

13th CIRP Global Web Conference (CIRPe 2025)

Accelerating Fleet Upgrade Decisions with Machine-Learning Enhanced Optimization

Kenrick Howin Chai^{a,*}, Stefan Hildebrand^{a,*}, Tobias Lachnit^b, Martin Benfer^b, Gisela Lanza^b, Sandra Klinge^a

^aDepartment of Structural and Computational Mechanics, TU Berlin, Straße des 17. Juni 135, Berlin and 10623, Germany

^bwbk Institute of Production Science, Karlsruhe Institute of Technology, Kaiserstraße 12, Karlsruhe and 76131, Germany

* Corresponding author. Tel.: +49 (0) 30 314 21481; E-mail address: stefan.hildebrand@tu-berlin.de, kenrick.howin.chai@campus.tu-berlin.de

Abstract

Rental-based business models and increasing sustainability requirements intensify the need for efficient strategies to manage large machine and vehicle fleet renewal and upgrades. Optimized fleet upgrade strategies maximize overall utility, cost, and sustainability. However, conventional fleet optimization does not account for upgrade options and is based on integer programming with exponential runtime scaling, which leads to substantial computational cost when dealing with large fleets and repeated decision-making processes. This contribution firstly suggests an extended integer programming approach that determines optimal renewal and upgrade decisions. The computational burden is addressed by a second, alternative machine learning-based method that transforms the task to a mixed discrete-continuous optimization problem. Both approaches are evaluated in a real-world automotive industry case study, which shows that the machine learning approach achieves near-optimal solutions with significant improvements in the scalability and overall computational performance, thus making it a practical alternative for large-scale fleet management.

© 2025 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>)

Peer review under the responsibility of the scientific committee of the CIRPe 2025

Keywords: Product upgrades; Replacement problem; Optimization; Machine learning; Integer programming

1. Introduction

With rising sustainability demands and the need for long-term cost-effective operations, extending the lifetime of capital equipment has become increasingly important. Rather than relying on costly and recurring replacements, companies are increasingly exploring individualized upgrades and adaptations to maintain or enhance fleet performance over time. In Product-Service Systems (PSS), enabling mid-life upgrades provides a promising strategy to reduce resource consumption, improve operational efficiency, and ensure continuous technological adaptation [1]. In fleet management, decisions still often focus on full replacements, although targeted upgrades represent a valuable alternative or addition to conventional renewal strategies. However, planning such upgrades at the fleet level

introduces substantial complexity for manufacturers. Varying usage patterns, system aging, and heterogeneous product states must be considered while maintaining economic viability and scalability. Due to the scale of product fleets and the multitude of upgrade possibilities, conventional optimization approaches like mixed-integer programming (MIP) often face significant computational challenges. To address this challenge, the present study explores how machine learning (ML) can accelerate decision-making in fleet renewal scenarios by enabling the efficient planning and control of individualized product upgrades and adaptations at scale. The conventional Fleet Renewal Problem (FRP), also referred to as fleet replacement problem [2] asks for the optimal timing and selection of vehicle or machine replacements and has been studied extensively [3–5]. Extended approaches focus to

capture uncertainty in key parameters [6–8], to include varying usage intensity, technological changes and asset deterioration over the product lifespan [9,10] and environmental considerations embedded into the objective function or modeled as constraints [11–13]. Overall, the existing research on combined fleet renewal and upgrade decisions is sparse as predominantly upgrades are addressed at the level of individual products, with limited attention to fleet-level considerations [14–16]. This results in fragmented strategies and suboptimal outcomes which are particularly limited in scenarios where both replacement and upgrade options are viable alternatives. A major challenge when including product upgrades in management strategies for large fleets is the computational effort as FRP is solved conventionally discretizing all variables including the age of the products into integer values, leading to a combinatoric problem in the complexity class NP. A wide range of optimization methods has been proposed, most commonly integer programming (IP) [9,13] and MIP [10], where the computational effort grows exponentially with the problem size, intensified by intense RAM usage. Recent strong baselines include SAT based constraint programming (for example OR Tools CP-SAT), classical decomposition and relaxation methods (Benders and Lagrangian), and learning to optimize for MIPs [17–19]. The present study offers an alternative approach and addresses the direct inclusion of upgrade strategies into FRP respecting both cost and sustainability objectives, by a unified ML approach based on Straight-Through Estimator (STE). The new approach allows for significantly improved scalability using gradient-based optimization with only polynomial computational complexity. The validation is carried out with a real-world automotive industry case study. Its results show that the ML approach delivers near-optimal solutions with significantly improved computational performance, making it a practical tool for modern fleet lifecycle management.

