arXiv:2304.00789v1 [math.OC] 3 Apr 2023

结合组合优化的机器学习求解带时间窗的动态车辆路径问题

Léo Baty*[1]、Kai Jungel*[2]、Patrick S. Klein*[2]、Axel Parmentier[1], 和Maximilian Schiffer[2,3]

[1]CERMICS，巴黎桥路学院，马恩拉瓦莱，法国[2]管理学院，慕尼黑工业大学，慕尼黑，德国[3]慕尼黑数据科学研究所，慕尼黑工业大学，慕尼黑，德国

摘要

随着电子商务的兴起和客户要求的不断提高，物流服务提供商在日常规划中面临着新的复杂性，这主要是因为必须高效处理当日配送。现有的多阶段随机优化方法能够求解底层的动态车辆路径问题，但要么计算成本过高，无法应用于在线场景；要么（对于强化学习而言）难以在高维组合问题上取得良好表现。为缓解这些缺点，我们提出了一种新颖的机器学习流水线，其中纳入了组合优化层。我们将这一通用流水线应用于带调度波次的动态车辆路径问题；该问题最近在 NeurIPS 2022 的 EURO Meets NeurIPS车辆路径竞赛中得到推广。我们的方法在该竞赛中排名第一，在求解所提出的动态车辆路径问题时优于所有其他方法。通过这项工作，我们提供了一项全面的数值研究，进一步凸显了所提流水线的有效性与优势，超越了竞赛中取得的成果，例如展示了编码策略对于未见过的实例和情景的鲁棒性。

关键词：车辆路径规划、结构化学习、多阶段随机优化、组合优化、机器学习

*前三位作者对本研究贡献相同。

1 引言

随着过去十年电子商务的兴起，物流服务提供商面临着日益提高的客户要求，尤其是在（快速）配送时效方面。因此，当日配送这一概念，即物流服务提供商承诺在收到订单的当天完成配送，已成为B2C领域的关键要素。事实上，2021年当日配送的占比增长了18.5%（Maida2022年）。由于电子商务和B2C配送市场仍有多家主要企业参与竞争，零售商和物流服务提供商不断力求超越彼此，导致交付周期持续缩短，在部分城市，某些类型的产品甚至可在短短两小时内送达（Nicolai 2016年）。

在如此短的规划时域内完成最后一公里配送，从效率角度看仍然颇具挑战，因为物流服务提供商通常要在缩短交付周期与配置过多资源之间权衡，例如，维持规模过大的车队，以便随时能够立即响应新订单。事实上，当日配送这一概念使得提前一天规划——数十年来一直是最后一公里物流的现状——变得荒谬。物流服务提供商不再是求解组合复杂但静态的规划问题，以确定成本效益高的配送路线，而是必须在在线问题设定下动态地将订单调度给车辆，并为这些车辆规划路径。在这种设定下做出决策，既需要预判调度或延迟订单所带来的收益，又要应对相应静态规划问题的组合复杂性，并处理未来订单到达的不确定性。

实践中此类规划问题所带来的挑战，推动了科学界对动态车辆路径问题（VRP）变体的研究兴趣。解决此类问题的最先进方法，将其底层规划任务建模为多阶段随机优化问题，并通过直接求解（Pillac 等，2013年；Soeffker、Ulmer 和 Mattfeld，2022年）或强化学习求解（Nazari 等，2018年；Hildebrandt、Thomas 和 Ulmer，2023年；Basso 等，2022年）实现。然而，这两种方法都存在一个重大缺陷。基于强化学习的算法擅长做出前瞻性决策，但应用于高维组合问题时往往表现不佳。相较之下，随机优化技术通常适用于组合问题设定，但在高维问题中难以满足计算效率要求，因此不切实际，无法用于动态环境（Pflug 和 Pichler，2014年；Carpentier 等，2015年）。

在此背景下，我们提出一种新的方法论方案，以缓解上述不足。具体而言，我们构建了一条集成组合优化（CO）层的机器学习（ML）流水线，从而能够高效求解动态车辆路径问题。该流水线在设计上缓解了多阶段随机优化问题带来的挑战：其机器学习层能够通过恰当参数化底层确定性组合优化问题的一个实例来纳入不确定性，随后可在组合优化层中高效求解。我们在 NeurIPS 2022 举办的 EURO Meets NeurIPS车辆路径竞赛中使用了这条流水线（Kool、Bliek 等，2022年），其表现优于所有其他方法。

1.1 相关研究

我们的研究对两个不同的研究方向作出了贡献：从应用角度看，它与动态车辆路径问题相关；从方法论角度看，它与运筹优化增强型机器学习相关。下文将概述这两个相关研究方向。关于车辆路径问题的一般概述，请参阅 Vidal、Laporte 和 Matl 2020年 ；关于运筹优化增强型机器学习的一般概述，请参阅 Bengio、Lodi 和 Prouvost 2021年，以及 Kotary 等人 2021年。

动态车辆路径问题 动态车辆路径问题反映了现实世界过程中动态变化的特性：一些问题数据（例如需要服务的客户、他们的需求或彼此之间的行驶时间）无法预先获知，而是随着时间推移逐步显现。因此，动态车辆路径问题的应用领域十分广泛，涵盖网约车服务（Jungel 等人 2023年）、生鲜配送（Fikar 2018年）以及应急服务（Alinaghian、Aghaie 和 Sabbagh 2019年）。为使本综述保持简洁，以下将重点讨论动态调度问题中的动态车辆路径问题，并建议参阅 Pillac 等人（2013年）和 Ojeda Rios 等人（2021年），以及 Ulmer、Soeffker 和 Mattfeld（2018年）关于该领域的全面综述。求解动态车辆路径问题的方法大致可分为基于运筹优化的方法，

这类方法利用底层问题的组合结构；以及基于机器学习的方法，这类方法通过学习预见性策略来应对未来客户的不确定性。

纯组合优化方法通常将为静态车辆路径问题开发的求解方法改用于动态情形。具体而言，这些组合优化方法将（元）启发式算法嵌入滚动时域框架中，即每当有新信息进入系统时，就求解所研究问题的一个静态变体（例如，参见 Ritzinger、Puchinger 和 Hartl 2016年；Ojeda Rios 等人2021年）。其中，短视方法（例如，参见 Gendreau 等人 1999；Steever、Karwan 和 Murray2019年）仅利用当前问题状态，而前瞻方法（例如，Flatberg 等人 2007；Bent 和 Van Hentenryck 2004）会考虑未来各时期可能出现的情况，例如通过采样来实现。

机器学习方法学习能够考虑未来观测不确定性的策略。因此，它们将底层动态问题建模为马尔可夫决策过程，并通过近似动态规划方法（例如，参见 Ulmer、Soeffker 和 Mattfeld 2018年），即策略近似或价值函数近似，或通过强化学习（例如，参见 Nazari 等人 2018年；Kool、Hoof 和 Welling 2019年；Joe 和 Lau 2020年）来求解。关于强化学习在动态车辆路径问题中的应用综述，可参见 Raza、Sajid 和 Singh 2022 以及 Hildebrandt、Thomas 和 Ulmer 2023 。

如图所示，已有多种研究致力于解决动态车辆路径问题。在这些研究中，大多数方法要么通过采样未来情景来使用经典组合优化算法，要么运用机器学习来近似能够计入未来期望奖励的决策值。所有这些方法至少存在以下缺陷之一：基于经典组合优化的算法难以满足动态问题设定中的实时要求，而基于机器学习的方法往往解的质量欠佳，因为它们没有考虑问题的组合结构。迄今为止，尚未有针对动态车辆路径问题提出真正整合组合优化与机器学习的方法，在利用机器学习于动态环境中的优势的同时，也不忽视问题的组合结构。

组合优化增强型机器学习 许多现实世界中的组合调度问题都会受到不确定的未来事件影响。为了找到能够考虑这些不确定事件的组合解，可以将组合优化层集成到基于机器学习的流水线中。我们将这类流水线称为组合优化增强型机器学习流水线。在机器学习流水线中使用组合优化层的主要障碍在于其分段常数性质。具体而言，梯度几乎处处为零，因此在此类环境中无法提供有用信息，使得直接反向传播无效。当前最先进的方法通过引入正则化技术来平滑组合优化层，从而解决这一问题，使梯度计算具有实际意义，并使其能够用于基于机器学习的流水线。在监督学习设定中，使用 Fenchel-Young 损失的组合优化增强型机器学习流水线已成功采用加法正则化（Berthet 等人2020年）和乘法正则化（Dalle 等人 2022年）方法（Blondel、Martins 和 Niculae2020年）。

这类流水线的一个常见应用场景是困难组合优化问题，例如带释放日期的单机调度（Parmentier and T'Kindt2021年），经典组合优化方法通常难以求解此类问题。在这种方法中，统计模型学习对一个嵌入式、可求解的组合优化问题进行参数化；该问题与难以求解的原始问题具有相同的可行解空间，因此能够生成对原问题有效的解。这种方法还可扩展到多阶段优化问题的参数化学习，例如两阶段随机最小生成树问题（Dalle et al. 2022年）或随机车辆调度问题（Parmentier 2021年；Parmentier 2022年）。Jungel et al. 2023年 使用运筹优化增强型机器学习流程，在动态自主按需出行系统中学习调度策略。乍看之下，Jungel et al. 2023年 的研究似乎与本文相似，但两者在几个根本方面有所不同：Jungel et al. 2023年 研究的调度问题可在多项式时间内求解，因此其组合优化层不依赖启发式算法。此外，Jungel et al. 2023年的统计模型采用广义线性模型的形式，而本文采用深度学习。

尽管运筹优化增强型机器学习流水线已广泛用于解决现实问题，但迄今为止，该领域的现有研究在组合优化层中依赖精确算法。然而，当组合优化问题难以求解时（大多数动态车辆路径问题正是如此），这种方法便难以应用。此外，现有研究假设组合优化层的目标函数为线性函数，且预测的目标成本与决策变量维度相同；而在我们的动态车辆路径问题场景中，情况并非如此。

1.2 贡献

为弥补上述研究空白，我们提出一种新颖的基于机器学习的流程，并在其中融入组合优化层；随后将其应用于 EURO Meets NeurIPS车辆路径竞赛所引入的一类新颖动态车辆路径问题。这类问题对学术界和实践领域都极具研究价值。具体而言，我们的工作包含若干项贡献。从方法论角度看，我们将运筹优化增强型机器学习流水线中的组合优化层推广至非线性目标函数。值得注意的是，在此背景下，我们还将开源库 InferOpt.jl 扩展至此类非线性场景。此外，我们提出了首个利用元启发式组件求解组合优化层的运筹优化增强型机器学习流水线。由此，我们表明，尽管一般的机器学习-组合优化范式在形式上要求使用最优解来推导真实梯度，但在实践中也能很好地利用启发式解。在此背景下，我们详细说明了如何精心设计元启发式算法，以得到启发式解，即近似梯度，从而使我们流水线中的机器学习层能够收敛。我们还展示了如何在监督学习场景下训练该流水线的机器学习层，也就是基于由前瞻性策略生成的训练集进行训练。我们通过一项全面的数值研究，在与最先进方法的基准测试中展示了该方法的有效性。此外，我们的研究验证了这种学习方法能够很好地泛化至未见场景。有趣的是，我们的结果表明，与通常做法背道而驰的是，模仿前瞻性策略对于高维多阶段随机优化问题也能取得良好效果。

