Explaining the performance impact of opportunity costs approximation in integrated demand management and vehicle routing

David Fleckenstein, Robert Klein, Vienna Klein, Claudius Steinhardt

Angaben zur Veröffentlichung / Publication details:

Fleckenstein, David, Robert Klein, Vienna Klein, and Claudius Steinhardt. 2025. "Explaining the performance impact of opportunity costs approximation in integrated demand management and vehicle routing." *EURO Journal on Transportation and Logistics* 14: 100166. https://doi.org/10.1016/j.ejtl.2025.100166.

EURO Journal on Transportation and Logistics 14 (2025) 100166

Contents lists available at ScienceDirect

EURO Journal on Transportation and Logistics

journal homepage: www.elsevier.com/locate/ejtl

Explaining the performance impact of opportunity costs approximation in integrated demand management and vehicle routing

David Fleckenstein [a], Robert Klein [a], Vienna Klein [a,*], Claudius Steinhardt [b]

[a] Chair of Analytics & Optimization, University of Augsburg, Universitaetsstr. 16, Augsburg, 86159, Germany
[b] Chair of Business Analytics & Management Science, University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, Neubiberg, 85577, Germany

ARTICLE INFO

Keywords:
Demand management
Vehicle routing
Markov decision process
Explainability
Opportunity cost
Attended home delivery

ABSTRACT

The widespread adoption of digital distribution channels both enables and forces more and more logistics service providers to manage booking processes actively to maintain competitiveness. As a result, their operational planning is no longer limited to solving vehicle routing problems. Instead, demand management decisions and vehicle routing decisions are optimized integratively with the aim of maximizing revenue and minimizing fulfillment cost. The resulting integrated demand management and vehicle routing problems (i-DMVRPs) can be formulated as Markov decision process models and, theoretically, can be solved via the well-known Bellman equation. Unfortunately, the Bellman equation is intractable for realistic-sized instances. Thus, in the literature, i-DMVRPs are often addressed via decomposition-based solution approaches involving an opportunity costs approximation as a key component. Despite its importance, to the best of our knowledge, there is no technique to systematically analyze how the accuracy of the opportunity costs approximation affects the performance, e.g., the profit, the revenue, the number of accepted customers, or the cost of the overall logistics system, which is represented by a vehicle routing problem, nor are there general guidelines on when to apply which class of approximation approach. In this work, we address this research gap by proposing an explainability technique that quantifies and visualizes the magnitude of approximation errors, their immediate impact on the underlying vehicle routing problem and its profitability, and the relevance of approximation errors in specific regions of the state space. Exploiting reward decomposition, it further yields a characterization of different types of approximation errors. Applying the technique to a generic i-DMVRP in a full-factorial computational study and comparing the results with observations in existing literature, we show that the technique contributes to better explaining algorithmic performance and provides guidance for the algorithm selection and development process.

1. Introduction

The proliferation of e-commerce and the progress of communication technology has led to the emergence and establishment of new business models that allow customers to book on-demand logistics services, mostly the delivery of goods (Waßmuth et al., 2023) or local transportation (Vansteenwegen et al., 2022). Prominent examples of these services are attended home delivery (AHD), same-day delivery (SDD), or mobility-on-demand (MOD). These business models have in common that customers expect a very high service level, e.g., in terms of the deviation from their desired service time (Amorim et al., 2024). Meeting these expectations makes demand consolidation challenging, which entails high fulfillment cost (Ulmer, 2020) of the underlying logistics system. To still operate these logistics system profitably, their operational planning has evolved: Instead of optimizing the associated vehicle routing alone, providers additionally apply demand management to achieve efficient fulfillment operations.

The resulting integrated demand management and vehicle routing problems (i-DMVRPs) are stochastic and dynamic with two types of integrated decisions: For each dynamically arriving customer request, the provider integratively makes a *demand control decision* and a *vehicle routing decision* with the overall objective of maximizing the expected profit, i.e., revenue net of operational fulfillment cost. Such an i-DMVRP can be modeled as a Markov decision process (MDP) and, theoretically, be solved by evaluating the well-known Bellman equation (Puterman, 2014). Practically, however, i-DMVRPs suffer from the curses of dimensionality (Powell, 2011) such that this is not tractable for realistic-sized instances. Consequently, in the literature,

* Corresponding author.
E-mail addresses: david.fleckenstein@wiwi.uni-augsburg.de (D. Fleckenstein), robert.klein@wiwi.uni-augsburg.de (R. Klein), vienna.klein@uni-a.de (V. Klein), claudius.steinhardt@unibw.de (C. Steinhardt).

https://doi.org/10.1016/j.ejtl.2025.100166
Received 12 March 2025; Received in revised form 3 October 2025; Accepted 7 October 2025
Available online 10 October 2025

demand control decisions for i-DMVRPs are often optimized with a decomposition-based solution approach. More precisely, two subproblems are solved sequentially for every incoming customer request (Fleckenstein et al., 2023; Ulmer, 2020; Gallego and Topaloglu, 2019, p. 25, Klein et al., 2018):

(1.) Approximating opportunity costs (OC) for each potential fulfillment option (e.g., a delivery time window, a delivery deadline, or a pick-up time) that could be offered. OC measure the expected future profit impact of selling this option given the current state of the system. In i-DMVRPs, OC capture not only displaced revenue but also the impact on logistics fulfillment cost.
(2.) Solving the actual demand control problem based on the approximated OC, i.e., deciding on the pricing or availability of fulfillment options, or the acceptance/rejection of the request, and thereby, integratively optimizing the underlying vehicle routing problem (VRP).

If the number of fulfillment options is sufficiently large, this decomposition into OC approximation and demand control problem is the prerequisite to efficiently optimize the demand control decision and, therewith, the integrated vehicle routing decision. E.g., it can then be modeled as an assortment optimization problem that can be formulated as a linear program (Heger and Klein, 2024). In addition to this computational advantage, the decomposition-based solution concept is flexible and extensible because any OC approximation algorithm suitable for i-DMVRPs (1) can be combined with any solution approach for the demand control problem (2). The observation that this basic structure is common to many solution algorithms for i-DMVRPs leads to the following conclusion: One of the main avenues for accelerating the development of practical solution approaches for i-DMVRPs is understanding and explaining the relation between the accuracy reached in (1), i.e., the *accuracy of the OC approximation*, and the quality of (2), i.e., the *quality of the resulting integrated demand control and vehicle routing decision*. However, despite the maturity of integrated demand management and vehicle routing as a research area, this relation has not been systematically explored, and therefore, largely remains a black box so far. Hence, we now close this research gap by comprehensively analyzing this relationship.

It is important to note that the strong interdependence between demand control and vehicle routing decisions makes them unique compared to similar demand management problems, e.g., in the field of revenue management. This is especially due to variable routing cost being neither negligible nor attributable to individual orders (Fleckenstein et al., 2025). This has two consequences for our analysis: First, the insights we aim to obtain are specific to the problem family of i-DMVRPs. Second, the analysis requires a novel explainability technique specifically tailored to i-DMVRPs. The technique we propose combines two building blocks:

B1: Chain of influencing factors – The first building block resembles a typical post-hoc explainability technique (Arrieta et al., 2020), i.e., we define metrics to evaluate the behavior of a given policy in certain states and the respective impact on solution quality. Thereby, we aim at answering the central questions that arise along the chain of influencing factors from OC approximation error to objective value loss as depicted in Fig. 1. This allows deriving insights on when (in which states) and why a certain policy performs especially good or bad.

B2: Reward decomposition – The second building block incorporates the idea of reward decomposition (Juozapaitis et al., 2019). For that, we exploit the finding that, in i-DMVRPs, opportunity costs can be decomposed into displacement cost (DPC) and marginal cost-to-serve (MCTS) (Fleckenstein et al., 2025). More precisely, we propose to apply OC approximations capturing only one of the two components, with the aim of assessing the importance of the respective component for approximation accuracy.

Afterward, in an extensive computational study, we apply the explainability technique to the generic i-DMVRP with accept/reject decisions and the stylized parameter settings from Fleckenstein et al. (2025). This focus on the simplest possible demand control problem is deliberate: First, it yields clean and isolated insights into the performance impact of OC approximation. Second, it captures the fundamental trade-off inherent to any i-DMVRP, namely, whether the revenue of a certain customer order exceeds the expected loss in future revenue and the expected increase of fulfillment cost. Therefore, the findings from the simple accept/reject setting can also be applied to more complex settings with multiple fulfillment options and dynamic prices. By complementing our numerical results with an extensive analysis of computational studies in the literature, we confirm the validity of our findings. In line with the theoretical arguments, our identified OC approximation error types can indeed also explain the observed performance of state-of-the-art solution approaches for richer, real-world i-DMVRPs. In summary, our work has four contributions:

(1.) To the best of our knowledge, we introduce the first explainability technique for the widely established decomposition-based solution approaches for i-DMVRPs.
(2.) We apply our explainability technique within a comprehensive computational study and identify fundamental OC approximation error types, i.e., OC approximation errors that can occur in a broad variety of real-world i-DMVRPs. Therewith, we are the first to systematically analyze the relation between the accuracy of OC approximation and the objective value.
(3.) We classify patterns in the occurrence of the fundamental approximation error types we identify, characterize which problem settings are prone to which error type, and propose algorithmic elements to successfully mitigate them. This yields insights that guide the selection and the design of OC approximation algorithms.
(4.) We compile indications for the occurrence of the identified error types from existing literature and show that our findings improve explainability of the reported results. Thereby, we transform the existing implicit knowledge about specific i-DMVRPs to explicit, high-level knowledge.

The remainder of this paper is structured as follows: In Section 2, we review the related literature both on algorithmic explainability and i-DMVRPs. In Section 3, we introduce and model the generic i-DMVRP under consideration. Then, in Section 4, we present our novel explainability technique for i-DMVRPs in detail, and we present our computational study in Section 5. In Section 6, we derive general insights for algorithm design. We discuss limitations in Section 7 and summarize our work in Section 8.

2. Literature review

Due to the cross-cutting nature of our study, the related literature spans across multiple distinct research areas. In Section 2.1, we review the literature on i-DMVRPs with a special focus on its origins in revenue management, dynamic pricing, and dynamic vehicle routing. In Section 2.2, we then discuss algorithmic explainability techniques, particularly from explainable reinforcement learning (RL), highlighting the techniques that we adapt and apply in the work at hand. Finally, in Section 2.3, we review the descriptive analytics that authors use to explain the observed performance of their i-DMVRP solution approaches.

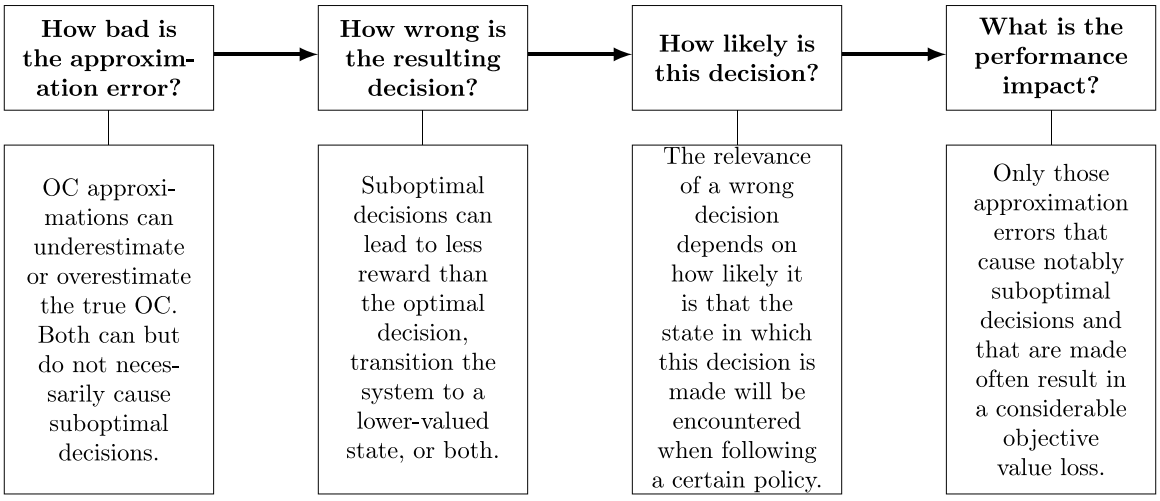


Fig. 1. Chain of influencing factors from OC approximation error to objective value loss.

2.1. Modeling and solving i-DMVRPs

In logistics, many companies dynamically collect orders for a transportation service that is fulfilled by a given fleet of vehicles. These companies face an i-DMVRP if they can both plan individual offers made in response to customer requests and plan the vehicle routes to feasibly fulfill the resulting orders. Hence, i-DMVRP research synthesizes two originally distinct research areas:

(1.) Research in *revenue management and dynamic pricing* addresses the dynamic optimization of offering decisions under the assumption that fulfillment is already pre-planned. For an extensive overview of this field, we refer the reader to the textbooks by Gallego and Topaloglu (2019) and Talluri and Van Ryzin (2004) as well as the reviews by Klein et al. (2020) and Strauss et al. (2018).
(2.) *Dynamic vehicle routing* investigates the optimization of fulfillment assuming given orders that arrive dynamically. For a deeper discussion of this research area, we refer the reader to the textbook by Toth and Vigo (2014) as well as the reviews by Hildebrandt et al. (2023), Soeffker et al. (2022), and Psaraftis et al. (2016).

Starting with the seminal work of Campbell and Savelsbergh (2005) on an AHD system, i-DMVRPs are considered in a variety of applications such as SDD (Azi et al., 2012), MOD (Atasoy et al., 2015), or mobile personnel booking (Avraham and Raviv, 2021). Indicative of the growing importance of this research area, there are several reviews that are either application-specific (Li et al., 2025; Waßmuth et al., 2023; Snoeck et al., 2020), or aim at i-DMVRP literature in general (Fleckenstein et al., 2023).

