

摘 要

随着汽车保有量的快速增长，交通安全隐患问题日益突出。为此，我国实行车辆年检制度，由交管部门对在用车辆定期检测和维修。传统汽车检测线由于检测容量有限，不能满足汽车安全性能检测需求量的大幅增长，多数汽车检测站引入了并行汽车检测系统（Parallel Vehicle Inspection System, PVIS）。现有并行汽车检测系统车辆调度多依靠人工来完成，待检车辆的检测周转时间较长，检测设备资源分配不均，PVIS 运行效率低下。

针对上述问题，本文以给定数量待检车辆的检测周转时间最短为优化目标，采用赋色 Petri 网（Colored Petri Net, CPN）对并行汽车检测系统进行建模，提出了基于此模型仿真的并行汽车检测线车辆调度策略，具体的研究内容如下：

1、通过分析并行汽车检测系统车辆调度的时空约束，建立了并行汽车检测系统（Timed Colored Petri Net, TCPN）模型，包括车辆分派过程模型和车辆检测过程模型两个层次，通过对模型的仿真运行，验证了模型的可用性和有效性。

2、针对并行汽车检测系统车辆调度的分派难题，提出车辆队列拆分算法（Vehicle Queue Splitting, VQS），该算法将待检车队按照 PVIS 高效率运作的原则合理拆分为两个子队列，分派至各车辆检测子系统（Vehicle Inspection Sub-System, VISS），并从子队列长度均衡性和 VISS 工位负载均衡性两个指标对所提 VQS 算法的性能进行分析；

3、为了减少 VQS 算法拆分后的各子队列的检测周转时间，进而实现给定数量待检车辆的检测周转时间最短的目标，提出基于 PSO 的检测线车辆调度算法，该方法属于求解调度问题的近似方法，对 VQS 算法拆分后的子队列进行调度得到近似最优的车辆序列，该车辆序列在上述 TCPN 模型中仿真运行得到检测用时。实验结果表明所提策略在并行汽车检测系统车辆调度优化问题中具有良好的性能。

关键词：并行汽车检测系统，汽车检测子系统，Petri 网，粒子群优化，车辆调度

Abstract

With the rapid growth of vehicles, traffic safety problems have become increasingly prominent. To this end, my country implements an annual vehicle inspection system, and the traffic control department regularly inspects and maintains vehicles in use. Due to the limited testing capacity of traditional automobile inspection system, it cannot meet the substantial increase in the demand for automobile safety performance inspection. Most automobile inspection stations have introduced Parallel Vehicle Inspection System (PVIS). At present, PVIS vehicle scheduling is mostly done manually, the inspection turnaround time of the vehicles to be inspected is long, the distribution of inspection equipment resources is uneven, and the PVIS operation efficiency is low.

In view of the above problems, this paper takes the shortest testing turnaround time for a given number of vehicles to be tested as the optimization goal, and uses Colored Petri Net (CPN) to model the parallel vehicle testing system, and proposes a parallel simulation based on this model. The vehicle scheduling strategy of the vehicle inspection system, the specific research content is as follows:

1. By analyzing the spatiotemporal constraints of vehicle scheduling in the PVIS, the Timed Colored Petri Net (TCPN) model of the parallel vehicle inspection system is established, including the vehicle dispatching process model and the vehicle inspecting process model. The simulation of the model verifies the usability and validity of the model.

2. Aiming at the problem of vehicle scheduling in PVIS, a Vehicle Queue Splitting (VQS) algorithm is proposed, which divides the fleet to be inspected into two sub-queues reasonably according to the principle of efficient operation of PVIS. It is then dispatched to each vehicle inspection subsystem (VISS), and analyzed the performance of the proposed VQS algorithm from the two indicators of sub-queue length balance and VISS station load balance.

3. In order to reduce the inspection turnaround time of each sub-queue split by the VQS algorithm, and then achieve the goal of the shortest inspection turnaround time for a given number of vehicles to be inspected, a vehicle scheduling algorithm for vehicle inspection systems (VIS) based on Particle Swarm Optimization (PSO) is proposed, which belongs to solving scheduling problems. The approximate method is to schedule the sub-queues splitted

by the VQS algorithm to obtain an approximate optimal vehicle queue. The vehicle queue is simulated and run in the above TCPN model to obtain the testing time. The experimental results show that the proposed strategy has good performance in the PVISvehicle scheduling optimization problem .

Key words: Parallel vehicle inspection system, vehicle inspection sub-system, Petri net, particle swarm optimization, vehicle scheduling

目 录

第一章 绪论	1
1.1 研究背景及意义	1
1.2 国内外研究现状	2
1.2.1 汽车检测系统调度问题的研究现状	2
1.2.2 Petri 网在调度问题中的应用	4
1.3 本文主要内容和章节安排	5
第二章 相关理论基础	7
2.1 调度问题概述	7
2.1.1 调度问题的分类	7
2.1.2 调度问题的研究方法	8
2.2 元启发式算法的调度方法	9
2.2.1 基于群体智能的算法	9
2.2.2 局部搜索算法	10
2.3 Petri 网的简介	11
2.3.1 CPN 的简介	11
2.3.2 HCPN 的简介	12
2.3.3 TCPN 的简介	13
2.4 本章小结	13
第三章 并行汽车检测线车辆调度的 TCPN 模型	15
3.1 并行汽车检测线车辆调度的场景介绍	15
3.1.1 并行检测线工位布局	15
3.1.2 并行检测线车辆调度的时间约束	16
3.1.3 并行汽车检测线的参数设置	17
3.1.4 并行检测线车辆调度问题描述	19
3.2 基于 TCPN 的并行汽车检测线调度系统建模	20
3.2.1 模型参数设置	20
3.2.2 车辆分派过程模型	21
3.2.3 车辆检测过程模型	22
3.3 本章小结	31
第四章 基于并行汽车检测系统的车辆分派算法与分析	33
4.1 并行汽车检测系统车辆调度框架	33
4.2 基于 PVIS 的车辆分派算法与分析	34
4.2.1 车辆队列拆分算法	35
4.2.2 算例分析	41

4.3 车辆队列拆分算法的实验分析.....	42
4.3.1 子队列长度均衡性.....	42
4.3.2 VISS 工位负载均衡性.....	43
4.4 本章小结.....	45
第五章 基于 PN 的并行检测线车辆调度策略及分析.....	47
5.1 基于粒子群的检测线车辆调度优化算法.....	47
5.1.1 粒子群算法基本原理.....	47
5.1.2 粒子群车辆调度优化算法.....	49
5.1.3 算例分析.....	52
5.2 基于 PN 的 VQS-PSO 车辆调度策略实验分析.....	54
5.2.1 有效性分析.....	54
5.2.2 复检率分析.....	56
5.3 本章小节.....	59
第六章 总结与展望.....	61
6.1 论文总结.....	61
6.2 工作展望.....	62
参考文献.....	63
攻读学位期间取得的研究成果.....	67
致 谢.....	69

分类号：TP39

10710-2019224016



长安大学

专业硕士学位论文

基于 Petri 网的并行汽车检测系统建模
与车辆调度

王贞

导师姓名职称	丁爱玲 教授		
申请学位类别	工程硕士	专业学位类别 及领域名称	计算机技术
论文提交日期	2022 年 4 月 29 日	论文答辩日期	2022 年 6 月 9 日
学位授予单位	长安大学		

第一章 绪论

1.1 研究背景及意义

我国经济的快速发展和交通基础设施的持续建设，促使汽车行业的蓬勃发展，随之增加的还有汽车逐年的保有量。据相关部门统计，我国汽车增量连续三年超过 1000 万辆^[1-3]。车辆数的大幅增长易引发交通堵塞和交通事故等交通安全相关问题。为了解决该问题，一方面，相关研究者们致力于研发安全系数高的车辆；另一方面，由检测机构负责定期对车辆进行安全性能检测。

汽车检测系统（Vehicle Inspection System, VIS）是一种自动化测量控制系统，主要用于对车辆的动力性能、安全性能、燃油经济性能及尾气排放性能等指标的检测。近年来，汽车检测需求不断攀升^[4-8]。为了在更短的时间内满足更多车辆用户的检测需求，一方面，研究人员致力于提高传统的汽车检测系统的检测效率^[9-10]；另一方面，检测机构在有限的场地里平行部署多个汽车检测系统，以期提高单位时间内的车辆的检测数量。

并行汽车检测系统（Parallel Vehicle Inspection System, PVIS）是指在同一个检测车间内平行部署两个或两个以上的汽车检测子系统（Vehicle Inspection Sub-System, VISS），各 VISS 所需的场地大小、检测设备及工位布局相同，且各 VISS 能够独立的、并行的运行。对于 PVIS 而言，待检车辆仍然按照到达的时间顺序排成队列，称之为待检车队。待检车队中的车辆有可能被分派到任意一个 VISS 上，这依赖于当前 PVIS 中各个 VISS 及各个工位的工作状态。

我国现有的汽车检测线车辆调度多依靠人工来完成，由于车辆调度需要获取各个工位和设备的检测状态以及车辆的检测需求后进行匹配，具有鲜明的组合优化的特征，而人工操作员所能观察到的检测状态又是有限的和局部的，据此做出的车辆调度方案有可能是低效的甚至是冲突的。此外，人工操作员基于某个时刻的当前状态做出决策，没有从待检车队整体的角度研究和判断，因此待检车队的整体检测效率可能较低。总之，依靠人工调度很难实现 PVIS 高效运转的目标。因此，急需一种切实可行的汽车检测线车辆调度方案以提高 PVIS 的高效性。

PVIS 车辆调度主要研究“待检车辆到 VISS 的合理分配”及“车辆的合理检测顺序”的问题，在此基础上通过合理的调度，优化“待检车辆的检测周转时间最短”这一目标。合理的调度可提高 PVIS 的有效性并减少不必要的等待时间，优化现有流程的同时充分

利用检测资源。因此，有必要对并行汽车检测系统的车辆调度问题进行深入研究。

Petri 网因其分布式的特征常被用于 PVIS 的建模，其可准确分析 PVIS 中不同工位之间复杂的逻辑关系，并从宏观角度上对检测过程中车辆的行为与彼此之间的依赖、约束和同步关系进行模拟与验证^[11]。因此，本文基于时延赋色 Petri 网构建了 PVIS 的 TCNP 模型，提出了合理的调度策略并通过仿真加以验证。由于并行汽车检测系统具有分布式系统的特性，而分布式系统的调度问题最主要的是合理的资源分配，针对 PVIS 车辆调度的分派难题，我们提出了车辆队列拆分（VQS）算法，以合理分配 PVIS 中各 VISS 检测资源。而粒子群优化算法具有较强全局寻优能力、搜索速度快，以及在求解大规模问题上的显著优势，我们提出了基于粒子群的车辆调度算法，优化 VQS 算法拆分后子队列中车辆的检测次序。可以看到，本文将传统汽车检测系统中的车辆调度研究延伸至 PVIS 中，同时可为分布式生产系统的调度问题提供一些新的解决思路。