2. Conceptual and Methodological Background

A product upgrade is a strategic approach to extend a product's lifecycle by enhancing its functionality and performance to meet evolving requirements, such as technological advancements or changing customer preferences. Unlike complete product replacement, upgrades focus on modifying or improving specific modules or components, allowing for continued use of the existing system [14,20]. Existing research covers financial aspects [1], design aspects [21,22], maintenance strategies for dealers and the second-hand market [23] as well as upgrade strategies for end-users [14]. The term fleet broadly refers to a collection of product entities that are grouped based on shared characteristics or a common operational purpose. While traditionally associated with ships or vehicles, the concept extends to encompass systems, sub-systems, or components within an industrial framework, depending on the focus of analysis. The defining feature of a fleet is the commonality among its members, whether technical, operational or contextual [24]. The FRP asks for an optimal schedule to replace units within a fleet which minimizes the total cost of ownership and operation over a defined planning horizon. As vehicles age or accumulate usage, their

performance deteriorates, leading to increased expenses such as operating, maintenance, repair and fixed overhead costs. The objective of the FRP is to strategically plan replacements in a way that balances these rising costs with investment in new assets, ensuring overall cost-efficiency and operational effectiveness of the fleet [9].

In the following, the framework to solve the conventional FRP is introduced [9,25]. IP Optimization is conducted over a planning horizon T , with replacement decisions made at each period $j \in \{0, \dots, T\}$. Each asset is characterized by its type $o \in \{0, \dots, O\}$ and age $i \in \{0, \dots, N\}$, where N denotes the maximum useful life and the following further notations are:

Decision Variables

- v_{oij} Number of type o , age i assets deployed in period j .
- b_{oij} Number of type o , age i assets purchased in period j .
- s_{oij} Number of type o , age i assets sold in period j .
- δ_j Binary variable, which takes a value of 1 if any asset is purchased in period j or 0 otherwise.

Parameters

- f Discount factor per period.
- p_{oij} Purchase price of a type o , age i assets in period j .
- k_j Fixed cost incurred in period j if a purchase is made.
- c_{oij} Operation and maintenance cost of a type o , age i asset in period j .
- r_{oij} Resale or salvage value of a type o , age i asset in period j .
- u_i Usage capacity of an asset of age i per period.
- d_j Fleet capacity requirement in period j .
- h_{oi} Initial number of type o , age i assets at the start of the planning horizon.

Objective Function

$$\begin{aligned} \text{minimize } C_{\text{objective,base}} &= \sum_{j=0}^{T-1} f^j \left(\sum_{i=0}^{N-1} \sum_{o=0}^O p_{oij} b_{oij} + k_j \delta_j \right) \\ &+ \sum_{j=0}^{T-1} \sum_{i=0}^{N-1} \sum_{o=0}^O f^{j+1} c_{oij} v_{oij} - \sum_{j=0}^T \sum_{i=0}^N \sum_{o=0}^O f^j r_{oij} s_{oij} \end{aligned}$$

To keep the current contribution concise, we omit the full set of constraints here. A detailed description is available as supplementary material at: <https://doi.org/10.48550/arXiv.2508.00915>

3. Fleet Upgrade Problem for sustainability

Fleet upgrading is particularly beneficial for high-investment, long-lifetime products, as it reduces costs and environmental impact associated with full disposal and repurchase. While maintenance can slow functional decline, targeted upgrades can not only preserve but enhance product capabilities beyond the original design, offering a cost-effective and sustainable alternative to traditional replacement cycles [14]. In the following, the framework of the FRP is extended to model product upgrades besides product replacements in a fleet and to consider environmental costs in the objective function. While the general structure of the model remains unchanged, several modifications are introduced. Specifically, each possible configuration of a product is treated

as a distinct type, denoted by $o \in O$. For example, $o = 0$ represents the base configuration, $o = 1$ the first upgraded variant, $o = 2$ the second upgrade, and so on. To accommodate this extension, additional decision variables and parameters are introduced, and the existing constraints are either modified or supplemented with new ones to accurately reflect the upgrade dynamics.