Since i-DMVRPs are dynamic and stochastic, the natural modeling approach is to formulate a Markov decision process (MDP) model (Puterman, 2014). It is important to note that MDP models not only serve as a formal problem definition. On top of that, model analysis, which can be done analytically or numerically (Bravo and Shaposhnik, 2020), yields domain knowledge that can be exploited by solution approaches. For i-DMVRPs, such model analyses can be found in Fleckenstein et al. (2025), Lebedev et al. (2021), and Asdemir et al. (2009). These three theoretical works addressing i-DMVRPs have in common that they mainly analyze models analytically. In contrast, our technique focuses on the numerical analysis of solution approaches. It also yields domain knowledge regarding a combination of a solution approach and a model of a specific i-DMVRP. However, we draw on the results of Fleckenstein et al. (2025), who extend the concept of OC for i-DMVRPs by tailoring a specific OC definition that captures and quantifies demand control effects on the underlying vehicle routing problem and show that, in i-DMVRPs, OC can be decomposed into MCTS and DPC.

Although i-DMVRPs can be solved to optimality by exact dynamic programming algorithms, this is impractical for realistic-sized instances. Hence, there exists a wide variety of heuristic solution approaches for specific i-DMVRPs. According to Fleckenstein et al. (2023), they can be classified into two broad solution concepts: First, there are static deterministic approaches that solve auxiliary models with a rolling-horizon to iteratively derive demand management and vehicle routing decisions (e.g. Klapp et al., 2020). Second, there are decomposition-based solution approaches that subdivide the demand management task into two sub-problems as already mentioned: OC approximation and demand control decision-making which integrates optimizing the underlying VRP (see Section 3).

In this work, we only consider the decomposition-based solution approaches, which are adopted by the majority of authors (Fleckenstein et al., 2023). The applied OC approximation approaches can be sub-divided further into sampling-based (e.g. Klein and Steinhardt, 2023) and learning-based (e.g. Ulmer, 2020) and are either targeted at approximating MCTS, DPC, or both (Fleckenstein et al., 2025). To evaluate the performance of these approximation approaches, authors of existing works resort to descriptive analyses. With our work, we aim at explaining the reasons for the observed results by investigating the impact of OC approximation errors.

2.2. Explainability of algorithmic performance and behavior

Explainability becomes increasingly relevant in analytics and optimization in general (see, e.g., the recent reviews by De Bock et al., 2024 and Goerigk and Hartisch, 2023). In particular, the field of explainable RL (XRL) has recently gained more attention (Milani et al., 2024). Due to the close relation between RL and the OC approximation approaches observed in i-DMVPR literature, the techniques developed for XRL are also applicable to most approaches tackling i-DMVRPs.

Milani et al. (2024) introduce a two-dimensional taxonomy of explainability techniques tailored to XRL. The first classification dimension proposed by the authors is borrowed from general explainable artificial intelligence:

(1.) Explainability can be *inherent* to a policy or restored *post-hoc*.
(2.) We can further distinguish *local* explanations that refer to individual states, and *global* explanations that holistically view the behavior of the policy.
(3.) Among the post-hoc techniques, a distinction can be made regarding the *degree of portability*, i.e., the range of solution approaches the technique can be readily applied to.

Regarding this classification dimension, our explainability technique is a *post-hoc* explanation since it is applied to a given (decomposition-based) policy. As discussed in Section 2.1, this type of policy is quite common, which makes our technique *portable*. Further, it features *local* (state-level) metrics but also involves *global* considerations since these local metrics are aggregated to explain the global behavior of the policy.

The second classification dimension specifically addresses XRL approaches and distinguishes explainability techniques based on the type of explanations they incorporate as follows:

(1.) *Feature importance explanations*: explaining individual actions by providing their context, e.g., state features. Typical approaches are, e.g., surrogate policies encoded as decision trees or saliency map explanations.
(2.) *Learning process and MDP explanations*: exploiting the definition of MDP model elements or training process steps to generate explanations. The aim is to identify critical drivers of the policy's individual decisions.
(3.) *Policy-level explanations*: identifying recurring sequences of decisions (e.g. by clustering states) to extract patterns of the policy's overall control behavior.

Regarding this second classification dimension, our technique can be viewed as a combination of *policy level explanations* (in B1) and *learning process and MDP explanations* (in B2). In the following, we briefly review the closest related literature for B1 and B2 separately.

B1 of our explainability technique is closely related to a technique called *strategy summarization* by Amir et al. (2019). They suggest identifying states of interest on the basis of importance, coverage, likelihood of encountering, and policy disagreement with the aim of aggregating these states to summarize the behavior of the policy. Applied to i-DMVRPs, measuring the OC approximation error itself can be considered equal to measuring policy disagreement with the optimal policy. To quantify state importance, we measure the impact of an approximation error in a certain state on the quality of the resulting decision. This can also be viewed as a special case of the state importance metric used by Torrey and Taylor (2013). Further, like (Amir et al., 2019), we consider the likelihood of encountering a state.

B2 of our explainability technique is a reward decomposition technique. It is first proposed by Russell and Zimdars (2003) with the aim of facilitating the learning process. With the same goal, it is also applied by Van Seijen et al. (2017) in the form of a hybrid reward architecture. However, as shown by Juozapaitis et al. (2019), reward decomposition can not only be applied for designing hybrid reward architectures but also as an explainability technique. Therefore, they analyze the influence of the different reward components for explaining the behavior of a given policy. In contrast, we analyze approximation errors that result from considering only one reward component for explaining the behavior of a given policy. This idea of analyzing approximation errors in RL is first presented by Mannor et al. (2007) with the aim of computing confidence intervals.

Regarding the application of our explainability technique, a distinguishing feature compared to most existing works in XRL is that we consider a large number of small problem instances and solve them to optimality. Thereby we derive generic domain knowledge, in the form of fundamental OC approximation error types, rather than analyzing heuristic policies for large instances. In this regard, we only found one similar approach by Bravo and Shaposhnik (2020). They use machine learning to analyze optimal policies for small problem instances of, amongst others, traditional revenue management problems.

In summary, our methodology combines a variety of existing RL explainability techniques in a novel way: Besides adapting them to the problem structure of i-DMVRPs, we introduce the new idea of combining strategy summarization and reward decomposition and applying both to derive characterizations of fundamental OC approximation errors.

2.3. Performance metrics in i-DMVRP literature

In contrast to "pure" revenue management and dynamic pricing, where explainability has already received some attention (e.g., Biggs et al., 2021; Bravo and Shaposhnik, 2020), we find no systematic application of techniques from XRL in the literature on i-DMVRPs. Instead, most authors evaluate the performance of their solution approaches by incorporating descriptive analytics, as we summarize in the following.
Aggregate metrics – Apart from the arithmetic mean of profit, which is the objective in most of the considered i-DMVRPs, many authors additionally report the following aggregate metrics describing the performance of policies: Among the most widely reported metrics are average or overall revenue, cost, and number of orders (Campbell and Savelsbergh, 2005). Further, some authors also report revenue per order (Klein et al., 2018), cost per order (Yang et al., 2016), average number of fulfillment options offered to each customer (Mackert, 2019), pooling rate (Anzenhofer et al., 2025), or fleet utilization (Klein and Steinhardt, 2023). In addition to the arithmetic mean, the standard deviation (Yang et al., 2016) or the coefficient of variation (Anzenhofer et al., 2025) are reported in a few studies.
Decision-making – For a more detailed analysis of a policies' performance, authors analyze how the resulting decision-making differs over time, i.e., over the course of the booking horizon, or for different types of requests: For any i-DMVRP, the acceptance rate or conversion rate (Mackert, 2019) or the cumulative revenue over time (Lang et al., 2021) can be reported. If customers can choose from a set of fulfillment options, the number (Abdollahi et al., 2023) or composition of offered fulfillment options (Klein and Steinhardt, 2023), or the chosen fulfillment options (Anzenhofer et al., 2025) can be analyzed. If dynamic pricing is applied, average prices of offered (Klein et al., 2018) or chosen fulfillment options (Yang et al., 2016) are reported.
Opportunity costs – If a parametric OC approximation is used, its parameter values (Lang et al., 2021) or the function values for certain parameter values (Avraham and Raviv, 2021) can be investigated. Only very rarely, authors directly consider approximated OC values for different groups of similar requests (Yang and Strauss, 2017) or over time (Koch and Klein, 2020).

In general, we identify three central problems that limit the explanatory power of the existing descriptive analyses: First, observations of the performance and the behavior of a policy do not provide direct evidence of whether or how exactly an OC approximation error influences the observed performance. Due to a lack of conclusive explanations, the reasoning is often limited to formulating hypotheses. Second, the metrics are only analyzed in an aggregate form, which does not allow distinguishing different types of errors that originate in certain regions of the state space. Third, since typically, a specific solution approach for a specific i-DMVRP is considered, the results are hardly attributable to certain characteristics of the problem structure, the instance structure, or the solution approach. This again limits conclusiveness and transferability.

Overall, there is a clear research gap regarding the development of explainability techniques for i-DMVRPs and the formulation of generalizable explanations for policy performance.

3. Problem definition and modeling

In this section, we formally characterize i-DMVRPs with a particular focus on the generic MDP model for i-DMVRPs by Fleckenstein et al. (2025).

Typically, an i-DMVRP is structured as follows: During a *booking horizon*, customers log-in to the business platform and place a service request by entering service parameters like pick-up/drop-off locations, desired fulfillment times, or vehicle types. In response, the provider either presents a set of suitable *fulfillment options* with different prices to choose from, or accepts/rejects the request. Then, a successfully placed customer request turns into a confirmed *customer order*. All customer

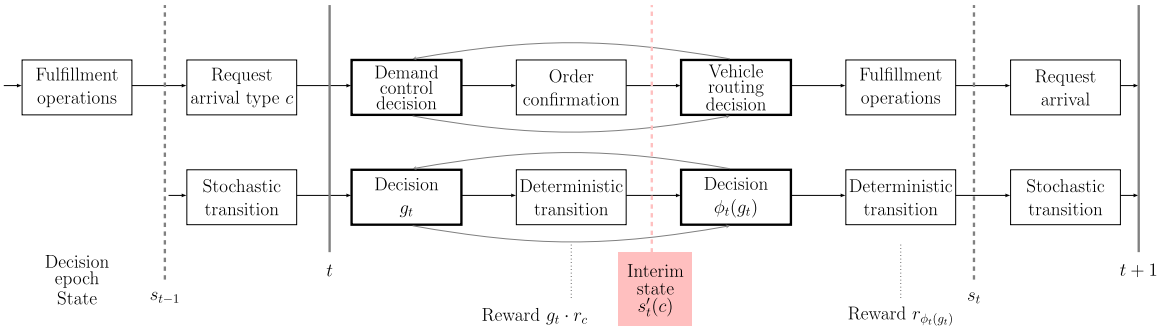


Fig. 2. Overview of the MDP model of the i-DMVRP booking and fulfillment process including the interim state (Fleckenstein et al., 2025).

orders are eventually served by the provider within the *service horizon*, which can either be *disjoint* or *overlapping* with the booking horizon. The former is typical for AHD, where customer and provider agree on a delivery time window for a certain day in advance. The latter is typical for SDD or MOD, where the customer expects to receive a service on short notice.

In the following, we consider the generic i-DMVRP model as in Fleckenstein et al. (2025) but adapt it specifically for the case of disjoint booking horizons and service horizons. Further, the underlying demand control subproblem features an accept/reject demand control. While the generalization to richer problems with multiple fulfillment options has a quite considerable impact on the demand control problem, it hardly impacts OC approximation. The only change is that there is no longer a single OC value for the entire request but one per fulfillment option that the customer could potentially order. To demonstrate this mathematically, we explicitly show how the considered problem can be generalized at the end of this section (see also Fleckenstein et al., 2025).

Analyzing the simplest possible demand control problem with only a single fulfillment option to derive insights for the entire family of i-DMVRPs is deliberate: First, precisely because of the demand control problem being so simple, we can obtain clean, isolated, and unambiguous insights on the performance impact of the OC approximation. In a more complex setting with multiple fulfillment options and customer choice, this kind of diagnostic clarity would be hard to obtain due to possible interactions between the demand control problem and the OC approximation. Second, because of the very high similarity between OC approximation in single-option and multi-option settings, the single-option setting is structurally representative of the core trade-off inherent in any of the more complex i-DMVRPs. More precisely, multi-option problems can be cast as a sequence of (albeit interdependent) accept/reject decisions on whether each fulfillment option should be offered based on its individual revenue net of its individual OC. Therefore, the insights gained from the single-option setting are fundamental in the sense that they also apply to the more complex problems within the family of i-DMVRPs.

Generic i-DMVRP model with accept/reject decisions

Decision epoch – A decision epoch marks the start of the MDP model's stages. In the considered problem, such stages correspond to (constant) time steps $t = 1; \dots; T$. A *customer request of type* $c \in C$ can arrive in stage t with a certain *arrival rate* λ^t_c. The type c of a customer request defines its associated *revenue* r_c and *location* l_c. Individual customer requests are then uniquely identified by combining this information with their *request time* . Arrival rates are assumed to be small enough that at most one customer request arrives per stage.

State – The system state $s_t = (C_t; \ _t)$ comprises two sets. The first set C_t consists of tuples $(c; \ )$, which store customer orders for which fulfillment has not yet started. The second set $_t$ stores the tour plan. Since we assume disjoint booking and service horizons, in our case, $_t$ is either preliminary or empty for all $t < T$. Please note that s_t defines a post-decision state. The state space of a decision epoch t is denoted as S_t and comprises all potential realizations of customer orders C_t and tour plans $_t$. Thus, $\forall t \in 1; \dots; T : s_t \in S_t$.

Action – An action in response to an arriving customer request of type c integrates an accept/reject decision for demand control $g_t \in G(s_{t-1}; c) \subseteq \{0; 1\}$, and a tour planning decision $_t(g_t) \in \ (s_{t-1}; c; g_t)$. Again, $_t(g_t)$ is either preliminary or empty for all $t < T$ due to the disjoint horizons. The action space for the tour planning, denoted as $(s_{t-1}; c; g_t)$, is defined by the routing constraints of the problem and depends on the preceding state s_{t-1}, the type c of the arriving request, and the demand control decision g_t. The action space for the demand control, denoted as $G(s_{t-1}; c)$, in turn, depends on $(s_{t-1}; c; g_t)$ since $g_t = 1$ is only feasible if $(s_{t-1}; c; g_t) \neq \emptyset$. Thus, $A_t(s_{t-1}; c) = \{(g_t; \ _t(g_t)) : g_t \in G(s_{t-1}; c); \ _t(g_t) \in (s_{t-1}; c; g_t)\}$.

