compared with the algorithm proposed by Baruah, which is promising to be applied in the auto-driving system for the security concern.

Key words auto-driving; safety-critical; directed acyclic graph; real-time scheduling; admission control

摘要 随着车载传感器设备数量的增多,交通设施和城市地标的快速变化、人车混行的复杂路况,对自动驾驶车辆实时反应的能力要求不断地提高.如何通过带有安全性保证的调度策略来应对物理环境中源源不断产生的传感器实时源事件输入,如何及时地控制传动系统来处理源事件并进行推理操作及其响应以规避危险是值得研究的问题.为此,将自动驾驶汽车视为安全攸关系统,提出了一种硬实时推理任务调度方法,首先为自动驾驶的推理过程建立了基于可并行有向无环图的推理任务模型;其次,提出了自动驾驶推理任务调度算法及其准入算法,保证了所调度的推理任务都能在满足硬实时约束的情况下完成自动驾驶推理操作及其响应动作.最后,进行了模拟实验,实验结果验证了该调度及其准入控制算法的有效性.实验结果表明:推理任务调度算法比直接调度算法和模型转换算法在调度成功率上分别高出9.62%和7.31%,该推理任务准入控制算法比Baruah的准入控制算法在任务集准入率上平均高出7.15%.

关键词 自动驾驶;安全攸关;有向无环图;实时调度;准入控制

中图法分类号 TP391

具备环境感知、自动变道功能的自动驾驶汽车是集成人工智能技术最显著的系统之一。随着云计算、大数据、物联网的快速发展,已经形成了万物互联、海量计算的超融合信息基础设施,并逐步形成了车联网^[1]。目前,已经能够通过摄像头、雷达、RFID、GPS等电子设备,实现对车辆、道路、交通环境等信息的采集,在车-路-人-环境-网络之间进行无线通信和信息交换,并将信息汇集在云端,计算出当前的交通态势以及不同车辆的最佳行驶路线,并及时汇报路况和安排信号灯周期,实现对人、车、路的智能监控、调度和管理。

但是,集成了众多人工智能最新研究成果的自动驾驶系统,仍然面临着诸多关键问题的挑战。即使是经过上百万公里测试的谷歌汽车,也无法辨别水坑、没盖的检修井^[2];已投产并使用的特斯拉 Model S 汽车也出现过伤亡事件。目前,虽然自动驾驶的总里程已经超过了 1.3 亿英里^[3],但是大部分的测试路段是路况较为简单的高速公路,在复杂的人-车混合道路上的测试并不充分。

从本质上看,自动驾驶汽车是集成了传动系统(包括转向控制、制动控制、档位控制、油门控制)、感知系统(例如雷达、摄像头)和推理系统(如图像识别、计算机视觉、模式识别、机器学习等)的轮式智能机器人。

自动驾驶汽车是技术高密度产品,能在路面行驶过程中感知当前路况,对周围环境中的车辆、行人、交通信号、障碍物等进行识别,并根据预设策略进行安全行驶。

自动驾驶汽车经历了 3 个发展阶段:

1) 专用实验型车辆横向/纵向驾驶。在该阶段,大多数的自动驾驶车辆都是以规则匹配为智能引擎、以地理信息系统为导航、以视觉传感器、距离传感器为路况识别、以自动传动为行驶控制的实验型车辆。该类实验车辆已经可以在行人稀少、路况明确、天气良好的环境中,进行无人工干预的自动驾驶行驶。如加州大学伯克利分校的 PATH 项目 1997 年在 7.6 英里公路上实现了由 8 辆车组成的自动驾驶车队^[4]。

2) 开放路况半自动驾驶。目前,谷歌汽车、特斯拉、宝马、奔驰等厂商已经在商用的车型上使用了自动驾驶的系统,能够在无人干预的情况下进行驾驶。

但是由于技术的不完善,仍在每隔一段时间后,提示由人手放在方向盘上进行操作。

3) 开放路况全自动驾驶。在开放路况上进行半自动驾驶的商用汽车,标志了全自动驾驶技术正在逐步成熟;如特斯拉、谷歌汽车等品牌计划推出可以在开放路况上使用的全自动驾驶汽车。

然而,尽管经过了几十万公里或上百万公里的测试,即使如谷歌和特斯拉这般强大的科技实力,仍无法保证自动驾驶汽车能够万无一失地运行。特斯拉在发布全自动驾驶系统后甚至于又进行了相应的降级^[3],将其变更回半自动的驾驶系统。这种情况的出现,主要是由于 3 方面原因:

1) 路况感知信息不完整。由于路况复杂性过大,汽车作为具有高度复杂性、智能化的自动控制系统,在运行过程中各种传感器、网络、传动设备等有发生故障的可能;自动驾驶系统很可能无法完整得到所有“正确”的信息。例如 2016 年 7 月特斯拉 Model S 在美国佛罗里达州的伤亡事件中,由于其车载传感器未能准确识别出一辆白色货车,导致 2 辆车发生了碰撞。针对自动驾驶系统中传感器感知不完全可靠的问题,国内外已经进行了一系列研究^[5-8]。其中,北京大学马连韬等人^[8]为路网数据的潜在错误提出了容错性方案,利用相同车辆及相同路段在误差上存在的内在关联,对缺失数据进行补充;并充分考虑到不同质量传感器对环境友好性评估的影响,确定了基于端设备质量加权的评估计算策略,获得了良好的效果。

2) 路况感知信息冲突。由于车联网的逐步部署,自动驾驶系统的信息来源包括了车载传感器(如摄像、雷达等)和通过地理信息系统获取到的驾驶路线,通过网络传递的信息有可能与车载传感器的信息相互冲突。例如高楼林立的街道,通过偏差的 GPS 定位信息获取到的街景导航地图与车载摄像头、雷达获取到的信息并不一致;道路临时施工封闭,根据原有的道路信息就无法正确实现导航;在路况良好、车辆高速行驶的情况下,出现行人横穿马路的突发事件等等。国内外也已经对路况感知信息冲突的问题开展了相关研究^[9-12]。其中,中国科学院计算技术研究所的陈浩等人^[12]为路况感知信息中出现的高冲突信息提出了 DSIT 的处理冲突方法,通过逻辑运算的证据组合,能较好地适应高冲突信息间的融合。