1.2 国内外研究现状

近年来，调度问题在各个领域受到了广泛关注，其核心是模型与算法^[12]，因此大量学者对其进行研究。本节对汽车检测线的调度问题现状和 Petri 网在调度问题中的应用现状进行介绍。

1.2.1 汽车检测系统调度问题的研究现状

汽车检测系统是一种特殊的生产系统，因此汽车检测线调度问题与生产系统调度问题有许多相似之处，汽车检测过程可看作生产系统的作业过程，故本节将汽车检测线调度问题作为生产系统调度问题的实例展开研究。

国内外相关学者对生产系统调度问题进行了系统而深入的调查与研究，取得了众多研究成果。现有关于调度问题的求解方法主要集中在理论层面，zuidai 通过精确的数学方法求全局最优解^[13]，常常使用的数学方法有分解法、分支定界、混合整数线性规划（mixed integer linear programming, MILP）等。

Hui 等人针对多级多产品批处理工厂的短期调度，提出了一种连续时间混合整数线性规划模型。该模型通过最大限度地减少订单完成的早慢来确定最佳顺序以及将客户订单分配到不相同的处理单元^[14]。Kopanos 等人针对多机批处理工厂短期调度问题，将拉格朗日分解技术应用用于 MILP 模型进行求解^[15]。Boom 等人通过切换最大加线性（SMPL）模型来描述半循环离散事件系统的调度问题，对 MILP 问题引入了一种新的重新参数化，

将 SMPL 系统的模型预测调度设计问题重新描述为一个混合整数线性规划 (MILP) 问题^[16]。

上述数学方法通过对目标函设立明确的约束条件, 实现了对调度问题的精确建模与求解, 然而多约束条件会使调度问题复杂化, 增加假设条件不足以满足大规模求解问题的需求, 致使精确方法在调度上呈现出局限性^[11]。近似寻优方法的出现打破了该局限性。近些年来, 部分学者受启发于自然界生物的行为模式而提出元启发式算法并引起了众多学者们的广泛关注。

Chen 等人针对柔性流水车间调度问题, 采用一种基于改进粒子群算法的新算法来搜索动态柔性流水车间调度问题的帕累托最优解^[17]。Pan 等人提出了一种新的混合离散粒子群优化算法, 用最小化最大完成时间 (makespan) 的准则解决无空闲置换流水车间调度问题^[18]。Zhang 等人针对有限缓冲区流水车间调度问题, 以最大完工时间为目标提出混合人工蜂群算法, 采用带权重的曲线拟合启发式算法改进初始种群生成质量, 在离散人工蜂群算法的引领蜂阶段嵌入遗传算法, 实验表明基于插入和交换邻域的邻域搜索算法进一步增强了混合算法的局部开挖能力^[19]。Zhang 等人结合混合零空闲置换流水车间调度问题, 将禁忌算法融入基于概率模型的分布估计算法中, 提出一种禁忌分布估计算法求解混合零空闲置换流水车间问题^[20]。Marichelvam 等人针对具有序列依赖性设置时间的流水车间调度问题, 开发了一种受生物启发的元启发式算法, 即萤火虫算法 (FA), 考虑了最小化制造跨度, 平均流速, 平均迟到和迟到作业的数量这四种绩效指标。实验结果表明, 所提出的 FA 比许多其他算法更有效^[21]。Shen 等人提出了一种基于多目标进化算法 (MOEA) 的主动-被动方法, 它能够同时处理包括效率和稳定性在内的多个目标, 通过结合启发式动态优化策略快速适应新环境, 并同时处理机器分配和操作顺序两种调度策略^[22]。Li 等人针对具有广义优先关系的资源均衡问题, 开发了一种新型遗传算法 (GA), 设计和实现具有高效的调度生成方案, 用于该算法能够在几分之一秒内获得小实例的最优度差距小于 2% 的最优解^[23]。Yu 等人针对多码头具有多个收货和装运月台的越库配送系统的卡车调度问题, 提出了一种新的混合遗传算法, 目的是最大限度地减少最大完成时间。评估结果表明, 与现有的启发式方法相比, 混合 GA 提供了更好的解决方案^[24]。

鉴于精确的数学方法和近似的元启发式方法各出所长, 越来越多的学者将研究重心转移到二者的融合或多种启发式算法的融合, 集各方式的优化机制于一体, 从深层次的角度对调度问题进行优化求解, 进而达到更高层次的优化目标。

Rajendran 等人针对两阶段流槽调度问题, 开发了一个分支绑定算法和两个启发式算法, 目的是在获得最佳制造跨度的情况下最小化总流速^[25]。Dai 等人提出了一种高效节能作业车间调度问题, 先建立了一个混合整数规划模型, 以最大限度降低能耗, 然后提出了一种增强的分布估计算法解决该问题, 采用分布估计算法进行全局搜索, 改进的模拟退火算法进行局部搜索, 结果表明该方法可以有效解决所提问题^[26-27]。Ritzinger 等人针对乘车调度问题, 提出了一种精确的动态规划算法和基于动态规划的元启发式算法用于限制解空间^[28]。Fonseca 等人将交叉对接中心的卡车调度问题建模为具有优先约束的双机流车间调度问题, 调度目标是最大限度地减少制造跨度。所提方法基于体积算法在时间索引公式上求解的拉格朗日松弛, 使用多项式时间启发式方法在计算效率高的时间内生成有效的上限和下限^[29]。

Ghaleb 等人针对无等待流水车间调度问题, 提出了一种有效的混合粒子群优化 (PSO) 元启发式算法, 该算法采用 Nawaz-Enscore-Ham (NEH) 启发式技术对种群进行高效初始化, 采用 PSO 机制引导的进化搜索以及基于局部邻域搜索的模拟退火, 避免了传统方法陷入局部最优的缺陷, 实现全局探索与局部开发的适当平衡^[30]。Bao 等人以最大完工时间为目标, 针对阻塞流水车间调度问题, 提出了一种混合和声搜索改进遗传算法, 并使用局部搜索和 NEH 启发式方法对混合算法进行了改进, 实验结果证明改进后的混合算法对于优化阻塞流水车间调度问题求解十分有效^[31]。Lourenço 等人针对公共汽车调度问题提出基于禁忌搜索和遗传算法的多目标元启发式, 克服了数学规划技术在多个目标相互冲突的问题, 这些方法已成功纳入 GIST 规划运输系统^[32]。Lee 等人针对具有紧急工作和有限等待时间的流水车间调度问题, 提出了一种迭代贪婪算法和一种模拟退火算法, 解决了混合整数规划 (Mixed Integer Programming MIP) 求解计算时间久的问题, 并验证了所提出算法的效率及有效性^[33]。

1.2.2 Petri 网在调度问题中的应用

VIS 和 PVIS 均为由离散事件驱动的调度系统, 因此, 汽车检测线车辆调度问题可以通过离散事件系统 (DES) 的方法进行建模和优化。Petri 网作为一种分布式系统模型, 被广泛应用于各类生产系统的建模, 因其强大的图形描述和逻辑分析能力, 已成为 DES 建模、分析和仿真的主要工具。

Yang 等人通过分析车辆检测过程的时间特性, 采用面向资源 Petri 网建模, 在此基础上提出一种启发式算法有效解决短期车辆调度问题^[10]。Gonzalo 等人针对汽车座椅罩

制造商的一个复杂的现实生产车间，介绍了一种基于 Petri 网的资源分配和调度方法，用 Petri 网进行过程和作业的建模，大量实例测试显示了 Petri 网在建模和优化现实系统方面的潜力^[34]。An 等人分析车辆检测过程中的驱动事件和状态演变，并采用赋色 Petri 网建立汽车检测线串行批处理调度模型^[11]。Mui 等人提出了一种基于特征 Petri 网的半导体制造系统模型，实现了半导体制造系统的建模及车产车间调度^[35]。Amira 等人研究了标记 Petri 网中最小初始标记（MIM）计算的 NP-Hard 组合优化问题，同时考虑一系列标签以最小化柔性制造系统（FMS）中的资源消耗，提出了一种基于蚁群优化（ACO）的方法，用于优化 FMS 中的资源分配和调度^[36]。Zhang 等人使用混合 Petri 网来描述原油操作的行为，给出了获得可行调度的初始状态要求，并提出了调度方法，大量工业案例研究表明所提方法计算效率较高^[37]。Wu 等人针对具有晶圆重新访问的单臂集群工具满足晶圆驻留时间限制的问题，发明对具有原子层沉积（ALD）工艺的单臂簇工具，开发 Petri 网络模型来描述系统的动态行为并给出了调度算法^[38]。Basak 等人提出一种灵活的汽车制造方法，用于利用高级 Petri 网对 FMS 的车间调度问题进行建模和分析，通过对土耳其法雷奥的灵活汽车制造系统（FAMS）的实时调度和控制进行建模，阐述了所提出的基于 PN 的方法^[39]。Zhu 等人介绍了用于柔性制造系统（FMS）控制的 Petri 网模型的设计与实现，提供了一种面向对象的 Petri 网方法来进行性能建模并实现有效的生产控制^[40]。Li 等人针对容易出现死锁的柔性装配系统（FAS）的调度问题，构建了 FAS 的 Petri 网模型，提出启发式波束搜索（HBS）算法来最小化制造跨度。为了在可接受的时间内找到更好的解决方案，将 HBS 算法与迭代策略集成得到 IWHBS 算法。实验结果验证了 IWHBS 算法相对于其他启发式搜索算法的有效性和优越性^[41]。

1.3 本文主要内容和章节安排

本文的研究内容主要有以下几个方面：

第一章，绪论。本章首先介绍了 PVIS 出现的背景及其研究的意义，然后介绍了目前国内外有关调度问题的研究现状，并针对这两部分内容阐述了相关研究现状，最后论述了论文的基本结构和创新点。

第二章，相关理论基础。本章主要介绍调度问题的相关理论基础，详细阐述调度的分类及研究方法，在基础上介绍了基于元启发式方法的调度问题；最后对 Petri 网的基本理论、基本操作及赋色 Petri 网（CPN）、分层赋色 Petri 网（HCPN）和时延赋色 Petri 网（TCPN）这三种高级 Petri 网进行介绍。

第三章，并行汽车检测系统（PVIS）车辆调度的场景介绍及 TCPN 模型。本章首先对 PVIS 场景进行介绍，详细介绍了 PVIS 的工位布局、约束条件和参数设置，并举例阐述了 PVIS 车辆调度问题；在此基础上提出 TCPN 模型，并介绍车辆在该模型中的检测过程。

第四章，针对并行汽车检测系统（PVIS）车辆调度的分派难题，提出车辆队列拆分算法（VQS）。首先，介绍了本文所提并行汽车检测系统车辆调度策略（VQS-PSO）的总体框架；然后，提出了车辆队列拆分（VQS）算法，并结合算例说明了算法的有效性，并从子车队长度均衡性和 VISS 工位负载均衡性两方面分析了 VQS 算法的性能。

第五章，针对车辆队列拆分算法（VQS）得到的子队列的优化问题，提出基于 PSO 的车辆调度优化算法。首先，提出了基于 PSO 的车辆调度优化算法，结合算例给出了该算法的优化过程；然后从有效性和复检率两个方面对本文所提的并行汽车检测线车辆调度策略（VQS-PSO）进行实验分析，并证明了该策略的良好性能。