有关复现本文所述结果的说明和所有必要材料，请参阅我们的 Git 仓库（https://github.com/tumBAIS/euro-meets-neurips-2022）。

1.3 文章结构

本文其余部分安排如下：第2节对问题设定作出形式化定义；第3 节介绍运筹优化增强型机器学习流程。随后，我们将详细介绍该流程的各个层。具体而言，第4 节详述组合优化层采用的算法框架，第5节详述机器学习层的学习方法。第6 节介绍计算研究的设计及其结果。最后，第7 节总结全文。

2 问题设定

我们研究的问题设定聚焦于 EURO Meets NeurIPS 车辆路径竞赛中提出的一种带时间窗的动态车辆路径问题（VRPTW）变体（参见 Kool、Bliek 等人，2022年）。在这一带时间窗的动态车辆路径问题中，我们旨在寻找一组成本最低的路径，这些路径从中央仓库出发并最终返回 d，由一支车辆车队为一组请求提供服务 $\mathcal{R}$，并在有限的规划时域内完成 $[0, T_{\max}]$。我们聚焦于在线问题设定，其中请求集 $\mathcal{R}$ 起初未知，且请求会持续到达 $[0, T_{\max}]$。在此规划时域内，车队运营商在（等间隔的）各个时间步作出调度决策 $\tau \in [0, T_{\max}]$，并需要服务规划时域内到达的所有请求。因此，我们将规划时域离散化为一组 n 时段 $E = \{[\tau_0, \tau_1], [\tau_1, \tau_2], \ldots, [\tau_{n-1}, \tau_n]\}$，并将一个时段的开始时间 e 记为 τ_e。

在每个时段，车队运营商都会针对该时段对应的请求集 $\mathcal{R}^e$求解一个调度与车辆路径问题。每个请求 $r \in \mathcal{R}^e$ 都有一定的需求量 q_r，而所有车辆的车辆容量相同，均为 Q，这限制了单条路径上可服务的最大请求数。服务请求 r需要 s_r 个时间单位，且必须在该请求专属的时间窗 $[\ell_r, u_r]$内完成。从请求 i 的配送地点前往另一个请求 j 的配送地点需要 t_{ij} 个时间单位，并产生成本 c_{ij}。我们可以直接在完全连接的有向图 $\mathcal{D}^e = (\mathcal{V}^e, \mathcal{A}^e)$上对某个时段的规划问题实例进行编码，其顶点集为 $\mathcal{V}^e = \mathcal{R}^e \cup \{d\}$，其中 d是仓库，弧集为$\mathcal{A}^e$。采用这一表示法，我们的问题表述如下。

系统状态 我们将决策时刻 τ_e 的系统状态 描述为 $x^e = \mathcal{R}^e$，其中 e 是从时间 τ_e开始的时段。在这里， $\mathcal{R}^e$ 包含所有已揭示但尚未调度的请求。在我们的具体设定中，不需要其他信息来描述系统状态，因为没有车队规模限制。

请注意，我们使用冗余记号 $x^e = \mathcal{R}^e$ ，以遵循机器学习和车辆路径规划领域中常用的惯例。我们可以将 $\mathcal{R}^e$ 中包含的请求分为两个互不相交的类别：必须调度的 请求需要在 e 中调度，因为若推迟到之后的时段调度，则会导致送达时违反这些请求的时间窗；可推迟的 请求可以在 e中调度，但并非必须。

可行决策 给定系统的当前状态 x^e，车队运营商会从 $\mathcal{R}^e$中选择一个子集，由本时段离开仓库的车辆负责服务，并计算相应路径以便调度车辆。在时段 e 调度的车辆于时间 $\tau_e + \Delta\tau$ 离开仓库，其路径一经调度便无法修改，即在时段 e 调度的车辆无法服务在时段 e'公布的请求，其中 $\tau_{e'} > \tau_e$。在此背景下，状态 x^e 下的可行决策 $y^e \in \mathcal{Y}(x^e)$对应一组路径，这些路径

(i) 包含所有必须调度的 请求，

(ii) 允许 每条路径在相应时间窗内访问其中包含的所有请求，并且

(iii) 且
每条路径上的累计客户需求不超过车辆容量 Q.

我们可以用向量 $(y^e_{i,\ j})_{(i,j)\in\mathcal{A}^e}$ 来表示一个可行决策 $y^e \in \mathcal{Y}(x^e)$，其中

$$y^e_{i,j} = \begin{cases} 1 & \text{if } (i,j) \text{ is in a route of the solution} \\ 0 & \text{otherwise.} \end{cases}$$

系统演化 车队运营商决定采用$y^e \in \mathcal{Y}(x^e)$后，系统便转入下一个时段 e' 。为描述 $x^{e'}$，我们从 $\mathcal{R}^e$中移除 y^e 所包含的所有请求，并加入在 τ_e 与 $\tau_{e'}$之间进入系统的所有请求，由此推导出 $\mathcal{R}^{e'}$ 。

策略 令 $\mathcal{X}$ 表示潜在系统状态的集合。那么，(确定性)策略 $\pi: \mathcal{X} \to \mathcal{Y}$ 是一个映射，它会为任意系统状态 $x^e \in \mathcal{X}$分配一个决策 $y^e \in \mathcal{Y}(x^e)$。

目标 我们的目标是找到一种策略，使规划时域 $[0, T_{\max}]$内服务所有请求 $\mathcal{R}$ 的期望成本最小。形式化地，

$$\min_{\pi} \mathbb{E}\left[\sum_{e\in E} c(\pi(x^e))\right], \tag{1}$$

其中， $c: \mathcal{Y} \to \mathbb{R}$ 给出了路径 $y \in \mathcal{Y}$的成本。

讨论 上述定义和形式化的问题设定包含若干假设，从实践者的角度来看，这些假设可能值得商榷。尤其是，假设车队规模不受限制且完全了解请求分布，似乎并不现实。本文聚焦于用于赢得 EURO Meets NeurIPS车辆路径竞赛的方法，因此为保持一致性和可复现性，我们决定保留这些假设，不作进一步探讨。不过，我们想强调，只要有一些历史数据，本文提出的方法流水线就可以直接应用于车队规模有限、且对潜在请求分布了解不完整的问题设定。

3 机器学习流水线与组合优化层

为解释我们运筹优化增强型机器学习流水线的原理，我们回顾一下，策略 π 将系统状态 x^e 映射为 $\mathcal{Y}(x^e)$中的一个可行决策 y^e 。状态 x^e 是一组请求 $\mathcal{R}^e$，而解 y^e 是一组可行

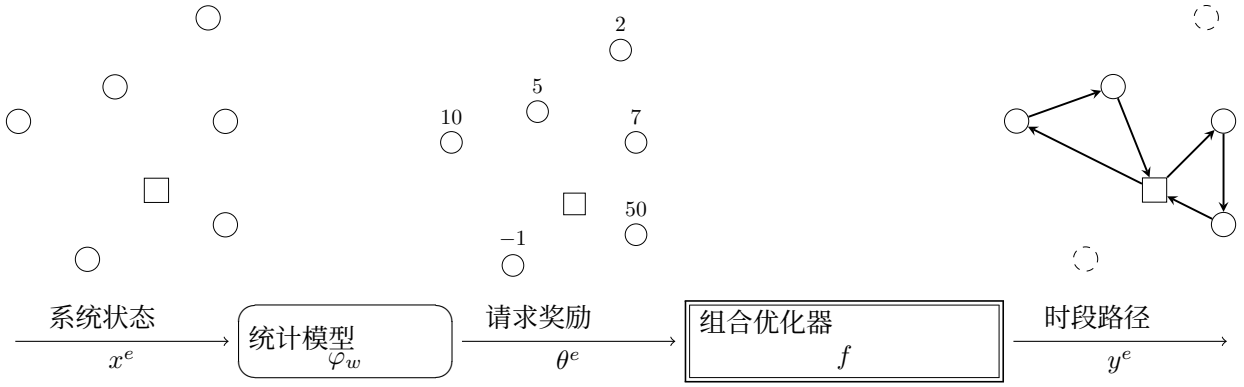


图 1：我们的CO增强型机器学习流程。

决策，这些决策编码了覆盖 $\mathcal{R}^e$子集的路径。因此，可行决策集合 $\mathcal{Y}(x^e)$与奖赏收集型带时间窗车辆路径问题的可行解集合一致。奖赏收集型带时间窗车辆路径问题是（静态）带时间窗车辆路径问题的一种变体：它不要求服务所有请求，但若服务请求 j，则会获得奖励 θ_j^e。本节将说明，在动态问题设定中，任何由最优策略得出的决策 y^e，都对应于针对精心选取的奖励向量 $\theta^e = (\theta_j^e)_{j\in\mathcal{R}^e}$ 的奖赏收集型带时间窗车辆路径问题的一个最优解。然而，确定奖励 θ_j^e并非易事，因此我们借助机器学习来完成这一任务。具体而言，我们引入一族策略 $(\pi_w)_w$，由图 1 所示的运筹优化增强型机器学习流水线进行编码：在机器学习层中，统计模型 φ_w根据给定的系统状态 x^e预测 θ^e。由此得到一个奖赏收集型带时间窗车辆路径问题实例 (x^e, θ^e)，我们在组合优化层中使用专门的算法 f求解该实例。该算法的输出 y^e 随后对应于我们的调度与路径规划决策。

接下来，我们首先正式介绍奖赏收集型带时间窗车辆路径问题，并证明时段 e 中的每个最优决策都可以表示为一个经过特殊构造的奖赏收集型带时间窗车辆路径问题实例的解。然后，我们将详细说明如何设计流水线，利用这一观察来寻找调度与路径规划决策。

奖赏收集型带时间窗车辆路径问题 回顾一下，$\mathcal{D}^e = (\mathcal{V}^e, \mathcal{A}^e)$ 是一个完全有向图，其顶点集为 $\mathcal{V}^e = \mathcal{R}^e \cup \{d\}$，由时段请求 $\mathcal{R}^e$ 和仓库 d 组成；奖赏收集型带时间窗车辆路径问题 $y^e \in \mathcal{Y}(x^e)$ 的一个可行解可以编码为向量 $(y_{i,j}^e)_{(i,j)\in\mathcal{A}^e}$，其中

$$y_{i,j}^e = \begin{cases} 1 & \text{if } (i,j) \text{ is in a route of the solution} \\ 0 & \text{otherwise.} \end{cases}$$

我们进一步考虑每条弧上的成本 $(c_{i,j})_{(i,j)\in\mathcal{A}^e}$，以及每个顶点上的奖赏 $(\theta_j^e)_{j\in\mathcal{R}^e}$。于是，我们可以将奖赏收集型带时间窗车辆路径问题的目标表述如下：

$$\max_{y\in\mathcal{Y}(x^e)} \underbrace{\sum_{\substack{(i,j)\in\mathcal{A}^e \\ j\neq d}} \theta_j^e y_{i,j}}_{\text{total profit}} - \underbrace{\sum_{(i,j)\in\mathcal{A}^e} c_{i,j} y_{i,j}}_{\text{total routing cost}}. \tag{2}$$

命题 3.1. 对于任意 x^e，存在一个 $\theta \in \mathbb{R}^{|\mathcal{R}^e|}$，使得 (2) 的任何最优解都是相对于 (1)的最优决策。