3) 道路复杂性过大. 在道路上可能会出现大量不可预知的事件, 并且对这些事件进行安全处理的时限不同. 比如, 高速公路上突然有鹿、牛等动物横穿; 在人流密集的地方拐弯时突然出现行人; 在高速路行驶时前方出现深坑. 理论上, 自动驾驶严格应该可以应对任何道路状况. 但是随着车载传感器设备的数量增多, 城市地标和交通设施的快速变化、人车混行的复杂路况, 对自动驾驶车辆实时反应的能力要求不断地提高; 而车载硬件的计算资源能力受限, 这给自动驾驶车辆在实时性上提出了严峻挑战. 车辆在自动驾驶过程中信息处理的资源需求是不一样的. 车辆在正常驾驶的过程中, 仅需比较平滑地处理相应的路况事件即可; 但在突发事件的应急反应过程中, 其处理时间短至秒级, 包括了计算应急措施时间和采取应急措施时间, 大量车载多模传感器所产生的数据没能得到及时推理操作及其相应指令输出. 自动驾驶汽车的计算资源是有限的, 车内的传动装置、刹车装置和转向装置都需要时间进行反应, 如果不通过带有安全性保证的调度策略来应对物理环境中源源不断产生的传感器实时源事件输入, 就无法及时地控制传动系统来处理源事件以规避危险, 而导致灾难性的后果. 如特斯拉 Model S 在 2016 年 1 月的伤亡事件中, 由于前车紧急避让, 导致特斯拉躲避前车时撞上了道路清扫车. 可以看出, 自动驾驶汽车是安全攸关的, 其进行自动驾驶的智能系统必须在一定的时间期限内完成对物理异常情况的推理操作及其响应; 如何保证自动驾驶机动装置能在硬实时约束下完成对大量车载多模传感器所产生数据的及时推理操作及其相应指令输出是个值得研究的问题.

然而目前国内外还缺少针对该问题的相关研究, 为此, 本文提出了一种自动驾驶硬实时推理任务调度机制. 在自动驾驶汽车发现异常事件时, 该机制首先尽可能综合感知当前全面的路况, 对自动驾驶智能推理操作所需的硬实时约束进行定义, 如以当前路况所需车辆传动系统的操作序列和碰撞预期时间为参考, 为自动驾驶推理任务制定截止期; 而后基于可并行的有向无环图定义自动驾驶推理任务模型, 并尽量在调度中利用该任务模型的结构、通过其在多处理器上的并行计算, 给出这些推理任务进行调度的策略, 再通过准入控制算法保证所有进入车载计算系统的推理任务都在其截止期之前完成, 使这些推理任务满足自动驾驶安全攸关系统的硬实时约束, 从而保证了自动驾驶的安全性. 本文对自动驾驶推理任务的调度算法和准入控制的算法进行了实验, 结果表明: 该推理任务调度算法的调度成功率比直接调度算法和模型转换算法的分别高出 9.62% 和 7.31%, 其准入控制算法比 Baruah 的准入控制算法在任务集准入率上平均高出 7.15%.

1 自动驾驶汽车的结构及其推理系统

1.1 自动驾驶汽车的总体结构

图 1 展示了自动驾驶汽车的总体架构模型. 自动驾驶汽车主要包括 3 种系统:

1) 感知系统. 感知系统是自动驾驶汽车的“眼睛和耳朵”, 如果不能准确地识别和感知周围环境, 那么, 路况的认知、驾驶策略和车辆控制都无从谈起. 目前, 车辆感知系统主要包括摄像头、毫米波雷达、超声波雷达、激光雷达等传感器, 获取测量周围

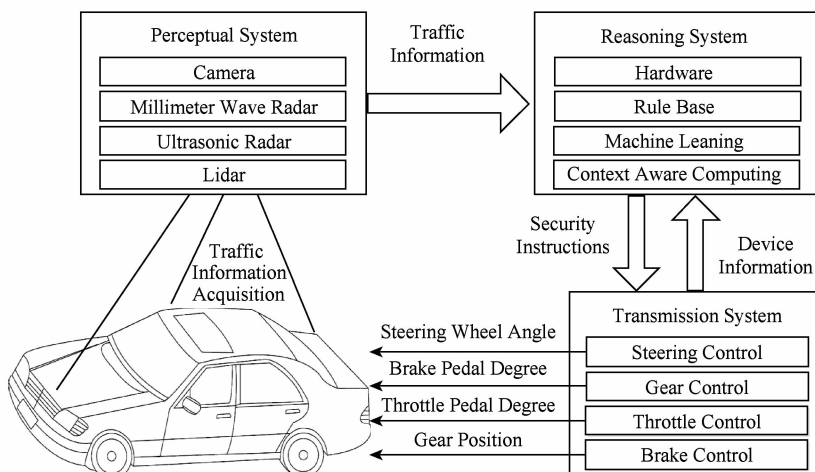


Fig. 1 Architecture of auto-driving cars

图 1 自动驾驶汽车的总体架构

车辆车速、探测周围障碍物等路况信息,并将这些信息传递给推理系统.

2) 推理系统. 推理系统是自动驾驶汽车的“大脑”. 当推理系统接收到路况信息后,将根据已有的规则库对路况进行识别和判断,通过智能推理来决定是否要执行转向、换挡、加/减速、刹车等行为.

3) 传动系统. 传动系统是自动驾驶汽车的“四肢”. 当传动系统接收到自动驾驶的响应动作指令后,会执行相应的转向、制动、加/减油门、换挡等操作.

因此,自动驾驶汽车安全行驶的核心问题是推理系统在当前路况发生改变时,如何通过感知系统接收的路况信息,从知识库中进行相应的规则匹配推理,并给出响应动作指令来指导传动系统,在其截止期到达前完成所有的响应动作.

