Uncertainty in Real-World Vehicle Routing (Extended Abstract)

Václav Sobotka, Hana Rudová

Faculty of Informatics, Masaryk University, Brno, Czech Republic
sobotka@mail.muni.cz, hanka@fi.muni.cz

Introduction

Vehicle routing problems (VRP) have been drawing the attention of researchers for decades. Especially heuristic approaches to deterministic VRP variants gradually evolved into a mature field with well-tested approaches capable of addressing a wider range of VRP variants, e.g., (Ropke and Pisinger 2006) or (Christiaens and Berghe 2020). Stochastic variants of VRPs have been extensively studied as well. However, there is a substantial gap between the scalability and generality of deterministic approaches and their stochastic counterparts. Typically, works addressing stochastic VRPs rely on exact approaches arising in stochastic programming or robust optimization and are substantially limited in scaling to larger instances and problem richness in terms of features and constraints. Similarly, approaches addressing uncertainty via simulations and sampling are inherently limited in their scaling capabilities. In our cooperation with freight transportation company Wereldo, we identified uncertainties in the amount of load to be transported, travel, and service times[1] as common and inherent aspects of the industrial problem. While the standard heuristic search solvers proved useful for the real-world problem, the ignored uncertainties frequently limit the direct applicability of the solutions they produce. To fill this gap, we aim to enrich the capabilities of these search algorithms with uncertainty-handling mechanisms that are (1) applicable to a variety of complex VRPs and (2) easy to integrate into existing state-of-the-art solvers without (3) unnecessarily compromising their scalability. We formulate four mechanisms fitting these criteria, including standard chance constraints, two data manipulation methods, and a novel penalty-based method. By leveraging existing efficient search methods, even large-scale VRPs by deterministic standards (several hundred to a few thousand customers) with complex constraints and features may benefit from uncertainty handling. In this extended abstract, we primarily focus on the key properties of the uncertainty handling methods and their impacts on the search space of the problem. We refer interested readers to the previously published full paper (Sobotka and Rudová 2024) for further details and complete results.

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]Time needed for (un)loading freight at a stop.

Methods

Our modeling builds on top of standard models for *capacitated VRP* (CVRP) and *VRP with time windows* (VRPTW). We introduce the stochastic aspects by assuming that the quantities representing customer loads, travel, and service times are independent random variables with known expected value $E(X)$ and variance $Var(X)$. Quantities in the deterministic models are replaced with the respective $E(X)$.

Resource deflation method is based on solving a modified problem instance. Particularly, we "deflate" vehicle capacities or impose earlier arrival deadlines. The severity of the modifications is driven by the *deflation factor* parameter.

Demand inflation method is also based on solving a modified problem instance. Particularly, we "inflate" the amount of transported load or travel and service times proportionally to their uncertainty. The severity of the modifications is driven by the *inflation factor* parameter.

Chance constraint method is a standard approach of stochastic programming. The method extends the deterministic model with probabilistic constraints, effectively limiting admissible probabilities of constraint violation with a parametrizable threshold. For capacities, we require that each vehicle keeps k standard deviations of its total load as slack capacity. For temporal aspects, we require k standard deviations of the arrival time as extra slack time before the arrival deadline on each vehicle stop. The k is a parameter.

Risk penalization method evaluates the slacks in analogy to the described chance constraint method. However, instead of considering insufficiently large slacks to be infeasible, too small slacks are progressively penalized in the objective function. First, the available slack is expressed as a k multiple of the standard deviation of the total uncertainty (total load, total temporal uncertainty). Then, the value of k is reduced by the parameter $shift$. The reduced value of k is then used to calculate a probabilistic bound using the Chebyshev inequality. Ultimately, all such penalties in the whole solution are summed and added into the objective function with a weight parameter w. For capacities, this penalty is calculated for each route based on the vehicle capacity and uncertainty in the total load. For temporal aspects, penalties are calculated and summed for each stop in each route by comparing the accumulated temporal uncertainty with the arrival deadline at the given stop.

Experimental Setup

The four described mechanisms for both load-related and temporal uncertainties were integrated into our implementation of the adaptive large neighborhood search (ALNS) solver (Sassmann et al. 2023) used by the company Wereldo in practice. The main goal of our experiments was to (1) test how well the mechanisms trade off total traveled distance with the probability of constraint violation in the produced solutions and (2) determine their impact on the scaling of the algorithm. The experiments were performed on instances derived from the standard CVRP[2] and VRPTW[3] benchmarks with various uncertainty settings. Moreover, we test the mechanisms on real-world instances from Wereldo.