Rewards – As a consequence of an acceptance decision $g_t = 1$, a revenue r_c is received. A rejection yields no reward. A routing decision $_t(g_t)$ entails a reward $r_{\ _t(g_t)}$. It equals the newly arising fulfillment cost, which, given the triangle inequality holds, is non-positive. Again, since we assume disjoint booking and service horizons, $\forall t < T : r_{\ _t(g_t)} = 0$. This means that the entire routing cost is incurred at the last decision epoch when the definitive tour planning decision is made.

Transition – When transitioning to state s_t, $_t$ is set to $_t(g_t)$. The first state component C_{t-1}, also changes. More precisely, if the newly arriving request of type c_t is accepted, the resulting customer order is added.

Objective – The provider aims at maximizing profit after fulfillment. Therefore, it is required to determine a policy that returns the optimal decision for each state that can potentially be reached. These decisions of a policy can be denoted as $a_t(s_{t-1}; c_t) = (g_t(s_{t-1}; c_t); \ _t(g_t(s_{t-1}; c_t)))$ at decision epoch t. Then, the objective function is:

$$\max \mathbb{E}\left[\sum_{t=1}^{T} (r_{c_t} g_t(s_{t-1}; c_t) + r_{\ _t(g_t(s_{t-1}; c_t))}) \mid s_0\right]: \tag{1}$$

Bellman equation – The objective function (1) can be expressed in the form of a Bellman equation, which defines a value $V_t(s_t)$ for each state s_t. Solving this equation yields the optimal policy *.

$$\begin{aligned} V_{t-1}(s_{t-1}) = \sum_{c \in C} \lambda^t_c \Big[& \max_{g_t \in G(s_{t-1};c)} \Big\{ g_t \Big(r_c + \max_{\ _t(g_t) \in \ (s_{t-1};c;g_t)} \Big\{ -r_{\ _t(g_t)} \\ & + V_t(s_t \mid s_{t-1}; \ _t(g_t)) \Big\} \Big) \\ & + (1 - g_t) V_t(s_t \mid s_{t-1}; \ _t(0)) \Big\} \Big] \\ & + \Big(1 - \sum_{c \in C} \lambda^t_c\Big) \max_{\ _t(0) \in \ (s_{t-1};0;0)} \Big\{ -r_{\ _t(0)} + V_t(s_t \mid s_{t-1}; \ _t(0)) \Big\}; \end{aligned} \tag{2}$$

with boundary condition:

$$V_T(s_T) = 0: \tag{3}$$

In Eq. (2), both types of decisions are represented in an integrated form. Thus, an interim state $s'_t \mid s_{t-1}; c; g_t$ can be defined to isolate the impact of the demand control decision from the impact of the vehicle

routing decision as also depicted in Fig. 2. Further, substituting the OC of accepting a request of type c, i.e., $V_t(s_{t-1}; c)$, we obtain the following reformulation. Note that we denote interim states $s'_t \mid s_{t-1}; c; 1$ by $s'_t(c)$ and interim states $s'_t \mid s_{t-1}; c; 0$, or $s'_t \mid s_{t-1}; 0; 0$, by $s'_t(0)$.

$$V_{t-1}(s_{t-1}) = \sum_{c \in C} \lambda_c \max_{g_t \in G(s_{t-1}; c)} \left(g_t \left(r_c - V_t(s_{t-1}; c) \right) + V'_t(s'_t(0)) \right); \tag{4}$$

with

$$V'_t(s'_t \mid s_{t-1}; c; g_t) = \max_{\gamma_t(g_t) \in \Gamma(s_{t-1}; c; g_t)} r_{\gamma_t(g_t)} + V_t(s_t \mid s_{t-1}; \gamma_t(g_t)) = r_{\gamma^*_t(g_t)} + V_t(s_t \mid s_{t-1}; \gamma^*_t(g_t)); \tag{5}$$

and

$$V_t(s_{t-1}; c) = V'_t(s'_t(0)) - V'_t(s'_t(c)) \geq 0. \tag{6}$$

Further, $\gamma^*_t(g_t)$ denotes the optimal routing decision for a given demand management decision g_t.

Generalization to multiple fulfillment options

Now, we briefly describe how the model introduced above can be generalized to multiple fulfillment options. The purpose is to strengthen the theoretical argument that findings regarding the OC approximation for the accept/reject demand control problem can likewise be generalized to richer problems with multiple fulfillment options. In Section 6, we provide further evidence for that this generalization is possible by evaluating computational results in the existing literature on richer i-DMVRPs.

We begin with the necessary adaptations to the definitions of action, transition, and reward before presenting the Bellman equation.

State – The set C_t, which stores pending customer orders, now consists of tuples $(c; \cdot; o)$, with o additionally denoting the fulfillment option chosen by the customer.

Action – In response to a customer request of type c submitted in state s_{t-1}, the provider now decides whether or not to offer each fulfillment option $o \in O(s_{t-1}; c)$, where $O(s_{t-1}; c)$ denotes the set of fulfillment options that can be offered feasibly. With these individual decisions, the complete action is to select an offer set $g_t \subseteq O(s_{t-1}; c)$.

Transition – Because the customer can now choose from multiple fulfillment options, the order confirmation step involves stochasticity in the form of customer choice probabilities $P_o(g_t)$.

Rewards – Since the revenue of an order can depend on the chosen fulfillment option $o \in g_t$, it now also features o as an index r_{co}.

These adaptations lead to the following formulation of the Bellman equation for the multi-option setting, generalizing the Bellman equation (4):

$$V_{t-1}(s_{t-1}) = \sum_{c \in C} \lambda_c \max_{g_t \in G(s_{t-1}; c)} \left(\sum_{o \in g_t} P_o(g_t) \left(r_{co} - V_t(s_{t-1}; c; o) \right) + V'_t(s'_t(0)) \right). \tag{7}$$

Crucially, it now features OC specific to each fulfillment option $o \in O(s_{t-1}; c)$:

$$V_t(s_{t-1}; c; o) = V'_t(s'_t(0)) - V'_t(s'_t(o)); \tag{8}$$

with interim state $s'_t \mid s_{t-1}; c; o$ denoted as $s'_t(o)$ and interim state $s'_t \mid s_{t-1}; c; 0$ denoted as $s'_t(0)$.

4. Explainability technique

In this section, we present our novel explainability technique for i-DMVRPs, which comprises two separate building blocks. Both are later applied for the comprehensive analysis of the relation between OC approximation error and the quality of the resulting demand management decisions.

4.1. Building block 1

The basic idea of B1 is to define metrics for each step in the chain of influencing factors behind the losses in objective value observed when following a certain policy (see Fig. 1). By this, we aim at identifying the regions of the state space that are especially relevant regarding the respective overall objective value loss. Therefore, we analyze the occurrence, the sign, and the magnitude of OC approximation errors in the respective states. Then, by suitable visualizations of the metrics, we compare them over various settings of problem parameter values (in the following referred to as *settings*) resembling different real-world i-DMVRPs. Based on that, we classify fundamental types of approximation errors, i.e., OC approximation errors that a broad variety of real-world i-DMVRPs are prone to. Therewith, we can eventually explain the performance of the considered policy. Both the metrics and the respective visualizations are generally valid, i.e., can be applied to any policies derived from different OC approximation approaches. In the following, we first describe the chain of influencing factors between OC approximation error and objective value loss. Afterward, we describe the metrics we use to quantify each step in this chain of influencing factors, propose visualizations of these metrics, and finally, discuss how to extend B1 to other performance indicators.

Chain of influencing factors

At the beginning of the chain of influencing factors, there is an approximation error in a certain state s_{t-1}, that could either be an underestimation or an overestimation of the true OC. Depending on the actual magnitude of such an approximation error, the magnitude of the true OC, i.e., $V_t(s_{t-1}; c)$, and the immediate reward r_c, this error can but not necessarily must result in a suboptimal decision.

Generally, a suboptimal decision in a certain state s_{t-1} can either yield less immediate reward than the optimal decision, transition the system to a lower-valued state than the optimal decision, or both. However, the respective negative effect on the objective value itself can vary from barely notable to considerable.

Whether the chain of influencing factors continues further, depends on the likelihood that s_{t-1} is encountered and the respective suboptimal decision is made when following the policy under consideration. We now define disaggregated metrics to quantify "how bad" an OC approximation error is, "how wrong" the resulting decision is, and also, "how likely" this decision is. Additionally, we define a fourth metric that captures the aggregated overall impact of OC underestimations or OC overestimations on the objective value, i.e., it quantifies the impact of the resulting demand control decision on the overall performance, e.g., the profit, the revenue, the number of accepted customers, or the cost of the underlying VRP.

Metrics

Fig. 3 shows which of the metrics presented in the following corresponds to which step in the previously described chain of influencing factors between OC approximation error and objective value loss. Please note, we assume that we examine an OC approximation relative to the true OC, i.e., a suboptimal policy relative to the optimal policy.

(1) Error magnitude – In general, an OC approximation can be inaccurate in both directions. By separately evaluating *overestimation* and *underestimation*, we can analyze whether the consequences are different. Hence, we calculate the magnitude of overestimation errors $e^o(s_{t-1}; c)$ and underestimation errors $e^u(s_{t-1}; c)$, separately, for all individual states s_{t-1} and request arrivals c as follows:

$$e^o(s_{t-1}; c) = \max\left(0, \hat{V}_t(s_{t-1}; c) - V_t(s_{t-1}; c)\right); \tag{9}$$

$$e^u(s_{t-1}; c) = \max\left(0, V_t(s_{t-1}; c) - \hat{V}_t(s_{t-1}; c)\right). \tag{10}$$

This metric allows us to identify regions of the state space, where an approximation systematically overestimates ($e^o(s_{t-1}; c) > 0$) or underestimates ($e^u(s_{t-1}; c) > 0$) the true OC. Hence, it provides information

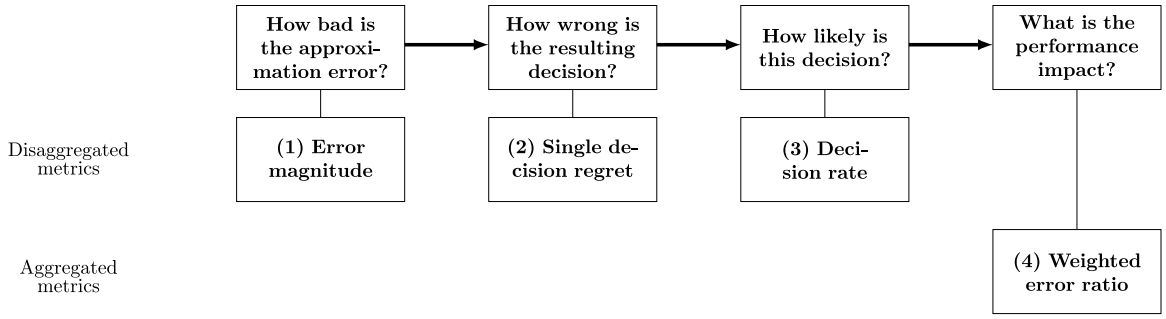


Fig. 3. Metrics among the chain of influencing factors from OC approximation error to objective value loss.

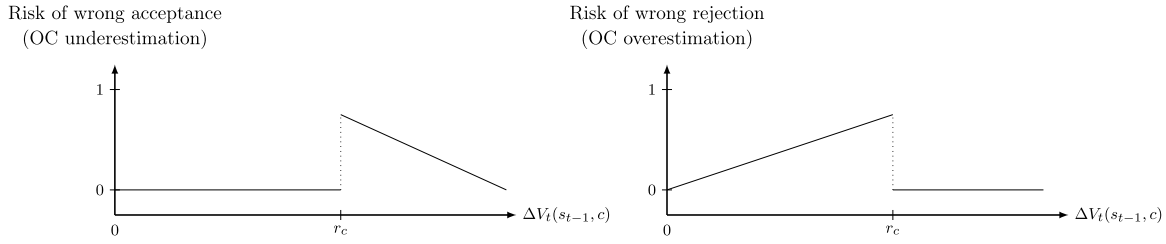


Fig. 4. Schematic representation of the risk of a wrong decision depending on the true opportunity costs $V_t(s_{t-1}; c)$ and the immediate reward r_c.

about where approximation errors originate and how strongly the chain of influencing factors is triggered.

(2) Single decision regret – As is well-known in revenue management (e.g., Talluri and Van Ryzin, 2004), an OC approximation error in itself is not problematic because the resulting decision may still be fairly accurate or even optimal. As previously described, the chain of influencing factors only continues if there is a suboptimal decision. Again, we can distinguish the consequences of an overestimation and an underestimation. Fig. 4 schematically plots the risk of a wrong decision dependent on the relation between immediate reward r_c and the true OC $V_t(s_{t-1}; c)$. In this context, the risk is to be interpreted as how large the error magnitude must be to cause a wrong decision. I.e., if the risk is high, even a slight error already causes a wrong decision. In case of an underestimation, only a wrong acceptance can result. However, if the true OC are equal to or below the immediate reward, the optimal decision is the acceptance, which is also the result of any arbitrarily erroneous OC approximation. Only if the true OC are larger than the immediate reward, the optimal decision is a rejection, which means that there is a risk of a wrong acceptance due to underestimation. This risk is greater the "tighter" the optimal acceptance decision is. In other words, the difference between true OC and the immediate reward acts like a buffer that absorbs underestimation errors up to a certain magnitude. Analogously, overestimation errors can exclusively cause wrong rejections, which is only possible if the true OC are at or below the immediate reward. Again, such a wrong decision is more likely the "tighter" the optimal acceptance decision is.