证明。 由于时间范围是有限的，且每一步的可行决策集合也是有限的，因此存在一个¯最优决策 $y^\star$ 对于 x^e。设 $\mathcal{R}^e$ 为 $\mathcal{R}^e$ 中在 $y^\star$被派发的请求子集。那么，任何¯解 y 只要其路径成本更低或相等，并且恰好覆盖 $\mathcal{R}^e$，也同样是最优的。这是因为

贝尔曼方程表明，路径不会影响状态的演化。我们可以通过在 $\mathcal{R}^e$ 上求解一个奖赏收集型带时间窗的车辆路径问题来构造 y，其中请求奖励为

$$\bar{\theta}_j = \begin{cases} M & \text{if } j \in \bar{\mathcal{R}}^e \\ -M & \text{otherwise,} \end{cases} \tag{3}$$

其中 $M = \left(|\mathcal{R}^e| \cdot \max_{(i,j)\in\mathcal{A}^e} c_{i,\ j} \right)$ 是一个较大的常数。相应的奖赏收集型带时间窗的车辆路径问题解 $\bar{y}$ 显然恰好覆盖 $\mathcal{R}^e$，且其路径成本不超过 $y^\star$。 □ □

组合优化层 我们将奖赏收集型带时间窗的车辆路径问题作为一个层 嵌入到运筹优化增强型机器学习流程中。因此，该组合优化层必须支持前向传播和反向传播，以确保与机器学习层兼容。

前向传播只需使用元启发式算法 f 求解由 4节中详述的$(x^e,\ \theta^e)$ 定义的奖赏收集型带时间窗的车辆路径问题实例。反向传播则通过该层反向传播学习算法所用损失的梯度。第 5节介绍该损失及其梯度。为便于陈述，我们将式（2）改写为

$$f\colon \theta^e \mapsto \underset{y\in\mathcal{Y}(x^e)}{\operatorname{argmax}}\ \theta^{e\top} g(y) + h(y) \quad \text{where} \quad g(y) = \left(\sum_{i\in\mathcal{V}^e} y_{i,j} \right)_{j\in\mathcal{R}^e} \text{and } h(y) = \sum_{(i,j)\in\mathcal{A}^e} c_{i,j} y_{i,j}. \tag{4}$$

机器学习层 求出命题 θ^e 中的最优奖赏3.1 并非易事。因此，我们使用统计模型 φ_w 根据系统状态 x^e预测奖赏向量 $\theta^e = \varphi_w(x^e) \in \mathbb{R}^{|\mathcal{R}^e|}$。从机器学习的角度看，唯一需要处理的技术问题是输入和输出的维度不固定。事实上，请求数量可能因状态而异，也可能因实例而异。在我们的计算研究中（参见第 φ_w 6节），我们对常用统计模型的不同选择进行了基准测试。

讨论 在 EURO Meets NeurIPS车辆路径竞赛中提出的其他基于机器学习的解决方案（例如 Doorn 等人 2022年）通常分两步进行。首先，他们使用二元分类器，分别决定调度哪些请求以及推迟哪些请求。然后，在第二步中，他们为这些已调度的请求构建路径，实质上将请求调度与路径构建解耦。这种方法的一大难点是学习一个能够隐式平衡当前路径成本和未来路径成本的统计模型，以找到最优的调度决策。我们通过在组合优化层（4）中同时做出调度与路径规划决策，绕过这一难点；这使我们能够基于路径规划决策而非调度决策来训练统计模型。

4 组合优化算法

为了在我们的组合优化层中求解奖赏收集型带时间窗车辆路径问题，我们提出了一种基于混合遗传搜索（HGS）的元启发式算法。我们的算法扩展了 Kool、Juninck 等人 2022年提出的 HGS算法实现；该实现改编自 Vidal 2022年 针对带时间窗的车辆路径问题提出的原始 HGS。

针对我们的问题设定，Kool、Juninck 等人2022年 的工作需要进行多项调整，以支持可选请求，例如调整局部搜索、初始化和交叉操作。我们还引入了新的变异机制，实现了奖赏收集型带时间窗车辆路径问题专用的邻域，并允许对种群进行热启动。接下来，我们将概述 HGS算法的核心概念，然后详细介绍针对奖赏收集型带时间窗车辆路径问题所做的修改。

带时间窗车辆路径问题的混合遗传搜索 HGS 算法是一种进化算法，它维护一个由解构成的种群，并将其划分为两个互不相交的子群体，分别包含可行解和不可行解。该算法使种群随时间不断进化，通过组合从种群中随机二元锦标赛选出的优质后代 解来生成子代父代 解，选取范围截至

种群中的解。随后，算法通过局部搜索过程改进生成的子代解。请注意，该算法采用基于惩罚的方法来探索解空间中的不可行区域。局部搜索过程会产生一个局部最优解，然后将其加入种群。如果种群规模超过某个阈值，可能会触发幸存者选择 过程。该过程会从种群中淘汰一部分解。幸存者选择和父代选择都基于解的适应度 ，这一指标反映了解的质量（即目标值）及其对种群多样性的贡献。这能确保种群具有足够的多样性，使遗传算法中的多样化与强化保持平衡。为将这一算法结构用于我们的问题设定，即奖赏收集型带时间窗车辆路径问题，我们进行了以下调整和扩展，形成了图2所示的算法。

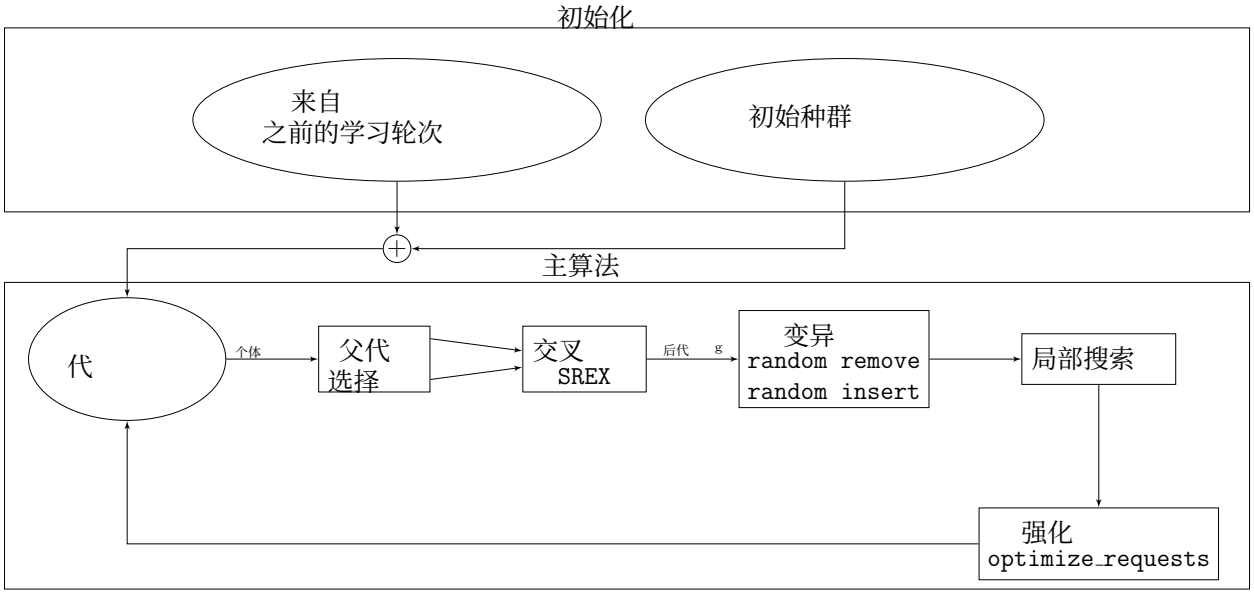


图2：我们元启发式算法的总体结构。圆形节点表示种群，方形节点表示算法组件。

解表示我们在解表示中隐式地表示哪些请求需要服务、哪些请求需要忽略。具体而言，我们采用与Vidal2022年相同的巨型巡回表示，但允许不完整的巨型巡回。在这种表示中，被认为无利可图的请求不会出现在巨型巡回中，因此传统局部搜索算子不会考虑这些请求。

在交叉过程中考虑可选请求Kool、Juninck等人2022年提出的混合遗传搜索通过两种交叉算子生成子代解：顺序交叉（OX）（Oliver、Smith和Holland2013年）和选择性路径交换（SREX）（Nagata和Kobayashi2010年）。作为第一项修改，我们移除了OX 算子，因为基准测试实验表明，该算子对我们的问题变体的算法性能没有显著影响。第二项修改是将SREX 算子调整为适用于奖赏收集型带时间窗车辆路径问题。具体而言，我们保留第一个父代所服务的请求集合，并在生成的每个后代中重新插入当前未服务的请求，这与常规SREX的做法相同，即按顺序重新插入。

额外的多样化与强化机制 我们进一步引入了两种新的变异算子，根据已服务请求的集合对解进行多样化和强化。第一种算子（随机移除/插入）根据掷硬币的结果，移除当前已服务请求中的 $\nu \cdot |R^{\sigma}|$ ，

或插入当前未服务请求中的 $\eta \cdot |C \setminus R^{\sigma}|$。此变异以 ρ 的概率在子代生成后立即进行。

第二个算子（优化_请求_集）用于优化给定解所服务的请求集。具体而言，该算子首先从解中移除所有会导致绕行成本高于相应请求利润的请求，然后重新插入当前解中尚未包含的所有有利可图的请求。该算子会使用从区间$[\zeta^{-}, \zeta^{+}]$中均匀抽取的随机因子扰动插入和移除成本。与第一个算子不同，我们会等到局部搜索收敛后再评估第二个算子，以避免因解的质量不佳而移除过多请求。我们以概率 μ运行该算子。为了加强对有前景解附近区域的搜索，我们还会对任何优于当前最优解的解运行该算子（不进行扰动）。

局部搜索 表 1 展示了我们算法中使用的局部搜索算子。我们考虑传统算子和奖赏收集型带时间窗车辆路径问题专用算子。具体而言，传统算子是指为带时间窗的车辆路径问题定义的算子（参见 Kool、Juninck 等人 2022年；Vidal2022年）。传统算子通过在解中交换弧来探索邻域，即改变当前解中一个或多个请求的位置。因此，这些算子会保留给定解中获得服务的请求集。另一方面，奖赏收集型带时间窗车辆路径问题专用算子只处理请求集，即从给定解中移除或插入请求。我们会在评估传统带时间窗车辆路径问题算子之后再评估这些算子，以避免过早移除有利可图的请求，例如因路径规划决策不佳而导致的移除。与 Vidal 2022 年的做法相同，我们进一步区分小邻域算子和大邻域算子：小邻域算子只考虑涉及地理位置接近且在时间窗方面相容的请求的移动；而大邻域算子则不受此类限制。

