International Journal of Mathematics Trends and Technology
ISSN: 2231-5373/ https://doi.org/10.14445/22315373/IJMTT-V72I6P106

Volume 72 Issue 6, 45-58, June 2026

Original Article

A Survey of Elevator Control Algorithms: From Classical Dispatching to Deep Reinforcement Learning with an Analytical Queueing Comparison

Nasir Ansari[1,2*], Ritu Khanna[3], Saurabh Mehta[4]

[1,3]*Department of Mathematics, Pacific Academy of Higher Education and Research University, Udaipur, India.*
[2]*Department of Mathematics, Vidyalankar Institute of Technology, Mumbai, India.*
[4]*Department of Electronics and Telecommunication, Vidyalankar Institute of Technology, Mumbai, India.*

[*]*Corresponding Author : nasir.ansari@vit.edu.in*

Received: 25 April 2026 Revised: 30 May 2026 Accepted: 18 June 2026 Published: 30 June 2026

Abstract *- Elevator cars assignment rules regulate passengers' waiting time, energy consumption, and satisfaction in high-rises. This paper reviews ten algorithm families applied to solving the elevator dispatching problem through analysis of the development of solutions, starting with simple mechanical algorithms in the 1950s up until deep reinforcement learning in current days. Families of algorithms are classified into Classical and Deterministic Control, Computational Intelligence, and Learning-based and Hybrid Control categories. In particular, Classical and deterministic control covers such families as Single Automatic Operation, Selective Collective Operation, Group Control, and Destination Control. Computational Intelligence includes fuzzy logic, neural networks, genetic algorithms, and model predictive control. Lastly, Reinforcement learning and hybrid meta-control belong to the third group. For each family, unified notations of mathematics are used to describe the queueing models and theoretical framework. Literature choice follows PRISMA 2020 guidelines, and the area is defined with taxonomy, chronology, and comparison tables. To make a quantitative comparison of different approaches possible, a set of queueing models (M/M/1, M/M/c, and M/G/c) was studied using a realistic twenty-four hours calls distribution for a fifteen-storey building with five elevators, including uncertainty bands calculated through resampling. It should be emphasized that a comparison was made basing on queueing models, but not the discrete-event elevator trajectories simulation. Performance improvements associated with advanced algorithm families were estimated according to cited literature, and therefore, the results show relative trends, but not the measurements. The hierarchy was discovered, where the lowest waiting times belong to hybrid and learning-based algorithm families, while Single Automatic Operation exceeds the upper ISO 4190 limit during the period of the rush hour. Unresolved issues and future directions for research are outlined below.*

Keywords *- Elevator control, Group control, Destination dispatching, Queueing models, Reinforcement learning, Model predictive control, survey, analytical queueing comparison.*

1. Introduction

A brief analysis of the lobby area of a high-rise office building illustrates the critical importance of the elevator system. Occupants rarely opt for the stairs beyond the fifth floor and only use the elevators in order to maintain operational efficiency. Waiting periods remain a significant problem, multiple elevator runs increase energy usage, and the lack of adequate elevator capacity at the peak hours in the morning leads to congestion in the lobby area. The solutions provided by the industry to these problems include the establishment of ISO 4190 standards, which stipulate the waiting period of about thirty seconds in office buildings, and VDI 4707 standards, which evaluate lifts based on energy usage. [1, 2].

The field of elevator control systems has undergone considerable transformation within the past one hundred years. Simple relay control was employed in the early stages of development, while the latter half of the twentieth century saw the development of SAO and SCO. A breakthrough happened in the 1980s due to the invention of microprocessors that made it possible to control an entire group of elevators using a single controller, leading to GCO. By the 2000s, Destination Control Systems (DCS) had emerged, requiring passengers to input their desired floors before boarding, and have since become the norm in tall buildings [3, 20]. Parallel to these conventional upgrades, researchers began exploring the potential of artificial Intelligence. Crites and Barto made an early breakthrough in 1998 by showing reinforcement learning could effectively manage elevator groups [5], followed shortly by Cortés et al., who applied genetic algorithms to the hall-call problem in 2004 [6]. Today, the field has shifted heavily toward deep reinforcement learning, specifically leveraging asynchronous actor-critic methods [7] and end-to-end deep Q-learning [24].

1.1. Motivation, Research Gap, and Novelty

Although there exists a significant amount of existing literature on this topic, there has not yet been a work that addresses the entire history and mathematics of each of these different families of control systems together in one publication. Earlier literature has treated classical control separately from modern artificial intelligence techniques and does not provide an effective way to compare the two methodologies. The main research gap here is the lack of an established mathematical base for the comparison of these algorithms.

There are four concrete aspects that characterize the uniqueness of this paper. Firstly, a consistent notation system is used in all ten algorithm families, thus allowing for the first-ever mathematical comparison between them. Secondly, the PRISMA 2020 methodology is used for literature search along with taxonomy and timelines that structure an otherwise scattered area. Thirdly, a comparative table is developed that contains information about the era, underlying method, strengths and limitations of every family. Finally, a reproducible analysis is performed through comparison of queueing in the different algorithms, along with uncertainty intervals and using ISO 4190 and VDI 4707 as benchmarks. This survey differs from prior comprehensive treatments of vertical transportation (e.g., Barney and Al-Sharif [3], Strakosch and Caporale [4]) by providing explicit analytical queueing models for each algorithm family and by proving a monotonicity result that establishes theoretical bounds on waiting-time improvement, whereas earlier works focused primarily on empirical design rules or simulation-based validation.

1.2. Scope and Limitations of the Analytical Model

In order to keep up with the rigor of mathematics, there has to be an explicit definition of the experimental boundaries. The experiments utilize concepts of queueing theory and resampling techniques for generating confidence intervals as well as statistical techniques. However, the efficiency of the algorithm families is based on observations in the literature, not independently measured. Further, the comparison is made using analytical models of queuing rather than simulation of elevator paths. Therefore, the figures presented as percentages have to be taken as relative performance measures rather than predictive ones.

2. Review Methodology and Literature Background

Literature search and selection adhered to the PRISMA 2020 guidelines (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) [26].

2.1. Search Strategy

The search was conducted on six different databases: IEEE Xplore, ScienceDirect, SpringerLink, ACM Digital Library, Scopus, and arXiv. The literature search covered papers between January 1990 and May 2024. Keywords were constructed using two different sets: topics-related keywords (elevator control, elevator group control, elevator dispatching, lift traffic) and methodology-related keywords (reinforcement learning, genetic algorithm, fuzzy logic, model predictive control, destination control, queueing).

2.2. Selection Process

Two reviewers conducted independent screening of titles and abstracts prior to performing a full-text assessment, disagreements being resolved by discussion. In the first search, 412 studies were found. Out of these, 71 were duplicates and 18 were irrelevant records, leaving 323 studies. Screening of title and abstract left out 196 studies. Out of the 127 studies that were obtained, 9 could not be accessed. A full-text assessment was done for 118 studies, excluding 70 studies which were either not focused on the control condition or had no performance measures or were in duplicate.

2.3. Review of Representative Work and State-of-the-Art

The literature reveals a progressive shift from static rules to dynamic optimization. In classical control, Pepyne and Cassandras proved that a simple threshold policy optimally dispatches vehicles during up-peak times [8] and later introduced an adaptive variant for fluctuating arrival rates [9]. Destination control was formalized into a mixed-integer assignment problem by Ruokokoski et al. [12] and evaluated for performance [11]. Nikovski and Brand developed an exact calculation of expected wait times for group control using Markov-chain dynamic programming, demonstrating up to seventy percent reductions in heavy traffic [10].

In computational Intelligence, Kim et al. developed early fuzzy group controllers [13], which Fernández et al. enhanced with dynamic relative waiting time considerations [14]. Genetic algorithms were introduced by Cortés et al. [6] and expanded to car dispatching by Bolat et al. [15], while Zhang and Zong focused on energy-saving regenerative strategies [22].
The learning-based domain originates from Crites and Barto's multi-agent reinforcement learning framework [5]. State-of-the-art developments include the asynchronous actor-critic controller by Wei et al. [7], occupancy-aware deep learning models [25], end-to-end deep Q-learning [24], and LSTM-based traffic-flow prediction for predictive controllers [23]. Hybrid meta-control, which switches strategies based on real-time conditions, has also been tested in office environments [18]. Compared to these existing findings, the present survey consolidates these isolated advancements into a singular

analytical taxonomy, allowing researchers to observe the theoretical performance gaps between state-of-the-art AI and traditional heuristic baselines. The taxonomy of the reviewed families is shown in Figure 1, and the historical evolution of the field is summarized in Figure 2.