For those states, in which the buffer cannot fully absorb the error, we must quantify the "suboptimality" of this single wrong decision. To this end, we introduce the metric *single decision regret*, denoted as $(s_{t-1}; c)$ for when a customer request of type c arrives in state s_{t-1}. It computes the overall reward difference between a single decision based on a, potentially wrong, OC approximation and the optimal decision. In less technical terms, to isolate the regret of one single demand control and integrated vehicle routing decision, all future decisions from decision epoch $t + 1$ onward are assumed to be made based on the optimal policy in both cases. Then, again depending on the observed underlying error magnitude $e^o(s_{t-1}; c)$ and $e^u(s_{t-1}; c)$, we can distinguish between overestimation regret and underestimation regret, even though both are calculated based on the same expression:

$$(s_{t-1}; c) = g_t^*(s_{t-1}; c) \quad r_c - V_t(s_{t-1}; c) \quad - g(s_{t-1}; c) \quad r_c - V_t(s_{t-1}; c) \; ; \tag{11}$$

with $g_t^*(s_{t-1}; c)$ denoting the optimal demand control decision and $g(s_{t-1}; c)$ denoting the demand control decision when following the policy under consideration. Then, we define the overestimation regret as:

$$^o(s_{t-1}; c) = \begin{cases} (s_{t-1}; c); & e^o(s_{t-1}; c) > 0 \\ 0; & \text{otherwise,} \end{cases} \tag{12}$$

and the underestimation regret as:

$$^u(s_{t-1}; c) = \begin{cases} (s_{t-1}; c); & e^u(s_{t-1}; c) > 0 \\ 0; & \text{otherwise.} \end{cases} \tag{13}$$

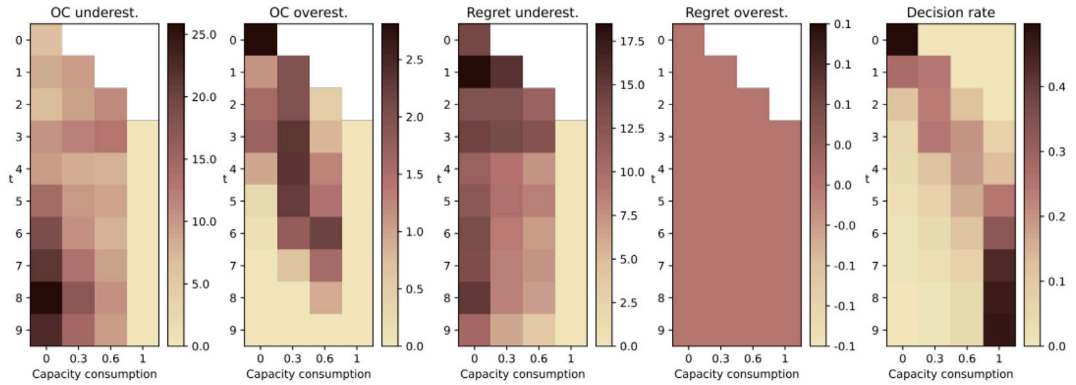
With this metric, we can assess whether an overestimation error or underestimation error leads to a suboptimal decision, and by which amount it causes the objective value to deteriorate assuming optimal decisions over the remaining booking process. Please note, for ease of readability, in the following, we refer to the single decision regret as *regret*.

(3) Decision rate – As a third step in the chain of influencing factors, the relevance of a suboptimal decision must be considered. It depends on how likely it is to visit the state in which the decision is made. We measure the likelihood in the form of the decision rate $P(s_{t-1}; c)$, which denotes the probability that a policy f visits state s_{t-1} and decides on the acceptance/rejection of a customer request of type c at decision epoch t. To calculate it, we simulate decision-making based on the considered OC approximation for a sufficiently high number of drawn sample paths. This metric provides information about the relevant areas of the state space, and thus, to what extent the regret in a certain state impacts the objective value.

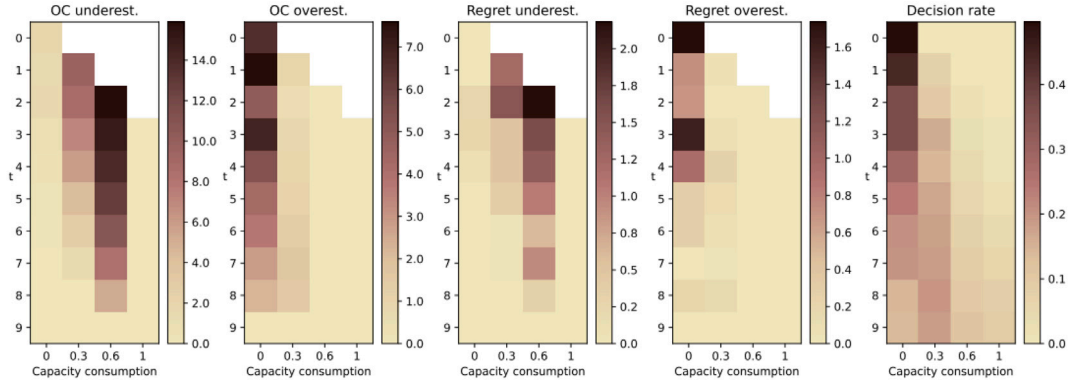
(4) Weighted error ratio – Additionally to the metrics involved in our chain of influencing factors, we propose an aggregate metric to approximate the share of the loss in objective value that is caused by overestimation and underestimation, respectively. We refer to this metric as *weighted error ratio* and define it for a certain setting as follows:

$$E = \frac{\sum_{s \in S} \sum_{c \in C} {}^o(s; c) \; P(s; c)}{\sum_{s \in S} \sum_{c \in C} \left({}^o(s; c) + {}^u(s; c) \right) \; P(s; c)} \tag{14}$$

Thus, a weighted error ratio of 1 means that the full observed objective value loss of a certain policy is caused by overestimation errors, and



(a) Policy A: Average objective value reached = -8.334



(b) Policy B: Average objective value reached = 12.447

Fig. 5. Technique of error characterization — exemplary for two policies (A and B) applied to the same setting.

a weighted error ratio of 0 means that only underestimation causes objective value loss.

Visualizations

Next, we propose a suitable visualization for each of the metrics in the chain of influencing factors. With these visualizations, we make the values of the metrics interpretable, such that common patterns across the state space and across different policies and problem settings can be identified. To illustrate the design of the proposed visualizations and how to interpret them, Figs. 5 and 6 serve as conceptual illustrations. Thus, the data shown are placeholders and not relevant for analysis. The actual analysis follows in Section 5.

Disaggregated metrics (1)–(3) — Due to the curses of dimensionality, analyzing the values of the metrics (1)–(3) for individual states as introduced above is impractical. Hence, we propose a state space aggregation for their visualization and the subsequent analysis. For this, we draw on two-dimensional lookup tables, which are well-known from reinforcement learning (e.g., Powell, 2022). We aggregate states according to the dimensions *decision epoch* t (y-axis) and *capacity consumption* (x-axis), where we measure the latter in percentage of the available capacity. Since we consider multiple instances per setting, we further aggregate the results of all instances per setting in one look-up table by averaging the respective numbers. In sum, these aggregations allow us to examine the average magnitude of overestimation and underestimation, the associated regret, and the decision rate in the different regions of the state space. Additionally, all instance-specific effects are averaged out. Further, this two-dimensional aggregation allows us to represent the resulting look-up tables in heatmaps, as exemplarily depicted in Fig. 5.

Now, it is possible to qualitatively analyze and compare the results for different policies as exemplarily demonstrated in the following: In the example presented in Fig. 5, we analyze two suboptimal policies A and B that, over the same set of instances, yield entirely different average objective values. However, by analyzing the heatmaps of our metrics (calculated based on the optimal policy), we can now explain these drastic results: Policy A suffers from severe underestimation errors that cause high regret in particularly relevant areas of the state space. Overestimation, in turn, also occurs but does not cause any regret. Contrary, policy B only exhibits mild underestimation with substantially lower regret in rather irrelevant states. Further, for policy B, overestimation also causes regret that is even slightly more relevant. Overall, the aggressive acceptance of (early) customers due to underestimation errors by Policy A causes a severe objective value loss. In comparison, the (slightly too) conservative behavior of Policy B leads to a much better performance. Hence, if we were to develop a policy for an i-DMVRP with this setting structure, we conclude that Policy A is missing crucial information. Hence, in the development process, we can now, e.g., integrate algorithmic elements to tackle the systematic underestimation by Policy A, or draw the conclusion to focus more development effort on Policy B due to its structural advantages.

Aggregated metric (4) — To visualize and interpret the fourth metric, i.e., the weighted error ratio, we propose a scatter plot, where the result of applying a policy to a certain setting is plotted as a point according to the weighted error ratio on the x-axis and the relative optimality gap on the y-axis (see Fig. 6). This visualization enables us to analyze in which settings a policy is prone to either underestimation or overestimation in combination with the resulting performance impact. The example given in Fig. 6 shows a policy that is mainly affected by underestimation errors in Setting 1 and Setting 2, with a more severe impact in Setting 1. Setting 3 is equally affected by overestimation and underestimation, whereas Setting 4 is mainly affected by overestimation.

Analysis of other performance indicators

Up to this point, we define B1 such that it analyzes an error's impact on the objective value, which is typically profit in i-DVMRPs. However, it depends on the specific definition of the objective function. We now

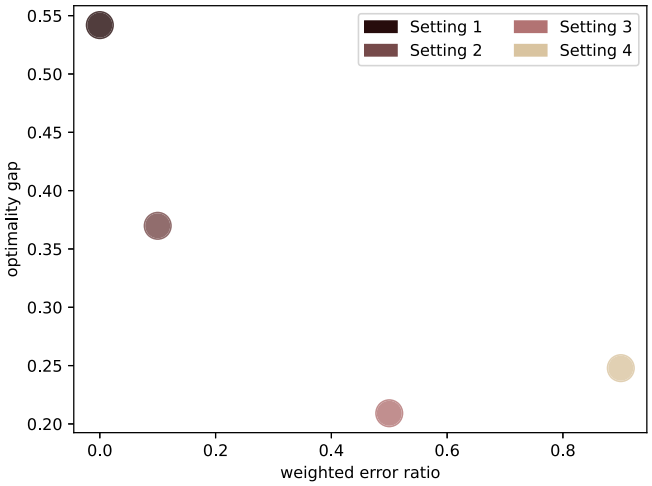


Fig. 6. Weighted error ratio — exemplary for four settings.

explain how B1 can be extended to also yield insights about the impact on any other performance indicator, since in practice, system performance in logistics and transportation is inherently multi-dimensional and also covers indicators like service level or the number of served orders/passengers.

To achieve this, the single decision regret can be additionally calculated for any performance indicator, irrespective of whether it is part of the objective (here revenue and cost) or not (e.g., number of orders, number of rejected customer requests, required fulfillment times, number of delays, workload distribution among drivers, etc.). From visualizations of these regret values (Fig. 5), we can assess whether erroneous decisions regarding the objective are also detrimental to these other indicators or even cause improvements. Moreover, these additional regret values can also be used to compute the associated weighted error ratio. Visualized as in Fig. 6, it reveals whether performance differences in terms of the additional indicators compared to the optimal policy are driven by underestimation or overestimation. E.g., a dominance of underestimation errors might lead to many suboptimal acceptance decisions negatively impacting profit but also causing an increase in the number of orders.

4.2. Building block 2

To apply the previously presented technique for gaining general insights on i-DMVRPs within our computational study, we now rely on a second explainability technique, namely on reward decomposition (Juozapaitis et al., 2019). More precisely, although B1 can be applied to *any* OC approximation in comparison to the optimal policy and already yields valuable insights, we apply it to *specific* OC approximations that base on the idea of reward decomposition. In the following, we first motivate the use of these approximations before we mathematically define the resulting policies and then explain how they can be computed.

Motivation

To design the policies we analyze in Section 5, we draw on the results of Fleckenstein et al. (2025). There, it is shown that the OC of an i-DMVRP can be decomposed into two components: DPC, which captures the loss of future revenue, i.e., a decision's impact on the positive rewards (e.g., delivery fees paid by accepted customers, revenues from selling shopping baskets, number of additionally accepted customers, or similar, depending on the objective function), and MCTS, which measures the respective increase of fulfillment cost, i.e., the impact on the negative rewards. Based on this finding, we derive and analyze OC approximations that only capture DPC or MCTS, i.e., completely neglect the other component. The pure DPC approximation is used by the DPC policy. Analogously, we design an MCTS policy that approximates OC purely based on MCTS.

These stylized policies are an effective tool for analyzing the entire space of (heuristic) policies tackling i-DVMRPs for the following reasons: First, single-component policies, which also target only one of both OC components, are common in both i-DVMRP research and practice. Most myopic policies only capture MCTS by design because approximating DPC would require at least some kind of information on future demand. A popular example are policies that use the insertion cost into a (tentative) route plan as an MCTS approximation (e.g., Campbell and Savelsbergh, 2006 or Atasoy et al., 2015). There are also anticipatory policies that use the anticipatory information about future requests only for refining the MCTS approximation and not for approximating DPC (e.g., Anzenhofer et al., 2025 or Azi et al., 2012).

Second, recent theoretical (Fleckenstein et al., 2025) and practical results (Abdollahi et al., 2023 or Anzenhofer et al., 2024) suggest that policies exploiting the decomposition into DPC and MCTS have a high potential. With such an approach, the policy can be intentionally tailored to precisely approximate one component while only coarsely approximating the other component, depending on their respective relevance in the considered setting.

Third, analyzing the DPC policy and the MCTS policy is valuable even with respect to policies that are based on a single-state value approximation including both components, without explicit decomposition. The reason is that certain policy design choices, e.g., feature definition or model specification in policies based on statistical learning, lead to the two components being approximated with a different accuracy implicitly. As an example, routing information can be integrated into a policy using various types of approximations, ranging from analytical formulae to skeletal route plans, with different accuracies (Klein et al., 2018; Koch and Klein, 2020; Vinsensius et al., 2020, or Yang and Strauss, 2017).

Based on these three observations, how precisely any policy approximates DPC and MCTS can be cast as a spectrum, ranging from entirely neglecting one component to a very precise approximation of both. Hence, separately investigating the extreme point of each component's spectrum, i.e., a policy that completely neglects one component, clearly reveals the fundamental types of errors caused by an inadequate approximation regarding this component.