第二章 相关理论基础

2.1 调度问题概述

调度，是实际工作中广泛存在的运筹学问题，它被广泛地应用于工业生产、交通运输和航天航空等多个领域。调度问题的关键在于模型和算法，前者的研究涉及了问题建模、目标函数和调度规则，后者的研究涉及问题可解性、计算复杂性以及算法有效性。如今，计算机的广泛应用对调度问题的求解产生了巨大影响。本节将从多个角度展开对调度问题的详细介绍。

2.1.1 调度问题的分类

Johnson 于 1954 年正式拉开了调度问题的序幕。典型的调度问题一般分为两种：流水车间调度（Flow-Shop Scheduling Problem, FSP）和作业车间调度（Job Shop Scheduling Problem, JSP）^[13]。

（1）流水车间调度

流水车间调度的特点：该问题中，待检工件的众多加工工序分别在不同的机器上完成，且在每台机器上执行的时间已知，加工工序之间的依赖关系已知，后一个工序需等前一工序执行完后才能加以处理，不同待检工件的执行方式一致。

流水车间调度适用场景：适用于在统一流水线模式下生产某种产品。像熟知的汽车装配线、食品生产线等都是经典的 FSP 问题。

（2）作业车间调度

作业车间调度的特点：该问题中，同样具有若干不同的机器，一台机器处理一道工序，不同于流水车间调度，该问题中工件的处理路线因工件而异。

作业车间调度的适用场景：应用领域广泛，凡是包含流水线作业调度的众多场景，都适用于该调度。

但是，汽车检测线相关的调度问题与流水车间有区别，但汽车检测线调度问题是一种类似流水线作业高速运转的过程，和流水车间又存在着很多的相似之处，故在本文中将其称作是一种特殊的流水车间问题。表 2.1 详细给出汽车检测线问题与 FSP、JSP 调度问题之间的对比。

表 2.1 检测线问题与两类经典调度问题的对比

调度分类	相同点	不同点
流水车间	加工顺序一定 加工时间一定 加工步骤已知	有重入过程 存在潜在的循环 在同一台机器的加工过程不止一次
作业车间	重入的步骤已知 重入步骤数量较少 重入次数较少	不允许改变加工顺序 无加工步骤的随机性 更多表现为流的结构

2.1.2 调度问题的研究方法

调度问题在现有研究中已被证明是复杂的 NP-hard 问题，针对不同的应用场景，其解决方法和侧重点也不尽相同。通常 NP-hard 问题的解法分为精确方法和近似方法两类 [13]。



图 2.1 调度问题的研究方法

图 2.1 为现有关于调度问题的主要解决方法，其中精确方法可以求解到调度问题的全局最优，但是过大的规模会大幅增加求解难度和复杂度，因此精确方法多用于小规模问题的求解。在大型调度问题中，采用调度问题中近似求解方法来确定最优或次优解可能更适合。因此，近几年来研究者对于求解调度问题的研究方向逐步从精确求解转向了多种近似解的方法。在求解调度问题的发展历程中，科学家们对自然界各种生物和人类行为典型特征的仿真研究，得出了一种新的研究思路。这类算法的核心是探索和利用，从有效信息中进行全局搜索找到近似最优解，它可以尽可能避免传统算法的缺陷，高效的获得可行解，因此受到了广泛的关注和研究。本文所提方法正是基于 Petri 网和粒子群优化算法用于求解并行汽车检测系统车辆调度问题的一种近似求解方法。

2.2 元启发式算法的调度方法

近年来，由于对自然界中的生物和人类行为等方面开展研究，相关学者们提出了一类新的求解方法即元启发式算法，这为求解组合优化问题提供了一种新的思路。元启发式算法通过搜索全局解空间实现在合理的成本范围内寻找最佳方案。本小节主要对基于群体智能的算法和局部搜索算法展开详细讨论。

2.2.1 基于群体智能的算法

研究者们受启发于自然界的演化机制和物理学原理，提出了基于智能群体的迭代寻优算法来解决调度问题。该类算法优化效率高、通用性强的优点，在多个领域均受到了越来越广泛的重视和应用。下面介绍群体智能的三种典型算法。

(1) 蚁群优化算法

意大利学者 Dorigo 在 1992 年受到蚂蚁系统的信息分享机制的启发提出了蚁群算法。学者发现，在没有任何线索提示的情况下，生物界的蚂蚁能找到从巢穴到食物来源的最优途径，原因是当它们在觅食时，它们所经过的地方会分泌一种特殊的“信息素”，蚁群中的其他蚂蚁根据路径上的信息素浓度选择自身的行进路线，信息素浓度越高被选择的几率越大，蚂蚁走过的路径又会留下新的信息素浓度增量，引导更多的蚂蚁对该路径的选择，蚁群就会按照“正反馈”的方式，聚集在一条路线上，导致该路径信息素的浓度就会越来越高，经过数次迭代后，该条路径的信息素浓度明显高于其它路径，该路径则为蚁群算法求得的近似最优解。

(2) 粒子群算法

1995年美国的 Kennedy 和 Eberhart 两位博士从鸟群捕食行为中受到启发,提出了粒子群优化算法 (Particle Swarm Optimization, PSO) [42]。粒子群算法是用一种没有重量的粒子来模拟鸟群体中的鸟类,每个粒子的位置和速度属性用以决定下一轮迭代的运动变化。每一个粒子都会在一个独立的搜索空间中寻找最优的结果,通过与自身的历史最优解相比较得到当前粒子的个体极值,并与所有的粒子共享。在整个粒子群中,选择众多粒子的个体极值点中的最优值,作为种群极值,下次迭代过程中,粒子会根据个体极值和群体极值来更新自身的速度和位置,使得自身更加趋向近似最优解。

(3) 遗传算法

遗传算法 (Genetic Algorithms, GA) 是受启发于自然选择和种群遗传而提出的算法,模拟自然选择过程中的繁殖、杂交和突变。在该算法中,问题的每个可能解对应一条“染色体”,众多个体组成一个群体 (即全部可能的解)。遗传算法通过随机的方式生成若干个体 (也就是最初的解),并对每个个体构建目标函数,用以后期性能评估,在此基础上,通过一些选择、交叉和突变等遗传操作,一定概率地选取部分好的个体用来修改种群,组成新的个体,目的是寻找问题的近似最优解。由于遗传了前一代的某些优良特性,这一代的个体表现出的性能比上一代更好,通过这种方法,可以将遗传算法看作是一种由可能的解构成的群体的演化过程。

2.2.2 局部搜索算法

局部搜索算法 (Local Search, LS) 是指将问题的解无限接近于最优解的一种算法,对于复杂的 NP 问题,解计算的难度也是呈指数增长的,因此,在搜索中通过向距离目标最近的方向搜索,寻求最优解的临近值。其中常用的算法包括:模拟退火 (Simulated Annealing, SA)、禁忌搜索 (Tabu Search, TS)、多点局部搜索 (Multi-start Local Search) 等。

(1) 模拟退火算法

1983年 Kirkpatrick 等人发现了最优问题和物理退火的相似之处,它是一种基于概率的局部搜索算法。为了实现模拟退火 (SA),需确定起点,通过某个目标函数缓慢降低温度,每个优化问题都有一个能量最小的最终状态,问题的目标都是达到最终状态。在终态时,系统将达到稳定状态。因此,模拟退火算法可以用于优化问题,因为它可使系统达到稳定状态。

(2) 禁忌搜索算法

禁忌搜索算法于 1986 年在美国被 Glove 教授提出,是一个模拟人智能过程的算法,改算法允许脱离当前解,使目标函数变差,以此避免出现局部最优,禁忌搜索(TS)在数次寻优过程中搜索到的最优解会存储在禁忌表格中,加入禁忌表中的解将不会在下次迭代搜索中被检索到,除非用赦免标准来赦免某些优秀的解,从而跳出局部最优解。

(3) 多起点局部搜索算法

多起点局部搜索算法最重要的特点之一是它的自适应特性,提高了算法的性能。该算法生成多个不同的初始解,实现自适应扰动机制使搜索过程搜索空间中的不同区域。多起点局部搜索算法适用于容易构造解而难以使局部搜索邻域在可行搜索空间内移动较远的问题。

2.3 Petri 网的简介

Petri 网是开发分布式调度方法并将这些开发与最先进的调度优化算法集成的有效工具,由于其分布式的特性,Petri 网常用来描述系统中复杂的逻辑关系,不仅能够清晰地描述系统的结构,而且能够阐述系统的动态行为^[43]。

2.3.1 CPN 的简介

赋色 Petri 网(CPN)是一种高级的 Petri 网,对复杂的分布式系统具有更强的分析处理能力,其可更加准确的贴合真实 VIS 各工位之间错综复杂的逻辑关系和车辆、检测场地之间的约束关系^[11]。针对 CPN 的形式化定义如下所示。

定义 1: 赋色 Petri 网表示为一个九元组^[44]:

$$CPN = (P, T, A, C, V, W, G, E, I)$$

各元组含义在表 2.2 中给出

表 2.2 赋色 Petri 网各元组的含义

元组名称	含义
P	表示库所的集合
T	表示变迁的集合
A	表示有项弧的集合
C	表示非空的有限颜色集
V	类型化变量的集合
G	表示守卫函数

表 2.2 赋色 Petri 网各元组的含义 (续)

元组名称	含义
E	表示弧表达式函数
I	表示初始化函数

2.3.2 HCPN 的简介

使用 CPN Tools 建模复杂且庞大的网系统时, 通过对模型分层以实现将网系统划分成多个子网, 这就是 HCPN 的本质^[11]。分层赋色 Petri 网 (HCPN) 的定义如下:

定义 2: 分层赋色 Petri 网可用一个四元组表示^[44]:

$$\Sigma_H = (S, SM, PS, FS)$$

其中:

S : 为网系统中模块的有限集合, 每个模块都表示为一个赋色 Petri 网;

$SM: T_{\text{sub}} \rightarrow S$ 表示子模块函数, 它为每个替换变迁分配一个子模型;

PS : 表示端口套接字函数, 它分配一个端口套接字函数 $PS(t) \subseteq P_{\text{sock}}(t) \times P_{\text{port}}^{SM(t)}$ 给每个替换变迁, 对于所有的 $t \in T_{\text{sub}}$ 和 $(p, p') \in PS(t)$ 都需要满足 $ST(p) = PT(p')$, $C(p) = C(p')$ 和 $I(p)\langle \rangle = I(p')\langle \rangle$;

$FS \subseteq 2^p$ 表示非空的融合库所集合, 因此 $C(p) = C(p')$ 和 $I(p)\langle \rangle = I(p')\langle \rangle$ 代表所有 $p, p' \in fs$ 和 $fs \in FS$ 。

下面介绍本文构建的并行汽车检测系统 TCPN 模型中涉及到有关分层赋色 Petri 网的一些术语:

(1) 替代变迁

通常使用替代变迁搭建一个 CPN 模型的顶层网络, 然后替代变迁内部则是子网具体实现。即将原来单一页面的复杂 CPN 模型的某一区域用替代变迁来表示, 这样原来复杂的模型就被改造成了逻辑结构简单的顶层网络模型。然后替代变迁内部则是某一区域的具体实现, 逻辑关系并没有被改变。

(2) 子页和父页

父页是指 CPN 模型的顶层逻辑结构, 父页面中会包含多个替代变迁, 子页是指替代变迁所替代的功能相对应的页面。

(3) 融合库所

融合库所和普通库所的不同在于融合库所中的 token 可以在网系统的不同页面上共享,也就是说如果一个库所需要不同的页面上执行不同的功能并需确保该库所包含的 token 在网中有一致性时,这时候就需要创建融合库所来实现该需求。

(4) 端口库所和槽库所

端口库所和槽库所的功能是为了实现模型中的父页面和子页面进行通信。端口库所和槽库所分别特指子页面中的关联库所和父页中的关联库所。端口库所包括三类:子网的输入库所(IN 库所)、子网的输出库所(OUT 库所)、同时负责子网输入和输出的库所(I/O 库所)。

2.3.3 TCPN 的简介

TCPN 是在原 Petri 网的基础上,增加了时间因素^[43-44]。时间因素值为附加在 token 上的数字时间戳,用以表示变迁触发所需要的时间,随着网系统的运行实时更新。这样,网系统的运行轨迹就会被投影到一个统一的时间坐标上。因此,为了更好地建模车辆在检测线中每个工位上的检测时间,本文采用 TCPN 对汽车检测线调度问题进行建模。下面给出 Jensen 在 1995 年提出的时延赋色 Petri 网(TCPN)的原始定义。

定义 3: 时延赋色 Petri 网是一个三元组^[44],形式化表示为:

$$TCPN = (CPN, R, r_0)$$

其中:

R : 表示时戳, $R \geq 0$;

r_0 : 表示初始时钟矢量, $r_0 \in R$ 。

2.4 本章小结

本章首先对调度问题做了简要概述,其次介绍了调度问题现有的研究方法,包括精确方法和近似求解方法;然后从基于群体智能算法和局部搜索算法两大方面对近似求解方法中的元启发式方法作以介绍;最后,介绍了 Petri 网的基本理论,包括 CPN、HCPN 及 TCPN 这三种高级 Petri 网的形式化定义及相关概念。

第三章 并行汽车检测线车辆调度的 TCPN 模型

3.1 并行汽车检测线车辆调度的场景介绍

汽车安全性能检测系统是汽车检测站内布设的一种常用生产系统，其主要用途是在汽车不解体的前提下，检查运行车辆的各项技术指标是否合格，主要包括制动、速度、灯光等检测项目，并且对第一次检查不合格的车辆维修后再次检测。

常用汽车安全检测系统是指在汽车检测站内设有三个检测工位，如图 3.1 所示，每个检测工位对应不同的检测项目，待检车辆依次通过检测线的一工位、二工位、三工位，在每个检测工位完成相应的检测任务。

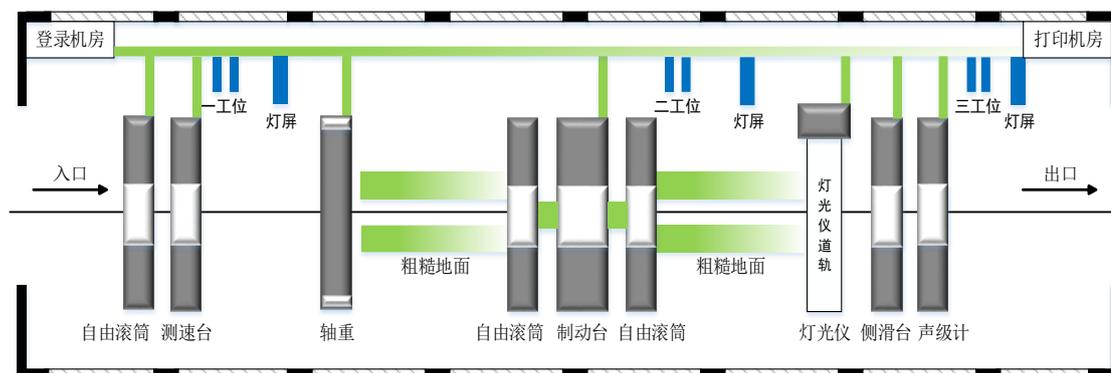


图 3.1 汽车安全性能检测线布局

3.1.1 并行检测线工位布局

PVIS 中每一个独立的 VISS 根据检测需要会设置 3 至 5 个检测工位。图 3.2 所示的 PVIS 由 2 个 VISS 和一个调度模块组成，且在每个 VISS 中设置了 3 个检测工位。第一工位的检测项目是速度和废气，第二工位的检测项目是轴重和制动，第三工位的检测项目是汽车前照灯，侧滑和声级。需要明确指出，每个工位的检测项目只能使用专用的检测设备。例如，一工位需要安装速度台和废气仪，二工位需要安装轴重和制动测试台架，三工位需要安装声级计，侧滑台和前照灯测试仪及其导轨，这些检测设备安装后位置无法更改，而且检测项目必须与检测设备相匹配。

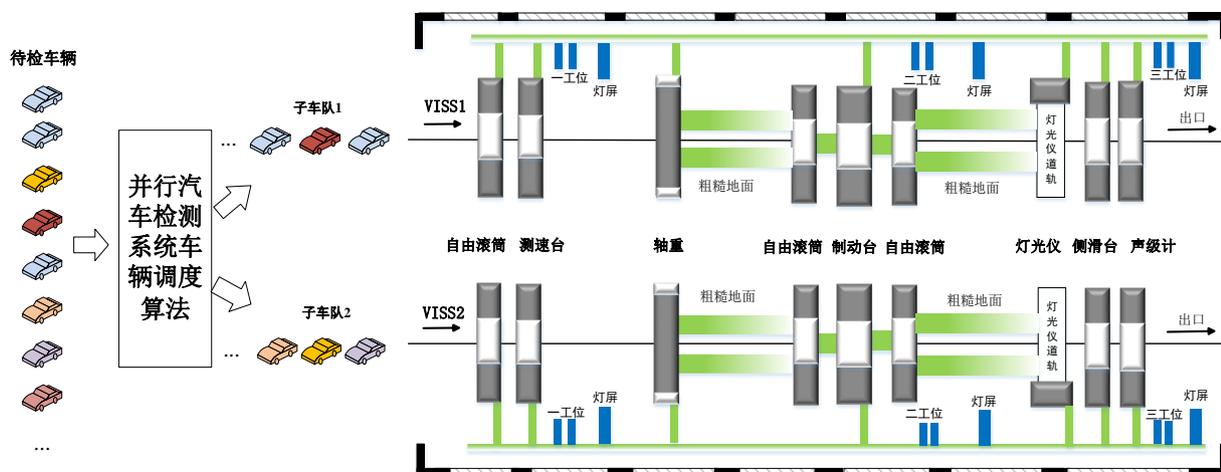


图 3.2 并行汽车检测系统布局

3.1.2 并行检测线车辆调度的时间约束

我们详细说明车辆检测过程应满足的时间约束,这是建立 PVIS 的 PN 模型的基础。令 $VISS = \{VISS_1, VISS_2, \dots, VISS_L\}$, $V = \{v_1, v_2, \dots, v_K\}$, $Ts = \{Ts_1, Ts_2, \dots, Ts_J\}$ 分别表示车辆检测子系统,待检车辆及检测工位,其中 L, K 和 J 是非负整数。符号 $T_{(VISS_l, v_k, Ts_j)}^{start}$ 和 $T_{(VISS_l, v_k, Ts_j)}^{end}$ 分别表示待检车辆进入 $VISS_l$ 中检测工位 Ts_j 的时间点和离开检测工位 Ts_j 的时间点,车辆在某个 $VISS$ 上的检测应满足以下约束:

- 1) 车辆 v_k 在 $VISS_l$ 的工位 Ts_j 检测结束的时间应不早于检测开始的时间:

$$T_{(VISS_l, v_k, Ts_j)}^{end} \geq T_{(VISS_l, v_k, Ts_j)}^{start} \quad (3.1)$$

- 2) 车辆 v_k 在 $VISS_l$ 中工位 Ts_{j+1} 的检测开始时间应不早于在工位 Ts_j 的结束检测时间:

$$T_{(VISS_l, v_k, Ts_{j+1})}^{start} \geq T_{(VISS_l, v_k, Ts_j)}^{end} \quad (3.2)$$

- 3) 相邻两辆车 v_k 和 v_{k+1} 在 $VISS_l$ 中工位 Ts_j 的检测时间应满足:

$$T_{(VISS_l, v_{k+1}, Ts_j)}^{start} > T_{(VISS_l, v_k, Ts_j)}^{end} \quad (3.3)$$

- 4) 相邻两辆车 v_k 和 v_{k+1} 在 $VISS_l$ 中相邻工位同时检测的时间约束:

$$T_{(VISS_l, v_{k+1}, Ts_j)}^{start} \geq T_{(VISS_l, v_k, Ts_{j+1})}^{start} \quad (3.4)$$

在 VISS 的检测过程中有很多并发的检测过程,当计算 K 辆车总的检测时间时,如果简单地累加每辆车的检测时间,则并发检测的时间消耗就会被重复计算。为了消除这种重复计算,我们定义了车辆的检测周转时间的概念,只要待检车辆的数量及其车辆的排队顺序是已知的,那么 K 辆车的检测周转时间 T 可以按以下式计算:

$$T = T_{(v_k, Ts_j)}^{end} - T_{(v_1, Ts_1)}^{start} \quad (3.5)$$

但是，由于在 PVIS 中调度单元将给定数量的待检车辆分为两个及两个以上车辆子队列，子队列中的车辆依次进入其对应的 VISS，我们将各 VISS 上车辆子队列的检测周转时间记为：

$$T_l = T_{(VISS_l, v_k, Ts_j)}^{end} - T_{(VISS_l, v_1, Ts_1)}^{start} \quad (3.6)$$

综上所述，给定数量的待检车辆的检测周转时间 MS 可以按以下式求得：

$$MS = \max(T_1, T_2, \dots, T_L) \quad (3.7)$$

为了定量分析 PVIS 的整体检测效率，其目标函数可以定义如下：

$$\min MS = \max(T_1, T_2, \dots, T_L) \quad (3.8)$$

除了上述时间约束外，PVIS 车辆调度还需满足以下约束：

- 1) 待检车辆进入某个 VISS 后，它在完成所有检测项目检测之前不能进入 PVIS 中并行布设的其他 VISS；
- 2) 根据文献^[10]的论证，对于表 3.1 而言，尽管每个工位的实际检测耗时不同，但是我们仍为每个工位设置了统一的工位周期，记为 Π ，对于实际检测时间小于 Π 的工位增加等待时间 w ，这种处理方法在实际的 VIS 中是可行的；
- 3) 分配到同一个 VISS 的多辆汽车，既可以有初检车辆也可以有复检车辆；
- 4) 对于复检车辆而言，前后两次可以进入不同的 VISS 进行检测。