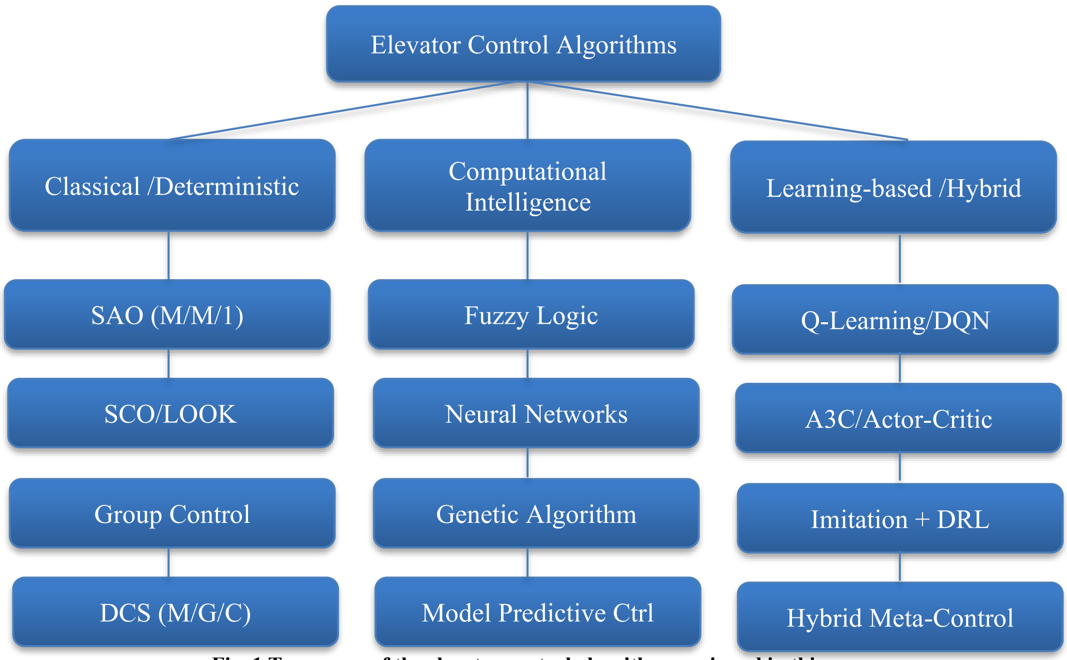

Fig. 1 Taxonomy of the elevator control algorithms reviewed in this survey

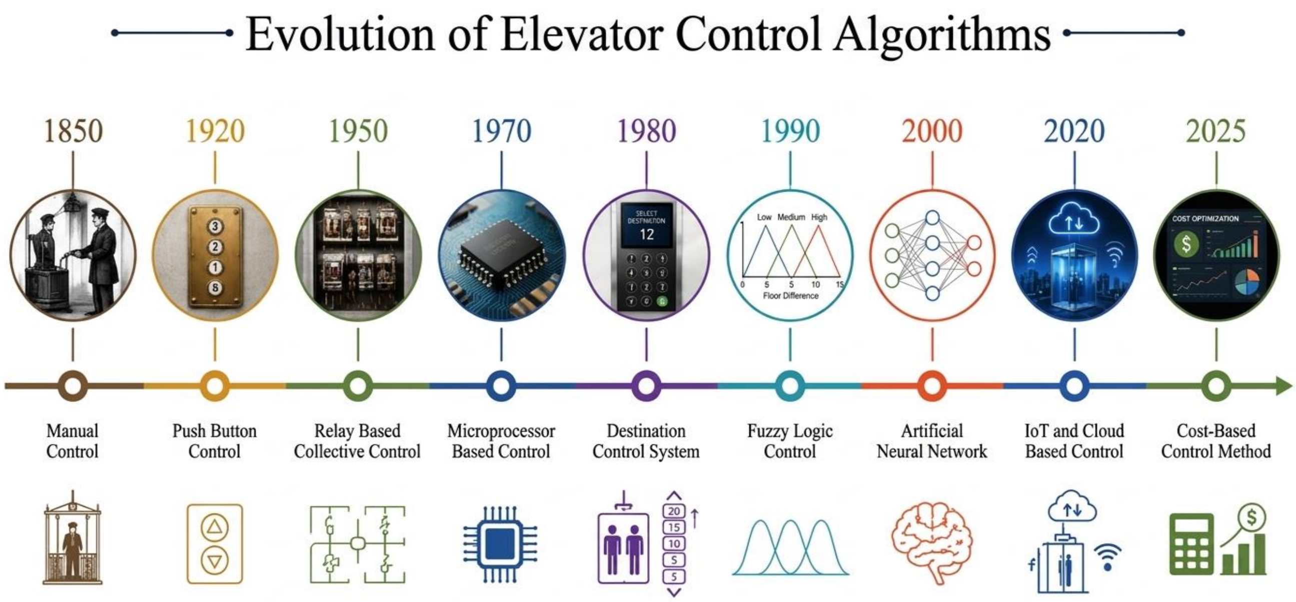

Fig. 2 Evolution of elevator control algorithms from 1850 to 2025

3. Mathematical Models of Elevator Control Algorithms

To ensure mathematical rigor and standardize terminology, a unified notation is established. Let N be the number of floors, E the number of elevators, and C the capacity of each car. Let $S_j(t)$ be the floor of elevator j at time t and $D_j(t)$ its direction. The set of active calls is $\mathcal{C}(t)$. Calls arrive as a Poisson process with rate $\lambda(t) = \lambda_0 \cdot \mu(t)$, where λ_0 is the base call rate and $\mu(t)$ is the traffic multiplier for hour t. Let t_f be the travel time per floor and t_d the door time. Throughout, a superscript names the family (for example W^{SAO} is the mean passenger waiting time under SAO) and the subscript q denotes the queue-only component, so $W = W_q + 1/\mu$ adds the mean service time.

Each family is mapped to a standard queue as a tractable first-order approximation. These mappings serve as analytical estimates rather than exact dynamical models, as real elevator service times depend on call distribution and car position. The queueing formulations follow standard results in queueing theory [27, 28].

Table 1. Unified Notation Used Throughout the Paper

Symbol	Meaning
N	Number of floors
E	Number of elevators (servers)
C	Passenger capacity of each car
C	Number of active servers in a queueing model
$\lambda(t)$	Call arrival rate at time t (calls/min)
λ_0	Base call rate
$\mu(t)$	Traffic multiplier for hour t
μ_{alg}	Service rate of family "alg" (calls/min)
P	Traffic intensity, $\rho = \lambda/\mu$
A	Offered load, $a = \lambda/\mu$
t_f, t_d	Travel time per floor; door time
W, Wq	Mean wait; queue-only mean wait
C(c,a)	Erlang-C delay probability
H	Efficiency factor for advanced families
Γ	RL discount factor, $\gamma \in (0,1)$
Q*(s,a)	Optimal action-value function

3.1. Modeling Assumptions

The analytical comparison rests on the following explicit assumptions, stated here for rigor and transparency.

Assumption 1. *Hall calls form a Poisson arrival process with hourly rate $\lambda(h) = \lambda_0\, \mu(h)$. Within each one-hour window, the rate is treated as stationary.*

Assumption 2. *Service times are exponentially distributed for the M/M/c families and generally distributed (finite mean and variance) for the M/G/c family (DCS). The mean service time is the reciprocal of the per-family service rate μ_{alg} defined in Equations (1)–(5).*

Assumption 3. *The cars in a multi-car family act as statistically identical parallel servers, so that classical M/M/c and M/G/c results apply as a first-order approximation.*

Assumption 4. *Each queue is stable, i.e. the offered load satisfies $a = \lambda/\mu < c$. Configurations violating this bound are reported as saturated rather than assigned a finite wait.*

Assumption 5. *For the advanced families, the efficiency factor $\eta \in (0,1]$ acts multiplicatively on the base-queue waiting time, $W_{\text{adv}} = \eta\, W_{\text{base}}$, with η taken from the cited empirical literature.*

The following result formalizes the central qualitative claim of the comparison: increasing the number of coordinated servers cannot increase the expected wait. It provides the analytical justification, requested in review, for the ordering observed in Section VI.