1.2 自动驾驶推理系统

图2展示了自动驾驶推理系统结构,其主要流程包括5个步骤:

1) 传感信息获取. 自动驾驶汽车拥有大量需要计算的信息,如通过车载传感器获取当前的车况和路况,通过网络传输获取当前的实时街景和路径规划.

2) 信息融合和预处理. 自动驾驶汽车获取到的信息是大量的、复杂的和异构的. 因此,需要通过信息融合技术,进行相互印证和整合,提取出必要的、

可靠的和准确的信息.

3) 判断当前路况. 在感知到当前路况后,需要判断当前汽车是处于正常行驶状态或应急处理状态. 正常行驶是指路况条件较好,车况正常的前提下,仅需保持规定速度和车道即可. 应急处理是指在突发情况下,需要进行紧急刹车、转向灯操作,以避免人员伤亡和经济损失. 应急处理状况需通过应急规则推理引擎,最终将输出截止期内可行的安全操作序列,并作用于传动系统.

4) 规则推理和调度. 应急规则推理引擎先根据规则库,将传感信息转化为连续的事件流,通过时间滑动窗口进行过滤,去除不必要的路况信息. 其次,基于自动驾驶推理任务准入控制算法安排进行车载计算系统的推理任务,并根据自动驾驶推理任务调度算法对已准入的推理任务进行调度,给出推理任务的安全操作序列,来保证所有自动驾驶推理任务均在其截止期之前完成推理操作及其响应动作.

5) 在传动系统中执行安全操作序列.

为了保证自动驾驶的安全性,本文将在第2节中基于可并行的有向无环图来定义自动驾驶推理任务模型,在第3节中基于该任务模型通过分析现有调度算法的问题并给出改进的调度算法及其对应的准入控制算法,并在第4节中通过实验对提出算法的有效性进行验证.

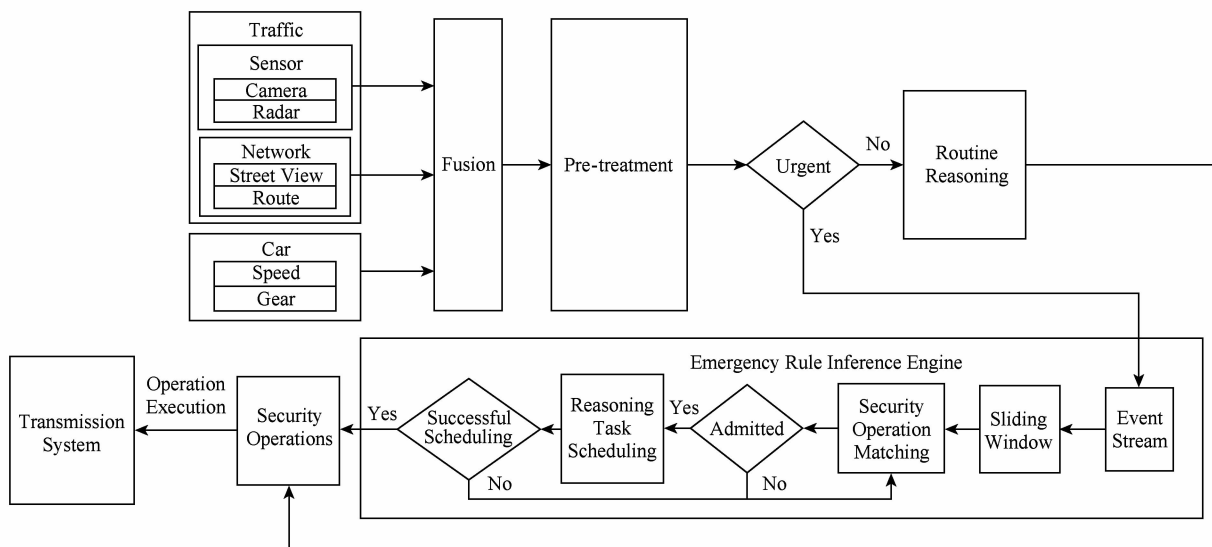


Fig. 2 Auto-driving reasoning system

图2 自动驾驶推理系统

2 自动驾驶推理任务模型

本文的目标是将自动驾驶应急操作规则推理和

决策框架抽象为具有硬实时约束条件的推理过程,并建立相应的模型和求解,以提升自动驾驶过程的安全性. 可并行的有向无环图(directed acyclic graph, DAG)用于任务建模,可在调度中利用该结构的特

点使推理任务在多处理器上实现并行计算且减少推理操作的重复执行,从而提高自动驾驶推理任务调度成功的可能性。为此,本节基于可并行的有向无环图,给出自动驾驶推理任务模型,如图3所示,实时推理任务 τ_i 可形式化地表示成 $(\{\tau_{i,j} \mid 1 \leq j \leq n_i\}, G_i, D_i, T_i)$,其中:

1) 偏序图 G_i . 该有向图定义了 n_i 个子任务 $\tau_{i,j}$ 之间的偏序关系(每个子任务定义了自动驾驶的一个推理操作,不同的子任务可在多处理器上并行或串行执行),决定了 τ_i 的执行流。

2) 相对截止期 D_i . τ_i 从就绪到执行完所允许的最大时间间隔,其值为车载传感器根据车速、制动距离、转向速度等参数所计算出来的值。

3) 周期或最小到达间隔 T_i . τ_i 不同实例就绪时间的间隔,反映了自动驾驶推理任务到达的不同情况。

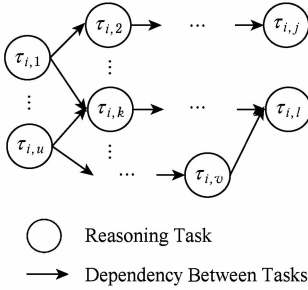


Fig. 3 Reasoning task model

图3 推理任务模型

偏序图 G_i 中,若有边从子任务 $\tau_{i,u}$ 指向子任务 $\tau_{i,v}$,则 $\tau_{i,u}$ 是 $\tau_{i,v}$ 的直接前驱, $\tau_{i,v}$ 是 $\tau_{i,u}$ 的直接后继。 $preddirect(\tau_{i,j})$ 和 $succdirect(\tau_{i,j})$ 分别是 $\tau_{i,j}$ 的直接前驱和直接后继集合。若有路径从子任务 $\tau_{i,u}$ 可达子任务 $\tau_{i,v}$,则 $\tau_{i,u}$ 是 $\tau_{i,v}$ 的前驱, $\tau_{i,v}$ 是 $\tau_{i,u}$ 的后继。 $pred(\tau_{i,j})$ 和 $succ(\tau_{i,j})$ 分别是 $\tau_{i,j}$ 的所有前驱和所有后继的集合。没有前驱的子任务称为源任务,没有后继的子任务称为终任务。