Type	Name	描述
传统（小型）	重定位	从其路径中移除单个请求，并将其重新插入到不同的位置在解中
	成对重定位	从其路径中移除一对连续请求，并将它们重新插入到解中的不同位置
	反向重定位 pair	从其路径中移除一对连续请求，并按相反顺序重新插入到解中的不同位置
	swap	交换解中两个请求的位置
	请求对交换	交换解中两个连续请求对的位置
	请求对与单个	将解中一个连续请求对与一个单个请求的位置互换解
	2-opt	反转一个路径片段
	2-opt*	将两条路径各拆分为两个片段，并将第一条路径中的一个片段与第二条路径中的一个片段交换
传统（大型）	relocate*	从其路径中移除一个请求，并将其插入到任何空间重叠路径中的最佳位置
	swap*	从存在空间重叠的路径 r_a 和 r_b 中移除请求 a 和 b，并插入 a 并分别将 b 和 r_b 插入 r_b 和 r_a中的最佳位置
PC-VRPTW	服务请求	将当前未服务的请求插入解中
	移除请求	从解中移除一个请求

表 1：局部搜索算子

初始化 我们采用一种预处理技术，以应对学习过程中遇到的极端权重。具体而言，我们根据所需绕行的最大值和最小值，分别维护一组确定有利可图和确定无利可图的请求。我们依据以下观察来确定无利可图的请求：如果对于某个请求 $r \in \mathcal{R}^{e}$，$\min_{j \in \mathcal{R}^{e} \cup \{d\}} c_{jr} + \min_{j \in \mathcal{R}^{e} \cup \{d\}} c_{rj} - \theta_r \geq \max_{i,j \in (\mathcal{R}^{e} \cup \{d\})^2} c_{i,j}$成立，那么对于任何包含 r 的路径 σ，都存在一条成本更低的路径 σ'。为此，设 $u, v \in \mathcal{R}^{e} \cup \{d\}$，使得 (u, r)、$(r, v) \in \sigma$。那么 $c_{u,r} + c_{r,v} - \theta_r \geq \max_{i,j \in (\mathcal{R}^{e} \cup \{d\})^2} c_{i,j} \geq c_{u,v}$，因此从 σ 中移除 r 就能得到一条成本更低的路径。类似地，如果 $c_{d,r} + c_{r,d} < \theta_r$ 成立，我们就强制纳入请求 $r \in \mathcal{R}^{e}$。此外，在评估阶段，此过程还允许通过相应设置 θ_r 来考虑必须派送的客户。

随着学习过程逐渐收敛，客户价格的变化幅度会减小。我们在算法中利用这一特性，并将之前学习轮次的解作为初始种群的种子；初始种群按照 Kool、Juninck 等人2022年的方法生成。具体而言，我们通过一种新的构造启发式算法生成有前景的解：先对解应用优化_客户算子，以适应请求奖励的变化，再通过局部搜索过程优化路径规划决策。请注意，这仅在训练流水线时可行。评估流水线时，我们只使用按照 Kool、Juninck 等人2022年的方法生成的初始解。

讨论 聚焦于我们的算法设计决策，我们注意到，针对学习方法和训练环境中的独特挑战，我们对算法进行了量身定制。具体而言，要成功计算出能够确保训练过程中收敛的近似梯度，需要低方差的解。在这里，算法需要能够很好地泛化到训练期间观察到的不同奖励分布。此外，为了使训练能在合理的时间内完成，算法还必须能够在严格的时间限制内提供此类解。

我们通过一种设计应对了这些挑战：通过积极的多样化（例如，随机插入与删除请求）和强化（优化_请求_集合）算子、显式的请求插入与删除邻域以及热启动，引导种群走向搜索空间中有希望的区域。与隐式方法（例如 Vidal、Maculan 等人于 2016年提出的方法）相比，所得算法的行为更加贪婪，但能在训练期间施加的严格时间限制内可靠收敛。我们还通过一个预处理步骤应对训练期间观察到的极端奖励。

5 学习方法

我们的学习问题的目标是找到参数 w，使统计模型 φ_w 能够预测奖励向量 θ，从而在组合优化层中产生“良好”的决策。为实现这一目标，我们训练运筹优化增强型机器学习流水线来模仿一种良好的策略。为此，我们采用监督学习设定，并据此构建数据集 $\mathcal{D} = \{(x^1, \bar{y}^1).....(x^n, \bar{y}^n)\}$，其中包含状态实例 x^i 及被模仿策略所采取的决策 $\bar{y}^i$。在这一监督学习设定中，我们定义损失 $\mathcal{L}(\theta, \bar{y})$，用于量化预测 $f(\theta)$而非 $\bar{y}$时产生的误差。随后，我们将学习问题表述为寻找使经验风险最小化的参数 $\hat{w}_{\mathcal{D}}$，即使训练数据集上的平均损失最小化的参数，

$$\hat{w}_{\mathcal{D}} = \operatorname*{argmin}_{w} \sum_{i=1}^{n} \mathcal{L}(\varphi_w(x^i), \bar{y}^i). \tag{5}$$

本节其余部分将介绍被模仿的策略、训练集、损失以及用于求解学习问题的算法（5）。

5.1 训练集中的前瞻决策

动态车辆路径问题（带时间窗）之所以困难，是因为做出决策时，未来请求尚不可知。如果我们知道未来，也就是说，如果在规划期开始时就知道所有请求，那么只需在规划期开始时求解相应的静态带时间窗车辆路径问题，然后在每个时段做出相应的调度决策即可。这个过程会为我们提供最优的前瞻策略。然而，这种前瞻策略当然无法在实践中使用，因为它依赖于无法获得的信息。尽管如此，我们可以在所有信息都已揭示之后，也就是利用历史数据，事后还原前瞻策略所做出的决策。因此，我们可以在学习问题（5）中学习模仿一种基于历史数据得到的前瞻策略。

具体而言，我们按如下方式重建预见性策略的决策。我们已知某个历史带时间窗的车辆路径问题实例中的所有请求。然后，我们为每个请求 i 关联一个<style id='1'>释放时间，该时间等于请求在动态环境中被揭示的时间τ_i，也就是请求i 最早可以被服务的时间点。随后，我们在遵守释放

时间和时间窗的前提下，寻找以最低成本服务所有这些请求的路径。我们使用 Kool、Juninck 等人提出的改进版混合遗传搜索 2022年 来求解这一静态带时间窗车辆路径问题的变体。由此得到一组路径 P，并按如下方式从中重建每个时段所采取的决策。对于路径 $p \in P$，令 $\tau_p = \text{最大值}_{r \in p}\, \tau_r$ 表示该路径所服务的任一请求的最晚释放时间 p。预见性策略在第一个时段派出路径 p，其中 e 满足 $\tau_e \geq \tau_p$；也就是说，这是所有已服务请求均已被揭示的第一个时段。

5.2 损失函数

我们回顾一下，我们的学习问题的目标是找到统计模型 w 的参数值 φ_w，使得对于任意状态 x，组合优化层都能为奖赏收集型带时间窗的车辆路径问题实例 $y = f(\theta)$ 预测出一个良好的决策$(x,\ \theta = \varphi_w(x))$。更准确地说，对于数据集中的每个状态-决策对$(x,\ \bar{y})$，我们希望目标决策 $\bar{y}$ 尽可能接近奖赏收集型带时间窗的车辆路径问题的最优解（2）。因此，自然可以将 $\bar{y}$ 作为 $f(\theta)$ 的解时的非最优性作为损失函数：

$$\mathcal{L}(\theta, \bar{y}) = \max_{y \in \mathcal{Y}(x)} \{\theta^\top g(y) + h(y)\} - (\theta^\top g(\bar{y}) + h(\bar{y})). \tag{6}$$

显然，一般而言，$\mathcal{L}(\theta, \bar{y}) \geq 0$；只有当 y 是（4）的最优解时，$\mathcal{L}(\theta, \bar{y}) = 0$。遗憾的是，多面体 $\mathcal{P}(\bar{y}) = \{\theta \in \mathbb{R}^{|x|} \mid \theta^\top g(\bar{y}) + h(\bar{y}) \geq \theta^\top g(y) + h(y)\ \ \forall y \in \mathcal{Y}(x)\}$ 通常高度退化。例如，如果 $h = 0$，那么对于所有 $\mathcal{Y}$中的 y，$\theta = 0$ 都属于 $\mathcal{P}(y)$。这意味着 $\theta = 0$ 将成为我们学习问题的一个最优解，这会带来问题，因为这样的 θ 会使我们的组合优化层能够返回 $\mathcal{Y}(x)$中的任意一个解。通过考虑扰动损失，可以消除这种退化

$$\mathcal{L}_\varepsilon(\theta, \bar{y}) = \mathbb{E}\Big[\max_{y \in \mathcal{Y}(x)} \{(\theta + \varepsilon Z)^\top g(y) + h(y)\}\Big] - (\theta^\top g(\bar{y}) + h(\bar{y})). \tag{7}$$

其中，$\varepsilon > 0$，且 $Z \sim \mathcal{N}(0, \mathbb{I}^{|x|})$ 为高斯扰动。文献中仅在 $g(y) = y$ 且 $h(y) = 0$ 的情况下研究过扰动损失 $\mathcal{L}_\varepsilon$（参见 Berthet 等，2020）。由于 h 非零，损失的几何结构发生了变化。值得注意的是，当 $h = 0$ 时，ε 的大小并不重要（参见 Parmentier 和 T'Kindt 2021），但当 h 非零时，其大小就很重要：ε 越大，h 对预测的影响就越小。命题 5.1 总结了在我们所考虑的更一般情形下这些损失的几何结构。

命题 5.1。设 $x \in \mathcal{X}$, $\bar{y} \in \mathcal{Y}(x)$。令 $\mathcal{C}(\bar{y}) = \{\theta \in \mathbb{R}^{|x|} \colon \theta^\top g(\bar{y}) \geq \theta^\top g(y)\ \ \forall y \in \mathcal{Y}(x)\}$ 为与 $g(\bar{y})$ 相关联的法锥。

1. $\theta \mapsto \mathcal{L}(\theta, \bar{y})$ 是分段线性且凸的，其次梯度为

$$\underbrace{g\Big(\operatorname*{argmax}_{y \in \mathcal{Y}(x)} \theta^\top g(y) + h(y)\Big)}_{= g(f(\theta))} - g(\bar{y}) \in \partial_\theta \mathcal{L}(\theta, \bar{y}).$$

2. $\theta \mapsto \mathcal{L}_\varepsilon(\theta, \bar{y})$ 是 $\mathcal{C}^\infty$ 且为凸函数，其梯度为

$$\nabla_\theta \mathcal{L}_\varepsilon(\theta, \bar{y}) = \mathbb{E}\left[g\left(\operatorname*{argmax}_{y \in \mathcal{Y}(x)} (\theta + \varepsilon Z)^\top g(y) + h(y)\right)\right] - g(\bar{y}) = \mathbb{E}[g(f(\theta + \varepsilon Z))] - g(\bar{y}).$$

3. $\mathcal{L}_\varepsilon(\theta, \bar{y}) \geq \mathcal{L}(\theta, \bar{y})$。

4. $\mathcal{C}(\bar{y})$ 是 $\mathcal{P}(\bar{y})$的衰退锥。

5. 设 $\theta \in \mathbb{R}^{|x|}$。若 η 属于 $\mathcal{C}(\bar{y})\backslash\{0\}$，则 $\lambda \mapsto \mathcal{L}(\theta + \lambda\eta, \bar{y})$ 单调不增。若进一步有 $\mathcal{C}(\bar{y}) \neq \mathbb{R}^{|x|}$，则 $\lambda \mapsto \mathcal{L}_\varepsilon(\theta + \lambda\eta, \bar{y})$ 单调递减。