Proposition 1 (Monotonicity of waiting time in server count): Consider an M/M/c queue under Assumptions 1-4 with fixed arrival rate λ and fixed per-server service rate μ, where $c\mu > \lambda$. Let $W_q(c)$denote the mean waiting time in the queue. Then $W_q(c)$ is non-increasing in c ; that is, $W_q(c+1) \leq W_q(c)$ every integer c with $c\mu > \lambda$.

Proof: For an M/M/c queue with offered load $a = \lambda/\mu$ and utilization $\rho = a/c < 1$ the mean waiting time in the queue is $W_q(c) = \frac{C(c,a)}{c\mu-\lambda}$, where $C(c,a)$ is the Erlang-C delay probability of Equation (3). Two effects act as c increases by one. First, the denominator $c\mu - \lambda$ strictly increases by $\mu > 0$, which alone decreases. W_q. Second, the Erlang-C probability $C(c,a)$ It is itself known to be strictly decreasing in c for fixed a, adding a server cannot raise the probability that an arriving call finds all servers busy. A standard way to see this is by recursion.

$$\frac{1}{C(c,a)} = 1 + \frac{c-a}{a}\frac{1}{C(c-1,a)}, \qquad 0 < a < c,$$

from which $C(c,a) < C(c-1,a)$ whenever $c > a$, because the added positive term increases $1/C(c,a)$. Since both the numerator $C(c,a)$ decreases (or stays bounded) and the denominator $c\mu - \lambda$ increases, their ratio $W_q(c)$ is non-increasing in c. Equality holds only in the degenerate limit of negligible load. Hence $W_q(c+1) \leq W_q(c)$.

Proposition 1 explains analytically why the single-server family (SAO, $c = 1$) is dominated by every multi-car family ($c = E$): with λ held fixed, moving from one server to servers can only reduce the expected wait. The same monotonicity underlies the period-stability behavior discussed in Section VI.

3.2. Single Automatic Operation (SAO)

SAO serves one call at a time on a first-come, first-served basis [4]. It is modeled as an M/M/1 queue. The service rate depends on the average travel distance of about N/2 floors:

$$\mu_{\text{SAO}} = \frac{60}{\frac{N}{2}t_f + 2t_d} \tag{1}$$

With traffic intensity $\rho = \lambda/\mu$ and $\rho < 1$ for stability, the mean wait in the queue is $W_q^{\text{SAO}} = \rho/[\mu(1-\rho)]$ and the total mean time adds the service time. SAO is simple, but its wait grows without bound as $\rho \to 1$.

3.3. Selective Collective Operation (SCO)

SCO serves every call in the current direction before reversing, like the LOOK disk-scheduling rule [14]. Calls are split by direction into $Q_{\text{up}}(t)$ and $Q_{\text{down}}(t)$, and the next stop is the nearest call ahead. Directional grouping lets every car serve several calls per pass, so the authors model SCO as an M/M/c queue over all E cars with a reduced per-call efficiency and service rate.

$$\mu_{\text{SCO}} = \frac{60}{0.65 \cdot \frac{N}{2}t_f + 2t_d} \tag{2}$$

The Erlang-C formula gives the probability that a new call waits:

$$C(c, a) = \frac{\frac{a^c}{c!}\frac{1}{1-a/c}}{\sum_{k=0}^{c-1}\frac{a^k}{k!} + \frac{a^c}{c!}\frac{1}{1-a/c}} \tag{3}$$

where $a = \lambda/\mu_{\text{SCO}}$ is the offered load. The mean queue wait is $W_q^{\text{SCO}} = C(c, a)/(c\,\mu_{\text{SCO}} - \lambda)$, and the total mean wait adds the service term$1/\mu_{\text{SCO}}$. SCO is modeled with c = E collective servers operating at a lower per-call efficiency than Group Control, rather than $c = \lfloor E/2 \rfloor$, because in practice every car participates in collective service.

3.4. Group Control Operation (GCO)

GCO uses all E cars as a coordinated pool, tracking the state $\Psi(t) = \left\{\left(S_j(t), D_j(t), L_j(t)\right)\right\}$, where $L_j(t)$ Is the load of the car j. A new call goes to the car that can reach it fastest:

$$j^* = \arg\min_{j=1,\dots,E} T_{\text{est}}(j, f, t), \tag{4}$$

and the controller minimizes the average wait $\overline{W} = \frac{1}{M}\sum_{i=1}^{M} w_i$ subject to $L_j(t) \leq C\mathrm{L}_{\mathrm{j}}$. Using all cars typically cuts the wait by thirty to forty-five percent against SCO [16].

3.5. Destination Control System (DCS)

With DCS, passengers register their floor before boarding, so the controller can batch passengers with shared destinations [20], [21]. With a stop-reduction factor $\beta_{\text{DCS}} \approx 0.40$,

$$\mu_{\text{DCS}} = \frac{60}{\beta_{\text{DCS}} \cdot \frac{N}{4}t_f + 1.5t_d} \tag{5}$$

DCS gives the highest throughput of any fixed-rule method, and its dispatching can be cast as a mixed-integer assignment problem [11], [12].

3.6. Fuzzy Logic Control (FLC)

FLC handles uncertain demand with fuzzy rules [13]. Inputs such as load, wait, position, and direction are fuzzified, combined through a rule base with Mamdani inference, and defuzzied into a crisp action. FLC gives about fifteen percent shorter wait than Group Control at medium load.

3.7. Neural Network Control

A neural network learns the map from the system state to dispatch action from past traffic [7]. Training minimizes the error between the network output and an ideal action. Once trained, the network handles new traffic well without needing destination entry.

3.8. Genetic Algorithm Control (GA)

A genetic algorithm encodes a dispatch schedule as a chromosome and evolves it [6], [15]. The normalized fitness to minimize is

$$F(\mathbf{C}) = \alpha \frac{\overline{W}_t}{\overline{W}_{\max}} + \beta \frac{\overline{T}_t}{\overline{T}_{\max}} + \gamma \frac{\overline{E}_c}{\overline{E}_{\max}}, \tag{6}$$

where $\overline{W}_t$,$\overline{T}_t$ and $\overline{E}_c$ are the mean wait, mean travel time, and mean energy of a candidate schedule, and $\overline{W}_{\max}$, $\overline{T}_{\max}$, and $\overline{E}_{\max}$ are normalizing constants (the worst observed values in the current population) that make the three terms dimensionless and comparable; the weights satisfy $\alpha + \beta + \gamma = 1$. GA gives near-optimal schedules when traffic is steady but adapts slowly to surges.

3.9. Model Predictive Control (MPC)

MPC solves a finite-horizon optimization at each step [17]. Over horizon H, it minimizes.

$$J = \sum_{i=t}^{t+H} \left[C\big(\Psi(i), \Phi(i)\big) + \lambda_E\, E\big(\Psi(i)\big) \right], \tag{7}$$

where $\Phi(i)$ is the set of outstanding calls at step i, $C(\cdot)$ is the waiting cost over those calls, and $E(\cdot)$ is the energy cost of the planned moves. Only the first action of the optimized sequence is applied, and the problem is re-solved at the next step with fresh measurements (receding horizon).

3.10. Reinforcement Learning Control (RL)

RL models the problem as a Markov Decision Process and learns a policy that maximizes long-run reward [5], [19]. The state encodes car positions, loads, and call queues; the action assigns a car to a call; the reward penalizes waiting time and energy; and $\gamma \in (0,1)$ is the discount factor that weights future reward. The optimal action-value obeys the Bellman equation.