令 τ_i 的偏序图 G_i 中从源任务到终任务的路径所构成的集合(可能存在多条路径)为 P_i ,则 P_i 中最长的路径称为关键路径 P_i^{crit} ,关键路径的长度记为 C_i^{crit} 。关键子任务 $\tau_{i,j}^{crit}$ 是 P_i^{crit} 上的子任务, $\tau_{i,j}^{crit}$ 的执行时间记为 $C_{i,j}^{crit}$ 。那么, G_i 的路径松弛时间为 SL_i : $SL_i = D_i - C_i^{crit}$,这是为保证 τ_i 满足 D_i 能被延迟的最大时间长度。 τ_i 的最坏执行时间 C_i 是 τ_i 所有的子任务执行时间之和, τ_i 的最佳执行时间 C_i^{opt} 是 τ_i 所有关键子任务的执行时间之和。那么, τ_i 的利用率 U_i 为 C_i/D_i 。其中, τ_i 的子任务 $\tau_{i,j}$ 可形式化地表示成 $(r_{i,j}, C_{i,j})$,其中:

1) 就绪时间 $r_{i,j}$. 若 $\tau_{i,j}$ 是源任务, $r_{i,j}$ 是 $\tau_{i,j}$ 实例到达系统的时间(即被系统检测到的时间);否则, $r_{i,j}$ 是 G_i 中 $\tau_{i,j}$ 所有的直接前驱结点都完成的时刻:即 $r_{i,j} = \max(f_{i,k}), \tau_{i,k} \in preddirect(\tau_{i,j})$;

2) 执行时间 $C_{i,j}$. 通过最坏执行时间分析技术(worst-cast execution time analysis)预定义的 $\tau_{i,j}$ 所需执行的时间长度,则时刻 t 的剩余执行时间 $C_{i,j}(t)$ 是 $C_{i,j}$ 减去到时刻 t 为止累积的已执行时间。

表1列出了每个任务 $\forall \tau_i (i \geq 1)$ 及其对应的子任务 $\tau_{i,j} (j \in \{1, 2, \dots, n_i\})$ 在调度中可以用到的参数:

1) 开始时间 s_i 和 $s_{i,j}$. τ_i 和 $\tau_{i,j}$ 开始执行的时间;

2) 完成时间 f_i 和 $f_{i,j}$. τ_i 和 $\tau_{i,j}$ 执行结束的时间,即 $f_i = \min\{t \mid C_i(t) = 0\}, f_{i,j} = \min\{t \mid C_{i,j}(t) = 0\}$;

3) 响应时间 R_i . τ_i 从就绪到完成的时间长度, $R_i = f_i - r_i, R_i \leq D_i$;

4) 绝对截止期 d_i 和 $d_{i,j}$. τ_i 和 $\tau_{i,j}$ 允许的最晚完成时刻, $d_i = r_i + D_i, f_i \leq d_i$;

5) 若 τ_i 所有的子任务 $\tau_{i,j}$ 都在其绝对截止期之前完成,则 τ_i 被成功调度。

Table 1 Parameters of Reasoning Tasks and Sub-Tasks

表1 推理任务及子任务的调度参数

Tasks	Start Time	Finish Time	Ready Time	Response Time	Absolute Deadline
τ_i	s_i	f_i	r_i	R_i	d_i
$\tau_{i,j}$	$s_{i,j}$	$f_{i,j}$	$r_{i,j}$		$d_{i,j}$

若任意实时推理任务集 τ 不能同时满足2个条件,则 τ 将不能在 m 个执行速度为1的处理器上被任何算法调度成功:

1) τ 的利用率不能超过 m ,即 $U(\tau) \leq m$,其中, $U(\tau) = \sum_{\tau_i \in \tau} U_i, U_i$ 是 τ_i 的利用率。

m 个执行速度为1的处理器在长度为 L_i 的时间段上所提供的系统容量最多不超过 $m \times L_i$ 个执行单元;然而,当 $U(\tau) > m$ 时, τ 的总处理需求已超过 $m \times L_i$,无论用什么调度算法,这些处理器都已过载,因此,不能成功调度 τ 。

2) 对 τ 中任意的 τ_i ,其偏序图 G_i 中关键路径的长度 C_i^{crit} 都不超过 τ_i 的相对截止期 D_i ,即 $\forall \tau_i (\tau_i \in \tau): C_i^{crit} \leq D_i$ 。

若 $C_i^{crit} > D_i$,无论用什么调度算法, τ_i 的执行都会超出其相对截止期,则 τ 被调度失败。

3 自动驾驶推理任务调度方法

第 2 节的实时推理任务模型是基于可并行的有向无环图,我们将这种任务简称为并行 DAG 任务,这类任务在多处理器上调度的难度表现为 2 点:

1) 一个推理任务被激活时,只有源任务被激活,其余的子任务在其所有直接前驱都完成时才被激活;即这些子任务的激活是随着不同调度策略下其前驱执行情况的不同而动态变化的.在实时推理任务集被调度的过程中,调度器只知道当前时刻已经激活的子任务,却无法预先得知所有子任务全部的调度信息.

2) 调度过程中推理任务的 DAG 结构在多处理器上的不同并行程度会产生不同的完成时间,即任务的完成时间具有不可预测性,给调度带来不确定性,对可调度性的分析是个挑战.