6. 设 $\theta \in \mathbb{R}^{|x|}$。若 η 位于 $\mathcal{C}(\bar{y})$ 的内部 $\mathring{\mathcal{C}}(\bar{y})$，则 $\lim_{\lambda \to \infty} \mathcal{L}(\theta + \lambda\eta, \bar{y}) = \lim_{\lambda \to \infty} \mathcal{L}_\varepsilon(\theta + \lambda\eta, \bar{y}) = 0$。

˜证明。 1 $\theta \mapsto \mathcal{L}\ \theta\ \bar{y}$ θ $\theta\ \theta\ \in$。(,) 是线性
$\mathbb{R}^{|x|}\times\mathbb{R}^{|x|}$ y^* $_{y\in\mathcal{Y}(x)}\theta^\top g\ y\ +h\ y$ $\tilde{y}$ $_{y\in\mathcal{Y}(x)}\tilde{\theta}^\top g\ y\ +h\ y$
且凸的，因为它是若干关于 的线性映射的最大值。设 (,) 。设 属于 argmax ( ) ( )，而 属于 argmax ( ) ( )。根据 $\tilde{y}$ 的定义，我们有 $\tilde{\theta}^\top g(\tilde{y}) + h(\tilde{y}) \geq \tilde{\theta}^\top g(y^*) + h(y^*)$，从而得到 $\mathcal{L}(\tilde{\theta}, \bar{y}) \geq \mathcal{L}(\theta^*, \bar{y}) + (\tilde{\theta} - \theta^*)^\top (g(y^*) - g(\bar{y}))$以及 $\partial_\theta \mathcal{L}(\theta, \bar{y})$ 中的次梯度。

2. 由于 $\mathcal{L}_\varepsilon(\theta,\ y) = \mathbb{E}[\mathcal{L}(\theta + \varepsilon Z,\ y)]$,，我们得到 $\theta \mapsto \mathcal{L}_\varepsilon(\theta,\ y)$ 是 C^∞，因为它是 $\theta \mapsto \mathcal{L}(\theta,\ \bar{y})$ 与高斯密度（即 C^∞）的卷积，并且作为凸函数的期望，它是凸的。其梯度可由 $\mathcal{L}(\theta,\ y)$ 的次梯度得到。

3. 设 $\theta \in \mathbb{R}^{|x|}$。对于所有 $\tilde{y}$ 属于 $\mathcal{Y}(x)$ 的情形，我们有

$$\text{for all } z \in \mathbb{R}^{|x|}, \quad \max_y (\theta + \varepsilon z)^\top g(y) + h(y) \geq (\theta + \varepsilon Z)^\top g(\tilde{y}) + h(\tilde{y})$$
$$\text{hence,} \quad \mathbb{E}[\max_y (\theta + \varepsilon Z)^\top g(y) + h(y)] \geq \theta^\top g(\tilde{y}) + h(\tilde{y}) \quad (\text{since } \mathbb{E}[Z] = 0)$$
$$\text{and,} \quad \mathbb{E}[\max_y (\theta + \varepsilon Z)^\top g(y) + h(y)] - (\theta^\top g(\bar{y}) + h(\bar{y})) \geq \theta^\top g(\tilde{y}) + h(\tilde{y}) - (\theta^\top g(\bar{y}) + h(\bar{y})).$$

最后： $\mathbb{E}[\max_y (\theta + \varepsilon Z)^\top g(y) + h(y)] - (\theta^\top g(\bar{y}) + h(\bar{y})) \geq \max_y \{\theta^\top g(y) + h(y)\} - (\theta^\top g(\bar{y}) + h(\bar{y}))$，即 $\mathcal{L}_\varepsilon(\theta,\ \bar{y}) \geq \mathcal{L}(\theta,\ \bar{y})$ 。

4. 设 $\theta \in \mathcal{P}(\bar{y})$。我们有 $\forall y,\ (\theta + \lambda\eta)^\top g(\bar{y}) + h(\bar{y}) \geq (\theta + \lambda\eta)^\top g(y) + h(y) \Leftrightarrow \forall y,\ \theta^T g(\bar{y}) + h(\bar{y}) + \lambda\eta^\top (g(\bar{y}) - g(y)) \geq \theta^\top g(y) + h(y)$。如果 η 属于 $\mathcal{C}(\bar{y})$，那么 $\eta^\top (g(\bar{y}) - g(y)) \geq 0$，且对于任意 $\lambda > 0$，$\theta + \lambda\eta$ 都属于 $\mathcal{P}(\bar{y})$。如果 η 不属于 $\mathcal{C}(\bar{y})$，那么 $\eta^\top (g(\bar{y}) - g(y)) < 0$，并且存在一个 $\lambda > 0$，使得 $\theta + \lambda\eta$ 不属于 $\mathcal{P}(\bar{y})$。

5. 设 $\eta \in \mathcal{C}(\bar{y})$。记 $\hat{y} = \text{argmax}_y (\theta + \lambda\eta)^\top g(y) + h(y)$。我们有

$$\text{for all } \lambda, \quad \mathcal{L}(\theta + \lambda\eta, \bar{y}) = (\theta + \lambda\eta)^\top g(\hat{y}) + h(\hat{y}) - [(\theta + \lambda\eta)^\top g(\bar{y}) + h(\bar{y})]$$

对 λ 求导，得到： $\eta^\top (g(\hat{y}) - g(\bar{y}))$。根据 η 的定义，该值为负。因此， $\lambda \mapsto \mathcal{L}(\theta + \lambda\eta,\ \bar{y})$ 具有非增性质。

类似地，对所有 Z，记 $\hat{y}_\varepsilon(Z) = \text{argmax}_y (\theta + \lambda\eta + \varepsilon Z)^\top g(y) + h(y)$，则可得到 $\lambda \mapsto \mathcal{L}_\varepsilon(\theta + \lambda\eta)$ 的导数：$\eta^\top (\mathbb{E}[g(\hat{y}_\varepsilon(Z))] - g(\bar{y}))$。如果 $\mathcal{C}(\bar{y}) \neq \mathbb{R}^{|x|}$，那么 $\mathbb{P}(\theta + \lambda\eta + \varepsilon Z \notin \mathcal{C}(\bar{y})) > 0$。因此，$\mathbb{E}[g(\hat{y}_\varepsilon(Z)] \neq g(\bar{y})$，$\eta^\top (\mathbb{E}[g(\hat{y}_\varepsilon(Z)] - g(\bar{y})) < 0$，所以 $\lambda \mapsto \mathcal{L}_\varepsilon(\theta + \lambda\eta)$ 是递减的。

6. 设 $\eta \in \mathring{\mathcal{C}}(\bar{y})$。根据 $\mathring{\mathcal{C}}(\bar{y})$ 的定义，我们有 $\eta^\top (\bar{y} - y) > 0$ 对所有 $y \neq \bar{y}$都成立。因此，存在 $M > 0$，使得对于所有 $\lambda \geq M$和 $y \neq \bar{y}$，都有 $(\theta + \lambda\eta)^\top \bar{y} + \geq (\theta + \lambda\eta)^\top y$。也就是说，对于所有 $\lambda \geq M$，都有 $\theta + \lambda\eta \in \mathcal{P}(\bar{y})$，即 $\mathcal{L}(\theta + \lambda\eta, \bar{y}) = 0$。因此 $\lim_{\lambda\to\infty} \mathcal{L}(\theta + \lambda\eta, \bar{y}) = 0$。

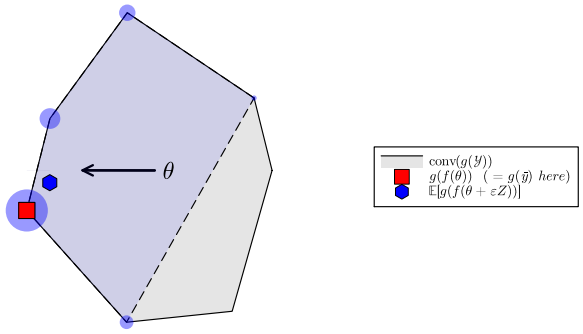
若 $\mathcal{C}(\bar{y}) \neq \mathbb{R}^{|x|}$，由第 5 点以及前述极限可知， $\lambda \mapsto \mathcal{L}(\theta + \varepsilon z + \lambda\eta, \bar{y})$ 对任意 z都单调收敛至零。因此，由单调收敛定理可得 $\lim_{\lambda\to\infty} \mathcal{L}_\varepsilon(\theta + \lambda\eta, \bar{y}) = 0$ 。

如果 $\mathcal{C}(\bar{y}) = \mathbb{R}^{|x|}$，结果显而易见。

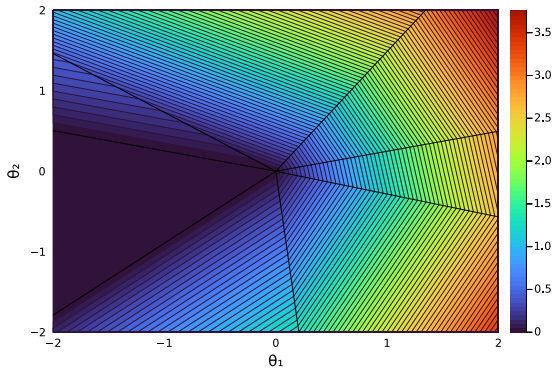
□

扰动损失 $\mathcal{L}_\varepsilon$ 具有适用于学习问题的性质。事实上，它是凸且光滑的；尽管其梯度中的期望难以计算，但通过采样 Z 即可获得随机梯度。因此，可以使用随机梯度下降来求解损失为 $\mathcal{L}_\varepsilon$ 的学习问题 (5)。此外，$\theta \mapsto \mathcal{L}_\varepsilon(\theta, \bar{y})$ 仅当 θ 远处于 $\mathcal{C}(\bar{y})$ 的内部时才趋于 0，这意味着对于 θ， $\bar{y}$ 是 (4) 的最优解这一点毫无歧义，并且消除了 $\theta \mapsto \mathcal{L}(\theta, \bar{y})$ 的退化问题。图 3 展示了二维情形。图 3a 表示与凸包络 $\text{conv}(g(\mathcal{Y}))) = \text{conv}(\{g(y)\ \forall y \in \mathcal{Y}\})$ 对应的多面体。由于组合优化层的目标函数在 $g(y)$ 中是线性的，算法 f 必然会输出该多面体的一个顶点（红色方块）。深蓝色六边形是

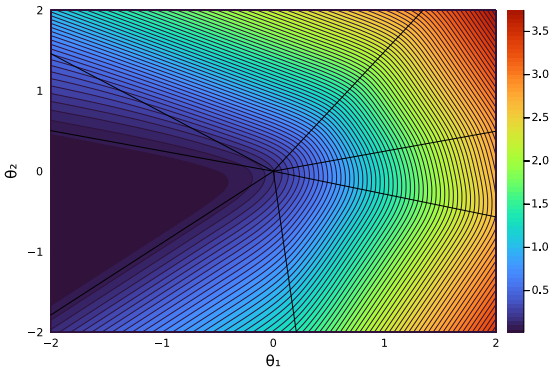
扰动组合优化层的输出 $\mathbb{E}[g(f(\theta+\varepsilon Z))]$，可视为多面体顶点上某个分布（蓝色圆圈）的期望。在图 3b 中，当 $h=0$ 且没有扰动时，可以看到，对于法锥中任意的 θ，损失 $\mathcal{L}(\theta,\bar{y})$ 均为 0；该法锥对应于 $\bar{y}$（其中 $g(\bar{y})$ 是红色方块），并且原点属于任一顶点 y 的法锥。在图 3c 中，加入扰动后可以看到，只有当 $\mathcal{L}(\theta,\bar{y})$ 位于锥内部足够深处时，它才趋于 0，这意味着组合优化层的输出没有歧义。最后，在图 3d 和 3e 中可以看到，非零的h 会改变较小 θ 值时的几何形状，但不会改变较大 θ 值时的几何形状。