3.1.3 并行汽车检测线的参数设置

由图 3.2 中并行汽车检测线的工位布局可知，VISS 由三个检测工位组成，且各工位的检测设备和检测项目都是固定的，因此每个检测项目所需的检测时间也是确定的，VISS 中检测项目及工位的检测耗时如表 3.1 所示。

表 3.1 VISS 检测项目和检测耗时

工位	检测项目	检测耗时(分钟)	工位耗时(分钟)
第一工位 (FS)	速度	4	4
第二工位 (SS)	轴重	2	5
	制动	3	
第三工位 (TS)	灯光	2	6
	侧滑	2	
	声级	2	

待检车辆被分派到某个 VISS 后，与传统的 VIS 类似，要依次在 VISS 的每个工位进行检测，所有项目的检测结果合格才视为完成检测。如果待检车辆是第一次进行检测，我们称之为初检车辆。有时，部分车辆会出现一个或者多个项目检测不合格的现象，必须对此类车辆进行维修，维修后的车辆将再次前往检测系统重新接受检测，这类待检车辆被称之为复检车辆。通常，待检区域的车辆既有初检车辆也有复检车辆。基于此，把初检车辆与复检车辆分开进行检测是 PVIS 中进行车辆调度的基本的，直观的方法。在第五章，这种调度方法将被作为一个参照与本文所提车辆调度方法进行比较。

由于待检车辆中既有初检车辆也有复检车辆，针对三个工位的 VIS 我们定义了待检车辆所有可能的七种检测需求 (TR)，TR1 代表一个完整的检测过程，初检车辆在三个检测工位顺序进行检测，依次进行了速度计，轴重和制动器，以及前照灯，侧滑和声级等项目的检测。TR2 到 TR7 分别表示 6 种复检车辆的检测工位组合。TR2 表示车辆只需要检测一工位的项目，同样，TR4 和 TR6 分别表示车辆只需在二工位和三工位检测。TR3 表示该车辆需完成一工位和二工位的检测项目，TR5 和 TR7 对应的检测需求则分别是检测二工位和三工位，以及一工位和三工位。待检车辆的七种检测需求各自对应的工位序列及检测周转时间见表 3.2。

表 3.2 车辆检测需求工位组合

检测需求	检测序列	检测时间 (分钟)
TR1	FS → SS → TS	18
TR2	FS	6
TR3	FS → SS	12
TR4	SS	6
TR5	SS → TS	12
TR6	TS	6
TR7	FS → TS	12

接着，我们给出每种检测需求的二进制表示，用三位二进制数来直观地描述每种检测需求对应的检测工位组合，数字“1”表示待检车辆在某个工位上有检测任务，数字“0”表示待检车辆在该工位上没有检测任务，第一位对应工位一，第二位表示第二工位，第三位表示第三工位，例如车辆的检测需求为 TR7，其对应的二进制表示为“101”，即检测需求为 TR7 的车辆只在一工位和三工位上有检测任务。

待检车辆的检测需求按照上述原则表示为三位二进制数后，三位二进制数中 1 的个数即为工位数。检测需求为 TR1 的全检车辆，其二进制表示为“111”，工位数为 3；检测需求为 TR3，TR5，TR7 对应的二进制表示分别为“110”，“011”，“101”，工位数均为 2；检测需求为 TR2，TR4，TR6 对应的二进制表示分别为“100”，“010”，“001”，工位数为 1。待检车辆的七种检测需求各自对应的二进制表示及工位数见表 3.3。

表 3.3 车辆检测需求二进制表示

检测需求	二进制表示	工位数
TR1	111	3
TR2	100	1
TR3	110	2
TR4	010	1
TR5	011	2
TR6	001	1
TR7	101	2

按照这种表示方法，只要待检车队中待检车辆的检测需求是已知的，任意一个车辆队列都可以用一组二进制数来表示。

3.1.4 并行检测线车辆调度问题描述

我们以三辆车的分派过程为例说明并行汽车检测线车辆调度问题，待检车辆检测需求的二进制表示如图 3.3 所示，并行汽车检测线车辆调度就是以给定三辆车的检测周转时间最短为目标。已知 VISS 的工位周期为 Π ， v_i 分派到 VISS1， v_j 分派到 VISS2，此时 v_k 有两种选择：



图 3.3 三辆车的待检车队

(1) 若 $v_k \rightarrow$ VISS1: VISS1 对应的车辆队列为 $v_k \rightarrow v_i$ ，VISS2 对应的车辆队列为 v_j ，车辆检测过程见图 3.4；

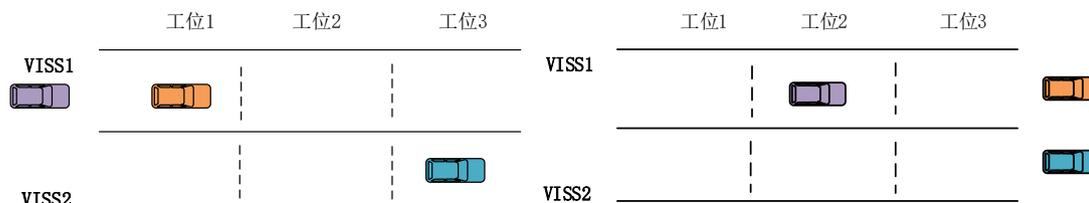


图 3.4 $v_k \rightarrow$ VISS1 车辆检测过程

如图 3.4 所示三辆车完成检测需要两个工位周期，第一个工位周期，橙色车辆在 VISS1 中工位 1 进行检测，紫色车辆在 VISS1 入口处等待，蓝色车辆在 VISS2 中工位 3 进行检测；第二个工位周期，橙色车辆在检测完成离开 VISS1 后，紫色车辆才能在工位 2 检测，蓝色车辆检测完成离开 VISS2，VISS2 三个检测工位均空闲。采取该种调度方法三辆汽车 v_i 、 v_j 、 v_k 所需的检测周转时间为 2Π 。

(2) $v_k \rightarrow$ VISS2: VISS1 对应的车辆队列为 v_i ；VISS2 对应的车辆队列为： $v_k \rightarrow v_j$ ，车辆检测过程图 3.5；

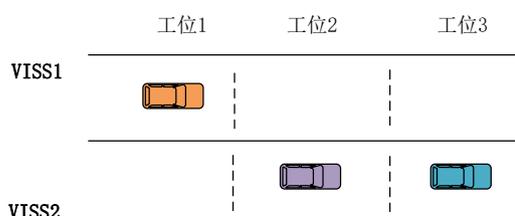


图 3.5 $v_k \rightarrow$ VISS2 车辆检测过程

如图 3.5 所示三辆车完成检测仅需一个工位周期，该工位周期内橙色车辆在 VISS1 中工位 1 检测，紫色车辆和蓝色车辆分别在 VISS2 中工位 2、工位 3 同时检测，采取该种调度方法三辆汽车 v_i 、 v_j 、 v_k 所需的检测周转时间为 Π 。

PVIS 中各条 VISS 是独立地运行，很多时候待检车辆在某一个 VISS 中需要等待，在另一条 VISS 则不需要等待直接检测。因此，车辆队列中哪些车去 VISS1 检测，哪些车去 VISS2 检测，会影响各 VISS 的检测效率，进而影响待检车队的检测用时。

3.2 基于 TCPN 的并行汽车检测线调度系统建模

赋色 Petri 网 (CPN) 作为一种高级 Petri 网，具有对复杂的分布式系统进行建模、分析和验证的能力，能够详细、准确的模拟 VIS 各工位之间待检车辆、检测场地和检测设备之间的依赖、约束和同步关系^[8-9]。本节采用赋色 Petri 网 (CPN) 建模并行汽车检测系统 (PVIS)，并在模型基础上提出并行检测系统车辆调度策略。

3.2.1 模型参数设置

本节介绍并行汽车检测系统 TCPN 模型中用到的颜色集及变量的定义，具体定义如表 3.4 所示：

表 3.4 TCPN 模型中颜色集和变量定义

名称	类型	描述	
		定义	含义
INT	Colset	int	整数颜色集
QUE	Colset	string	字符串颜色集
DATA	Colset	string timed;	带时戳的字符串颜色集
intLIST	Colset	list DATA	表示待检车辆元素的颜色集
XT	Colset	product DATA*INT timed	时戳字符型和整型表示的颜色集
E	Colset	with e	输入文件颜色集
di(i=1,2,3)	Var	DATA	检测线工位 i 中检测设备组的变量
c	Var	DATA	检测过程中车辆移动轨迹的变量
oq	Var	intLIST	待检车辆列表以及待检车辆列表中每辆车的变量
ml	Var	intLIST	待检车辆列表中进入检测线以外余下其他车辆的变量

3.2.2 车辆分派过程模型

如图 3.6 所示, 在初始标识下, Infile 变迁使能, 当该变迁触发时调用代码段读入待检车辆队列文件, 构造待检车辆列表后存入库所 Que。之后, Dispatch 变迁使能, 在其对应的代码段中车辆队列拆分算法被执行, 其结果是将原来的待检车辆队列拆分为两个待检车辆子队列。然后, 变迁 Schedule1 和 Schedule2 使能, 在其对应的代码段中基于 PSO 的车辆调度优化算法均被执行, 得到了两个车辆排队顺序已被优化的子队列, 分别存入库所 Osque1 和 Osque2, 这样优化后的子队列分别进入 VISS1 和 VISS2。通过 VISS 模型的仿真运行, 得到各待检车辆子队列的检测周转时间。需要特别说明, 变迁 Schedule1 和 Schedule2 的代码段可以调用不同的车辆分派算法, 从而得到不同的待检子队列。

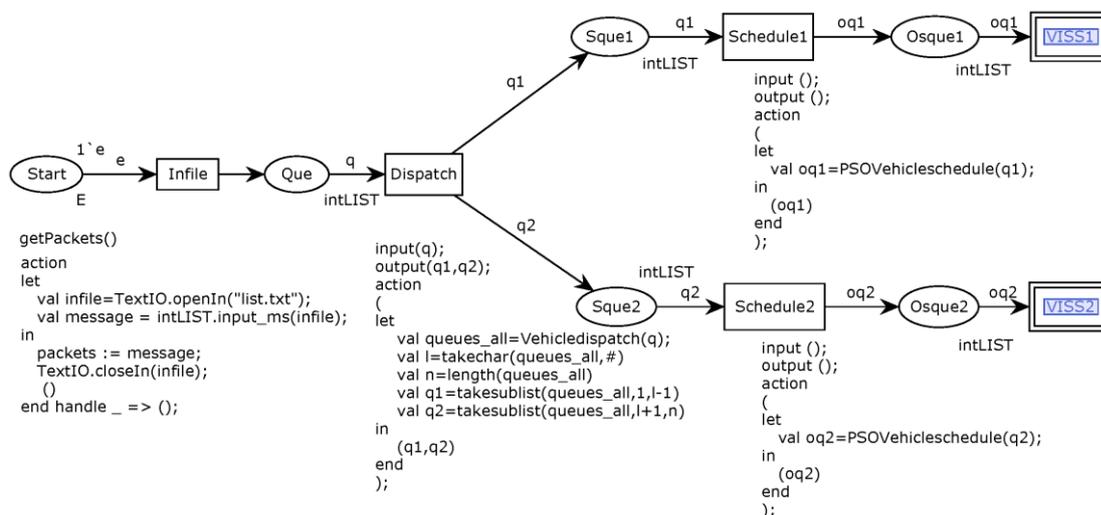


图 3.6 并行检测线车辆分派过程示意图

根据第二节中 PVIS 组成和特征可知 PVIS 中各 VISS 的工位布局，检测项目及检测顺序是相同的。因此，我们以车辆在某个 VISS 上的检测过程为例，建模车辆在每个工位及每个检测设备上的检测过程，详细说明以变迁序列的触发过程表示检测过程中必须满足的时间约束和场地约束，以及如何利用 CPN 模型的仿真运行来得到该待检车辆队列的检测周转时间。