多处理器上带偏序约束的(如 DAG 偏序约束)可抢占任务调度已被 Ullman^[13]证明是 NP-Hard 问题.那么,多处理器实时推理任务调度也是 NP-Hard,即能判定实时推理任务集可调度性的多项式级或是拟多项式级的调度算法不存在.因此,本节主要是寻求对实时推理任务的有效近似调度算法而非最优算法.

把推理任务整体作为调度算法的输入,将其当成普通任务来考虑调度策略,只在执行推理任务时保证其子任务间的时序约束的调度方法,称其为直接调度方法.这种方法简单易行、复杂度较低,但没

在调度中根据推理任务模型的结构特点给出有意义的指导,只是消极地在推理任务执行时才去满足其子任务的时序约束,使调度失去了灵活性;当推理任务的最坏执行时间超过其相对截止期时,这种方法无法适用,通用性较差.

已有研究者对并行 DAG 任务提出了模型转换的调度思想,这类调度的特点是通过计算所有子任务的局部偏差和局部截止期,把推理任务转换成独立的任务进行等价调度.如 Saifullah 等人提出了任务分解^[14]调度算法,把任务的松弛时间按一定策略分配给所有的子任务后,为子任务分配局部偏差和局部截止期,把 DAG 任务分解转成独立的串行任务进行等价调度.而 Qamhieh 等人则提出了任务延展^[15]调度算法,把任务中的部分可并行子任务按一定策略延展至其截止期(即把关键路径所在的路径延展至该任务的利用率到 1 为止)后,为剩余的子任务分配局部偏差和局部截止期,把 DAG 任务转成独立的串行任务进行等价调度. Qamhieh 等人也对比了任务延展调度算法和任务分解调度算法的可调度性,结果表明任务延展调度算法始终优于任务分解调度算法.

但是,对并行 DAG 任务的模型转换调度使 DAG 任务丢失了部分模型特性(即转换后的任务可调度,原并行 DAG 任务也可调度;但反之不成立),转换后的任务模型不如原 DAG 任务模型通用.而且,文献[16]通过 2 个调度的例子证明了模型转换调度方法和直接调度方法是不可兼容的:如图 4(a)是一个用模型转换调度方法可调度而用直接调度方

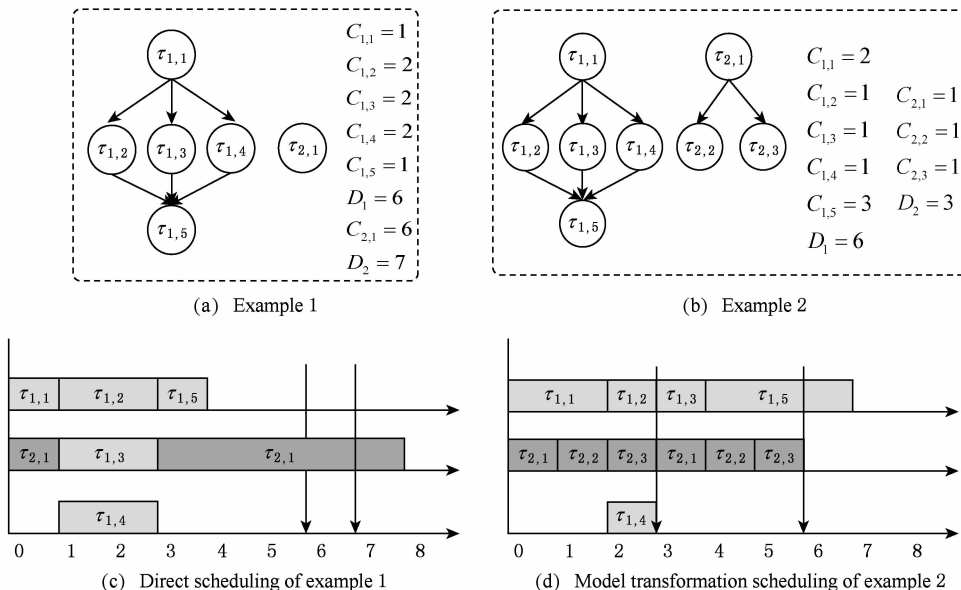


Fig. 4 Incomparability of model transformation scheduling and direct scheduling

图 4 模型转换方法和直接调度方法的不可兼容性

法不可调度的例子,如图 4(b)是一个用直接调度方法可调度而用模型转换调度方法不可调度的例子。

从图 4(c)和(d)调度失败的例子可以看出,直接调度算法和模型转换调度算法调度失败的原因都是调度过程中没有根据当时处理器的空闲情况灵活地调整任务 DAG 结构的并行执行程度.反过来看就是调度算法要通过为 DAG 任务分配不同的并行程度来决定处理器在不同时刻的空闲时间情况。

若一个时间单元在 2 个或多个处理器上都是空闲的(不妨设同一时间单元上空闲的处理器个数为 k),则称该时间单元为并行空闲时间单元,其并行度为 k . 处理器的空闲时间预留不当是因为没有成功做到 3 点:

- 1) 尽可能不多留处理器的空闲时间;
- 2) 在不得不留出处理器的空闲时间时,尽可能不保留并行的空闲时间;
- 3) 已经出现的并行空闲时间尽可能让活动任务及时占用执行。

因此,本节算法的优化目标是使得并行空闲时间单元总数尽可能少而且并行空闲时间单元的并行度也尽可能小.这里,我们用任务比例公平和近似执行速率的思想,先给出优化问题的形式化描述及其优化目标和相应的约束条件,再提出一个有效的近似算法来趋近优化目标。

任务比例公平和近似执行速率的思想可有效地避免贪婪算法引起的处理器空闲时间预留不当问题,并为后续优化目标减少计算规模做准备,因此,先在多处理器上划分时间片,再计算并保证各个时间片内每个活动推理任务的局部工作量,具体分为 3 步:

1) 根据推理任务的截止期不同划分时间片 $TS_j = [t_{j_s}, t_{j_e}]$,并计算 TS_j 内各任务的局部工作量 $LW_{i,j}$;不妨设 $length(TS_j)$ 是 TS_j 的区间长度, $PW_{i,j} = C_i/D_i \times length(TS_j)$,如果 $PW_{i,j}$ 是整数,那么, $LW_{i,j} = PW_{i,j}$;否则, $LW_{i,j} = \lfloor PW_{i,j} \rfloor$ 或 $\lceil PW_{i,j} \rceil$.