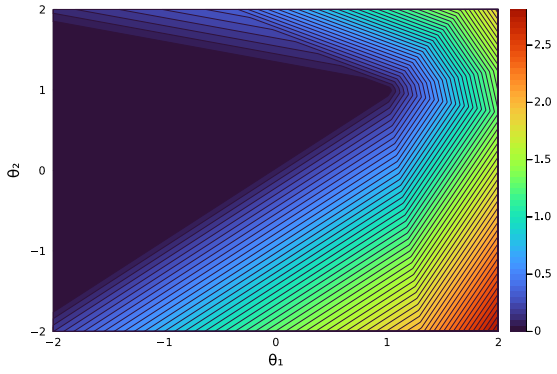
(a) 多面体 conv($g(\mathcal{Y})$)



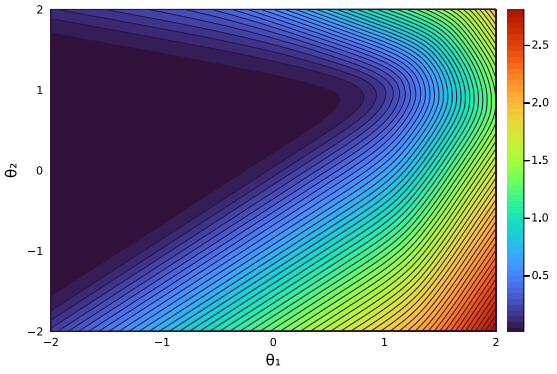
(b) $\theta \mapsto \mathcal{L}(\theta,\ \bar{y})$，适用于 $h=0$



(c) $\theta \mapsto \mathcal{L}_\varepsilon(\theta,\ \bar{y})$，适用于 $h=0$



(d) $\theta \mapsto \mathcal{L}(\theta,\bar{y})$，适用于 $h\neq 0$



(e) $\theta \mapsto \mathcal{L}_\varepsilon(\theta,\ \bar{y})$ 用于 $h\neq 0$

图 3：二维示例 θ 及其多面体 3a。损失的等高线图 $\mathcal{L}$ 以及其扰动版本 $\mathcal{L}$ 对应两个不同的 h。3b 和 3c中的粗线表示与该多面体相关联的法扇。我们可以看到，经过扰动的正则化将损失“推入”法锥 $\mathcal{C}(\bar{y})$。

非最优组合优化层 请注意，我们的组合优化层 f 是一种（元）启发式算法，因此无法保证获得最优解，这与本节所回顾理论的假设并不一致。然而，我们观察到，这在实践中并不是问题，因为 f 输出的解足够接近最优解。

6 计算研究

我们的计算研究有两个目标。首先，我们通过基准测试，将运筹优化增强型机器学习流水线的性能与若干最先进方法进行比较并加以验证。其次，我们开展广泛的数值实验，评估不同训练设置对运筹优化增强型机器学习流水线性能的影响。具体而言，我们研究 i）特征集，ii）训练集中实例的规模，iii）训练集大小，iv）模仿目标策略，以及 v）所用统计模型类型的影响。

为此，我们首先在第 6.1 节详细介绍计算研究的设计，并在第 6.2 节介绍不同的基准测试。随后，我们在第 6.3 节展示基准研究的结果，并在第 6.4 节讨论不同训练设置下实验的结果。

6.1 实验设计

我们的计算研究按照 EURO Meets NeurIPS车辆路径竞赛（Kool、Bliek 等人，2022年）中提出的问题设定进行设计。我们考虑一组源自美国杂货配送服务真实数据的带时间窗的车辆路径问题实例。我们将这些实例称为静态实例。静态实例包含一组请求，每个请求都有特定的位置、服务时间、需求和时间窗。因此，每个实例都隐含地定义了请求位置、服务时间、需求和时间窗的分布。我们从一个静态实例生成动态问题实例的方法如下：首先，将静态实例的规划时域划分为若干个一小时时段。然后，对于每个时段 e，我们随机采样 m 个请求：分别从静态实例的位置、需求、服务时间和时间窗分布中独立抽取，构造这些请求。在本节余下部分，我们将 m 称为样本量。对于相对于当前时段起始时间和所选时间窗不可行的请求，我们将其剔除。剩余请求构成在时段 e中揭示的请求集合。这种采样方法有两项影响：第一，随着时段推进，由于采样到可行时间窗的概率降低，每个时段到达的预期请求数量也会减少。第二，结束时间越接近规划时域末端的时间窗越可能出现，因为这类时间窗可行的概率更高。按照竞赛的问题设定，我们仅最小化行驶时间，因此车辆数量、服务时间和等待时间不属于目标。此外，我们不限制车辆数量，并假设已知静态实例的请求分布。

最后，我们指出，通过改变从请求分布中采样所用的随机种子，可以从一个静态实例生成多个动态实例。下文将此随机种子称为实例随机种子。

6.2 基准策略

我们将运筹优化增强型机器学习流水线的性能与一组基准策略进行比较。这些策略采用两阶段方法：首先确定要派发的请求集合，然后在第二步中使用 HGS算法求解已派发请求的带时间窗的车辆路径问题（参见 Kool、Juninck 等人，2022年；Vidal，2022年）。除非另有说明，每种基准策略每个时段的决策时间限制为 90 秒。下文将简要介绍本计算研究中考虑的基准策略。

Gree贪心策略 会在所有请求进入系统后立即将其派发 .

Lazy 策略 贪心策略会在所有请求进入系统后立即派遣。 .

随机策略 随机策略以 50% 的概率派遣可延期请求。此外，它会派遣所有必须派遣请求。

滚动时域策略 滚动时域策略会为剩余时段采样一个情景，并以与前瞻性策略类似的方式应用混合遗传搜索，假设采样的情景就是真实情景。由此得到一组路径，并据此决定调度哪些请求。具体而言，当且仅当当前时期的某个请求与当前时期的一个必须调度请求共用一条路径时，才调度该请求。我们将时间限制设为 600 秒，以确保混合遗传搜索收敛，因为该策略需要在一个完整情景上求解带时间窗的车辆路径问题。请注意，这将 90 秒的时间限制延长至

蒙特卡洛策略 蒙特卡洛策略会为剩余时段采样九个情景，并按照滚动时域策略的方式分别求解。它依据多数决策来调度请求，即当且仅当滚动时域策略在至少 50% 的采样情景中调度该请求时，才调度该请求。因此，我们相应地提高时间限制，总计允许 5400 秒。

ML-CO策略 ML-CO策略编码于第 3 节介绍的运筹优化增强型机器学习流程中。除非另有说明，我们使用一组包含 15 个训练实例的数据集训练 ML-CO策略，每个时段的样本量为 50 个请求，这些样本通过第 5.1 节详述的前瞻性策略生成。我们将用于生成前瞻性解的混合遗传搜索运行时间限制为 3600 秒。我们的机器学习层采用包含四个隐藏层的前馈神经网络，每个隐藏层包含十个神经元。对于 ML-CO策略，我们将每个时段寻找决策的时间限制设为 90 秒。表 2 表明，该时间限制不会显著影响 ML-CO策略的性能。

	评估期间的决策时间限制 [秒]					
	30	60	90	120	180	240
与前瞻性基准的相对距离：	5.66%	5.18%	5.15%	5.01%	4.89%	4.87%

表2：评估期间为 PC-HGS 设置不同时间限制时，我们的 ML-CO策略的性能。

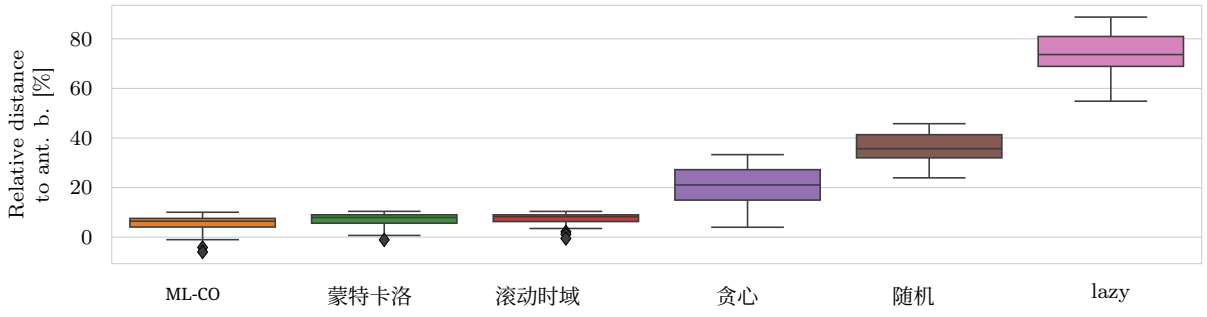
我们注意到，贪心、惰性和随机策略是 EURO Meets NeurIPS车辆路径竞赛中使用的基准启发式算法，感兴趣的读者可参阅 Kool、Bliek 等人2022年以了解更多详情。

我们还将训练期间使用的前瞻性策略（参见第5.1节）纳入作为基准。为得出前瞻性基线，我们使用 Kool、Juninck 等人提出的 HGS算法2022年，并将时间限制设为 3600 秒，以求解离线带时间窗车辆路径问题。请注意，HGS算法是一种启发式算法，这意味着有可能找到优于前瞻性基线的解。

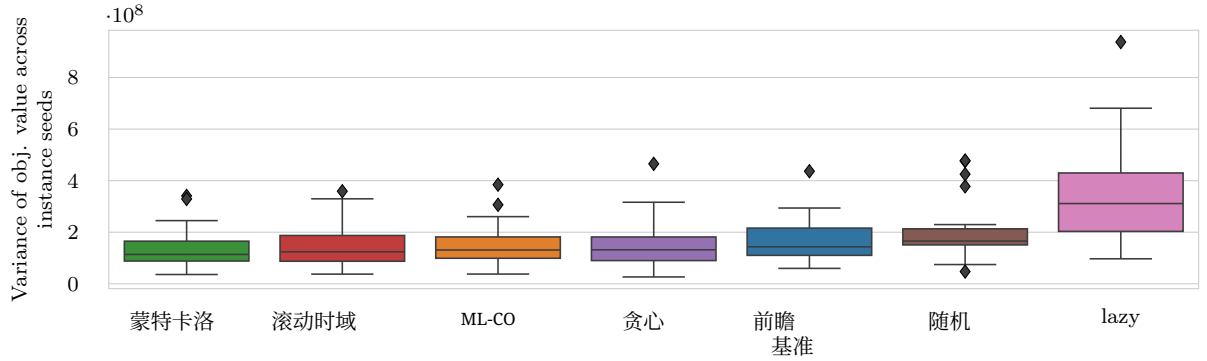
我们在 25 个动态测试实例上评估基准策略的性能，并报告每个实例在 20 个不同实例种子下的平均结果。为创建动态测试实例，我们设定每个时段的样本量为 100 个请求。