Decision Variables

m_{oij}	Number of type o components installed on age i assets in period j .
dm_{oij}	Number of type o components removed from age i assets in period j .
bk_{oij}	Number of new components of type o purchased in period j .
sk_j	Number of used components of type o sold in period j .
i_{oj}	Inventory of type o components held in period j .

Parameters

ep_{oj}	Emissions from producing a type o asset in period j (tons CO ₂ -eq).
en_{oj}	Emissions during operation of a type o asset in period j (tons CO ₂ -eq).
es_{oj}	Emissions from disposal of a type o asset in period j (tons CO ₂ -eq).
ec_j	CO ₂ price in period j (cost per ton CO ₂ -eq).
hk_o	Initial inventory of type o components before planning starts.
pk_{oj}	Purchase price of a type o component in period j .
rk_{oj}	Selling price of a type o component in period j .
cm_{oj}	Installation cost of a type o component in period j .
cd_{oj}	Removal cost of a type o component in period j .
u_{oij}	Usage capacity of a type o , age i asset in period j .

Objective Function

minimize $C_{\text{objective,extended}} = C_{\text{objective,base}} + C_{\text{environment}} + C_{\text{upgrade}}$

$$C_{\text{environment}} = \sum_{j=0}^{T-1} \sum_{i=0}^{N-1} \sum_{o=0}^O f^j \cdot ep_{oj} \cdot ec_j \cdot b_{oij}$$

$$+ \sum_{j=0}^{T-1} \sum_{i=0}^{N-1} \sum_{o=0}^O f^{j+1} \cdot en_{oj} \cdot ec_j \cdot v_{oij} + \sum_{j=0}^T \sum_{i=0}^N \sum_{o=0}^O f^j \cdot es_{oj} \cdot ec_j \cdot s_{oij}$$

$$C_{\text{upgrade}} = \sum_{o=0}^O \sum_{j=0}^T f^j \cdot pk_{oj} \cdot bk_{oj} + \sum_{o=0}^O \sum_{i=0}^N \sum_{j=0}^T f^j \cdot cm_{oj} \cdot m_{oij}$$

$$+ \sum_{o=0}^O \sum_{i=0}^N \sum_{j=0}^T f^j \cdot cd_{oj} \cdot dm_{oij} - \sum_{o=0}^O \sum_{j=0}^T f^j \cdot rk_{oj} \cdot sk_{oj}$$

The complete constraints are also included in the supplementary material referenced earlier.

4. Accelerating the Optimization with Machine Learning techniques

IP is well known for being NP-hard [26], i.e. the required computational complexity grows exponentially with the problem size. To mitigate this computational burden, an

alternative approach is suggested which reformulates the IP into a mixed discrete-continuous optimization problem. This reformulated problem is then solved with established ML techniques. Therefore, the STE is applied which introduces a differentiable approximation for discrete variables to enable the use of gradient based optimization algorithms with improved convergence behavior [27,28]. The extended model taking product upgrades into account entails disjunctive or logical constraints which cannot be directly included with standard IP solvers but requires explicit modeling techniques such as Big-M or hull transformation [29]. In preliminary tests, these increased the runtimes of the solver by more than three orders of magnitude even for small tasks, rendering this approach practically infeasible. In contrast, the ML framework presented here allows to include these directly. Leveraging this flexibility, the present formulation treats the asset allocation v as the central decision variable to be directly optimized. All other decision variables are not optimized independently but are instead derived from v . This way, the original flow balance constraints can be omitted, which govern variable transitions across time periods in the IP formulation. This transformation reduces the complexity of the feasible set and eliminates potential multicollinearity among the optimized decision variables, thereby enhancing the efficiency and numerical stability of the optimization process. Constraints that do not govern flow balance are reformulated as soft constraints and penalties for violations are calculated using the Rectified Linear Unit (ReLU) function, resulting in continuous functions facilitating the application of gradient-based optimizers. The detailed expressions defining the mapping from the central variable v to all decision variables in the extended formulation are omitted here for brevity. These equations can be found in the supplementary material referenced earlier. The optimization process is structured into four sequential phases; each aimed at iteratively improving the central decision variable v to find an optimal solution within the feasible region:

1. Initialization

The process begins with the initialization of the central decision variable v , usually randomly sampled and as far as possible based on problem-specific heuristics. A suitable optimization algorithm such as Adam is selected to iteratively update v . Adam is a widely used adaptive stochastic optimization method. It extends the stochastic gradient descent optimizer by computing individual learning rates from estimating first and second moments of the gradients [30].

2. Projection and Derivation of Decision Variables

To facilitate optimization, v is first transformed into a feasible space by eliminating negative values and ensuring no assets are active in the final time period or at their maximum lifetime:

$$v = \max(0, v)$$

$$v_{oij} = 0 \quad \forall o; \forall i; j = T,$$

$$v_{oij} = 0 \quad \forall o; i = N; \forall j$$

Next, to allow gradient flow while rounding values down to the nearest integer, v is discretized using STE

incorporating the definition from Fan et al. [31] $STE_{\text{round}}(v) = \text{stop_grad}([v]) - \text{stop_grad}(v) + v$.

The operator `stop_grad` blocks gradient flow during backpropagation, so that the derivative is considered as 0. In the forward pass, this yields $STE_{\text{round}}(v) = [v]$. During backpropagation, gradients are computed using the surrogate $\nabla STE_{\text{round}}(v) = \nabla v$.

After projection and discretization, all remaining decision variables are derived from both the continuous and rounded versions of v using predefined equations.

3. Cost Function and Penalty Term Calculation

The cost function is computed for both the fractional and rounded solutions. Additionally, a penalty term is added for the rounded solution to account for any constraint violations not handled during projection or derivation, for inequality constraints: $p_{\text{ineq}} = c \cdot \text{ReLU}(z)$, for equality constraints: $p_{\text{eq}} = c \cdot |z|$ with the penalty weight c , and the respective expression z . If the penalty is zero and the rounded solution yields the lowest objective value so far, it is logged as the current best solution.

4. Backpropagation

The loss function combining the cost of the fractional solution and the penalty from the rounded solution is computed together with its gradients w.r.t. the optimizable parameters. One optimization step is carried out by the optimizer to update v . The process then loops back to Phase 2 for the next iteration.

5. Case Studies

5.1. Case 1: Basic Formulation

To allow for direct comparison and benchmarking, the first case study applies the data from Parthanadee et al. [9] considering a vehicle fleet across eight scenarios, varying in vehicle prices, operation and maintenance (O&M) costs, and vehicle utilization. Vehicle prices follow either constant depreciation or real price trends from used-car markets. O&M costs are either fixed or based on data from 177 passenger cars. Two usage models are considered: one with constant mileage, the other based on user preferences. Table 1 outlines the scenario structure.

Table 1. Overview of the scenarios.

Scn.	Vehicle Price	O&M	Utilization	Demand (km)
1	Linear	Constant	Constant	300.000
2	Linear	Constant	User Preference	228.000
3	Linear	Empirical	Constant	300.000
4	Linear	Empirical	User Preference	228.000
5	Empirical	Constant	Constant	300.000
6	Empirical	Constant	User Preference	228.000
7	Empirical	Empirical	Constant	300.000
8	Empirical	Empirical	User Preference	228.000

The planning horizon is three years, with a maximum vehicle age of ten years and a 5% discount rate. The initial fleet consists of vehicles aged 1, 3, 5, 7, and 9 years, with 3 vehicles per age group. Each vehicle can provide 20,000 km of services. In scenarios with constant mileage, total annual demand is 300,000 km; in preference-based scenarios, 228,000 km. Table

2 summarizes vehicle prices (in THB), O&M costs (THB/km), and lifetime mileage.