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s'} P\,(s'|s,a) \max_{a'} Q^*(s',a') \tag{8}$$

Deep variants such as DQN and A3C handle large state spaces and keep improving with experience [7], [24], [25]. Under standard assumptions (finite state-action space, ergodicity, and a sufficiently small learning rate), the tabular Q-learning update converges to Q^* with probability 1 [19]; for deep variants, convergence is approximate and requires additional assumptions on the function-approximation architecture and exploration schedule.

3.11. Hybrid Control

A hybrid system keeps a set of strategies and selects the best for the current moment by minimizing a combined cost. $J_\sigma(t) = \alpha W(t) + \beta E(t) + \gamma Q(t)$[18]. It may pick DCS at peak, MPC when quiet, and safe fixed rules in emergencies, with a learning meta-controller choosing among them.

4. Comparative Analysis

Table 2 maps each family to its queueing model and computational cost. Table 3 compares the families by era, technique, strengths, and limitations.

Table 2. Queueing model and complexity of each algorithm family

Family	Queue Model	Technique	Complexity
SAO	M/M/1	FCFS	O(1)
SCO	M/M/c	LOOK	O(n log n)
Group Control	M/M/c	Time estimate	O(E·N)
DCS	M/G/c	Assignment	$O(N^2)$
Fuzzy Logic	Fuzzy queue	Rule inference	O(k)
Neural Network	M/M/c adaptive	Gradient	O(m·n)
Genetic Algorithm	M/M/c offline	Evolutionary	O(pop·gen)
MPC	M/G/c predictive	Horizon QP	$O(n^3)$
RL Adaptive	MDP	Policy opt.	O(|S|·|A|)
Hybrid	Composite	Meta-select	O(K)

Table 3. Comparative Summary of Elevator Control Algorithm Families

Family	Era	Core Technique	Strengths	Limitations
SAO	1950s	First-come, first-served	Very simple, cheap, predictable	Poor throughput; wait grows fast under load
SCO	1960s	Directional collective (LOOK)	Reliable; serves many calls per pass	Still limited at high demand
Group Control	1980s	Coordinated multi-car dispatch	Good throughput; uses all cars	Needs a central controller; more complex
DCS	2000s	Destination pre-registration	Best fixed-rule throughput; batches trips	Needs passenger input; higher install cost
Fuzzy Logic	1990s	Mamdani rule inference	Handles uncertainty; smooth tuning	Rule base is hand-crafted
Neural Network	1990s–2020s	Supervised the learning of dispatch	Generalizes to new traffic	Needs training data; less transparent
Genetic Algorithm	2004	Evolutionary schedule search	Near-optimal offline schedules	Slow to adapt to sudden surges
MPC	2000s–now	Receding-horizon optimization	Energy awareness; predictive	Heavy computation per step
RL Adaptive	1998–now	Markov decision process learning	Adapts and keeps improving	Long training; careful reward design
Hybrid	present	Meta-selection of strategies	Best overall; robust across regimes	Most complex to design and verify

5. Analytical Queueing Comparison

To make this comparison practical and reproducible, the reference building is defined as having fifteen floors, five elevators each holding twelve passengers, with a travel time between floors of three seconds and a door opening time of four seconds. The analytical queueing models are then tested from each family (Section III) against a realistic call arrival pattern of twenty-four hours. To clarify the scope of this comparison, it is necessary to stress that this is neither a simulation of elevator dynamics nor an agent-based simulation, but a comparison of analytical queueing models with uncertainty analysis. Hence, this approach is referred to as an analytical queueing comparison, not a Monte Carlo simulation of elevator dynamics.

5.1. Call-Arrival Profile

The call rate fluctuates hourly according to the pattern $\lambda(h) = \lambda_0\, \mu(h)$, with the rate λ0 being 2.6 calls per minute at a busy mid-rise office building. The multipliers include: μ = 1.45 (up-peak in the morning), 1.28 (lunch time), 1.00 (base afternoons), 1.40 (down-peak in the evening), and 0.30 (night). The load lies within the stable zone of each family while preserving the single-server SAO near saturation at its peak.

5.2. Explicit Family Models and Assumptions

For the four basic families, the waiting function $f_{\text{alg}}(\lambda)$ is calculated directly according to the queue models described in Eqs. (1) - (5). For the six advanced families, the waiting functions are given through the efficiency factor η that modifies the base queue; these families are described in detail in Table 4. It is worth stating that the above-mentioned efficiency factors are assumptions made based on the trends described in the experimental studies [7], [14], [17], [25] and not based on the data measured in this study. This is the key restriction in the performance comparison.

Table 4. Explicit model definitions for each family

Family	Base queue/source	Efficiency η
SAO	M/M/1, single fast server	—
SCO	M/M/c, c=E collective	—
Group Control	M/M/c, c=E	—
DCS	M/G/c, c=E, β=0.40	—
Fuzzy Logic	Group base × η [14]	0.85
Neural Network	DCS base × η [7]	0.82
Genetic Algorithm	Group base × η [6]	0.90
MPC	DCS base × η [17]	0.88
RL Adaptive	DCS base × η [5]	0.78
Hybrid	DCS base × η [18]	0.72

5.3. Uncertainty Quantification

To show how parameter uncertainty propagates, each model output is resampled R = 1000 times with multiplicative noise of eight percent standard deviation and the mean and standard deviation band are reported:

$$\overline{W}_{\mathrm{alg}}(h) = \frac{1}{R}\sum_{r=1}^{R} \max\left(1,\ f_{\mathrm{alg}}\big(\lambda(h)\big)\,(1+\varepsilon_r)\right), \quad \varepsilon_r \sim \mathcal{N}(0, 0.08^2) \tag{9}$$

The resampling quantifies sensitivity to the model parameters; it does not add dynamical realism. All counts, code, and the resulting tables are provided as supplementary material for full reproducibility.

6. Results and Discussion

The experimental results are supported by both graphical and tabular data. In direct response to the requirement for editable display items, every results figure is accompanied by an editable source table: the period-average waiting times of Figure 4 are listed in Table 5, and the energy and handling-capacity values of Figures 6 and 7 are listed in Table 6. All tables are typeset as native text, not images, so the numerical data remains fully editable.

6.1. Waiting Times Across the Day

Hourly average waiting times are displayed in Figure 3. Three clear peaks occur: morning, afternoon, and evening. SAO exceeds the ISO 4190 standard of thirty seconds, while the other families remain below fifteen seconds.