Policy definition

As shown in Fleckenstein et al. (2025), OC, $V_t(s_{t-1}; c)$, can be decomposed into DPC, formally denoted as $R_t(s_{t-1}; c)$, and MCTS, formally denoted as $F_t(s_{t-1}; c)$. For the formal definition of both components, we first define the expected future revenue of a given interim state s'_{t-1} at decision epoch $t-1$ as $R'_{t-1}(s'_{t-1})$ and the expected future fulfillment cost of a given interim state s'_{t-1} at decision epoch $t-1$ as $F'_{t-1}(s'_{t-1})$. Then, we can define DPC and MCTS as follows:

$$R_t(s_{t-1}; c) = R'_t(s'_t(0)) - R'_t(s'_t(c)) \tag{15}$$

and

$$F_t(s_{t-1}; c) = F'_t(s'_t(0)) - F'_t(s'_t(c)); \tag{16}$$

and $V_t(s_{t-1}; c) = R_t(s_{t-1}; c) + F_t(s_{t-1}; c)$ holds.

The DPC-based approximation $\hat{R}_t(s_{t-1}; c)$ results from:

$$\hat{R}_{t-1}(s_{t-1}) = \sum_{c \in C} \left(\max_{g_t \in G(s_{t-1}; c)} \left(g_t^0 \, r_c - \hat{R}_t(s_{t-1}; c) \right) + \hat{R}'_t(s'_t(0)) \right) \tag{17}$$

Since the DPC-based approximation neglects fulfillment cost, i.e., $r_{t(s'_t)} = 0 \quad \forall t = 1; \ldots; T$, the revenue is the only type of reward, and the future cost impact of an acceptance decision is ignored. Solving Eq. (17), we obtain the DPC policy π^R, the DPC-based decision $g_t^R(s_{t-1}; c) = \arg\max_{g_t \in G(s_{t-1}; c)} \left(g_t^0 \, r_c - \hat{R}_t(s_{t-1}; c) \right)$ being in state s_{t-1} and observing the arrival of a request of type c, and the DPC-based objective value $J^{\pi^R} = \hat{R}_0(s_0)$.

The MCTS-based approximation $\tilde{H}_t(s_{t-1}; c)$ results from:

$$\tilde{H}_{t-1}(s_{t-1}) = \sum_{c \in C} \left(\max_{g_t \in G(s_{t-1};c)} \left(g_t \cdot r_c - \tilde{H}_t(s_{t-1}; c) \right) - \tilde{H}_t(s_{t-1}; c) + \tilde{H}'_t(s'_t(0)) \right). \quad (18)$$

Since the MCTS-based approximation neglects displacement cost, formulating the Bellman equation of this approximation requires an argmax( ) operator. This prevents the revenue from being included in the state value while still comparing it to the future cost impact for deciding on each single request's acceptance. Then, the result of the argmax( ) operator, which encodes the binary demand control decision, is multiplied with the future cost impact, i.e., the MCTS. Solving Eq. (18), we obtain the MCTS policy $\pi^{\tilde{H}}$, the MCTS-based decision $g_t^{\pi^{\tilde{H}}}(s_{t-1}; c) = \arg\max_{g_t \in G(s_{t-1};c)} \left(g_t \cdot r_c - \tilde{H}_t(s_{t-1}; c) \right)$ being in state s_{t-1} and observing the arrival of a request of type c, and the MCTS-based objective value $J^{\pi^{\tilde{H}}} = \tilde{H}_0(s_0)$.

Policy computation

Calculating the metrics of building block B1 (Section 4.1) as defined in Eqs. (9) to (14) requires the computation of the optimal policy. For this purpose, we can readily draw on Bellman equation (4), which formulates the problem as a dynamic program. Hence, we apply the standard backward recursion algorithm for solving dynamic programs to optimality (see, e.g., Bertsekas, 2012 for an in-depth explanation). For determining the final route plan, i.e., $\pi^*_T(g_T)$, we solve the VRP instance that comprises all orders stored in state s_{T-1} to optimality. Consequently, every demand control decision that is based on the i-DMVRP-specific OC definition implicitly optimizes the underlying VRP such that an optimal demand control decision integrates an optimal routing decision.

For computing the DPC policy and the MCTS policy, we use the same algorithm to optimally solve Bellman equation (17) and Bellman equation (18), respectively. With this approach, we avoid random errors that would be caused by the inconsistent solution quality of any heuristic algorithm for policy computation. This is an important feature of our explainability technique because it ensures that the errors observed in the results are purely systematic errors that can be attributed to neglecting one of the OC components.

5. Computational study

We now apply the explainability technique consisting of B1 and B2 to a generic i-DMVRP as introduced in Section 3. The aim of this numerical analysis is to identify and characterize fundamental types of approximation errors that can occur in any real-world i-DMVRP. In Section 5.1, we first present the experimental design, the tested parameter settings, and the benchmarks that we consider next to the already introduced DPC-based and MCTS-based approximations. Then, in Section 5.2, we characterize the observed types of approximation errors. Finally, we investigate the impact of these types of errors on the objective value in Section 5.3 to further refine the characterization.

5.1. Experimental design

Overall, we draw on the full-factorial study design proposed in Fleckenstein et al. (2025), which consists of 66 different settings of an i-DMVRP with disjoint booking horizon and service horizon and pure accept/reject decisions. The integrated VRP is a distance-constrained, capacitated VRP in all settings. Further, all settings have in common that we assume a single fulfillment vehicle and a booking horizon of $T = 10$ potential decision epochs with 10 potentially arriving customer requests such that at most one customer request arrives per decision epoch.

The settings differ in the following parameters: *location distribution*, *revenue distribution*, general *profitability* of a setting, and binding *capacity constraints*. The former two parameters are customer-related, the latter two are provider-related. In detail, the following parameter values are possible:

Location distribution – Regarding the customers' location distribution, we generally consider a one-dimensional line segment. This simplification is commonly used in literature analyzing problem elements other than the vehicle routing decisions (e.g., Yildiz and Savelsbergh, 2020) to prevent any distortion by suboptimal vehicle routing decisions. For our study of opportunity cost approximation, this is particularly important. We consider two realizations: A stream of customer request locations l_c with $c = 1; \ldots; 10$ are either drawn from (1) a uniform distribution over a line segment with length 50 in the interval $[-25; 25]$ representing a single urban area, or (2) in random order from two truncated normal distributions with means -10 and 20 and the same standard deviation of 2.5 to generate two equal-sized clusters representing, e.g., two villages in a rural area. The first location distribution is referred to as *unif* for uniform distribution. The second location distribution is referred to as *clust* since there is two clusters of customers.

Revenue distribution – Regarding the customers' revenue distribution, we consider four realizations: Besides (1) homogeneous customer streams with all revenues equal to 15 monetary units referred to as *homog*, (2) heterogeneous customer streams are generated by randomly assigning a revenue of 25 monetary units to 30% of the customers in a stream of customer requests. The other 70% of the customers are assigned revenues equal to 15 monetary units. In practice, heterogeneous revenues can arise, e.g., due to different shopping basket values in e-commerce applications or differently high delivery fees resulting from a customer subscription model. If heterogeneous customers are considered in a setting, the high-revenue customers can either (2.1) strictly arrive in the beginning (referred to as *h-b-l* for high-before-low), (2.2) randomly (referred to as (*rand*)), or (2.3) strictly in the end (referred to as *l-b-h* for low-before-high) of the booking horizon. Further, for each of the realizations (2.1)–(2.3), an additional clustered setting is considered in which all high-revenue customers are located in the distant cluster. We refer to these settings as *clust_sort*. These settings resemble variable transportation fees depending on the distance, which are common in mobility-on-demand systems. Also, if required (as in h-b-l and l-b-h settings), we sort the drawn customer streams.

These customer-related parameter realizations yield 11 meaningful combinations (referred to as *customer settings*), as depicted as leaf nodes in Figure A.8 in Appendix A. Each of these settings is then considered six times, according to six combinations of the two provider-related parameters' realizations that we explain in the following.

Profitability – To vary the general profitability of the settings, we modify the routing cost factor. More precisely, each customer setting is considered three times with different routing cost factors of 0.2, 0.6, or 1 monetary units per distance unit. We refer to these settings by addressing the profitability as *high*, *med*, or *low*. High-profitability settings are common in business models in which customers purchase goods (e.g., groceries) or services (e.g., installation services) coupled with the logistics service. In contrast, low-profitability settings occur, e.g., in publicly funded mobility-on-demand services with comparatively low fees paid by the customers.

Capacity constraints – Regarding the logistics capacity constraints, we assume two different realizations. We either limit the route length to 50 length units and refer to these settings as *distance-constrained* (*dist*), or we limit the physical capacity of the fulfillment vehicle to 3 units. In the latter case, we assume unit demand for all customers and refer to the respective settings as *load-constrained* (*load*). The distance-constrained settings mimic situations in which the available vehicle hours are the binding constraint and the physical capacity of vehicles is rarely exhausted. This is a typical assumption, e.g., in same-day delivery. Conversely, the load-constrained settings represent applications in which physical vehicle capacity is more scarce, e.g., attended home delivery of bulky goods.

For each of the resulting 66 settings, we consider 50 different instances in our computational study. Each instance is defined by an individual customer stream of 10 customers, sampled according to the above-mentioned customer-related parameter values in advance. Of these customer streams, each customer then places a request with probability $^{t}_{c} = 0{:}5$ if $c = t$ and $^{t}_{c} = 0$, otherwise. This is the only source of stochasticity once an instance is fully specified. We apply all considered policies (see next paragraph) to all instances, calculate the metrics introduced in Section 4.1, and average the respective results over all 50 instances per setting to derive the setting-specific results we report.

Note, the choice of such a basic problem is deliberate, and typically done in the literature, to obtain findings that are valid for a broad variety of real-world problems (e.g., Ulmer and Thomas, 2020).

Considered policies — In the computational study, we apply the following policies:

(1.) DPC policy: With this policy, we analyze the errors resulting from an insufficient consideration of MCTS. For its computation, we optimally solve Bellman equation (17) for the entire state space using backward recursion.

(2.) MCTS policy: This policy allows analyzing errors due to an insufficient consideration of DPC. It is computed by optimally solving Bellman equation (18) for the entire state space using backward recursion.

(3.) Optimal policy: Access to the optimal policy is an essential prerequisite for our explainability technique because it is the basis for calculating the metrics that quantify the influencing factors introduced in Section 4.1 (see Eqs. (9) to (14)). The optimal policy can be computed by solving Bellman equation (4) for the entire state space using backward recursion.

(4.) Myopic policy: This policy, we consider additionally due to its high relevance as a benchmark policy in the i-DMVRP literature (e.g., Arian et al., 2022; Klein and Steinhardt, 2023, or Yang et al., 2016). Instead of a state value difference, the respective OC approximation is defined as the insertion cost into the current myopic route plan, i.e., a route plan based on only confirmed customer orders.

The objective values of all policies are depicted in Figure B.9 in Appendix B.

5.2. Identification of fundamental approximation errors

In the following, we identify and characterize the fundamental types of approximation errors that we observe for the DPC policy and the MCTS policy. This analysis is based on the heatmaps introduced in Section 4.1. Since it is not possible to include the full set of heatmaps for all settings in the paper at hand, we provide two types of supplementary material for the interested reader. First, we present carefully selected heatmaps in Appendix C. These are intended to be used as "textbook" examples that show a certain error and its characteristics particularly clearly. Second, to allow for full reproducibility, the complete set of heatmaps can be viewed or downloaded at zenodo.org/full-set-of-heatmaps.

When discussing an error type's characteristics in the following, especially regarding the influence of setting parameters, we only mention those error types that are reasonably pronounced and occur over many different settings. Thereby, we ensure generalizability beyond the basic settings we consider. Then, we explain the prevalence of these error types in the different settings as well as their interplay. Afterward, in Section 6, we discuss numerical results from recent literature and show that our findings generalize to larger settings in richer i-DMVRPs.

DPC policy

The DPC policy lacks information on how well a request can be consolidated with other orders to a profitable fulfillment tour. The reason is that this information is encoded in the MCTS: The tour becoming only slightly longer (and costlier) indicates good consolidation (low MCTS). If marginal cost is high, the order request cannot be consolidated well. However, the DPC policy can only observe this information indirectly if it is reflected in the request's capacity consumption in the underlying VRP, and thus, affects the displacement of future revenue. The mechanism behind this works as follows: In the distance-constrained setting, a high marginal increase of the tour length not only means high MCTS but also a high consumption of the scarce logistical resource, which is the distance traveled by the vehicle. Hence, accepting the current request likely causes the displacement of several future requests, i.e., a large amount of revenue. Apart from this indirect information gain, the policy assumes perfect consolidation of orders, i.e., MCTS of zero for each request. In the following, we introduce the three distinctive types of approximation errors resulting from this DPC policy. More precisely, we discuss two types of underestimation errors and one type of overestimation error, that occur when applying the DPC policy.

Underestimation error type 1: neglecting better consolidation of future requests

This error occurs when the acceptance of the current request causes the displacement of expected future requests but not the displacement of expected future revenue. To explain why such a situation can occur, we first make the subtle but important distinction between request displacement and revenue displacement. The former means that, given the limited capacity, one or more future requests cannot be accepted due to accepting the currently arriving request. The latter refers to the future revenue that is lost in total due to the acceptance of a request now compared to a rejection. The relationship between both is as follows: Request displacement can, but not necessarily does, cause revenue displacement. The situation in which it does not occurs when the displaced future requests have a *similar revenue* as the currently arriving request. Therefore, even if there is request displacement, the expected revenue displacement is most likely below the current request's revenue and not enough to cause a rejection.

However, the rejection would still be optimal if future requests can be consolidated better with other future requests or orders already received, i.e., are less costly to serve. The DPC policy cannot access this information since it is encoded by the MCTS, which is why the underestimation error leads to a wrong acceptance decision. In the case that future requests have a higher revenue than the current request, as especially visible in our *l-b-h* settings, the DPC alone already exceed the current request's revenue even without additionally considering the MCTS. This reduces the error's regret since the DPC are sufficiently high to reject requests, which is the optimal decision (compare Figure C.10a and Figure C.10b).