Table 2. Vehicle Price, O&M-Cost and Utilization.

Age	Vehicle Price (THB)		O&M-Cost (THB/km)		Utilization (km)	
	Linear	Empirical	Constant	Empirical	Constant	User Preference
0	1.000.000	1.000.000	3	3,44	20.000	20.000
1	946.315	910.000	3	3,53	20.000	20.000
2	892.630	830.000	3	4,75	20.000	20.000
3	838.945	820.000	3	3,75	20.000	20.000
4	785.260	630.000	3	4,81	20.000	18.000
5	731.575	620.000	3	3,82	20.000	16.000
6	677.890	600.000	3	3,73	20.000	14.000
7	624.205	590.000	3	4,91	20.000	12.000
8	570.520	580.000	3	3,83	20.000	10.000
9	516.835	510.000	3	5,02	20.000	8.000
10	463.150	330.000	3	4,05	0	0

5.2. Case 2: Extended Formulation

An electric vehicle with exchangeable battery modules is introduced into the second case study to represent an upgrade option. The chosen example product is the Nio ES8 which is currently available in the market with two battery configurations: a 75-kWh battery providing a range of 375 km, and a 100-kWh battery with a range of 500 km. The base vehicle costs €60,000, excluding the battery. The battery costs amount to €12,000 for the 75-kWh version and €21,000 for the 100-kWh version [32]. In addition, each battery swap incurs a cost of €30 [33]. Annual maintenance costs are assumed to be €576. The annual mileage per vehicle is set at 11,733 km.

Depreciation is modeled as follows: the vehicle loses 25% of its value in the first years. After three years, 50% of the original value remains. From the fourth year onward, a constant annual depreciation rate of 5% is assumed. The parameters for environmental costs are summarized in Table 3, based on Wietschel et al. [34]. Emissions from disposal are neglected, as their contribution is negligible compared to other phases of the vehicle life cycle. The assumed CO₂ prices are €45 per ton in 2024, €55 per ton in 2025, and €65 per ton in 2026 [35].

Table 3. CO_{2eq}-Emission from diesel and electric cars

Type	2024	2025	2026
Diesel car production (t CO _{2eq} per car)	8,1	8,1	8,1
Diesel car operation (g CO _{2eq} /km)	134,68	136,36	138,04
Electric car production (t CO _{2eq} per car)	12,6	12,5	12,4
Electric car operation (g CO _{2eq} /km)	5,12	5,05	4,97
CO ₂ price (€ / t)	45	55	65

The planning horizon spans three years, with a maximum vehicle service life of ten years. At the start of the planning period, the fleet consists of no vehicles. The initial range requirement of the fleet is 5,000 km, increasing by 200 km annually.

6. Results

Optimization was performed using CPLEX to represent the IP method and the Adam optimizer for the ML approach. Experiments were conducted on a workstation (“WSA”) with

an AMD Ryzen 7 5800H CPU, an NVIDIA GeForce RTX 3050 Ti GPU, and 16 GB of RAM. For the IP method, the default CPLEX configuration via Pyomo was used. The ML model employed Adam with a learning rate of 0.5, a penalty weight of 100, and initialized v as a zero matrix. Training proceeded for 10,000 epochs or until negligible changes were observed; no random seed was set due to the deterministic initialization. Discrepancies between the IP and ML solutions were evaluated using the absolute percentage error

$$\text{Discrepancy} = \frac{|\text{obj} - \text{obj}_{\text{optimal}}|}{|\text{obj}_{\text{optimal}}|} \cdot 100\%$$

The optimization results are presented in Table 4. The objective values for the first base case are negative, as the initial fleet consists of 15 vehicles and the investment costs are excluded from the objective function. The results indicate that the ML approach achieved the same optimal value as IP in 5 out of 9 scenarios. In the remaining cases, the ML method produced near-optimal solutions, with a maximum discrepancy of approximately 1%.

Table 4. Optimization result.