6.3 性能分析

图 4a 对比了所介绍基准的性能。图中显示了各策略的目标值相对于前瞻性基线（ant. b.）的差距。总体而言，我们可以看到，考虑未来请求不确定出现情况信息的基准策略，即滚动时域、蒙特卡洛、ML-CO 和前瞻性基线，其表现优于仅依据当前时段做出调度决策的基准策略，即惰性、随机和贪心。具体而言，这些策略的表现平均分别比前瞻性基线差 74.05%、36.29% 和 20.99%。滚动时域策略的表现比前瞻性基线差 7.12%。我们发现，仅基于单一情景考虑调度决策对未来的影响，就已显著改善了相较于贪心策略的性能。鉴于样本空间的规模，抽取的情景往往无法很好地代表之后各时段实际观察到的请求，这一结果



（a）相对于前瞻基准解的性能。



（b）目标值相对于实例种子的方差。

图 4：基准策略的性能。

令人惊讶。不过，抽取情景与真实情景之间的差距，可能会被将未来请求与实际请求结合起来规划低成本路径所带来的益处所抵消。蒙特卡洛策略优于滚动时域策略，与前瞻性基线的差距仅为 6.97%。这表明，通过增加采样情景的数量来更好地近似未来不确定性，能够带来更好的调度决策，从而提升性能。然而，与贪心策略到滚动时域策略所带来的提升相比，相较于滚动时域基准的改进幅度较小。我们的*ML-CO* 策略优于所有其他在线策略，其表现仅比前瞻性基线差 5.15%。这令人惊讶，因为多阶段随机问题领域的文献普遍认为情况恰恰相反，即模仿前瞻性策略并不能很好地泛化。

结果 1. 我们的 *ML-CO* 策略在所有在线基准策略中表现最佳，目标值比蒙特卡洛策略高 *1.57%*。这表明，通过模仿前瞻性策略来学习策略，在此问题设定下能够取得良好表现。

图 4b 展示了所考虑的基准策略在多个实例种子下的目标值方差。比较图 4a 和 4b 可以看出一个明显趋势：低方差策略优于高方差策略。这一趋势源于找到的解具有更强的鲁棒性，能够在不确定的未来观测下良好泛化。需要注意的是，尽管表现较差，蒙特卡洛策略的方差却低于我们的 ML-CO 策略。这与蒙特卡洛策略所依据的采样方法有关：该方法得出的鲁棒解在所有采样场景下都表现良好，因此过于侧重方差最小化。另一方面，我们的 ML-CO 策略则在鲁棒性与解的质量之间取得了平衡。

结果 2。蒙特卡洛策略和我们的 *ML-CO*策略能够很好地泛化到不确定的未来观测。

*ML-CO*策略中的学习组件平衡了鲁棒性与解的质量之间的权衡，从而带来了更优的性能。

6.4 扩展分析

本分析评估不同训练设置对我们 ML-CO策略性能的影响。具体而言，我们考察了 i) 特征集，ii) 每个训练实例的规模，iii) 训练集规模，iv) 模仿目标策略，以及 v) 所用统计模型的类型所产生的影响。我们按照第 6.2 节所述评估各模型的性能，并报告与前瞻性基准的相对差距。

	特征集		
	完整（基准）	模型感知	无模型
与前瞻性基准的相对距离：	5.15%	13.83%	6.78%

表3：使用不同特征集的ML-CO策略性能。

不同特征集 表 3 比较了我们的 ML-CO 策略在三种特征集（即 完整、模型感知和无模型）上的性能。表 4 详细列出了各特征集包含的特征。具体而言，无模型特征集包含可根据当前状态计算的特征 x^e，而 模型感知特征集仅包含包含静态实例分布信息的特征。我们还纳入了完整特征集，它结合了 模型感知和无模型特征集中的特征。挑战赛提交模型使用的是完整特征集，即我们的基准。结果表明，我们的 ML-CO 策略的性能并非仅依赖于模型感知信息，而这些信息在现实场景中并不可用。具体而言，考虑由静态实例推导出的模型感知特征后，我们的 ML-CO 策略与前瞻性基准之间的差距平均仅缩小了 1.63 个百分点。

结果3. 我们的 *ML-CO* 策略即使不考虑由静态实例推导出的模型感知特征，也能表现良好。

无模型		模型感知	
位置的 x 坐标	x_r	到所有位置的行程时间分布分位数:	
位置的 y 坐标	y_r	1%分位数	$Pr[X < x] \leq 0.01, X \sim t_{r,:}$
需求	q_r	5%分位数	$Pr[X < x] \leq 0.05, X \sim t_{r,:}$
服务时间	s_r	10%分位数	$Pr[X < x] \leq 0.1, X \sim t_{r,:}$
时间窗开始	l_r	50%分位数	$Pr[X < x] \leq 0.5, X \sim t_{r,:}$
时间窗结束	u_r	所有时间窗松弛时间分布的分位数:	
从配送中心到请求的时间	$t_{d,r}$	0%分位数	$Pr[X < x] \leq 0, X \sim u_{:} - (l_r + s_r + t_{r,:})$
仓库到请求点的相对时间	$t_{d,r}/(u_r - s_r)$	1%分位数	$Pr[X < x] \leq 0.01, X \sim u_{:} - (l_r + s_r + t_{r,:})$
时间窗开始 / 剩余时间	$l_r/(T_{max} - \tau_e)$	5%分位数	$Pr[X < x] \leq 0.05, X \sim u_{:} - (l_r + s_r + t_{r,:})$
时间窗结束 / 剩余时间	$u_r/(T_{max} - \tau_e)$	10%分位数	$Pr[X < x] \leq 0.1, X \sim u_{:} - (l_r + s_r + t_{r,:})$
必须派遣	$\mathbb{1}_{\tau_e + \Delta\tau + t_{d,r} > u_r}$	50%分位数	$Pr[X < x] \leq 0.5, X \sim u_{:} - (l_r + s_r + t_{r,:})$

表4：不同的特征集。

不同训练实例样本量的影响 表 5 展示了在不同样本量的训练实例上训练 ML-CO 策略时，其相对于前瞻性基线的性能。请注意，无论训练时使用的样本量是多少，我们评估训练所得模型时，使用的都是由样本量为 100 的设置生成的实例。我们的结果表明，不同样本量之间存在权衡。具体而言，我们的流水线在样本量为 50 的实例上训练时性能最佳，相对于前瞻性基线的平均差距为 5.15% 。增大

	训练实例的样本量				
	10	25	50	75	100
与前瞻性基准的相对距离：	8.71%	6.31%	5.15%	7.41%	13.08%

Ta表 5：在不同规模的训练实例上训练时，ML-CO策略的性能 .

或减小样本量都会降低性能。我们将此归因于偏差-方差权衡：较小的实例有助于训练出在训练集上高度准确的统计模型，但该模型无法泛化到评估时使用的较大实例。另一方面，使用较大的实例进行训练则无法提取足够的结构信息，因而难以得到准确的统计模型。

结果 4. 训练 *ML-CO* 策略时，平衡学习准确性和模型泛化至关重要。

	训练实例数量							
	1	2	5	10	15	20	25	30
与前瞻性基准的相对距离：	10.19%	7.77%	7.26%	5.89%	5.15%	5.23%	4.50%	4.94%

表6： 在不同数量的训练实例上训练时，ML-CO策略的性能 s.

不同数量的训练实例表6 展示了在不同训练集大小下训练ML-CO策略时，该策略相对于前瞻性基准的性能。我们的结果表明，ML-CO策略的性能会随着训练集大小的增加而提升。然而，边际提升会随着训练集大小的增加而递减。具体而言，当训练集大小达到10个实例时，性能趋于饱和，这表明即使训练集很小，我们的ML-CO策略也能学会良好的泛化。这说明我们的方法只需少量实例，就能提取问题设定中包含的大部分结构信息；这与经典监督学习的发现形成鲜明对比，后者通常需要大得多的训练集。

Re结果5。*ML-CO*策略只需少量训练实例，就能学会通用策略。

		模仿的前瞻性策略						
	最佳随机种子	3600 秒	900 秒	300 秒	240 秒	180 秒	120 秒	60 秒
与前瞻性基准的相对距离：	6.18%	5.15%	4.79%	7.61%	6.01%	5.25%	5.33%	5.81%
平均目标值 [×10000]:	20.49	20.56	20.70	20.78	20.80	20.87	20.97	21.21

表 7：采用不同目标策略时 ML-CO策略的性能。

改变目标策略 为研究 ML-CO策略所模仿的解的质量对其的影响，我们调整用于推导底层前瞻性解的 HGS 的时间限制。其基本思路是：在较短时间限制下得到的解，质量应低于在较长时间限制下得到的解。由于 HGS算法会进行随机决策，我们还加入了一个训练集，该训练集由每次运行时间限制为 3600 秒的 10 次运行中找到的最优解构成。表 7 展示了在不同目标解（其解的质量各不相同）上训练 ML-CO策略时，该策略相对于前瞻性基准的性能。出乎意料的是，训练集的解的质量与 ML-CO策略的性能之间并无明显趋势。因此，我们推测，提高前瞻性目标解的质量并不会提升经训练的 ML-CO策略的性能，也就是说，前瞻性策略或许并非最适合模仿的策略。

结果6。 前瞻性策略可能并非最适合模仿的策略，也可能存在能带来更好性能的目标解。

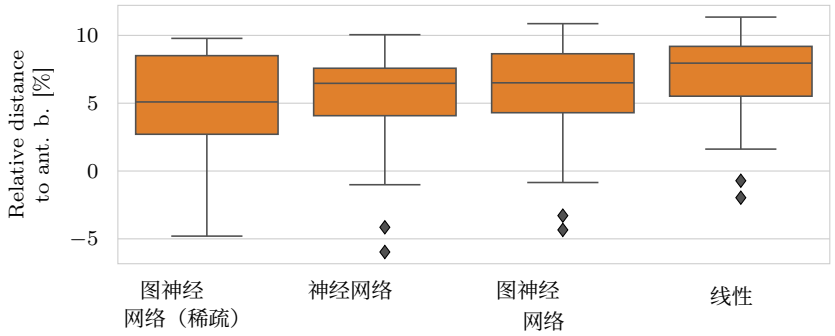


图5：使用不同统计模型的 ML-CO策略性能。

不同的统计模型 图5比较了使用不同统计模型时我们 ML-CO 策略的性能 φ_w。所有统计模型都依赖于特征映射 ϕ，该映射将状态 x^e 及该状态的请求 i 映射为特征向量 $\phi(i, x^e)$，位于 $\mathbb{R}^{|\phi|}$中。我们的线性模型 $\varphi_w(x^e) = (w^\top \phi(i, x))_i$ 通过对每个维度 i分别应用相同的线性模型 $\phi \mapsto w^\top \phi$，从而能够处理大小可变的输入和输出。类似地，我们的神经网络 $\varphi_w(x^e) = \left(g_w(\phi(i, x))\right)_i$ 对每个维度 i分别应用辅助神经网络g_w。最后，我们提出了两种图神经网络。两种图神经网络都将请求视为节点，并将请求之间的连接视为边。我们纳入了一种常规图神经网络和一种稀疏化图神经网络；后者仅包含根据请求时间窗和行驶时间判断为可行的边。两种图神经网络都为每个节点 i接收一个输入向量 $\phi(i, x)$，使得 $\varphi_w(x^e) = \left(h_w(\phi(i, x)_{i \in I}\right)$。