The regret of the DPC policy also decreases with greater correlation between consolidation and capacity consumption in the underlying VRP, as observable in our *dist* constrained settings (compare Figure C.10b and Figure C.10c). The root cause of this is that DPC in this setting indirectly encode information about the request's consolidation potential to a profitable fulfillment tour through the mechanism explained above. In general, this type of underestimation error becomes smaller over time as there are less opportunities for collecting orders that allow better consolidation (see Figure C.10a). Further, the error occurs in a broad region of the state space with a relatively high decision rate (see Figure C.10).

Underestimation error type 2: neglecting inherent request unprofitability

This error mainly occurs when the current request is neither profitable based on consolidation with the orders already received nor is it expected to become profitable taking future consolidation opportunities into account. Then, the request's marginal increase in routing cost is so large that MCTS cannot be below the revenue, and the optimal decision is a rejection. The DPC policy, without access to this information, erroneously accepts the request. The error becomes stronger over time with shrinking future consolidation opportunities and tends to occur in states with low capacity consumption where not much consolidation

is already established (see Figure C.10a). The associated regret is only dampened when high-revenue requests arrive in the affected states as especially observable in our *l-b-h* settings (see Figure C.10b). This is because the optimal decision for these high-revenue requests is the acceptance and therefore the underestimation error cannot cause a wrong decision. Compared to underestimation error type 1, considering the decision rate reveals that this error type has much less impact because the region of late states with low capacity consumption is hardly visited in any of our settings (see Figure C.10a–C.10c). However, if revenues reflect the distance from the depot as in our *clust_sort* settings, the error can occur earlier, i.e., in states with a higher decision rate (see Figure C.10d).

In extreme cases, the underestimation due to these two types of errors can become large enough that an acceptance decision is made in every state. Then, the policy behaves exactly like a first-come-first-served policy.

Overestimation error: protecting capacity for high-revenue demand with poor consolidation

The missing information on consolidation opportunities can also lead to an OC overestimation. This error occurs when the policy anticipates demand that has indeed higher revenue but cannot be consolidated as well as the current request. In such a state, it is better to accept the current customer request despite its lower revenue than to reserve capacity for the higher-revenue demand that is less profitable in terms of revenue net of routing cost due to poor consolidation. If the high-revenue demand arrives early (setting *h-b-l*), DPC are low enough such that no capacity is reserved and the DPC policy makes the correct acceptance decision despite the overestimation error. Consequently, a heterogeneous revenue distribution that is not strictly *h-b-l* is prone to this error also causing regret (compare Figure C.11a and Figure C.11b). We can locate the error in all states early in the booking horizon but it only leads to regret in states in which displacement effects are sufficiently strong, i.e., states with high capacity consumption in case of *load*-constrained settings or states with low capacity consumption in case of *dist*-constrained settings (compare Figure C.11a and Figure C.11c). It remains constant over time until the high-revenue demand is expected to realize. Its magnitude is proportional to the mismatch between revenue and consolidation opportunities offered by the demand, or in other words, between revenue displacement observed by the policy and the actual profit displacement. A particularly strong occurrence can be observed in *low*, *clust_sort* settings in which the high-revenue requests also cause the highest routing cost (see Figure C.11d). In the region of the state space, in which this error type occurs, the decision rate is comparatively high (see Figure C.11).

MCTS policy

The MCTS policy has information about all parameters of expected future requests including their revenue. However, it only uses the information about the revenue when making a decision on an individual request, as revenues generally do not enter its value function. Thus, the policy can only anticipate the routing cost of future customers but not the associated cumulative revenue. For this policy, we find three distinctive estimation errors.

Underestimation error type 1: neglecting future high-revenue demand

Since the MCTS policy cannot observe heterogeneity in revenue, it fails to reserve capacity if there is more profitable, high-revenue demand arriving in a later phase of the booking process as especially observable in our *l-b-h* settings. The resulting regret due to earning less revenue per order is roughly proportional to the magnitude of the underestimation error. The reason is that the amount of logistics capacity that should be reserved for late high-revenue demand is also proportional to the revenue of this demand that would be displaced. The error becomes stronger the more capacity is consumed and the earlier a state is (see Figure C.12a). This is because logistics capacity becomes more valuable the scarcer it is and the more time there is left to find a highly profitable customer to allocate it to. The error magnitude is insensitive toward the profitability indicating that relation between the revenue of early and late requests matters and not the relation between revenue and cost (compare Figure C.12a and Figure C.12b).

Underestimation error type 2: missing information on the volume of future demand

Even if the revenues are homogeneous, underestimation occurs when applying the MCTS policy since its OC approximations do not fully reflect the volume of demand to come. This, however, is an important piece of information because it reflects how many chances there are left for finding a "better" request than the current one. Hence, early in the booking process the optimal decision generally tends to be a rejection while late in the booking process, the acceptance tends to be the optimal decision. In earlier states, this information can only be derived from the expected revenue to come, and thus, the DPC. Only in the end of the booking horizon, the low volume of demand to come impacts the MCTS policy's OC approximation via the decreasing number of consolidation opportunities it notices, which are reflected in an MCTS increase.

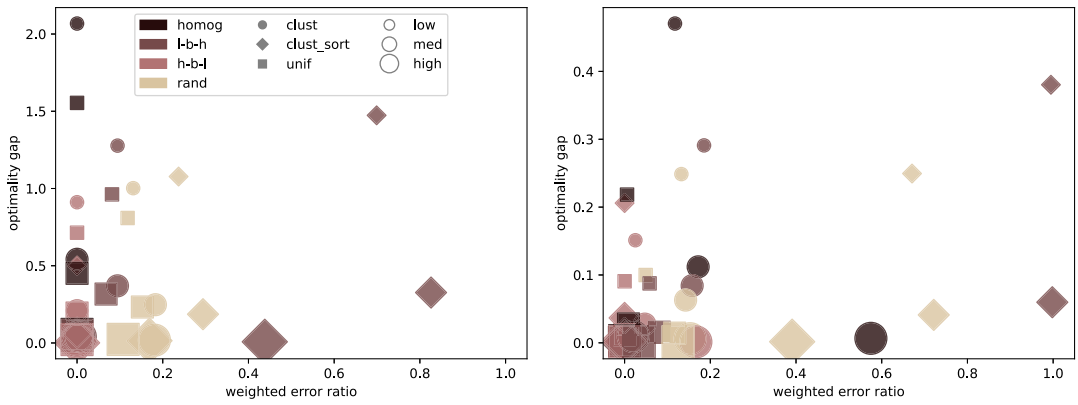
The consequence of this type of underestimation error is that the policy does not protect capacity for requests offering better consolidation, i.e., are less costly to serve. This can either lead to less efficient routing, which reduces profitability on the cost side, or it may also reduce the number of accepted orders due to the higher capacity consumption, which affects the revenue side. The error and the associated regret occur predominantly in early states (see Figure C.12c), which can be directly explained by the discussion of the error mechanism above. As for underestimation error type 1, the error magnitude does not change with profitability (compare Figure C.12c and Figure C.12d).

Both types of MCTS underestimation errors are usually superimposed on each other. Similar to the type 1 underestimation error of the DPC policy, they occur in a fairly large region of the state space with a high decision rate. In the worst case, the combination of both errors leads to solutions with less accepted orders, less revenue per order, and an inefficient routing.

Overestimation error: wrong cost attribution

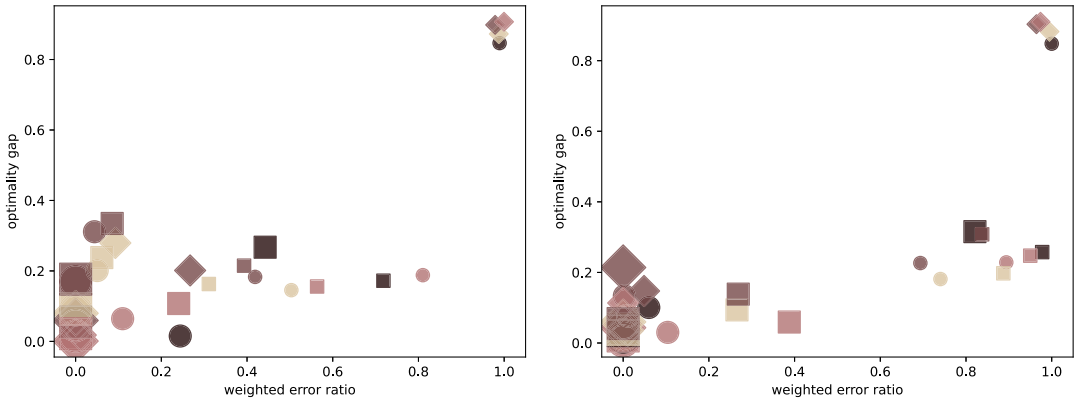
The missing information on future revenue can also lead to overestimation errors by the MCTS policy. This is caused by the following mechanism: If none or only very few orders are confirmed already, accepting a request momentarily entails a high increase in routing cost compared to rejecting it. If the instance is sufficiently profitable, anticipating optimal future decision-making would reveal the following: Since further requests will be accepted, consolidation to profitable fulfillment tours is possible and the increase in routing cost is shared among all these requests, with the MCTS of each request being much lower than the initial cost for opening the tour with only one customer. This means that the final cost difference between accepting the current request and rejecting it would not be as high as it appears in the moment of the acceptance decision. The MCTS policy, however, anticipates that in the reject-case, no or only few additional requests would be accepted, again because of their momentarily high increase in routing cost compared to a rejection. In summary, the policy wrongly attributes high routing cost to early arriving requests. For the policy, it appears as if these early requests served in the tour are responsible for the high cost increase, which creates a self-fulfilling prophecy recursively causing erroneous rejections.

The error and the resulting regret become more severe with lower profitability (compare Figure C.13a and Figure C.13b or Figure C.13c and Figure C.13d), and its recursive character makes this error particularly harmful. At worst, it can lead to the policy stalling in states with no or few orders, i.e., the tour is never really opened. This becomes apparent from the plots showing the decision rate (see Figure C.13b and Figure C.13d). The reason is that suboptimal rejections due to overestimation lead to successor states in which the overestimation error is likely to occur again. Thus, the error occurs mainly in states with no or a very short tentative route and becomes more severe with



(a1) *load* (a2) *dist*

(a) DPC policy



(b1) *load* (b2) *dist*

(b) MCTS policy

Fig. 7. Weighted error ratio and average optimality gap.

decreasing profitability. By contrast, error and regret are reduced by heterogeneous revenues. In addition, the region of high decision rates shifts away from the states with low capacity consumption, where the error occurs, which indicates that the self-fulfilling prophecy is broken early in the booking process. Responsible for this are high-revenue requests that are accepted despite the overestimation (compare Figure C.14a and Figure C.14b). Here, the buffer introduced in Fig. 4 absorbs the error for some requests. This has a positive knock-on effect because these orders then serve as seed customers to establish consolidation independent from their distribution over time. Similarly, a uniform distribution of locations (*unif*) can reduce the error due to orders located close to the depot serving as seed customers (compare Figure C.14a and Figure C.14c)

5.3. Resulting performance impact

Using the weighted error ratio and its visualization introduced in Section 4.1, we now analyze the contribution of underestimation errors and overestimation errors to the losses in objective value relative to the optimal solution. Again, we consider the DPC policy and the MCTS policy in the 66 different settings. The results are depicted in Fig. 7. As discussed in the following, there are some clearly observable patterns between the optimality gap, the weighted error ratio, and certain setting parameter values. From these patterns, we can derive insights on how location distribution, revenue distribution, and profitability cause underestimation errors or overestimation errors when applying a given policy and how this affects the optimality gap. However, depending on the policy applied, there may hardly be any patterns, as we observe for our MCTS policy. This shows that there can be complex interactions between underestimation and overestimation errors, e.g., both can offset. In the following, we further elaborate on that for each policy individually.

DPC policy

When applying the DPC policy, the average optimality gap generally decreases with increasing profitability, but the weighted error ratio stays almost constant with only a slight shift toward underestimation. The type of the binding capacity constraints also has an influence on performance since the average optimality gap is smaller for *dist*-constrained settings, and there are more observations of settings in which the optimality gap results mainly from overestimation. Both scenario parameters combined, the observations reach from an average optimality gap below 1% in *high*-profitability, *dist*-constrained settings, to even negative objective values in some *low*-profitability, *load*-constrained settings.

Generally, in *clust* settings, the average optimality gap tends to be larger than in *unif* settings. Since we only observe a slight shift toward overestimation, this is mainly caused by more severe regret from underestimation errors. If the requests' revenues are proportional to their distance from the depot, i.e., in *clust_sort* settings, the share of overestimation increases but without a clear impact on the average optimality gap. For *med*-profitability settings, the gap becomes smaller but it increases for *low*-profitability settings.

Regarding the revenue distribution, we find the highest average optimality gap for our *homog* settings and the smallest for *h-b-l* settings. In both types of settings, there is no overestimation in case of *load*-constrained settings and only a small impact of overestimation in

dist-constrained settings. In case of heterogeneous revenues, i.e., in our *h-b-l*, *l-b-h*, or *rand* settings, the average optimality gap and the share of overestimation decrease the earlier the high revenue arrives, as especially observable in our *h-b-l* settings.

MCTS policy

For the MCTS policy, the patterns are less clear. In *high*-profitability settings, underestimation is by far the dominant error. Due to the dominating underestimation error, the average optimality gap is higher the more high-revenue demand arrives late, i.e., in *l-b-h* settings, as well as in *load*-constrained settings, where less consolidation is possible. Despite the error ratio shifting toward overestimation with the setting becoming less profitable, this has no coherent impact on the average optimality gap because of the complex interactions of underestimation errors and overestimation errors. E.g., we observe average optimality gaps increasing consistently with decreasing profitability in *h-b-l* settings that are *dist*-constrained with *unif* or *clust* locations, and *load*-constrained, *clust_sort* settings. In *load*-constrained settings with *unif* or *clust* location distribution, however, the average optimality gap decreases between *med*-profitability and *low*-profitability settings due to offsetting errors for *rand* and *l-b-h* settings. Another example for these inconsistencies are settings with *homog* revenues. Here, the average optimality gap is much higher for *unif* settings compared to *clust* settings, when considering *med* profitability. In *low*-profitability settings, we observe an inverse relation.

Regarding the location distribution, the general findings are that *unif* settings tend to have a higher average optimality gap compared to *clust* settings and that *low*-profitability, *clust_sort* settings are prone to overestimation.