Case	Sc.	Integer Programming	Machine Learning	Discrepancy
Base	1	-5.274.136	-5.274.136	0,00%
	2	-6.029.618	-5.990.568	0,65%
	3	-4.486.399	-4.486.399	0,00%
	4	-5.565.468	-5.506.788	1,05%
	5	-6.285.552	-6.285.552	0,00%
	6	-7.023.553	-7.006.278	0,002%
	7	-5.668.737	-5.668.737	0,00%
	8	-6.234.137	-6.223.408	0,17%
Extended	-	257.454	257.454	0,00%

Furthermore, an empirical analysis is performed to investigate how computational complexity scales with problem size. For this purpose, scenario 7 from the base model is employed as an exemplary case study. The problem size is scaled by extending the planning horizon following a logarithmic progression, allowing a systematic investigation of the scalability of the optimization methods across multiple orders of magnitude. The discount rate is omitted in this analysis, as its application would lead to cost values approaching zero over extended planning periods, thereby distorting the results. Fig. 1 compares the computation times of the IP solver and the ML solver depending on problem size. The results demonstrate that the IP solver handles small problems very rapidly. However, its computation time increases rapidly with the problem size as expected due to IP being an NP hard problem. In contrast, although the ML method requires more time for smaller problems, its computational time grows at a significantly lower rate. This corresponds to the upper bound of training complexity with Adam not being significantly dependent on the problem size but being dominated by the required final tolerance [36]. Furthermore, the IP solver was unable to provide a solution starting at a planning horizon of 10^5 due to complete exhaustion of available RAM, whereas the ML method continues to provide reliable solutions.

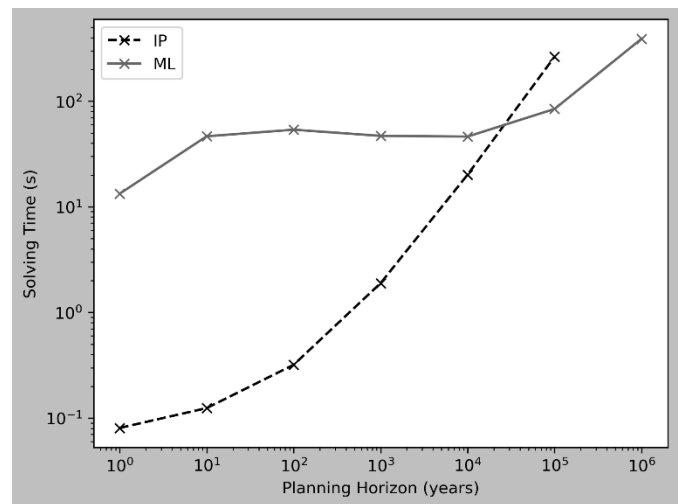


Fig. 1. Comparison of computational times of solvers, performed on the workstation “WSA” as described above.

7. Discussion

The results demonstrate in five out of the nine scenarios examined, that the ML-based optimization method successfully reproduces the optimal solutions obtained using IP. In the remaining cases, deviations are minimal, with a maximum discrepancy of approximately 1%, while no constraint violations occur in any scenario. These findings confirm that enforcing constraints through penalty terms is sufficient to maintain feasibility, highlighting the robustness and reliability of the ML-based approach. Table 5 exemplifies cases where the ML method does not reach exact optimality, predominantly in period 0, (except Scenario 6 where period 2 deviates). In most cases, discrepancies between IP and ML solutions are minimal, with only Scenario 2 showing a pronounced deviation.

Table 5. Comparison of decision variables v between IP and ML method for scenarios with suboptimal solutions

Sc.	Optimizer	Age										
		0	1	2	3	4	5	6	7	8	9	10
2	IP	0	0	0	11	0	0	0	0	0	1	0
	ML	0	0	1	9	1	0	0	1	0	0	0
4	IP	0	0	0	10	0	0	2	0	0	0	0
	ML	0	0	0	9	0	3	0	0	0	0	0
6	IP	0	0	0	0	12	0	0	1	0	0	0
	ML	0	0	0	0	13	0	0	0	0	0	0
8	IP	0	0	0	0	1	0	15	0	0	0	0
	ML	0	0	0	0	0	2	14	0	0	0	0

Although the IP solver always delivers optimal solutions, its computational performance degrades sharply as problem size grows. Additionally, the large number of branches which need to be stored leads to very high memory requirements. In contrast, the ML-based method reliably provides solutions for very large problems, as it only requires information from the current and previous optimization step, avoiding memory bottlenecks. This illustrates the clear advantage of the ML approach over IP for large-scale optimization problems.