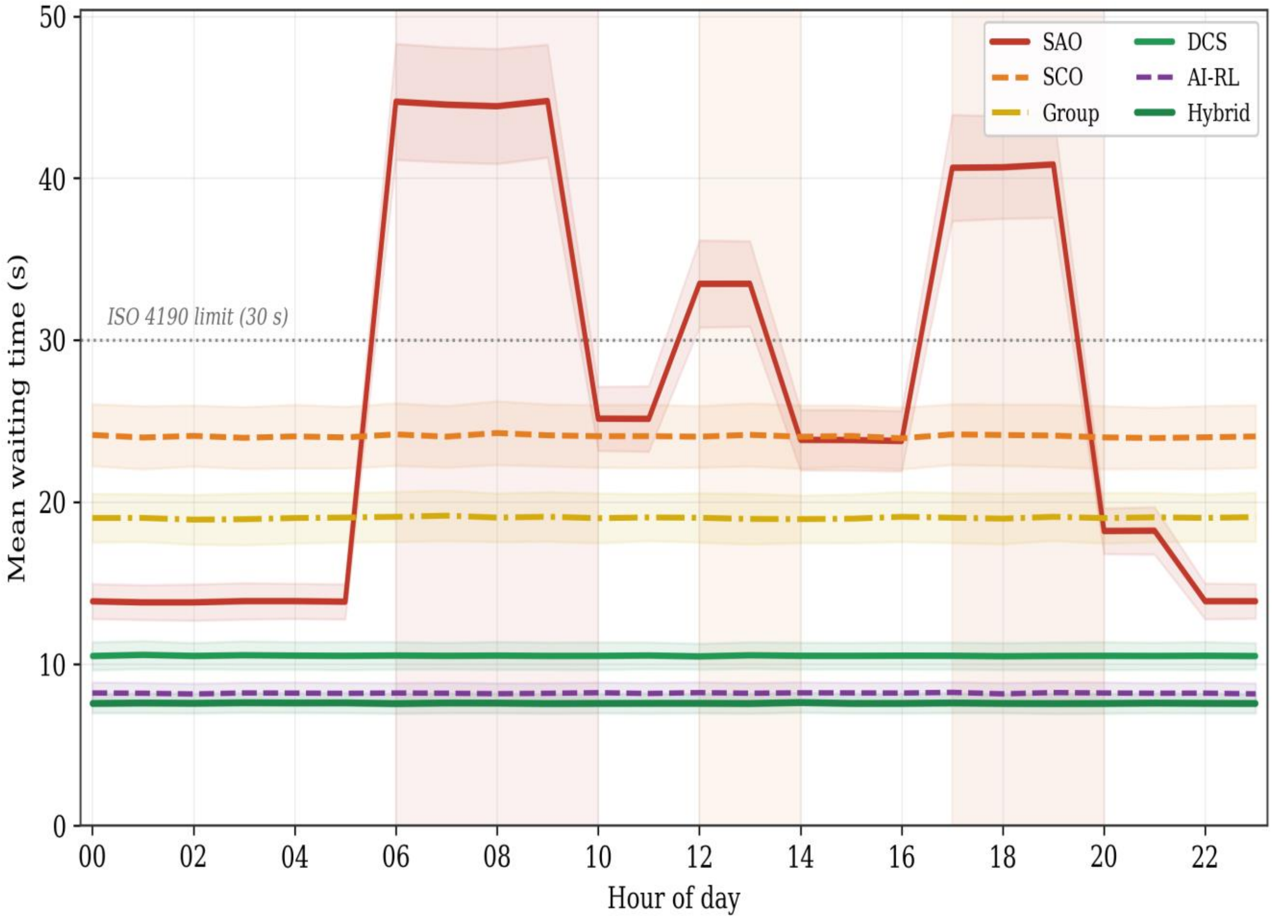

Fig. 3 Mean waiting times across a 24-hour cycle for six representative families, with ±1 standard deviation bands from 1000 parametric resamples

6.2. Average Wait by Period

Figure 4 and Table 5 average the wait over five daily periods. The performance hierarchy remains constant: Hybrid is consistently quickest, and SAO is consistently slowest. The Hybrid model reduces morning-peak wait time by approximately eighty-three percent compared to SAO, while DCS reduces it by seventy-seven percent.

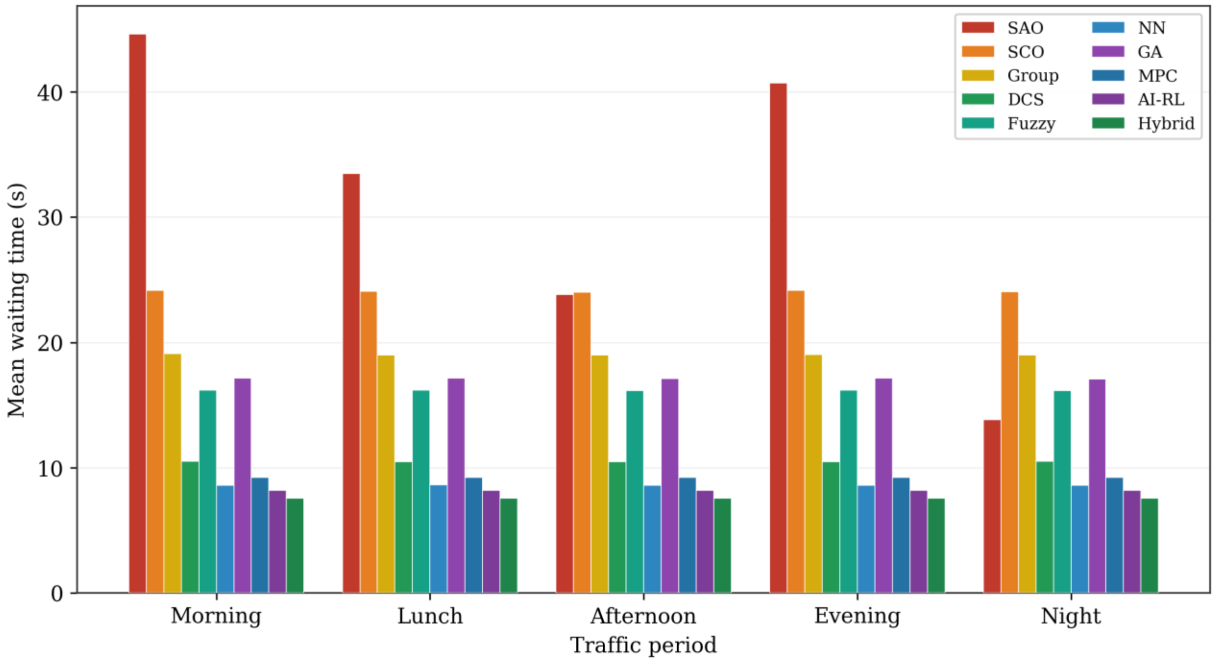

Fig. 4 Average wait by traffic period for all ten families

Table 5. Mean Waiting Times (s) by Traffic Period and Reduction vs. SAO at Morning Peak

Family	Morn.	Lunch	Afternoon	Evening	Night	% red.
SAO	44.6	33.5	23.8	40.7	13.8	—
SCO	24.1	24.1	24.0	24.1	24.0	45.9%
Group Control	19.1	19.0	19.0	19.0	19.0	57.2%
DCS	10.5	10.5	10.5	10.5	10.5	76.5%
Fuzzy Logic	16.2	16.2	16.1	16.2	16.2	63.7%
Neural Network	8.6	8.6	8.6	8.6	8.6	80.7%
Genetic Algorithm	17.1	17.1	17.1	17.2	17.1	61.6%
MPC	9.2	9.2	9.2	9.2	9.2	79.3%
RL Adaptive	8.2	8.2	8.2	8.2	8.2	81.7%
Hybrid	7.6	7.6	7.6	7.6	7.6	83.1%

A particular feature of Table V that could be considered somewhat paradoxical is that the multi-server families (SCO, Group Control, DCS, and Hybrid) have the lowest degree of variation throughout the day, while the SAO family demonstrates the highest variation, falling from 44.6 seconds in the morning peak to 13.8 seconds at night. However, such a distribution is not accidental; it can be easily explained with the help of the queuing theory model. As can be seen, the SAO family models the single-server queue, which approaches saturation in the peak period.

At the same time, the multi-server queues have an equal rate of arrival and, therefore, have enough extra capacity. Thus, the waiting times for such vehicles are determined mostly by the constant service time, $1/\mu$. That is why the waiting time does not depend on variations in the traffic rate to a large extent. In other words, the key problem in multi-vehicles and smart-control scenarios is reducing traffic sensitivity, which adversely affects the single-vehicle queue. This idea can be easily observed from Figure 3.

6.3. Spread and Significance

Figure 5 represents the spread of waiting times at the morning peak. A one-way ANOVA on the ten families yields $F(9,9990) = 42,912$ and a p-value below 10^{-300}, confirming that the deterministic model outputs are statistically distinct. This significance indicates that the relative ordering of the algorithms is robust against parameter variations within the defined model.