Considering the distribution of revenues, we find that *homog* or *l-b-h* settings tend to have a higher average optimality gap.

6. Insights for algorithm selection and algorithm design

In this section, we formulate six actionable insights based on our findings that support developers and users of OC approximation approaches in selecting and designing algorithmic components for specific i-DMVRPs. By closely incorporating the key computational results of the existing literature, we now show that our findings explain much of the performance differences observed in the literature. Therewith, we also compile evidence for that our results derived from the numerical analysis of a generic i-DMVRP are fundamental and generalize to the entire family of i-DMVRPs.

Opportunity costs approximation errors can be grouped into a few distinct types: Neglecting one of the OC components, DPC or the i-DMVRP specific MCTS, causes systematic errors that can be grouped into a small number of fundamental types consistently observable across a variety of settings. Furthermore, we find many patterns regarding their occurrence in certain regions of the state space and in different settings. While some of those patterns were already suspected in existing literature based on the decision-making behavior of the respective solution approaches, we are the first to provide direct, numerical evidence to characterize these patterns and are also able to identify entirely novel ones. E.g., the existence of underestimation errors when applying an MCTS policy has been discussed but not conclusively proven by Mackert (2019) and Yang and Strauss (2017). Overestimation, however, has only been briefly mentioned by one author (Mackert, 2019). Overall, our main finding is that neglecting either component can lead to underestimation errors as known from traditional revenue management applications (e.g. airline or car rental revenue management) but, more importantly, can also lead to overestimation errors. This distinguishes the discussion of OC in i-DMVRPs from existing discussions of OC in traditional revenue management, as it is a new observation that is specific to i-DMVRPs and results from the integration of the underlying VRP.

As we show in the work at hand, both OC underestimation and OC overestimation are systematic in the sense that they occur even if an exact algorithm, i.e., applying backwards recursion to the (modified) Bellman equations (4), (17), and (18), is used to compute the respective OC estimate.

Underestimation is the dominant error when neglecting DPC or MCTS: The fundamental issue with neglecting DPC or MCTS is that the resulting approximation does not reach the correct absolute level of the true OC since the contribution of the neglected component is missing. Hence, underestimation is by far more common than overestimation (see Fig. 7): For the DPC policy, underestimation (weighted error ratio $<0{:}5$) is dominant in 89.4% of all settings. For the MCTS policy, this occurs in 69.7% of the settings. This dominance of underestimation leads to greedy decision-making overall. Many authors report such greedy behavior in studies with realistic-sized instances, which shows the general validity of this result. This accounts for both availability control (e.g., Arian et al., 2022; Lang et al., 2021; Mackert, 2019, and Campbell and Savelsbergh, 2005) in i-DMVRPs, and dynamic pricing in i-DMVRPs (e.g., Abdollahi et al., 2023; Klein and Steinhardt, 2023; Klein et al., 2018; Yang and Strauss, 2017; Yang et al., 2016). In the former case, the policy accepts too many customers or offers too many, or even all of the feasible fulfillment options to each requesting customer resulting in a first-come-first-served decision-making behavior. In the latter case, the average price level is too low and can drop markedly during the booking horizon. In MOD applications, this observation is also known as the "wild goose chase" (Castillo et al., 2025). In accordance with our findings, which show the potential severity of the underestimation errors, some works addressing i-DMVRPs find that the resulting greedy behavior can lead to a worse performance than static pricing (Abdollahi et al., 2023; Yang and Strauss, 2017).

Due to the absence of research on pure DPC policies, there are no direct indications for respective underestimation errors in the literature. However, there is an indirect observation that can now be better explained in the light of our findings: Koch and Klein (2020) observe that average OC increase quite strongly at the end of the booking horizon, an effect also reported by Vinsensius et al. (2020). This can be attributed to only unpopular fulfillment options remaining available, for which only few orders are collected already. Thus, collecting an additional order for such an option incurs high MCTS. If MCTS is not adequately captured by the OC approximation, this results in severe underestimation errors.

Another important observation from Fig. 7 is that neglecting either MCTS or DPC tends to cause a higher performance loss in *load*-constrained settings than in *dist*-constrained settings. This demonstrates a link of the two components in some settings by capturing the consumption of logistics capacity associated with an order in both components. This attributes to settings in which high capacity consumption causes both strong displacement effects and a high marginal increase of fulfillment cost, i.e., DPC and MCTS will both be high. This means that an approximation based on one of the components can still correctly determine the relative ordering of the OC across different requests or different fulfillment options for one request. However, if the absolute value of the OC approximations is too low due to the high relevance of the missing component, this information gain does not come into effect. In Abdollahi et al. (2023), we find further evidence for this mechanism. They report that their MCTS policy performs well in terms of the number of collected orders and the cost per order, which indicates that the policy correctly captures the "right" orders to achieve high consolidation. However, their policy still fails to improve profit due to the general price level being too low.

Regarding the design of solution algorithms, the explicit consideration of both components is the safest way of avoiding structural underestimation. However, this may be associated with a much higher computational effort. Instead, the described findings related to underestimation suggest two low-threshold algorithmic strategies for mitigating its performance loss:

(1.) The OC approximation can be raised to the correct level by adding a rough estimate of the other component. An example for this can be found in Yang et al. (2016). They introduce a cost penalty as an additional rough DPC estimate if an order cannot be feasibly inserted into the sampled route plan (indicating strong displacement effects). The main challenge with such an approach is the correct adjustment of the rough DPC estimate. Klein et al. (2018) benchmark their own approach, which features a more sophisticated DPC approximation, against the approach by Yang et al. (2016), and the results show that its performance does not decline monotonically with a tighter capacity restriction. Based on our results, the reason for this is that the performance depends on how well the penalty is adjusted to result in the correct level of DPC rather than on the general magnitude of displacement effects.

(2.) The demand control approach can be made more robust against the consequences of underestimation rather than tackling the underestimation error itself. Hence, the goal is to reduce the regret resulting from a given underestimation error. E.g., Ulmer (2020) present an example for such an approach. They introduce a basis price that is charged even if the OC approximation would suggest an even lower price. This curtails greedy control behavior resulting from underestimation. With the same intention, Vinsensius et al. (2020) formulate a budget constraint that limits the incentive levels their algorithm can offer to incoming customers.

Neglecting DPC or MCTS can cause severe overestimation errors: In contrast to underestimation, which is presumed to cause performance differences in several existing works, there are only few publications discussing possible overestimation errors. Mackert (2019) observes that using a less accurate approximation of routing cost leads to the policy making less offers on average, which points toward an increase of average OC, i.e., toward a potential overestimation. Vinsensius et al. (2020) observe that their algorithm yields a very high incentive cost which they attribute to an overestimation of OC due to neglecting "the opportunity cost of capacity" (Vinsensius et al., 2020, p. 582), i.e., DPC, without further numerical evaluation of OC approximation errors.

With our computational study, we provide definitive proof that overestimation errors as a consequence of neglecting DPC or MCTS exist. In most settings, overestimation contributes less than underestimation to the optimality gap. However, if overestimation is the dominant error, the optimality gap is particularly large in most cases. This is especially true for the MCTS policy, which can completely stall due to the recursive nature of the type of overestimation error it exhibits. At the same time, this recursiveness is what makes the error manageable by providing some form of anticipatory information on future consolidation opportunities. This explains the success of skeletal or sampled route planning that is applied as a well-performing algorithmic component in several existing publications (e.g., Anzenhofer et al., 2025; Koch and Klein, 2020; Yang et al., 2016). However, none of these publications contains a conclusive explanation of why it is beneficial.

The early phase of the booking process is critical: Computational experiments in the existing literature suggest that demand control in the early phase of the booking process has a critical influence on solution quality (Anzenhofer et al., 2025; Campbell and Savelsbergh, 2006). Our computational results support this finding as we observe that the regret of a suboptimal demand control decision decreases over time for most types of overestimation and underestimation errors. Since the effect is observable independent of the revenue distribution, we can derive that the underlying main reason is not the loss in immediate reward but the transition to a state that has a much lower value than the successor state resulting from the optimal decision.

Anticipation is not an end in itself: The existing literature studying specific i-DMVRPs consistently emphasizes the importance of anticipation for the performance of solution approaches (Fleckenstein et al., 2023). However, our computational study shows that myopic policies can be hard to beat in certain settings (see Figure B.9 in Appendix B, e.g., *med*-profitability or *low*-profitability, *load*-constrained settings). This result becomes even more relevant given that, in the existing literature, anticipatory policies only achieve maximal improvements of a around 10–15% over myopic benchmarks for some specific i-DMVRPs (e.g., Azi et al., 2012; Heitmann et al., 2023; Koch and Klein, 2020), and that these numbers may even be subject to a negative results bias (Fanelli, 2012). From our computational study, we can derive some of the underlying reasons:

(1.) In some settings, an i-DMVRP can be "easy" to solve in general, e.g., *high*-profitability, *h-b-l* settings (see Figure B.9). Highly profitable demand arriving early in the booking process hardly impacts estimation errors but substantially reduces the resulting regret. Therefore, a more accurate OC approximation by an anticipatory policy may not necessarily translate into a reduced regret.

(2.) Myopic policies benefit from the fact that estimation errors can offset. Therefore, higher overestimation errors compared to an anticipatory policy can, counter-intuitively, be beneficial because the underestimation errors are then compensated better (e.g., see Figure C.15).

(3.) In settings with *low* profitability or low consolidation, overestimation, which myopic policies are prone to, is generally less harmful than underestimation because it leads to a more conservative acceptance behavior, or higher offered prices in dynamic pricing.

In summary, we conclude that investing a substantial amount of the scarce computation time at each decision epoch into an anticipatory OC approximation may not be the most efficient approach for certain i-DMVRPs. Instead, a greater performance improvement might be possible by investing it into solving the other complex subproblems of i-DMVRPs, e.g., the demand control subproblem, or the dynamic vehicle routing subproblem.

Algorithms must be aligned with strategic–tactical decisions on system parameters: Our analysis reveals that the suitability of a given OC approximation, and the resulting policy, depend strongly on the structural characteristics of the underlying problem instance. These characteristics are not exogenous but, in turn, shaped by the provider's strategic and tactical decisions: The demand distribution depends on the configuration of the service (e.g., frequency of fulfillment tours), the design of the fulfillment options (e.g., narrow time windows vs. wide ones), and on the definition of the service area that could, e.g., cover an urban area (most likely uniform distribution) or a rural area (most likely demand clusters). Revenue distribution and profitability depend on coupled goods or services sold together with the logistics service and the basic pricing scheme defined by a company's demand management strategy (e.g., reduced delivery fees for higher-valued shopping baskets, distance-based pricing, "early-bird" discounts vs. "last-minute" discounts). Capacity constraints are influenced by fleet composition and fleet size that determine the supply of logistics resources.

Given these interdependencies, algorithm selection and design can also be viewed as a strategic decision (Anzenhofer et al., 2025) that must be aligned with other strategic decisions shaping the operational-planning policy's environment. By enabling the identification of approximation error types that are most critical for a given setting, our analysis contributes to improving this alignment. If the setting of a certain system varies strongly among different service days (weekday vs. weekend) or regions (urban vs. rural), the provider may even use a set of different policies or policy variants, each tailored to a certain setting.

7. Limitations

The work at hand offers various insights into the nature of OC approximation errors and shows how they can be used to explain the performance of existing approximation algorithms and inform the development of new ones. However, there are limitations to our findings, which can be grouped into three areas: generalization to richer i-DMVRPs, prediction of problem setting, and suggestion of problem-specific algorithms. In the remainder, we outline each of these areas and explain our efforts to mitigate them within the scope of this study.

Generalization to richer i-DMVRPs: Due to the stylized nature of our experimental design, the question whether our findings generalize to richer i-DMVRPs arises naturally. The basic argument in favor of our approach is that our reduced definition of problem and instances captures the essential structure of i-DMVPRs in a controlled environment. This ensures that the identified error types occur as clearly as possible and are fundamental in the sense that they can occur in any i-DMVRP. The successful identification of the identified error types in existing computational studies underscores the validity of this argument (see Section 6). However, the reverse does not necessarily hold: Richer i-DMVRPs may involve additional, problem-specific error types that cannot be observed in our stylized experimental design. Our analysis is therefore targeted at isolating the core error types rather than providing a comprehensive taxonomy of all possible error types.

Nevertheless, the experimental design of such a stylized approach must be carefully chosen to limit the risk of missing a fundamental error type. In the following, we justify our design choices along the four main avenues for potential generalizations. However, regarding each of the avenues, it is possible that, despite these efforts, we could not identify all fundamental types of errors:

Larger instances – The instance size to which our explainability technique can be applied is limited by the requirement that the optimal policy must be computable in reasonable time. Due to the exponential growth of the state space with the instance size, the marginal computational burden of increasing the instance size is also quite high. This ultimately makes the consideration of realistic-sized instances impossible. In addition, there is a conflict between investing computation time into larger instances or into more settings and more runs per setting. To maximize the likelihood of reliably capturing fundamental errors, we therefore prioritize the latter in our experimental design.

Overlapping horizons – While we only consider a setting with non-overlapping booking and service horizons, there is a strong theoretical argument for that the findings generalize well to settings with overlapping horizons. As shown in Fleckenstein et al. (2025), it is possible to transform the model for non-overlapping settings into the model we use as a basis for our analyses. The basic idea is to deliberately delay the routing cost such that they only become effective at the final decision epoch.

Richer vehicle routing problems – Due to the wide variety of the underlying routing problems that occur in real-world i-DMVRPs, an exhaustive consideration would not be possible. Hence, we consider a basic one-to-many delivery problem with two types of constraints that mimic scarcity of physical vehicle capacity (load) and driving distance (time) as key logistics resources.

Richer demand control problems – We study a simple accept/reject problem for two reasons: First, precisely due to the demand control problem being so simple, we can obtain isolated insights on the performance impact of the OC approximation. In a more complex setting with multiple fulfillment options and customer choice, this kind of diagnostic clarity would be hard to obtain due to possible interactions between the demand control problem and the OC approximation. Second, the single-option setting is structurally representative of the core trade-off inherent in any more complex i-DMVRP (Section 3). More precisely, multi-option problems can be cast as a sequence of (albeit interdependent) accept/reject decisions on whether each fulfillment option should be offered based on its revenue net of its OC.