8. Conclusion and Outlook

In this work, an extended IP formulation is introduced to integrate fleet renewal and upgrade options, which can be solved by established solvers such as CPLEX. To overcome the extensive needs of IP w.r.t. RAM and computation time for more complex problem statements, an alternative approach based on ML is proposed. The results indicate that the IP method with a branch-and-cut solver like CPLEX is preferable only for smaller cases. For extensive and complex product fleet management with individual upgrade strategies and additional constraints, the ML version is better scalable than the conventional solution. A machine learning approach may therefore be more suitable for vehicle fleets with numerous configurable components and variants and additional constraints. The current contribution focuses on deterministic optimization with a predefined cost function. Future research could explore the integration of data-driven methods to better capture uncertainties and dynamic behaviors inherent in real-world systems, enabling a more nuanced and realistic modeling approach. Furthermore, recent studies by Lee and Kim [17] and Tang et al. [18] have introduced novel ML-based techniques for solving MIP problems. A comparative analysis with these emerging methods could yield valuable insights. Moreover, the development of hybrid approaches that synergistically combine classical mathematical optimization with ML techniques holds potential for enhancing both scalability and solution quality.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used DeepL and ChatGPT to ensure correct translations and grammar. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

- [1] Khan MA, West S, Wuest T. Midlife upgrade of capital equipment: A servitization-enabled, value-adding alternative to traditional equipment replacement strategies. *CIRP J Manuf Sci Technol* 2020;29:232–44.
- [2] Ansaripoor AH, Oliveira FS, Liret A. Recursive expected conditional value at risk in the fleet renewal problem with alternative fuel vehicles. *Transp Res Part C Emerg Technol* 2016;65:156–71.
- [3] Karabakal N, Lohmann JR, Bean JC. Parallel Replacement under Capital Rationing Constraints. *Manage Sci* 1994;40:305–19.
- [4] Hartman JC. The parallel replacement problem with demand and capital budgeting constraints. *Naval Research Logistics (NRL)* 2000;47:40–56.
- [5] Chang P-T. Fuzzy strategic replacement analysis. *Eur J Oper Res* 2005;160:532–59.
- [6] Hsu C-I, Li H-C, Liu S-M, et al. Aircraft replacement scheduling: A dynamic programming approach. *Transp Res E Logist Transp Rev* 2011;47:41–60.
- [7] Ansaripoor AH, Oliveira FS. Flexible lease contracts in the fleet replacement problem with alternative fuel vehicles: A real-options approach. *Eur J Oper Res* 2018;266:316–27.
- [8] Forootani A, Zarch MG, Tipaldi M, et al. A stochastic dynamic programming approach for the machine replacement problem. *Eng Appl Artif Intell* 2023;118:105638.
- [9] Parthanadee P, Buddhakulsomsiri J, Charnsethikul P. A study of replacement rules for a parallel fleet replacement problem based on user preference utilization pattern and alternative fuel considerations. *Comput Ind Eng* 2012;63:46–57.
- [10] Büyüktaktakın İE, Hartman JC. A mixed-integer programming approach to the parallel replacement problem under technological change. *Int J Prod Res* 2016;54:680–95.
- [11] Figliozzi M, Boudart JA, Feng W. Economic and Environmental Optimization of Vehicle Fleets: Impact of Policy, Market, Utilization, and Technological Factors. *Transp Res Rec* 2011;2252:1–6.
- [12] Boudart J, Figliozzi M. Key Variables Affecting Decisions of Bus Replacement Age and Total Costs. *Transp Res Rec* 2012;2274:109–13.