3.2.3 车辆检测过程模型

本节构建的并行汽车检测系统 TCPN 模型中所涉及到的库所、变迁及其物理意义均在表 3.5 中给出。

表 3.5 TCPN 模型中库所、变迁的物理意义

元素	元素类型	物理意义
ENT	库所	检测线入口
OUT	库所	检测线出口
RDTij, i=1,2,...,7, j=1,2,3	库所	检测需求为 i 的车准备在 j 工位检测
ARRij, i=1,2,...,7, j=1,2,3	变迁	检测需求为 i 的车辆到达第 j 工位
TSij, i=1,2,...,7, j=1,2,3	库所	检测需求为 i 的车辆正在检测第 j 工位
TSEij, i=1,2,...,7, j=1,2,3	库所	检测需求为 i 的车辆在工位 j 检测完成
Waitij, i=1,2,...,7, j=1,2,3	变迁	检测需求为 i 的车辆在工位 j 等待
RDLij, i=1,2,...,7, j=1,2,3	库所	检测需求为 i 的车准备离开 j 工位
Passij, i=1,2,...,7, j=1,2,3	库所	检测需求为 i 的车辆通过第 j 工位
LEAi3, i=1,2,...,7	变迁	检测需求为 i 的车离开第三工位
TSDi, i=1,2,...,7	变迁	检测需求为 i 的车辆完成检测任务
LEAi, i=1,2,...,7	变迁	检测需求为 i 的离开检测线
TFDi, i=1,2,3	库所	三个工位的检测场地
EQGi, i=1,2,3	库所	三个工位的检测设备

本节中，我们利用 CPN 提供的层次化建模方法，把图 3.6 中用双框表示的替代变迁 VISS1/2 的实际执行过程展开。展开后的 VISS1/2 模型仍然可以分为两层，上层模型如图 3.7 所示，我们称之为 VISS 父页，父页面中包含了 Input、TR1、TR2、TR3、TR4、TR5、TR6、TR7、Output 这 9 个替代变迁，这些替代变迁分别对应着下层的 9 个子页。Input 子页先读入待检车辆子队列，车辆根据其检测需求进入相应子页执行检测任务，TR_i (i=1,2,...,7) 子页的功能是模拟检测需求为 TR_i 的车辆在汽车检测线中的检测过程，并将检测结果传给 Output 子页，TR_i(i=1,2,...,7)的具体定义见表 3.2，Output 子页的功能是接收并记录待检车辆子队列中车辆的检测周转时间。

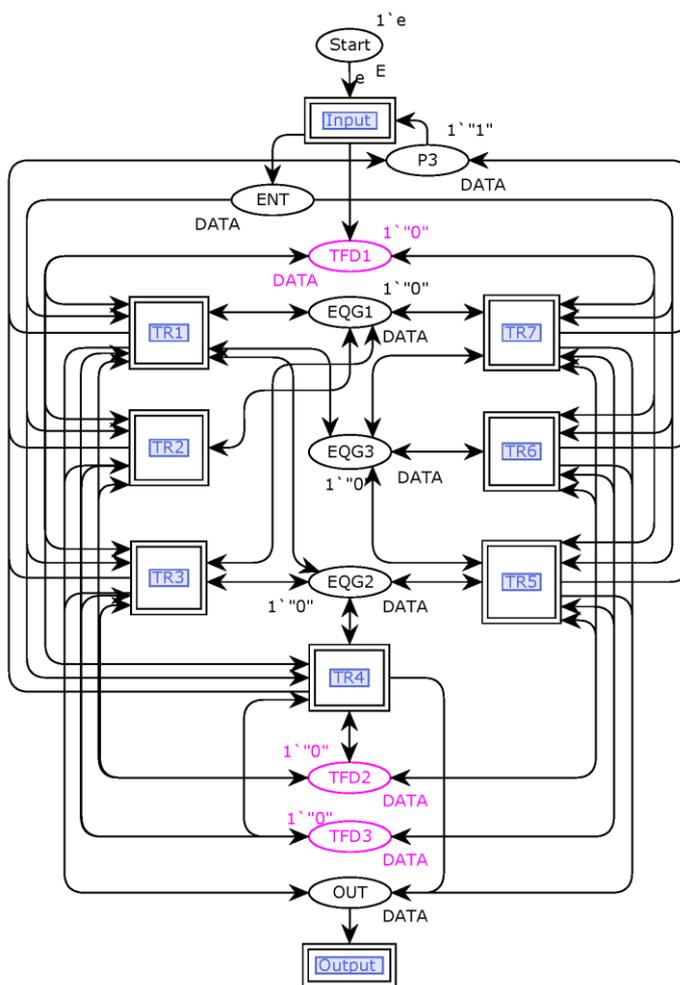


图 3.7 VISS 父页

下面我们依次介绍 Input 子页、TR_i(i=1,2,...,7)子页及 Output 子页，并以两辆车为例说明车辆根据自身检测需求进入相应子页面，执行检测任务，得到检测周转时间的全过程。

(1) Input 子页

如图 3.8 所示，在初始标识下，变迁 $OpenIn$ 使能，当该变迁触发时调用代码段读入待检车辆子队列文件，库所 $Osque$ 得到待检车辆列表。之后变迁 $q\ to\ c$ 使能，当该变迁触发时调用代码段，待检车辆列表中第一个车辆元素从列表中拆分出来，存入库所 car ，库所 $P3$ 中的初始 $token$ 被使用，与此同时，去掉第一个车辆元素后余下的待检车辆列表存入库所 $Osque$ 。然后，变迁 In 使能，当该变迁触发时库所 car 中表示第一个车辆元素的 $token$ 进入库所 ENT ，此时该 $token$ 根据其车辆检测需求通过融合库所 ENT 进入 TRi ($i=1,2,\dots,7$)中的某个相应子页进行检测， $token$ 代表的第一个车辆元素进入 $TFD1$ （第一工位场地）后，融合库所 $P3$ 再次得到 $token$ ，此时变迁 $q\ to\ c$ 再次使能，重复前面拆分待检车辆列表中首个车辆元素的流程，直到库所 $Osque$ 的车辆列表中元素数为 0 即待检车队中最后一辆车进入检测线。

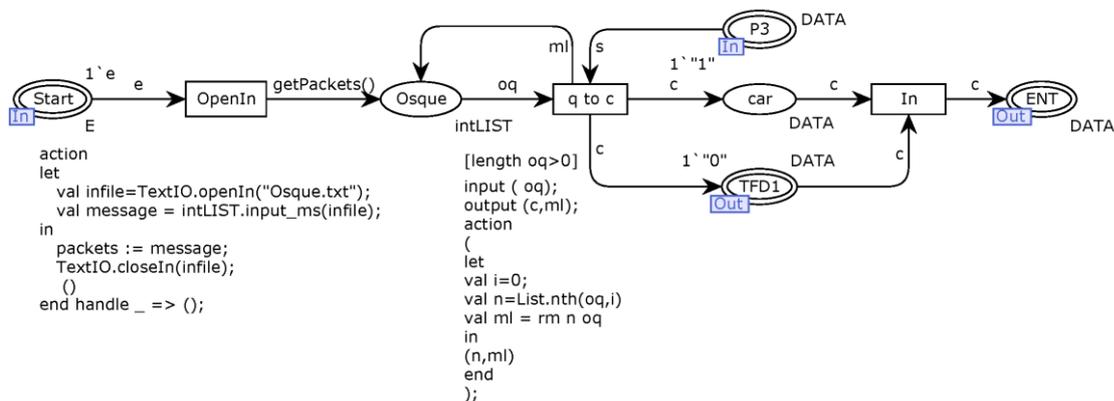


图 3.8 Input 子页

(2) $TRi(i=1,2,\dots,7)$ 子页

车辆在检测线中的具体检测执行过程如图 3.9 所示。(a)-(g)分别表示七种检测需求 TRi , $i=1,2,\dots,7$ 的对应的子页，每种检测需求对应的检测工位组合如表 3.2 所示。

VISS 模型中各个子页包括检测设备（图中黄色库所表示）、检测场所（图中绿色库所表示）及待检车辆（Input 子页给出）三部分，如图 3.9(a)所示， ENT 库所建模了车辆进入汽车检测子系统的状态， OUT 库所建模了车辆离开汽车检测子系统的状态，分别在这两个库所处建立监视点，观察这两处库所中 $token$ 出现的时间，库所 ENT 中 $token$ 出现的时间即车辆进入 VISS 的时间，库所 OUT 中 $token$ 出现的时间即车辆离开 VISS 的时间。下面说明检测需求为 $TR1$ 即全检的车辆在模型中具体的检测过程。

当检测需求为 $TR1$ 车辆进入检测子系统， $ARR11$ 变迁触发，库所 $TFD1$ 的 $token$ 被移除，表示第一工位场地被占用， ENT 库所中的 $token$ 移入 $RDT11$ 库所中，表示检测需求为 $TR1$ 即全检的车辆到达工位一， $RDT11$ 库所中的 $token$ 则表示准备在第一工位

检测的车辆，变迁 TS11 表示全检的车辆正在第一工位进行速度项目的检测，变迁 TS11 上附加的时戳“@+4”表示第一工位速度项目的检测用时为 4 分钟。变迁 TS11 触发后，EQG1 库所中的 token 被移除，表示第一工位的检测设备此时处于忙碌状态。4 分钟后，TSE11 库所得到 token，表示全检的车辆完成了第一工位的检测任务，此时，Wait11 变迁使能，Wait11 变迁触发时车辆在原地等待，附加在此变迁上的时戳“@+2”表示车辆原地等待了两分钟，检测时间加上等待时间即前文提到的工位周期。然后，RDL11 库所得到 token，此时，变迁 ARR12 使能，RDL11 库所中含有 token 表示全检的车辆准备离开第一工位。变迁 ARR12 触发后，RDL11 库所中的 token 移入 TFD1 库所，表示场地 TFD1 进入空闲状态；与此同时 TFD2 库所中的 token 被移除，RDT12 库所得到该 token，表示检测场地 TFD2 被占用，说明车辆离开工位一后进入工位二。