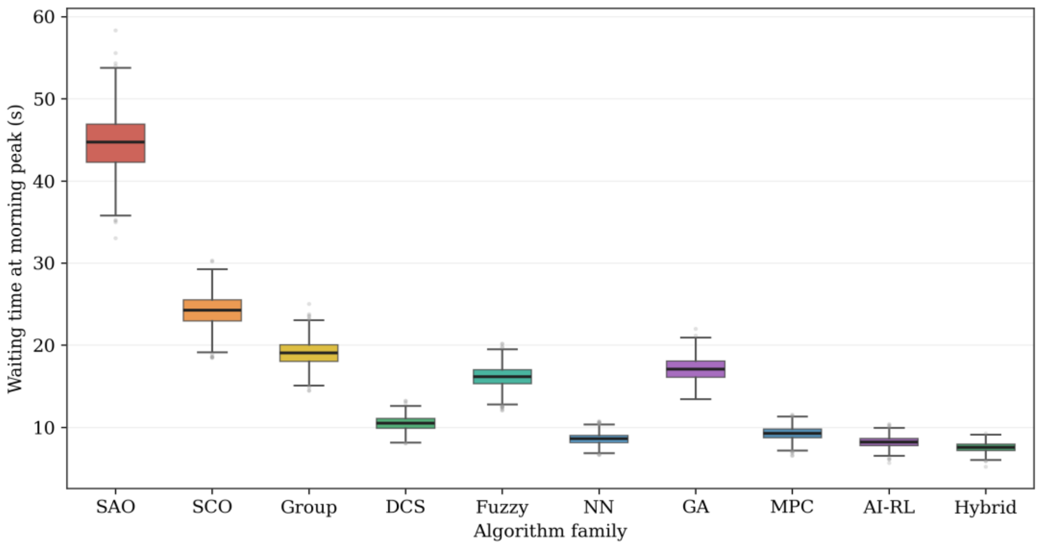

Fig. 5 Waiting time distributions at the morning peak over 1000 resamples. The box is the interquartile range, and the whiskers span the 5th to 95th percentile

6.4. Energy and Handling Capacity

Energy consumption at each stop is estimated in Fig. 6 and Table 6. The MPC, DCS, and Hybrid belong to the Class A range of the VDI 4707 standard, while SAO and SCO demonstrate higher values due to unnecessary travel. In Fig. 7, Handling capacity is formally defined as the percentage of the building's total occupant population that the elevator system can transport within a five-minute interval during peak traffic, computed from the analytical round-trip time of the queueing model [29]; SAO is below the ISO 4190 threshold of 12%, while the learning-based families exceed 17%.

Table 6. Estimated Energy per Stop (Wh) and Handling Capacity

Family	Peak (Wh)	Off-peak (Wh)	Handling (%)
SAO	21.1	7.2	6
SCO	15.8	5.4	9
Group Control	12.3	4.2	13
DCS	8.8	3.0	16
Fuzzy Logic	11.4	3.9	12
Neural Network	8.4	2.9	17
Genetic Algorithm	10.5	3.6	11
MPC	7.9	2.7	15
RL Adaptive	7.4	2.5	18
Hybrid	7.0	2.4	20

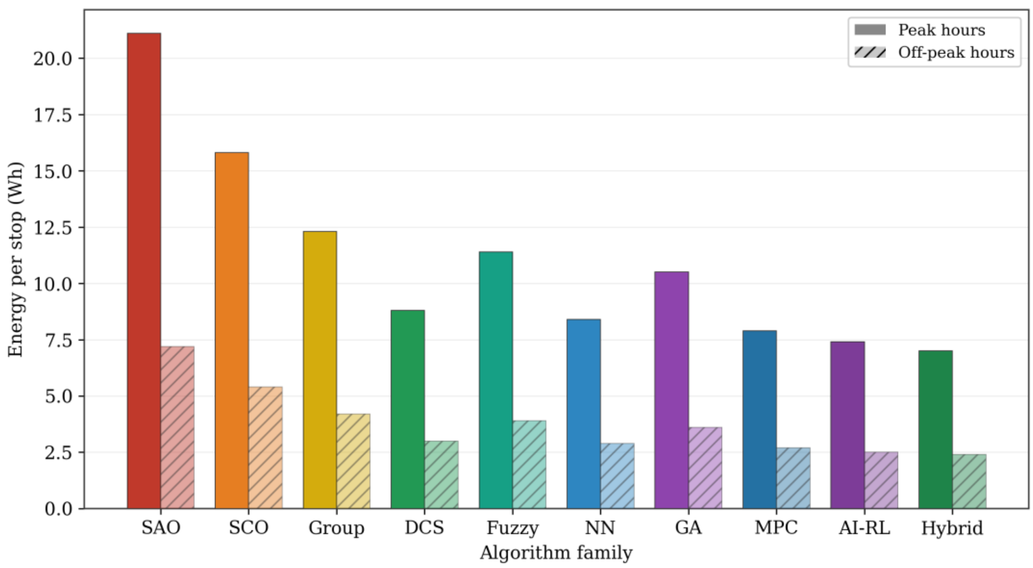

Fig. 6 Estimated energy use per stop at peak and off-peak

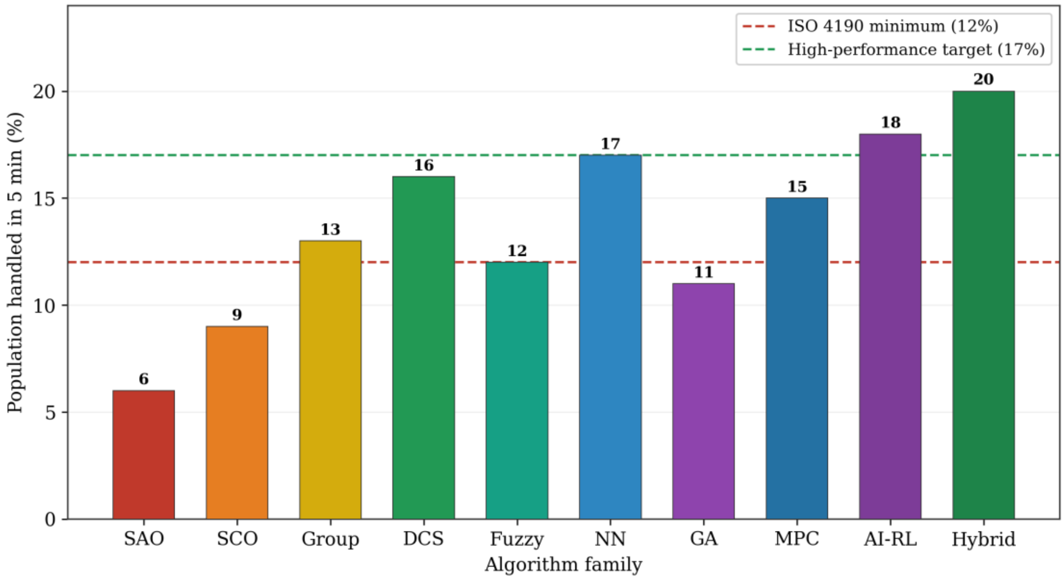

Fig. 7 Handling capacity as the percent of building population moved in five minutes, against ISO benchmarks

6.4.1. Discussion on State-of-the-Art Comparison

The analytical advantage of learning-based and hybrid families over classical methods admits a precise explanation. Two mechanisms operate jointly. The first is structural and is captured by Proposition 1: coordinating all cars as parallel servers, rather than dispatching a single car, can only reduce the expected queueing wait. This bounds the achievable wait from below for every multi-car family relative to SAO. The second mechanism is the effective increase in service rate: by optimizing the state-action mapping over a receding horizon (MPC) or by exploiting a learned value function (RL), these controllers reduce empty-car travel and raise batching efficiency, which in the model corresponds to a smaller efficiency factor and hence a faster effective service time. The combination of more servers (Proposition 1) and a higher effective service rate explains why the advanced families simultaneously achieve the shortest waits, the highest handling capacity, and the lowest energy per stop, whereas the rigid thresholds of SAO and SCO cannot exploit either mechanism. This is consistent with the empirical gains reported for deep actor-critic and occupancy-aware controllers in the recent literature [7, 25].

6.5. Sensitivity

Figure 8 demonstrates that SAO degrades rapidly as building height increases, while DCS and Hybrid scale gracefully; the figure uses a logarithmic vertical axis and marks the point beyond which the single-server SAO queue becomes unstable. Figure 9 confirms that simple methods reach analytical congestion limits long before advanced methods as arrival rates climb.