- [13] Emiliano W, Alvelos F, Telhada J, et al. An optimization model for bus fleet replacement with budgetary and environmental constraints. *Transportation Planning and Technology* 2020;43:488–502.
- [14] Chung W-H, Kremer GO, Wysk RA. A dynamic programming method for product upgrade planning incorporating technology development and end-of-life decisions. *Journal of Industrial and Production Engineering* 2017;34:30–41.
- [15] Wu B, Jiang Z, Zhu S, et al. Data-Driven Decision-Making method for Functional Upgrade Remanufacturing of used products based on Multi-Life Customization Scenarios. *J Clean Prod* 2022;334:130238.
- [16] Khan MA. Capital Equipment Upgrade Strategy in the Context of Servitization. West Virginia University, 2020.
- [17] Lee T-H, Kim M-S. RL-SPH: Learning to Achieve Feasible Solutions for Integer Linear Programs 2025.
- [18] Tang B, Khalil EB, Drgoña J. Learning to Optimize for Mixed-Integer Non-linear Programming with Feasibility Guarantees 2025.
- [19] Lennartson B. Optimization of Timed Petri Nets using CP-SAT. *IFAC-PapersOnLine* 2024; 58:90–5.
- [20] Albers A, Düser T, Kuebler M, et al. Upgradeable Mechatronic Systems - Definition and Model of Upgrades in the Context of the Model of SGE - System Generation Engineering. *FISITA World Congress 2023 Proceedings*, 2023.
- [21] Martinez M, Xue D. A modular design approach for modeling and optimization of adaptable products considering the whole product utilization spans. *Proceedings of the Institution of Mechanical Engineers, Part C* 2018;232:1146–64.
- [22] Gadalla M, Xue D. An efficient optimisation method based on weighted AND-OR trees for concurrent reconfigurable product design and reconfiguration process planning. *Int J Prod Res* 2023;61:859–79.
- [23] Su C, Wang X. Optimal Upgrade Policy for Used Products Sold with Two-dimensional Warranty. *Qual Reliab Eng Int* 2016;32:2889–99.
- [24] Medina-Oliva G, Voisin A, Monnin M, et al. Predictive diagnosis based on a fleet-wide ontology approach. *Knowl Based Syst* 2014;68:40–57.
- [25] Hartman JC. A GENERAL PROCEDURE FOR INCORPORATING ASSET UTILIZATION DECISIONS INTO REPLACEMENT ANALYSIS. *Eng Econ* 1999;44:217–38.
- [26] Kannan Ravindran and Monma CL. On the Computational Complexity of Integer Programming Problems. In: Henn Rudolf and Korte B and OW, editor. *Optimization and Operations Research*, Berlin, Heidelberg: Springer Berlin Heidelberg; 1978, p. 161–72.
- [27] Bengio Y, Léonard N, Courville A. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation 2013.
- [28] Shah R, Yan M, Mozer MC, et al. Improving Discrete Optimisation Via Decoupled Straight-Through Gumbel-Softmax 2024.
- [29] Bynum ML, Hackebeil GA, Hart WE, et al. *Pyomo — Optimization Modeling in Python*. Springer International Publishing; 2021.
- [30] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization 2017.
- [31] Fan T-H, Chi T-C, Rudnicky AI, et al. Training Discrete Deep Generative Models via Gapped Straight-Through Estimator 2022.
- [32] Rudschies W. Nio ES8: Erst mal nur in Norwegen n.d. <https://www.adac.de/rund-ums-fahrzeug/autokatalog/markenmodelle/nio/nio-es8/> (accessed March 30, 2025).
- [33] Lin C. The Chinese EV Company That Made Battery Swapping Work n.d. <https://knowledge.insead.edu/strategy/chinese-ev-company-made-battery-swapping-work> (accessed March 30, 2025).
- [34] Wietschel M, Moll C, Oberle S, et al. Klimabilanz, Kosten und Potenziale verschiedener Kraftstoffarten und Antriebssysteme für Pkw und Lkw. *Endbericht* 2019.
- [35] Mendelevitch R, Repenning J, Matthes FCh, et al. Treibhausgas-Projektionen 2024 für Deutschland - Rahmendaten. *Dessau-Roßlau: Umweltbundesamt*; 2024.
- [36] Wang B, Fu J, Zhang H, et al. Closing the Gap Between the Upper Bound and the Lower Bound of Adam's Iteration Complexity 2023.