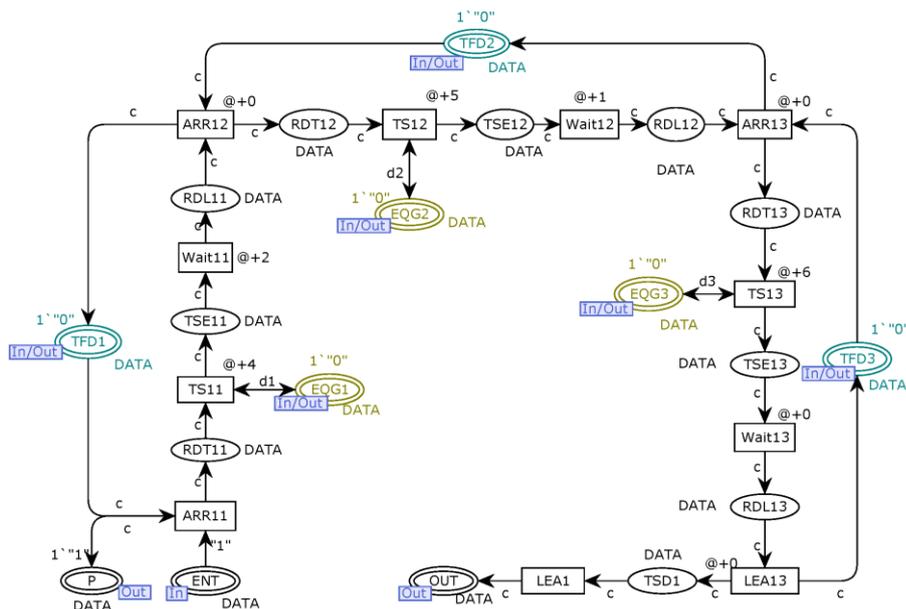


图 3.9(a) TR1 子页

检测需求为 TR1 的车辆到达第二工位后，场地 TFD2 处于被占用的状态，库所 RDT12 中含有 token 表示车辆准备检测第二工位对应的检测项目，此时，变迁 TS12 使能，变迁 TS12 表示车辆正在检测工位二对应的检测项目，而变迁 TS12 上附加的时戳“@+5”表示车辆完成工位二对应的检测项目所需检测时长为 5 分钟。变迁 TS12 触发后，EQG2 库所中的 token 被移除，表明第二工位的检测设备此时处于忙碌状态即正在执行工位二的车辆检测任务；库所 TSE12 得到 token 表示车辆完成了第二工位的检测任务，此时，变迁 Wait12 使能，变迁 Wait12 表示车辆在原地等待，附加在此变迁上的时戳“@+1”表示车辆在第二工位等待了 1 分钟，紧接着，库所 TSE12 中的 token 移入库所 RDL12，

表示车辆此时的状态是准备离开工位二,此时,变迁 ARR13 使能,变迁 ARR13 触发后,库所 RDL12 中的 token 移入库所 TFD2,表示该车辆离开工位二,场地 TFD2 被释放处于空闲状态,TFD3 库所中的 token 移入 RDT13 库所中表示车辆进入工位三,库所 RDT13 中的 token 表示准备接受第三工位的检测车辆。

检测需求为 TR1 的车辆到达第三工位, RDT13 库所中包含 token 表示车辆准备检测第三工位对应的检测项目,此时,变迁 TS13 使能,变迁 TS13 表示车辆正在检测工位三对应的项目,变迁 TS13 上附加的时戳“@+6”表示车辆检测完成工位三对应的项目需要 6 分钟的时间,变迁 TS13 触发时, EQG3 库所的 token 被移除,表示工位三的检测设备组处于忙碌状态即在正在执行工位三的车辆检测任务。6 分钟后, TSE 库所得到 token,表示车辆完成了工位三相应的检测任务,此时变迁 Wait13 使能,附加在此变迁上的时戳“@+0”表明该变迁触发时车辆没有在三工位等待,库所 TSE 中的 token 移入库所 RDL13 中,表示车辆准备离开工位三。LEA13 变迁触发时库所 RDL13 中的 token 被移除,库所 TFD3 和库所 TSD1 中移入 token,表示车辆离开工位三,工位三的场地被释放处于空闲状态,车辆此时处于检测完成状态。此时, LEA1 变迁使能, LEA1 变迁触发时,库所 TSD1 中的 token 被移除,融合库所 OUT 得到 token,则表示检测需求为 TR1 的车辆离开检测子系统, OUT 库所中含有 token 表示车辆处于驶离检测线的状态。到这里,全检的车辆用了 18 分钟的时间依次进行了 VISS 三个工位共计 6 个项目的检测,车辆的检测时间值通过 OUT 融合库所传送至 Output 子页中。

下面结合图 3.9 (b),说明检测需求为 TR2 的车辆(仅检测第一工位)在模型中具体的检测过程。

当检测需求为 TR2 的车辆进入检测子系统后, ARR21 变迁使能, ARR 表示检测需求为 TR2 即仅检测第一工位的车辆到达工位一, ARR21 变迁触发后, ENT 库所中的 token 移入 RDT21 库所中, RDT21 库所中的 token 则表示准备在第一工位检测的车辆,库所 TFD1 的 token 被移除,表示第一工位场地被占用,变迁 TS21 表示车辆正在第一工位进行速度项目的检测,变迁 TS21 上附加的时戳“@+4”表示第一工位速度项目的检测用时为 4 分钟。变迁 TS21 触发时, EQG1 库所中的 token 被移除,表明第一工位的检测设备此时处于忙碌状态即正在执行工位一的车辆检测任务;库所 TSE21 得到 token 表示该车完成了第一工位的检测任务,此时,变迁 Wait22 使能,变迁 Wait22 表示车辆在原地等待,附加在此变迁上的时戳“@+2”表示车辆在第一工位等待了 2 分钟,紧接着,库所 TSE21 中的 token 移入库所 RDL22, RDL22 库所含有 token 表示车辆准备离开工

位一，此时，ARR22 变迁使能，ARR22 变迁触发后，库所 TFD2 中的 token 被移除，表示工位二的场地 TFD2 处于被占用状态，车辆到达工位二；同时库所 TFD1 中移入 token，表示工位一的场地 TFD 被释放，而库所 Pass22 中移入 token 的过程表示车辆不在工位二检测，只是经过工位二。

库所 TFD3 中含有 token 表示工位三的检测场地此时处于空闲状态，此时，ARR23 变迁使能，当 ARR23 变迁触发时，库所 TFD3 中的 token 被移除，说明工位三的检测场地 TFD3 处于忙碌状态，表示检测需求为 TR2 的车辆到达工位三，与此同时库所 TFD2 移入 token，说明工位二的检测场地 TFD2 被释放，Pass23 库所移入 token 的过程表示该车辆经过在工位三无检测任务，仅仅经过工位三而已。然后，变迁 LEA3 使能，LEA23 变迁触发时，库所 TFD3 中移入 token，工位三的检测场地 TFD3 被释放，处于空闲状态，表示车辆离开工位三，TSD2 库所得 token 表示该车完成了检测任务，此时，变迁 LEA2 使能，LEA2 变迁触发时表示该车离开检测子系统，至此，检测需求为 TR2 的车辆用时 6 分钟完成了其检测任务，检测时间值通过 OUT 融合库所传送至 Output 子页中。