线性模型的表现最差，与前瞻性基准的平均差距为 6.85%；而稀疏化图神经网络的表现最佳，与前瞻性基准的差距为 5.04%。将线性模型的性能与神经网络的性能进行比较，可以看出特征生成器的重要性。事实上，仅使用简单的神经网络，就已将与前瞻性基准的差距降至 5.15%。正如预期，使用能够计算结构特征的图神经网络进一步提升了 ML-CO策略的性能。然而，与神经网络的性能相比，这一提升幅度相当小。这表明，大部分结构信息已经包含在结构化学习方法中。

结果7。 生成特征的统计模型能够产生性能最佳的 *ML-CO*策略。

7 结论

我们提出了一种用于动态车辆路径问题的新型运筹优化增强型机器学习流水线，该问题由 EURO Meets NeurIPS车辆路径竞赛引入。具体而言，我们的工作包含若干方法论贡献，并进行了广泛的计算研究。从方法论角度来看，我们将基于机器学习的流程扩展到了预测的目标成本维度与决策变量维度不匹配的目标函数。这些目标函数适用于其他问题设定，因此我们已将其纳入开源库 InferOpt.jl。此外，我们提出了首个利用元启发式组件求解组合优化层的流水线，并展示了如何谨慎设计元启发式算法，以找到可用于计算近似梯度的启发式解。我们还展示了如何以监督学习的方式训练该流水线的机器学习层，即基于由前瞻策略生成的训练集进行训练。我们开展了全面的数值研究，并表明，通过运筹优化增强型机器学习流水线编码的策略，平均行驶时间比贪心策略少 13.18%，甚至比运行时间更长的蒙特卡洛策略少 1.57%。值得注意的是，我们的结果表明，与常见做法所暗示的相反，对于高维多阶段随机优化问题，模仿前瞻策略可能效果良好，即使该前瞻策略未必是最值得模仿的策略。

致谢

我们感谢 WouterKool 以及 EURO Meets NeurIPS 2022车辆路径竞赛的组织委员会提供了本文所研究的问题设定。

参考文献

Alinaghian, M., M. Aghaie, 和 M. S. Sabbagh（2019）。“临时救援中心选址与空中救援车辆动态路径规划的数学模型”。载于：《计算机与工业工程》131，第 227–241 页（参见第 2 页）。Basso, R., B. Kulcsár, I. Sanchez-Diaz, 和 X. Qu（2022）。“采用安全强化学习的动态随机电动车辆路径问题”。载于：Transportation Research Part E: Logistics and Transportation Review 157，第 102496 页（参见第 2 页）。Bengio, Y., A. Lodi, 和 A. Prouvost（2021）。“用于组合优化的机器学习：方法论概览”。英文。载于：《欧洲运筹学杂志》290.2，第 405–421 页（参见第 2 页）。Bent, R. W. 和 P. Van Hentenryck（2004）。“面向随机客户的部分动态车辆路径问题的基于情景的规划”。载于：《运筹学》52.6，第 977–987 页（参见第 3 页）。Berthet, Q., M. Blondel, O. Teboul, M. Cuturi, J.-P. Vert, 和 F. Bach（2020）。“通过可微扰动优化器进行学习”。载于：神经信息处理系统进展 33，第 9508–9519 页（参见第 3、11 页）。Blondel, M., A. F. Martins, 和 V. Niculae（2020）。“使用 Fenchel-Young 损失进行学习”。载于：The Journal of Machine Learning Research 21.1，第 1314–1382 页（参见第 3 页）。Carpentier, P., J.-P. Chancelier, G. Cohen, 和 M. De Lara（2015）。“随机多阶段优化”。载于：Probability Theory and Stochastic Modelling 75（参见第 2 页）。Dalle, G., L. Baty, L. Bouvier, 和 A. Parmentier（2022）。使用组合优化层进行学习：一种概率方法。arXiv:2207.13513（参见第 3 页）。Doorn, J. van, L. Lan, L. Pentinga, 和 N. Wouda（2022）。“使用混合遗传搜索和仿真求解带时间窗的静态和动态车辆路径问题”。载于：EURO Meets NeurIPS车辆路径竞赛。网址：https://github.com/ortec/euro-neurips-vrp-2022-quickstart/blob/main/ papers/OptiML.pdf（参见第 7 页）。Fikar, C.（2018）。“用于研究电子杂货配送食品损耗的决策支持系统”。载于：《计算机与工业工程》117，第 282–290 页（参见第 2 页）。Flatberg, T., G. Hasle, O. Kloster, E. J. Nilssen, 和 A. Riise（2007）。“实践中的动态随机车辆路径规划”。载于：动态车队管理：概念、系统、算法与案例研究。V. Zeimpekis, C. D. Tarantilis, G. M. Giaglis, 和 I. Minis 编。波士顿，马萨诸塞州：施普林格美国，第 41–63 页（参见第 3 页）。Gendreau, M., F. Guertin, J.-Y. Potvin, 和 É. Taillard（1999）。“用于实时车辆路径规划与调度的并行禁忌搜索”。载于：《交通科学》33.4，第 381–390 页（参见第 3 页）。Hildebrandt, F. D., B. W. Thomas, 和 M. W. Ulmer（2023）。“强化学习在随机动态车辆路径问题中的机遇”。载于：《计算机与运筹学》150，第 106071 页（参见第 2、3 页）。Joe, W. 和 H. C. Lau（2020）。“采用深度强化学习求解具有随机客户的动态车辆路径问题”。载于：自动规划与调度国际会议论文集 30.1，第 394–402 页（参见第 3 页）。Jungel, K., A. Parmentier, M. Schiffer, 和 T. Vidal（2023）。用于自主按需出行车队控制的基于学习的在线优化。arXiv:2302.03963（参见第 2、3 页）。Kool, W., L. Bliek, D. Numeroso, R. Reijnen, R. R. Afshar, Y. Zhang, T. Catshoek, K. Tierney, E. Uchoa, T. Vidal, 和 J. Gromicho（2022）。EURO Meets NeurIPS 2022车辆路径竞赛（参见第 2、4、14、15 页）。

Kool, W., H. van Hoof 和 M. Welling（2019）。“注意力机制：学习解决路径规划问题！”载于：《国际学习表征会议》（引自第 3 页）。Kool, W., J. O. Juninck, E. Roos, K. Cornelissen, P. Agterberg, J. van Hoorn 和 T. Visser（2022）。《带时间窗车辆路径问题的混合遗传搜索：高性能实现》（引自第 7–11、14、15 页）。Kotary, J., F. Fioretto, P. Van Hentenryck 和 B. Wilder（2021）。《端到端约束优化学习：综述》。arXiv:2103.16378（引自第 2 页）。Maida, J.（2022）。《美国当日送达市场：2021 年同比增长率为 18.50：2020 至 2025 年的市场规模》。网址：%7Bhttps://www.prnewswire.com/news- releases/same- day-delivery- market- in- the- us-18- 50-y- o- y- growth- rate- in- 2021-- market- size- from- 2020- to- 2025--301510886.html%7D（引自第 2 页）。Nagata, Y. 和 S. Kobayashi（2010）。“一种用于带时间窗的取送货问题的模因算法：采用选择性路径交换交叉”。载于：自然启发式并行问题求解，PPSN XI。施普林格·柏林·海德堡，第 536–545 页（引自第 8 页）。Nazari, M., A. Oroojlooy, L. Snyder 和 M. Takac（2018）。“用于解决车辆路径问题的强化学习”。载于：《神经信息处理系统进展》。S. Bengio、H. Wallach、H. Larochelle、K. Grauman、N. Cesa-Bianchi 和 R. Garnett 编。第 31 卷。Curran Associates, Inc.（引自第 2、3 页）。Nicolai, B.（2016）。《亚马逊在两小时内送达包裹》。网址：
h
t
t
ps://
www.welt.de/wirtschaft/article153690106/Amazon-liefert-Pakete-innerhalb-von-zwei-Stunden.html（引自第 2 页）。Ojeda Rios, B. H., E. C. Xavier, F. K. Miyazawa, P. Amorim, E. Curcio 和 M. J. Santos（2021）。“近期动态车辆路径问题：综述”。载于：《计算机与工业工程》160，第 107604 页（引自第 2、3 页）。Oliver, I., D. Smith 和 J. R. Holland（2013）。“旅行商问题中的排列交叉算子研究”。载于：《遗传算法及其应用》。心理学出版社，第 224–230 页（引自第 8 页）。Parmentier, A.（2021）。《学习运筹学问题的结构化近似》。arXiv:2107.04323（引自第 3 页）。—（2022）。“通过运筹学经典问题学习近似工业问题”。载于：《运筹学》70.1，第 606–623 页（引自第 3 页）。Parmentier, A. 和 V. T’Kindt（2021）。《学习求解具有释放时间和完工时间总和目标的单机调度问题》。arXiv:2101.01082（引自第 3、11 页）。Pflug, G. C. 和 A. Pichler（2014）。《多阶段随机优化》。第 1104 卷。施普林格（引自第 2 页）。Pillac, V., M. Gendreau, C. Guéret 和 A. L. Medaglia（2013）。“动态车辆路径问题综述”。载于：《欧洲运筹学杂志》225.1，第 1–11 页（引自第 2 页）。Raza, S. M., M. Sajid 和 J. Singh（2022）。“基于强化学习的车辆路径问题：近期进展”。载于：《先进机器智能与信号处理》。D. Gupta、K. Sambyo、M. Prasad 和 S. Agarwal 编。新加坡：施普林格·自然新加坡，第 269–280 页（引自第 3 页）。Ritzinger, U., J. Puchinger 和 R. F. Hartl（2016）。“动态与随机车辆路径问题综述”。载于：《国际生产研究杂志》54.1，第 215–231 页（引自第 3 页）。Soeffker, N., M. W. Ulmer 和 D. C. Mattfeld（2022）。“处方性分析视角下的随机动态车辆路径规划：综述”。载于：《欧洲运筹学杂志》298.3，第 801–820 页（引自第 2 页）。Steever, Z., M. Karwan 和 C. Murray（2019）。“食品配送服务的动态快递员路径规划”。载于：《计算机与运筹学》107，第 173–188 页（引自第 3 页）。Ulmer, M. W., N. Soeffker 和 D. C. Mattfeld（2018）。“动态多周期车辆路径规划的价值函数近似”。载于：《欧洲运筹学杂志》269.3，第 883–899 页（引自第 2、3 页）。Vidal, T.（2022）。“CVRP 的混合遗传搜索：开源实现与 SWAP* 邻域”。载于：《计算机与运筹学》140，第 105643 页（引自第 7–9、14 页）。

Vidal, T., G. Laporte, 和 P. Matl（2020年）。“现有及新兴车辆路径问题变体简明指南”。载于：《欧洲运筹学杂志》286.2，第 401–416 页（引自第2页）。Vidal, T., N. Maculan, L. S. Ochi, 和 P. H. Vaz Penna（2016年）。“面向利润车辆路径问题的隐式客户选择大邻域”。载于：《交通科学》50.2，第 720–734 页（引自第10页）。