To measure the probability of failure, all produced solutions are post-processed by simulating 10,000 realizations of the uncertain quantities. We sample from normal distributions parameterized by the available $E(X)$ and $Var(X)$, truncated at 0. The probability of failure is approximated as the proportion of realizations resulting in constraint violation. The scaling properties are assessed by comparing the runtimes of the ALNS solver with individual mechanisms to the baseline runtimes of the original ALNS.

Method Properties

The first key observation from the experiments in (Sobotka and Rudová 2024) is that standard deterministic solutions are prone to failures (constraint violations) when confronted with either capacity-related or temporal uncertainties. Critically, even in the mildest uncertainty setups, around 60 % of solutions evinced failure probabilities above 80 %. Moreover, setups with higher uncertainty further deepen this issue. These findings illustrate how (near)-optimal solutions are commonly very close to the infeasible regions of the deterministic search space, directly corresponding with their high failure probabilities. Consequently, a different search strategy and/or space is clearly needed to account for uncertainties, underscoring the main motivation of our paper.

Regarding the distance-risk tradeoff capabilities, all mechanisms proved to be viable in case the uncertainty is rather mild and homogeneously spread within the problem instance. The tradeoff capabilities are then comparable, with minor benefits of the risk penalization mechanism. However, the gaps among the strategies become more visible in setups with more uncertainty, and especially if the uncertainty is spread unevenly within the problem instance. Moreover, the gaps are generally bigger in the case of the temporal aspects as the uncertainty accumulates from more sources (travel and service times) compared to capacity-related uncertainty.

Across all of the tested mechanisms, the risk penalization strategy clearly evinced the best tradeoff capabilities. We attribute this dominance to several factors in which its search space differs from the other mechanisms. In the case of resource deflation, demand inflation, and chance constraint mechanisms, the concept of uncertainty is propagated into the search space by further restricting the set of feasible solutions. While the mechanisms differ in how precisely the additional restriction copies actual risks, these mechanisms solely split the previously feasible solutions into those where the present risks are acceptable and those where not. The fundamental difference and advantage of the risk penalization method is its finer reasoning about the present risks. While it still keeps a hard boundary (infinite penalty) driven by the $shift$ parameter, it provides an additional layer of preference over the acceptable solutions. This subtle difference has important implications. If two acceptable solutions are being compared qualitatively, their riskiness is reflected in the comparison. This is especially important if the candidate solutions are competitive in terms of distance but vastly differ in terms of risks. For example, we observed solutions in which a customer could have been served by two vehicles, both with acceptable risks. While the choice of the vehicle had no impact on the traveled distance, risks were affected considerably. In such a case, the risk penalization mechanism consistently selected the preferable solution, while the remaining mechanisms chose arbitrarily, as their preference is based solely on traveled distance.

Regarding the scaling properties, we tested all mechanisms on larger instances with up to 1,000 customers under different parameterizations. Critically, the measured average relative overheads of the mechanisms do not evince dependence on either parameterization or instance size and have generally minor impacts on runtimes with at most 117 % of baseline. The only exceptions are chance constraint and penalty methods for temporal uncertainties, with around a sevenfold increase in runtimes due to an increase in the asymptotic complexity. To alleviate this scaling issue at least on mildly constrained instances, where slowdowns are the most critical, we propose a lightweight modification of the penalty mechanism. This modification avoids evaluating the costly penalties during neighborhood inspection and only uses the penalty calculation at the end of the iteration to possibly discard a too risky candidate solution.

[2]http://vrp.atd-lab.inf.puc-rio.br/index.php/en/

[3]https://www.sintef.no/projectweb/top/vrptw/

Acknowledgments

Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.
We thank the company Wereldo for providing us with the problem motivating the paper and the real-world instances.

References

Christiaens, J.; and Berghe, G. V. 2020. Slack induction by string removals for vehicle routing problems. *Transportation Science*, 54: 417–433.

Ropke, S.; and Pisinger, D. 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4): 455–472.

Sassmann, V.; Rudová, H.; Gabonnay, M.; and Sobotka, V. 2023. Real-world vehicle routing using adaptive large neighborhood search. In Pérez Cáceres, L.; and Stützle, T., eds., *EVO**, 34–49. Springer Nature Switzerland.

Sobotka, V.; and Rudová, H. 2024. Uncertainty in real-world vehicle routing. In *ECAI 2024*, 4311–4318. IOS Press.