2) 若 $\sum_{i=1}^n \sum_{j=1}^{n_j} LW_{i,j} > m \times length(TS_j)$,则以 t_{j_e} 为 d_i 的任务保持 $LW_{i,j}$ 不变;在剩余的任务中取 C_i/D_i 值最小的任务,将该任务的 $LW_{i,j}$ 从 TS_j 中剔除,重复这个过程直至 $\sum_{i=1}^n \sum_{j=1}^{n_j} LW_{i,j} \leq m \times length(TS_j)$. 被剔除的任务需重新计算其在后续各时间片的局部工作量。

3) 推理任务 τ_i 的松弛度 L_i 初始化为其偏序图 G_i 的路径松弛时间 Sl_i ; $L_i = Sl_i = D_i - C_i^{cri}$. 那么,在

每个 TS_j 内,若 $L_i \neq 0$,所有的任务需优先满足 $LW_{i,j}$.

我们把自动驾驶推理任务调度问题描述为:将不同任务看成带各自颜色的棋子,把多处理器看成可填入棋子的棋盘,空白格子是处理器的空闲时间.棋盘矩阵行数是处理器个数 m ,时间片 TS_j 上的棋盘矩阵列数是 p ,在 TS_j 内要分配 $LW_{i,j}$ 的任务个数是 q . 令 $t_{j_s} \leq r \leq t_{j_e}$, $1 \leq t \leq m$,若棋盘矩阵 r 行 l 列的单元格没有填入棋子,则 $P_{r,l} = 0, S_{r,l,k} = 0$ ($1 \leq k \leq q$). 若棋盘矩阵 r 行 l 列的单元格填入棋子 i ($1 \leq i \leq q$),则 $P_{r,l} = i$ (表示该单元格对应的处理器单元是由 τ_i 来占用执行), $S_{r,l,i} = 1, S_{r,l,k} = 0$ ($1 \leq k \leq q$ 且 $k \neq i$).

优化目标:每个时间片 TS_j 内分配的任务需满足 $\max(\min(M_l))$, $M_l = \text{count}(S_{r,l} \neq 0)$,即每一列填入棋子的单元格个数满足其最小值最大,以最大化利用车载处理器的计算能力。

约束条件:

- 1) $\sum_{r=t_{j_s}}^{t_{j_e}} \sum_{l=1}^m \sum_{k=1}^q S_{r,l,k} = LW_{i,j}$ (局部工作量的约束);
- 2) $r_{i,q} = \max(f_{i,p}), \forall \tau_{i,p} \in \text{predirect}(\tau_{i,q})$ (子任务的时序约束)。

根据上述的优化目标和约束条件,我们给出如下的近似算法思想:设定推理任务初始的并行程度,再调整部分任务的执行序列以趋近优化目标.也就是,初始状态尽可能并行地占用处理器,再在调度中动态调整,不断将并行的空闲时间单元减少或将其并行度降低,从而不断接近优化目标.这里不求每个时间片都局部最优,只需满足时间片上的局部工作量即可.该近似算法的核心步骤有 3 步:

1) 计算推理任务 τ_i 的松弛度 L_i ,先按其初始值即其最佳执行时间(最大并行度,上限是 m)的情况来调度,后随着任务执行或被调整(即错开延后来减少并行度)而重新计算 L_i .

2) 任务的 L_i 越小,优先级越高.按优先级高低给任务先后分配 TS_j 上的 $LW_{i,j}$.

3) 推理任务按照工作保留式(work-conserving)调度来占用多处理器.先按照第 2 步执行,若不满足所有任务的 $LW_{i,j}$,则选取 $L_i \neq 0$ 且优先级最低的任务使其错开地延后(使其可并行于子任务部分串行化),重复这个过程直至满足所有任务的 $LW_{i,j}$. 调整后的任务需满足 $L_i \geq 0$,以使所有任务完成 $LW_{i,j}$ 的执行。

为保证上述近似算法的安全性,实时推理任务其准入控制的算法思想是:对加入新任务后的任务集,

计算任务 τ_k 的最坏响应时间;若该任务集中所有任务的最坏响应时间都小于相对截止期,则准入新任务;否则,拒绝新任务.

为计算 τ_k 的最坏响应时间 R_k ,我们需分析 τ_k 的关键连锁干扰 I_k ,那么, R_k 的上限为 $R_k^{UB} = C_k^{cri} + I_k (R_k \leq C_k^{cri} + I_k)$. 其中, I_k 既包括除 τ_k 外的其他推理任务对 τ_k 产生的干扰 I_k^{inter} ,也包括 τ_k 的非关键子任务对其关键子任务在截止期满足所造成的干扰 I_k^{intra} ,即 $I_k = I_k^{inter} + I_k^{intra}$. I_k^{inter} 和 I_k^{intra} 的计算为

$$I_k^{inter} = 1/m \times \sum C_i(R_k^{UB}), \tau_i \in \tau \ \& \ i \neq k.$$

借鉴 Baker 所提出的问题窗口分析方法^[17],可得

$$I_{i,k} = C_i(\lceil t_{ks}, t_{ke} \rceil) =$$

$$\lceil (t_{ke} - t_{ks} + R_i - C_i/m)/T_i \rceil \times C_i;$$

$$I_k^{intra} = I_{k,k}/m = 1/m \times (C_k - C_k^{cri}).$$

为了保证安全性, I_k^{inter} 和 I_k^{intra} 的最坏场景都是造成干扰的推理任务或子任务在 τ_k 的 $[r_k, d_k]$ 时间段的执行长度都是其最坏执行时间,考虑到处理器个数,这些任务在时间轴上对 τ_k 的干扰为它们的最坏执行时间均分在每个处理器的长度.