Prediction of problem setting: To meaningfully apply our findings to a specific i-DMVRP in order to guide the design of an OC approximation algorithm, it is necessary to predict key structural parameters in advance (location distribution, revenue distribution, and profitability). While precise prediction is inherently difficult – especially in early-stage or start-up scenarios – we argue that a reasonably accurate prediction is both sufficient and possible in most practical situations. The reason is that the parameters can all be predicted based on demand data and data on fulfillment cost. In the best case, the provider introduces an active demand management policy when the system is already running, allowing for direct use of historical order data and cost data (e.g., Anzenhofer and Fleckenstein, 2024). Even in the absence of historical data, suitable proxies are often available: for instance, brick-and-mortar grocery sales can be used if an attended home delivery service is investigated. Scheduled public transport data may serve as a reference for a newly introduced MOD service. Moreover, sociodemographic data, such as population density, can be used for a coarse prediction of demand density and consolidation potential.

Suggestion of problem-specific algorithms: Another crucial limitation is that our approach is not intended to prescribe specific solution algorithms for particular i-DMVRPs. Rather, as an explainability technique, it serves as a descriptive diagnostic tool to support the algorithm selection and design process across different stages. Thus, we see our approach as a contribution toward the long-term vision of aligning algorithms with problem characteristics in a systematic and explainable manner.

8. Conclusion and future research opportunities

With this work, we present a novel post-hoc explainability technique for i-DMVRPs as well as the results of its first systematic application. The proposed explainability technique allows quantifying and visualizing the extent of OC approximation errors, the regret associated with suboptimal decisions, and the rate at which such decisions are made. Applying this technique to a generic i-DMVRP and considering many settings with a broad variety of parameter values and their combinations, we derive a comprehensive identification and analysis of fundamental types of approximation errors. Finally, we show that this knowledge can be used to explain performance differences observed in the existing literature and to guide algorithm selection and algorithm design. In the following, we conclude by discussing how developers and users can benefit from our findings:

(1.) Our computational results provide a guideline for how important an accurate consideration of DPC and MCTS is in certain problem settings, and how likely the different fundamental types of errors are to occur. This provides indications for the selection of a basic solution concept.
(2.) The technique presented in Section 4 can be directly applied to small instances of the specific i-DMVRP under consideration to compare variants of the pre-selected solution concept with the optimal policy. Thereby, developers and users can gain detailed insights into which types of estimation errors are most relevant for their problem and make targeted adaptions to their algorithms to tackle them.
(3.) Providing training for users based on our findings can enable them to better diagnose anomalies in the behavior of policies and react accordingly.

Since algorithmic explainability is still an emerging concept that is of high relevance both from a theoretical and a practical perspective, we believe that there is great potential for future research connected to our work:

(1.) It would be interesting to apply post-hoc explainability techniques similar to ours to related problem classes, such as network revenue management problems arising, e.g., in the airline industry (Klein et al., 2020). In particular, also given the promising results of related research (Bravo and Shaposhnik, 2020), we believe that optimal policies for small instances are a valuable source of domain knowledge for problems that cannot be solved to optimality when considering realistic-sized instances.

(2.) Complementary to post-hoc techniques based on OC, we see a similar potential for the development of inherently explainable solution approaches for i-DMVRPs, such as inherently explainable policy function approximations.

(3.) Our numerical results suggest that myopic policies deserve more attention. While they are currently mainly considered for pure benchmarking purposes in comparison to anticipatory approaches, their potential as carefully designed, full-fledged solution approaches should also be investigated more thoroughly.

(4.) Finally, there are other elements of solution approaches besides the OC approximation that involve predictions or heuristics. Deriving insights on the impact of their errors on the resulting overall solution quality is certainly also of interest. One example are estimation errors in discrete choice models used for modeling customer choice behavior in response to an offer by the provider. Another example are heuristic solution algorithms for the demand control problem, whose errors could be analyzed by assuming the true OC are given.

CRediT authorship contribution statement

David Fleckenstein: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation. **Robert Klein:** Writing – review & editing, Supervision. **Vienna Klein:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Claudius Steinhardt:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ejtl.2025.100166.

References

Abdollahi, M., Yang, X., Nasri, M.I., Fairbank, M., 2023. Demand management in time-slotted last-mile delivery via dynamic routing with forecast orders. European J. Oper. Res. 309 (2), 704–718.

Amir, O., Doshi-Velez, F., Sarne, D., 2019. Summarizing agent strategies. Auton. Agents Multi-Agent Syst. 33, 628–644.

Amorim, P., DeHoratius, N., Eng-Larsson, F., Martins, S., 2024. Customer preferences for delivery service attributes in attended home delivery. Manag. Sci. 70 (11), 7559–7578.

Anzenhofer, F., Fleckenstein, D., 2024. Extended Booking Horizons in Rural Shared Mobility-on-Demand Systems: Insights and Implications for Demand Management. Working paper, University of Augsburg.

Anzenhofer, F., Fleckenstein, D., Klein, R., Steinhardt, C., 2024. Sustainable Dynamic Pricing for Rural Shared Mobility-on-Demand Systems. Working paper, University of Augsburg.

Anzenhofer, F., Fleckenstein, D., Klein, R., Steinhardt, C., 2025. Analyzing the impact of demand management in rural shared mobility-on-demand systems. OR Spectrum 1–69.

Arian, E., Bai, X., Chen, X., 2022. Joint Pricing and Routing for a Ride-Sharing Platform in Low-Density Rural Areas. Working paper, University of Illinois.

Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al., 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Inf. Fusion 58, 82–115.

Asdemir, K., Jacob, V.S., Krishnan, R., 2009. Dynamic pricing of multiple home delivery options. European J. Oper. Res. 196 (1), 246–257.

Atasoy, B., Ikeda, T., Song, X., Ben-Akiva, M.E., 2015. The concept and impact analysis of a flexible mobility on demand system. Transp. Res. Part C: Emerg. Technol. 56, 373–392.

Avraham, E., Raviv, T., 2021. The steady-state mobile personnel booking problem. Transp. Res. Part B: Methodol. 154, 266–288.

Azi, N., Gendreau, M., Potvin, J.-Y., 2012. A dynamic vehicle routing problem with multiple delivery routes. Ann. Oper. Res. 199 (1), 103–112.

Bertsekas, D., 2012. Dynamic Programming and Optimal Control: Volume I. Athena scientific.

Biggs, M., Sun, W., Ettl, M., 2021. Model distillation for revenue optimization: Interpretable personalized pricing. In: International Conference on Machine Learning. PMLR, pp. 946–956.

Bravo, F., Shaposhnik, Y., 2020. Mining optimal policies: A pattern recognition approach to model analysis. INFORMS J. Optim. 2 (3), 145–166.

Campbell, A.M., Savelsbergh, M., 2005. Decision support for consumer direct grocery initiatives. Transp. Sci. 39 (3), 313–327.

Campbell, A.M., Savelsbergh, M., 2006. Incentive schemes for attended home delivery services. Transp. Sci. 40 (3), 327–341.

Castillo, J.C., Knoepfle, D., Weyl, E.G., 2025. Matching and pricing in ride hailing: Wild goose chases and how to solve them. Manag. Sci. 71 (5), 4377–4395.

De Bock, K.W., Coussement, K., De Caigny, A., Slowiński, R., Baesens, B., Boute, R.N., Choi, T.-M., Delen, D., Kraus, M., Lessmann, S., et al., 2024. Explainable ai for operational research: A defining framework, methods, applications, and a research agenda. European J. Oper. Res. 317 (2), 249–272.

Fanelli, D., 2012. Negative results are disappearing from most disciplines and countries. Scientometrics 90 (3), 891–904.

Fleckenstein, D., Klein, R., Klein, V., Steinhardt, C., 2025. On the concept of opportunity cost in integrated demand management and vehicle routing. Transportation Science 59 (1), 125–142.

Fleckenstein, D., Klein, R., Steinhardt, C., 2023. Recent advances in integrating demand management and vehicle routing: A methodological review. European J. Oper. Res. 306 (2), 499–518.

Gallego, G., Topaloglu, H., 2019. Revenue Management and Pricing Analytics. Springer.

Goerigk, M., Hartisch, M., 2023. A framework for inherently interpretable optimization models. European J. Oper. Res. 310 (3), 1312–1324.

Heger, J., Klein, R., 2024. Assortment optimization: a systematic literature review. OR Spectrum 46 (4), 1099–1161.

Heitmann, R.-J.O., Soeffker, N., Ulmer, M.W., Mattfeld, D.C., 2023. Combining value function approximation and multiple scenario approach for the effective management of ride-hailing services. EURO J. Transp. Logist. 12, 100104.

Hildebrandt, F.D., Thomas, B.W., Ulmer, M.W., 2023. Opportunities for reinforcement learning in stochastic dynamic vehicle routing. Comput. Oper. Res. 150, 106071.

Juozapaitis, Z., Koul, A., Fern, A., Erwig, M., Doshi-Velez, F., 2019. Explainable reinforcement learning via reward decomposition. In: IJCAI/ECAI Workshop on Explainable Artificial Intelligence.

Klapp, M.A., Erera, A.L., Toriello, A., 2020. Request acceptance in same-day delivery. Transp. Res. Part E: Logist. Transp. Rev. 143, 102083.

Klein, R., Koch, S., Steinhardt, C., Strauss, A.K., 2020. A review of revenue management: Recent generalizations and advances in industry applications. European J. Oper. Res. 284 (2), 397–412.

Klein, R., Mackert, J., Neugebauer, M., Steinhardt, C., 2018. A model-based approximation of opportunity cost for dynamic pricing in attended home delivery. OR Spectrum 40 (4), 969–996.

Klein, V., Steinhardt, C., 2023. Dynamic demand management and online tour planning for same-day delivery. European J. Oper. Res. 307 (2), 860–886.

Koch, S., Klein, R., 2020. Route-based approximate dynamic programming for dynamic pricing in attended home delivery. European J. Oper. Res. 287 (2), 633–652.

Lang, M.A., Cleophas, C., Ehmke, J.F., 2021. Anticipative dynamic slotting for attended home deliveries. Oper. Res. Forum 2, 70.

Lebedev, D., Goulart, P., Margellos, K., 2021. A dynamic programming framework for optimal delivery time slot pricing. European J. Oper. Res. 292 (2), 456–468.

Li, Y., Archetti, C., Ljubić, I., 2025. Emerging optimization problems for distribution in same-day delivery. Comput. Oper. Res. 107105.

Mackert, J., 2019. Choice-based dynamic time slot management in attended home delivery. Comput. Ind. Eng. 129, 333–345.

Mannor, S., Simester, D., Sun, P., Tsitsiklis, J.N., 2007. Bias and variance approximation in value function estimates. Manag. Sci. 53 (2), 308–322.

Milani, S., Topin, N., Veloso, M., Fang, F., 2024. Explainable reinforcement learning: A survey and comparative review. ACM Comput. Surv. 56 (7), 1–36.

Powell, W.B., 2011. Approximate Dynamic Programming: Solving the Curses of Dimensionality. John Wiley & Sons.

Powell, W.B., 2022. Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions. John Wiley & Sons.
Psaraftis, H.N., Wen, M., Kontovas, C.A., 2016. Dynamic vehicle routing problems: Three decades and counting. Networks 67 (1), 3–31.
Puterman, M.L., 2014. Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons.
Russell, S.J., Zimdars, A., 2003. Q-decomposition for reinforcement learning agents. In: Proceedings of the 20th International Conference on Machine Learning. ICML-03, pp. 656–663.
Snoeck, A., Merchán, D., Winkenbach, M., 2020. Revenue management in last-mile delivery: state-of-the-art and future research directions. Transp. Res. Procedia 46, 109–116.
Soeffker, N., Ulmer, M.W., Mattfeld, D.C., 2022. Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. European J. Oper. Res. 298 (3), 801–820.
Strauss, A.K., Klein, R., Steinhardt, C., 2018. A review of choice-based revenue management: Theory and methods. European J. Oper. Res. 271 (2), 375–387.
Talluri, K., Van Ryzin, G., 2004. The Theory and Practice of Revenue Management. Springer.
Torrey, L., Taylor, M., 2013. Teaching on a budget: Agents advising agents in reinforcement learning. In: Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems. pp. 1053–1060.
Toth, P., Vigo, D., 2014. Vehicle Routing: Problems, Methods, and Applications. SIAM.
Ulmer, M.W., 2020. Dynamic pricing and routing for same-day delivery. Transp. Sci. 54 (4), 1016–1033.
Ulmer, M.W., Thomas, B.W., 2020. Meso-parametric value function approximation for dynamic customer acceptances in delivery routing. European J. Oper. Res. 285 (1), 183–195.
Van Seijen, H., Fatemi, M., Romoff, J., Laroche, R., Barnes, T., Tsang, J., 2017. Hybrid reward architecture for reinforcement learning. Adv. Neural Inf. Process. Syst. 30.
Vansteenwegen, P., Melis, L., Aktaş, D., Montenegro, B.D.G., Vieira, F.S., Sörensen, K., 2022. A survey on demand-responsive public bus systems. Transp. Res. Part C: Emerg. Technol. 137, 103573.
Vinsensius, A., Wang, Y., Chew, E.P., Lee, L.H., 2020. Dynamic incentive mechanism for delivery slot management in e-commerce attended home delivery. Transp. Sci. 54 (3), 567–587.
Waßmuth, K., Köhler, C., Agatz, N., Fleischmann, M., 2023. Demand management for attended home delivery–a literature review. European J. Oper. Res. 311 (3), 801–815.
Yang, X., Strauss, A.K., 2017. An approximate dynamic programming approach to attended home delivery management. European J. Oper. Res. 263 (3), 935–945.
Yang, X., Strauss, A.K., Currie, C.S., Eglese, R., 2016. Choice-based demand management and vehicle routing in e-fulfillment. Transp. Sci. 50 (2), 473–488.
Yildiz, B., Savelsbergh, M., 2020. Pricing for delivery time flexibility. Transp. Res. Part B: Methodol. 133, 230–256.