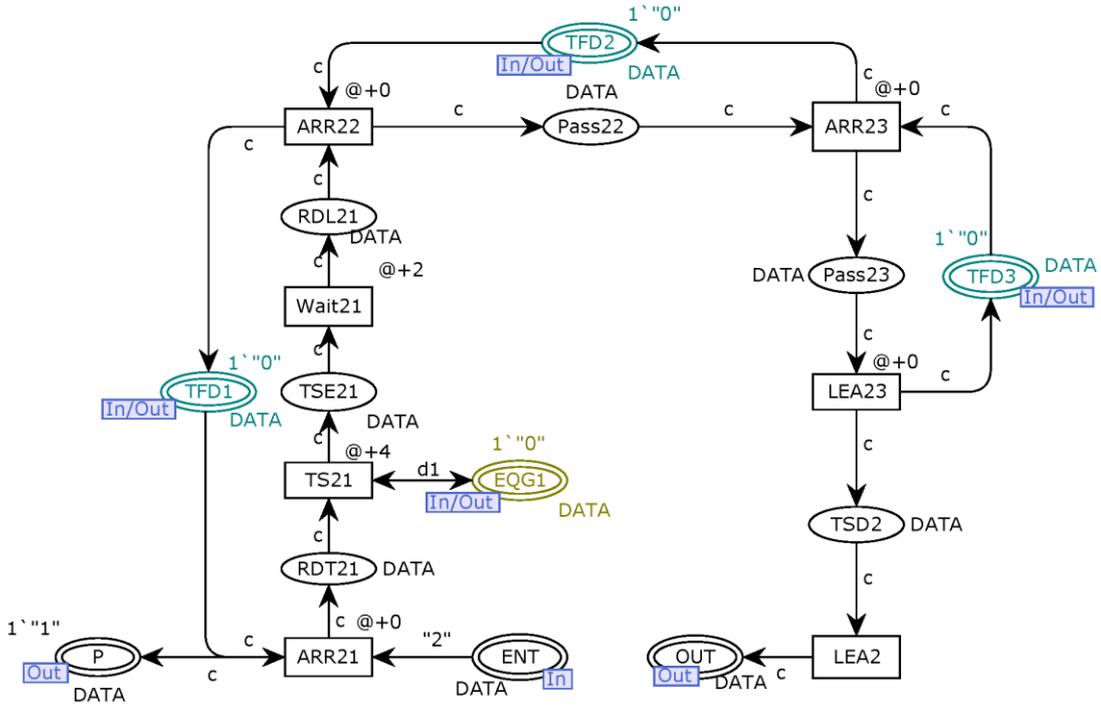


图 3.9(b) TR2 子页

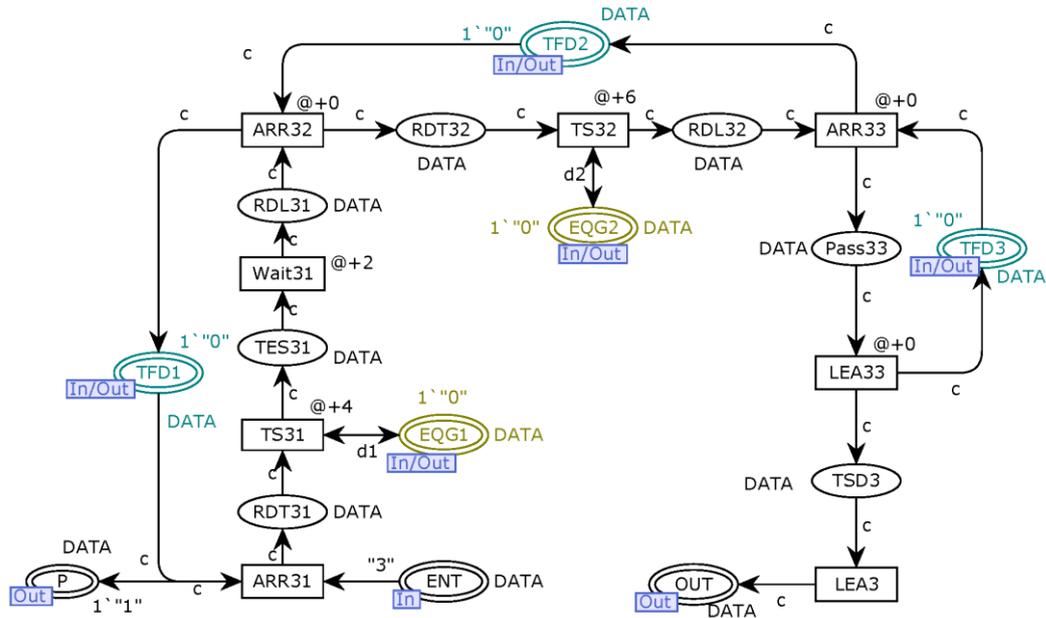


图 3.9(c) TR3 子页

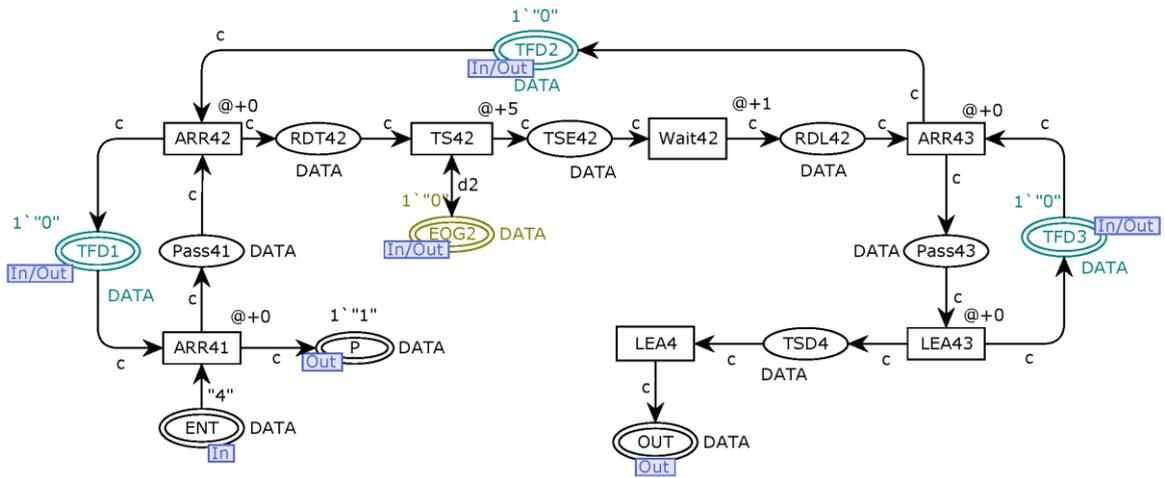


图 3.9(d) TR4 子页

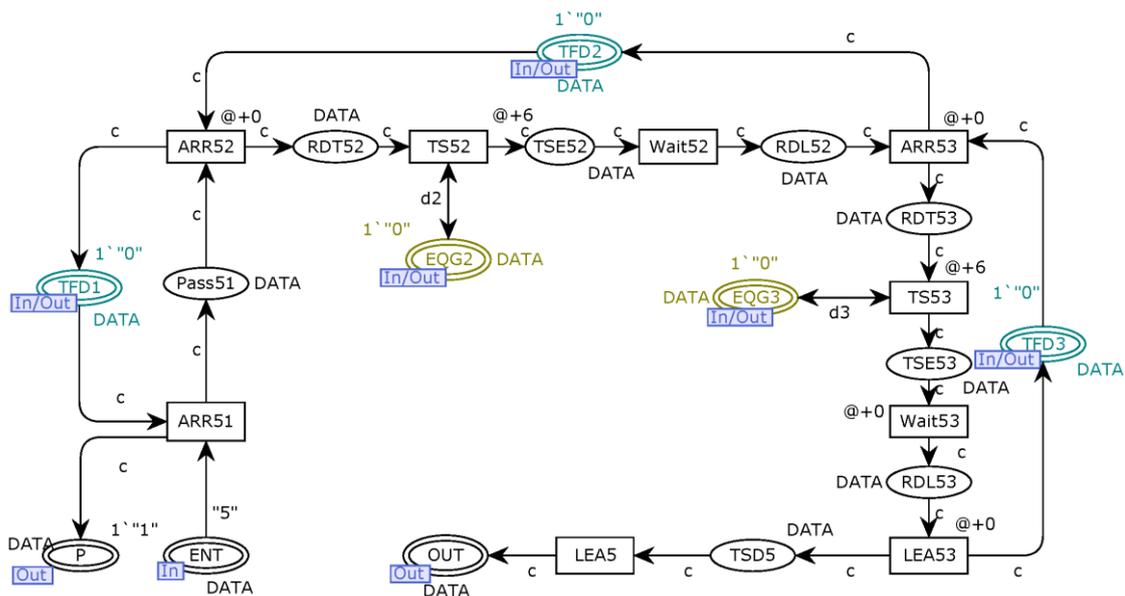


图 3.9(e) TR5 子页

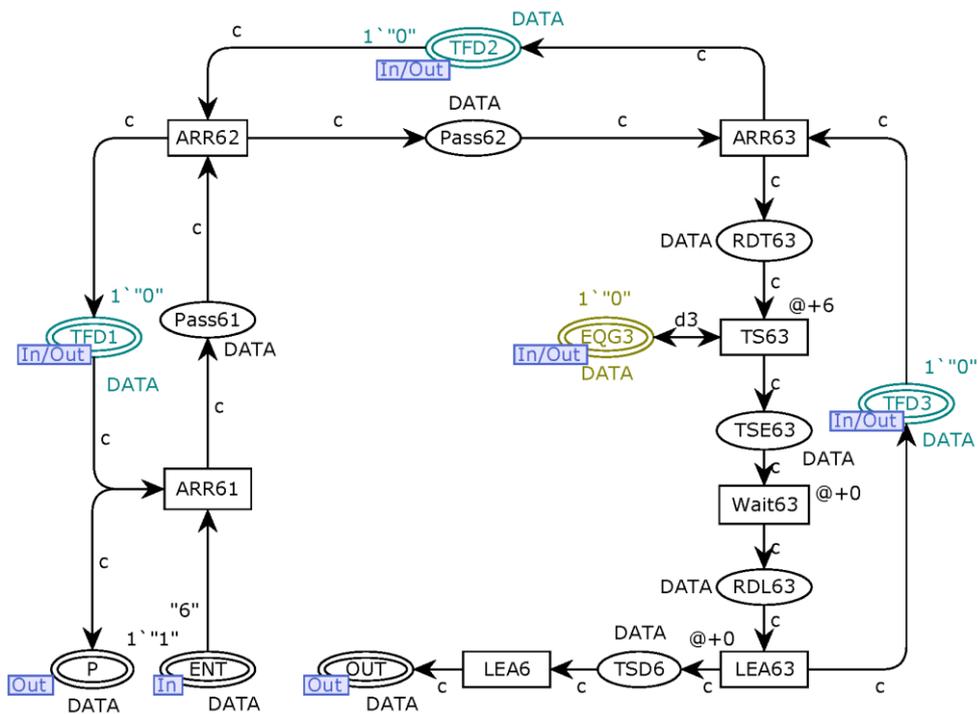


图 3.9(f) TR6 子页

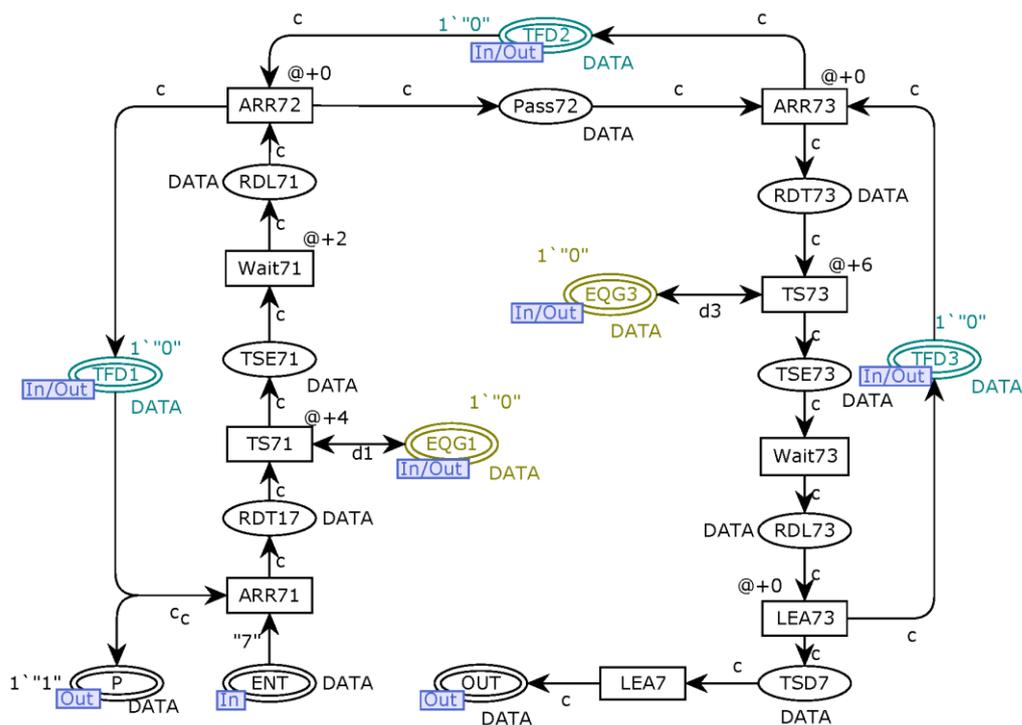


图 3.9(g) TR7 子页

(3) Output 子页

如图 3.10 所示待检车队中的第一辆车在相应 $TR_i(i=1,2,\dots,7)$ 子页完成检测的时间值通过融合库所 out 传送至 Output 子页后变迁 c to tc 使能, 当该变迁触发时, 其结果是将库所 P5 中初始 token 的时间戳 $@0$ 更新为此时库所 out 中 token 的时间戳后得到的 token 存于库所 $tc1$ 中, 然后, transfer 变迁发生后, 通过调用代码段, $tc2$ 库所中得到 token 的时间戳, 以及该时间戳的字符型和整型表示。rd_time 变迁触发后, time 库得到字符类型的时间值。此时, link 变迁触发时库所 time 中的时间值与字符 “,” 连接后存入库所 que_time 中, 变量 $e1$ 加 1, 库所 num 中的 token 值由 0 变为 1。之后, 融合库所 out 收到 $TR_i(i=1,2,\dots,7)$ 子页传送的新时间值即待检车辆列表中第二辆车的检测结果, 然后变迁序列 c to tc -> $transfer$ -> rd_time -> $link$ 依次触发, 重复上述步骤至待检车辆列表中最后一辆车完成检测, 最终, 库所 num 中保存待检车辆列表的车辆数, 所有待检车辆的检测周转时间以字符串形式存于库所 que_time 中。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/088072073051006035>