4 实验与结果

本节基于自动驾驶推理任务模型,随机生成推理任务的测试集,分别用直接调度算法(direct scheduling, DS)、模型转换算法(model transformation, MT)和第3节的推理任务调度算法(RS)进行调度,对比其任务的调度成功率;用第3节的准入控制算法(RS-AC)以及 Baruah^[18]对多处理器 DAG 任务的准入控制算法对这些测试集分别进行准入,对比两者准入能力的高低.

4.1 度量标准

调度成功率为

$$SuccRatio = SuccNum / TaskNum,$$

其中, $SuccNum$ 是测试集中成功被调度的任务个数, $TaskNum$ 是测试集中的任务个数.

任务集准入率为

$$AcRatio = AcTsNum / TasksetNum,$$

其中, $AcTsNum$ 是成功被调度的测试集个数(测试集中的所有任务均被准入), $TasksetNum$ 是随机产生的测试集个数.

4.2 数据集

推理任务进行实验的测试集都是按照以下方式

随机生成的 100 个推理任务. 每个推理任务的结构均是随机的 DAG,其生成方式与规则推理过程类似(从随机源任务开始到生成终任务为止),其中,源任务具有泊松分布(泊松分布参数取在 $[10, 150]$ 间的随机值)的就绪时间. 推理任务的子任务,其执行时间 $C_{i,j}$ 取值为 $[1, 9]$ 之间的随机值;该任务的所有子任务生成完,计算该任务的最佳执行时间 C_i^{cri} 和最坏执行时间 C_i ,设定其 T_i 服从均匀分布 $U[C_i^{cri}, C_i]$,其 D_i 服从均匀分布 $U[C_i^{cri}, T_i]$.

4.3 实验结果

本节将对随机的任务测试集在其负载^①递增的情况下,3种调度算法的调度成功率和2种准入控制算法的任务集准入率,如图5和图6所示.

调度成功率的变化曲线如图5所示,这里比较的是在每个任务集负载下生成的任务集通过直接调度算法(记作 DS)、模型转换算法(记作 MT)和第3节的推理任务调度算法(记作 RS)成功调度的任务比例.图5中 DS 和 MT 曲线的对比表明了直接调度算法和模型转换算法是不兼容的,RS 曲线始终位于 DS 和 MT 曲线的上方,可以看出第3节的推理任务调度算法优于直接调度算法和模型转换算法,且调度成功率平均分别比这两者高出 9.62% 和 7.31%. 这是因为本文提出的推理任务调度算法在调度过程中不断调整任务并行程度以逼近优化目标,因此,它比直接调度算法和模型转换算法出现处理器空闲时间预留不当的几率要小得多.

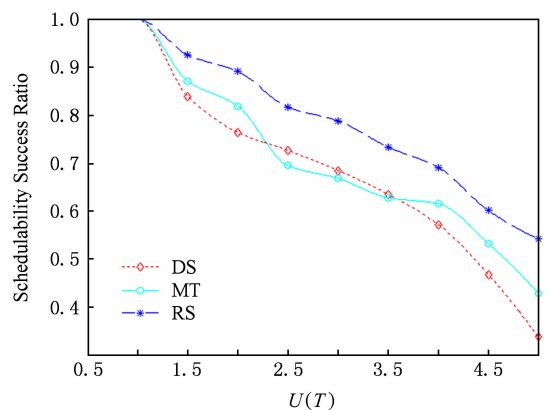


Fig. 5 Comparison of DS, MT, and RS

图5 DS, MT 和 RS 调度成功率对比

任务集准入率的变化曲线如图6所示,第3节的准入控制算法(RS-AC)与 Baruah 的准入控制算法成功准入并调度测试集的数目随着任务负载的递增

① 任务负载为密度之和与处理器个数之比: $U(T) = \sum_{\tau_i \in \tau} U_i/m$, 其中, U_i 是 τ_i 的利用率.

而逐渐减少.图 6 中 RS-AC 曲线始终位于 Baruah 曲线的上方,表明了本文的推理任务准入控制算法比 Baruah 的准入控制算法的准入能力更强,任务集准入率平均高出 7.15%.

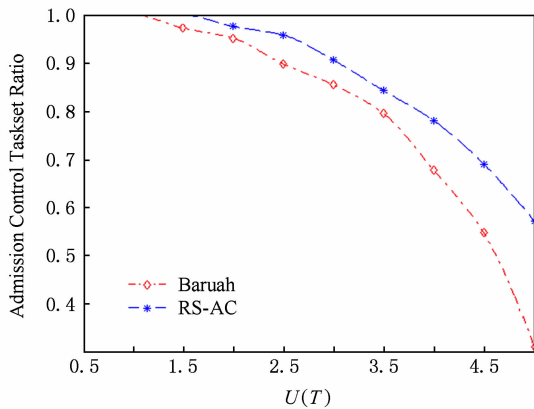


Fig. 6 Comparison of Baruah and RS-AC

图 6 Baruah 和 RS-AC 准入任务率对比

从实验结果可以看出,通过为自动驾驶系统引入硬实时推理任务调度机制,基于可并行的有向无环图定义自动驾驶推理任务模型,并在调度中利用该任务模型的结构根据当前车载处理器的空闲情况灵活安排推理任务的并行计算,且以准入控制算法保证所有进入车载计算系统的推理任务都能在其截止期之前完成,从而提升了自动驾驶时应急避险的安全性.