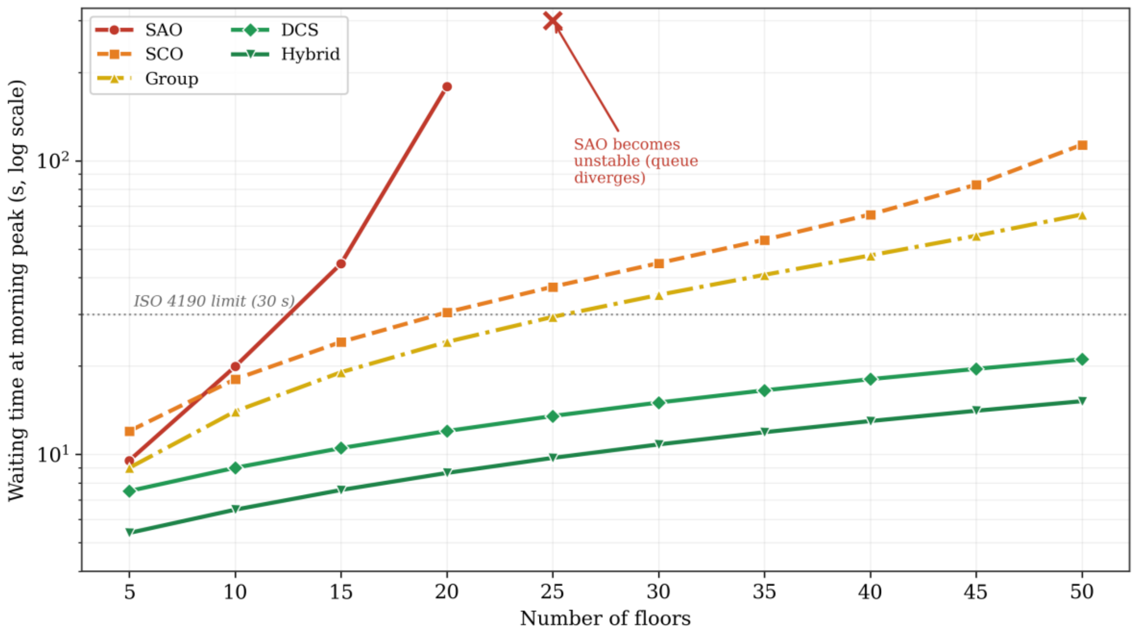

Fig. 8 Waiting time versus building height at the morning peak.

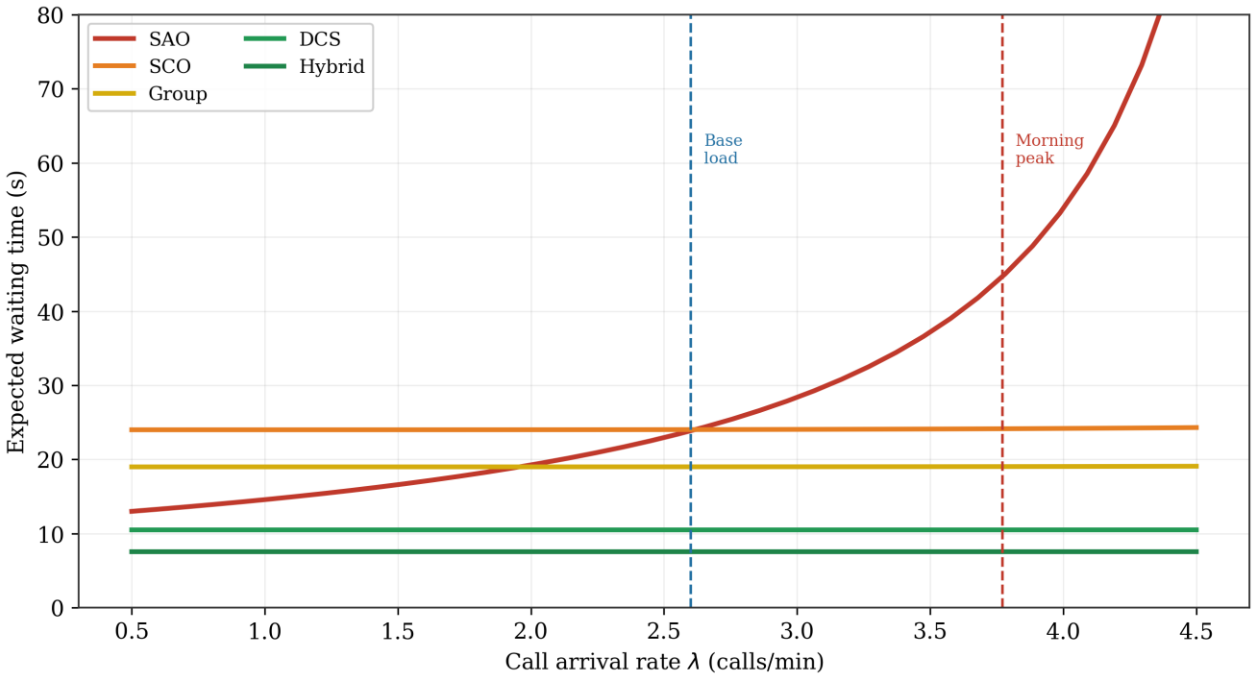

Fig. 9 Waiting time versus arrival rate, with peak and base markers.

6.6. Multi-Criteria Synthesis

Figure 10 consolidates six evaluation criteria for five representative families. The performance envelope expands from SAO toward Hybrid on every axis except implementation simplicity, where the single-server design retains an advantage. Among the non-hybrid families, MPC attains the strongest energy score and reinforcement learning the strongest adaptivity score, consistent with the analytical ordering established by Proposition 1.

.

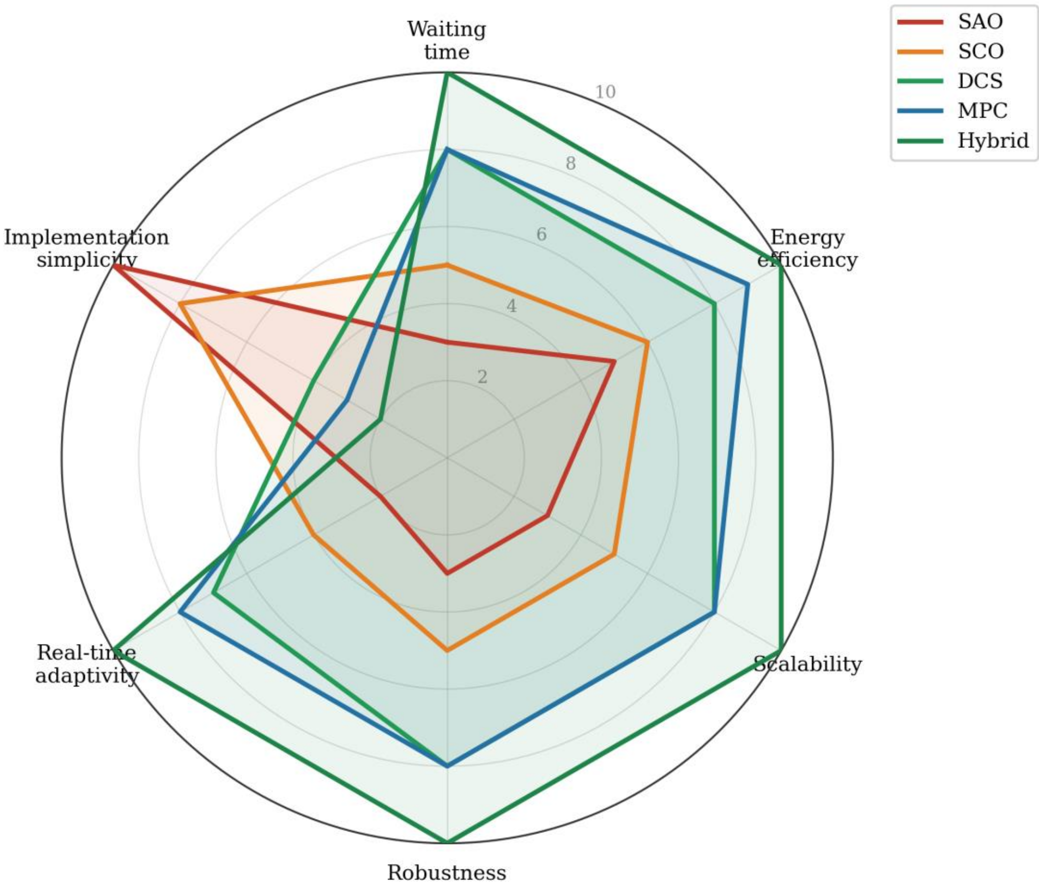

Fig. 10 multi-criteria radar chart for five families (0–10, higher is better).