5 总 结

本文提出了一种自动驾驶硬实时推理任务调度机制,用于在路况发生变化时,自动驾驶汽车可以尽可能综合当前路况,给出满足硬实时约束的自动驾驶智能推理操作及其响应指令序列.针对自动驾驶推理安全操作序列求解问题,本文首先基于可并行的有向无环图定义了自动驾驶的推理任务模型,接着分析了自动驾驶推理任务调度所面临的挑战,并给出了自动驾驶推理任务调度问题的形式化定义、优化目标以及约束条件,再通过近似算法解决了该问题,也提出了相应的准入控制算法.最后,通过实验结果验证自动驾驶推理任务调度模型及其求解算法有效地提升了自动驾驶的安全性.在下一步工作中我们将结合自动驾驶中传感器对路况的感知不完全可靠和可能存在相互冲突信息的情况以及自动驾驶技术中出现的新方法细化本文的推理任务模型,并在此基础上,探索新的自动驾驶推理任务调度方法.

参 考 文 献

- [1] Liu Xiaoyang, Wu Minyou. Vehicular CPS: An application of IoT in vehicular networks [J]. Journal of Computer Applications, 2012, 32(4): 900-904 (in Chinese)
(刘小洋, 伍民友. 车联网: 物联网在城市交通网络中的应用 [J]. 计算机应用, 2012, 32(4): 900-904)
- [2] Lee G. Major limitations hidden in Google automatic driving vehicles [J]. China Incubator, 2014, 16(8): 14-15 (in Chinese)
(李·戈梅斯. 谷歌自动驾驶汽车隐藏着一些重大的局限 [J]. 科技创业, 2014, 16(8): 14-15)
- [3] Parkin S, Li Yumeng. The paradox of automatic driving vehicle [J]. China Civil and Commercial, 2016, 12(8): 82-83 (in Chinese)
(Simon Parkin, 李雨蒙. 自动驾驶汽车面临的悖论 [J]. 中国民商, 2016, 12(8): 82-83)
- [4] Chen Xiaobo. Challenges and prospects for the development of automatic driving [J]. Integrated Transportation, 2016, 17(11): 9-13 (in Chinese)
(陈晓博. 发展自动驾驶汽车的挑战和前景展望 [J]. 综合运输, 2016, 17(11): 9-13)
- [5] Rohani M, Gingras D, Vigneron V, et al. A new decentralized Bayesian approach for cooperative vehicle localization based on fusion of GPS and VANET based inter-vehicle distance measurement [J]. IEEE Intelligent Transportation Systems Magazine, 2015, 7(2): 85-95
- [6] Hostettler R, Djurić P M. Vehicle tracking based on fusion of magnetometer and accelerometer sensor measurements with particle filtering [J]. IEEE Trans on Vehicular Technology, 2015, 64(11): 4917-4928
- [7] Boada B L, Boada M J L, Diaz V. Vehicle sideslip angle measurement based on sensor data fusion using an integrated ANFIS and an unscented kalman filter algorithm [J]. Mechanical Systems & Signal Processing, 2016, 72(3): 832-845
- [8] Ma Liantao, Wang Yasha, Peng Guangju, et al. Evaluation of GPS-environment friendliness of roads based on bus trajectory data [J]. Journal of Computer Research and Development, 2016, 53(12): 2694-2707 (in Chinese)
(马连韬, 王亚沙, 彭广举, 等. 基于公交车轨迹数据的道路 GPS 环境友好性评估 [J]. 计算机研究与发展, 2016, 53(12): 2694-2707)
- [9] Vanderhaegen F. A rule-based support system for dissonance discovery and control applied to car driving [J]. Expert Systems with Applications, 2016, 65(3): 361-371
- [10] Almodfer R, Xiong S, Fang Z, et al. Quantitative analysis of lane-based pedestrian-vehicle conflict at a non-signalized marked crosswalk [J]. Transportation Research Part of Traffic Psychology & Behaviour, 2016, 42(5): 468-478

- [11] Morando A, Victor T, Dozza M. Drivers anticipate lead-vehicle conflicts during automated longitudinal control; Sensory cues capture driver attention and promote appropriate and timely responses [J]. *Accident Analysis & Prevention*, 2016, 97(6): 206-219
- [12] Chen Hao, Wang Rui, Sun Rongli, et al. DSIT: An evidence reasoning method for information fusion in wireless sensor networks [J]. *Journal of Computer Research and Development*, 2015, 52(4): 972-982 (in Chinese)
(陈浩, 王睿, 孙荣丽, 等. DSIT: 面向传感网信息融合的证据推理方法[J]. *计算机研究与发展*, 2015, 52(4): 972-982)
- [13] Baruah S, Bonifaci V, MarchettiSpaccamela A, et al. A generalized parallel task model for recurrent real-time processes [C] //Proc of the 33rd IEEE Real-Time Systems Symposium. Piscataway, NJ: IEEE, 2012: 63-72
- [14] Saifullah A, Agrawal K, Lu C, et al. Multi-core real-time scheduling for generalized parallel task models [C] //Proc of the 32nd IEEE Real-Time Systems Symposium. Piscataway, NJ: IEEE, 2011: 217-226
- [15] Qamhieh M, George L, Midonnet S. A stretching algorithm for parallel real-time DAG tasks on multiprocessor systems [C] //Proc of the 22nd Int Conf on Real-Time Networks and Systems. New York: ACM, 2014: 13-22
- [16] Qamhieh M, Midonnet S. Simulation-based evaluations of DAG scheduling in hard real-time multiprocessor systems [J]. *ACM SIGAPP Applied Computing Review*, 2015, 14(4): 27-39
- [17] Baker T P. Multiprocessor EDF and deadline monotonic schedulability analysis [C] //Proc of the 24th IEEE Real-Time Systems Symposium. Piscataway, NJ: IEEE, 2003: 120-129
- [18] Baruah S. Improved multiprocessor global schedulability analysis of sporadic DAG task systems [C] //Proc of the 26th Euromicro Conf on Real-Time Systems. Piscataway, NJ: IEEE, 2014: 97-105



Wang Juanjuan, born in 1989, PhD candidate. Her main research interests include real-time intelligence and real-time scheduling.



Qiao Ying, born in 1973, PhD, professor. Her main research interests include real-time intelligence and real-time scheduling.



Wang Hongan, born in 1963, PhD, professor and PhD supervisor. His main research interests include real-time intelligence, real-time scheduling and human-computer interaction.