7. Research Gaps and Future Directions

The four remaining open problems define the field to some extent. Firstly, there is the problem of empirical validation. Since many works based on learning algorithms provide results that were achieved through simulation and not actual measurements in functioning buildings, it is hard to compare the gains obtained by different studies in terms of one common scale, as the coefficients used by advanced families of learning algorithms reviewed above have the same validation problem. Another difficulty connected to empirical validation is the lack of standard benchmarks because there is no common open database and no commonly recognized simulator based on discrete events. The last two open problems concern the energy consumption-ride comfort vs. waiting time trade-off and deployment.

Future work should replace efficiency heuristics with data through the use of true discrete event simulation of SCO, GCO, and DCS logic, along with reinforcement learning (RL) agents and model predictive control (MPC) optimization algorithms. In addition to the above, it is important that the community move towards developing open benchmarks based on real-world traffic traces, safe RL policies for emergency situations, and hardware in the loop testing of hybrid controllers.

8. Conclusion

The current research reviews ten families of elevator control algorithms, classified into classical, computational Intelligence, and learning approaches. With the help of a common set of queueing models and an analytical comparison in terms of a sample building, the theoretical performance bounds of these families are assessed. The order produced by the comparison is clear-cut – hybrid and learning families provide the best performance in terms of waiting time reduction and energy efficiency, while the single-car SAO algorithm is unable to satisfy the ISO 4190 criterion during traffic peaks. It should be specified that the efficiency values of advanced families are calculated based on the literature data trends, and the comparison itself is made by means of queueing analysis, not by simulation. As a result, the figures presented are the relative theoretical baselines. The suggested classification, along with the comparative tables and the outlined research issues, becomes the basis for further development of vertical transportation control.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding Statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgement

The authors thank Vidyalankar Institute of Technology for its support in conducting research and for its support under its higher study Policy.

References

[1] International Organization for Standardization, ISO 4190-1: Lifts (Elevators) and Service Lifts-Part 1: Planning and Selection, Geneva, Switzerland, 2010. [Publisher Link]

[2] Verein Deutscher Ingenieure, VDI 4707 Part 1: Energy Efficiency of Lifts, Düsseldorf, Germany, 2009. [Publisher Link]

[3] Gina C. Barney, and Lutfi Al-Sharif, *Elevator Traffic Handbook: Theory and Practice*, 2nd Ed., London, U.K.: Routledge, 2015. [Google Scholar] [Publisher Link]

[4] George R. Strakosch, The Vertical Transportation Handbook, 4th Ed., Hoboken, NJ, USA: Wiley, 2010. [Google Scholar] [Publisher Link]

[5] Robert H. Crites, and Andrew G. Barto, "Elevator Group Control using Multiple Reinforcement Learning Agents," *Machine Learning*, vol. 33, no. 2-3, pp. 235-262, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[6] Pablo Cortés, J. Larrañeta, and Luis Onieva, "Genetic Algorithm for Controllers in Elevator Groups: Analysis and Simulation during Lunchpeak Traffic," *Applied Soft Computing*, vol. 4, no. 2, pp. 159-174, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[7] Qinglai Wei et al., "Optimal Elevator Group Control via Deep Asynchronous Actor-Critic Learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5245-5256, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] David Pepyne, and Christos G. Cassandras, "Optimal Dispatching Control for Elevator Systems during Uppeak Traffic," *IEEE Transactions on Control Systems Technology*, vol. 5, no. 6, pp. 629-643, 1997. [CrossRef] [Google Scholar] [Publisher Link]

[9] David Pepyne, and Christos G. Cassandras, "Design and Implementation of an Adaptive Dispatching Controller for Elevator Systems during Uppeak Traffic," *IEEE Transactions on Control Systems Technology*, vol. 6, no. 5, pp. 635-650, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[10] Daniel Nikovski, and Matt Brand, "Exact Calculation of Expected Waiting Times for Group Elevator Control," *IEEE Transactions on Automatic Control*, vol. 49, no. 10, pp. 1820-1823, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[11] Mirko Ruokokoski et al., "Assignment Formulation for the Elevator Dispatching Problem with Destination Control and Its Performance Analysis," *European Journal of Operational Research*, vol. 252, no. 2, pp. 397-406, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[12] Mirko Ruokokoski, Harri Ehtamo, and Panos M. Pardalos, "Elevator Dispatching Problem: A Mixed Integer Linear Programming Formulation And Polyhedral Results," *Journal of Combinatorial Optimization*, vol. 29, no. 4, pp. 750-780, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[13] ChangBum Kim et al., "Design and Implementation of a Fuzzy Elevator Group Control System," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 28, no. 3, pp. 277-287, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[14] Joaquin R. Fernandez et al., "Dynamic Fuzzy Logic Elevator Group Control System With Relative Waiting Time Consideration," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 9, pp. 4912-4919, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[15] Berna Bolat et al., "Optimal Car Dispatching for Elevator Groups Using Genetic Algorithms," *Intelligent Automation & Soft Computing*, vol. 16, no. 1, pp. 89-99, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[16] Marja-Liisa Siikonen, "*Planning and Control Models for Elevators in High-Rise Buildings*," Ph.D. Dissertation, Espoo, Finland, 1997. [Google Scholar] [Publisher Link]

[17] Eduardo F. Camacho, and Carlos Bordons, *Model Predictive Control*, 2nd Ed., 2007. [CrossRef] [Google Scholar] [Publisher Link]

[18] Bao Ding et al., "A Hybrid Approach for the Analysis and Prediction of Elevator Passenger Flow in an Office Building," *Automation in Construction*, vol. 35, pp. 69-78, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[19] Richard S. Sutton, and Andrew G. Barto, *Reinforcement Learning: An Introduction*, 2nd Ed., Cambridge, MA, USA: MIT Press, 2018. [Google Scholar] [Publisher Link]

[20] Lutfi Al-Sharif et al., "Establishing the Upper Performance Limit of Destination Elevator Group Control using Idealised Optimal Benchmarks," *Building Services Engineering Research & Technology*, vol. 36, no. 5, pp. 546-566, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[21] Jörg Lauener, Traffic Performance of Elevators with Destination Control, Elevator World, vol. 55, no. 9, 2007. [Google Scholar]

[22] Jinglong Zhang, and Qun Zong, "Energy-Saving-Oriented Group-Elevator Dispatching Strategy for Multi-Traffic Patterns," *Building Services Engineering Research & Technology*, vol. 35, no. 5, pp. 543-568, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[23] Mo Shi et al., "Prediction and Analysis of Elevator Traffic Flow under the LSTM Neural Network," *Intelligent Control and Automation*, vol. 15, no. 2, pp. 63-82, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[24] Nathan Vaartjes, and Vincent Francois-Lavet, "Novel RL Approach for Efficient Elevator Group Control Systems," *arXiv preprint arXiv Preprint*, pp. 1-15, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[25] Shu Wang et al., "Smart Dispatching and Optimal Elevator Group Control through Real-Time Occupancy-Aware Deep Learning of Usage Patterns," *Advanced Engineering Informatics*, vol. 48, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[26] Matthew J Page et al., "The PRISMA 2020 Statement: An Updated Guideline for Reporting Systematic Reviews," *BMJ*, vol. 372, pp. 1-9, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[27] Leonard Kleinrock, *Queueing Systems*, Volume 1: Theory. New York, NY, USA: Wiley-Interscience, 1975. [Google Scholar] [Publisher Link]

[28] Gordon Frank Newell, *Applications of Queueing Theory*, 2nd Ed., London, U.K.: Chapman and Hall, 1982. [Google Scholar] [Publisher Link]

[29] Lutfi Al-Sharif, Ahmad M Abu Alqumsan, and Rasha Khaleel, "Derivation of a Universal Elevator Round Trip Time Formula under Incoming Traffic," *Building Services Engineering Research & Technology*, vol. 35, no. 2, pp. 198-213, 2015. [CrossRef] [Google Scholar] [Publisher Link]