УДК 519.872

АНАЛИЗ МНОГОЛИНЕЙНОЙ СИСТЕМЫ С МАРКОВСКИМ ВХОДНЫМ ПОТОКОМ И ДИНАМИЧЕСКИМ ЦЕНООБРАЗОВАНИЕМ

О. С. ДУДИНА[1)]***, С. А. ДУДИН***[1)]

[1)]*Белорусский государственный университет, пр. Независимости, 4, 220030, г. Минск, Беларусь*

Аннотация. Исследована математическая модель многолинейной системы массового обслуживания с марковским входным процессом (*Markovian arrival process*, MAP) и механизмом адаптивного управления доступом, основанным на динамическом ценообразовании и политике «отпусков» приборов. Поведение системы описано трехмерной цепью Маркова с непрерывным временем, выписан ее инфинитезимальный генератор. Приведены явные формулы для основных характеристик производительности, включая среднее число активных и занятых приборов, вероятности потерь. Численные эксперименты показывают, что использование динамического ценообразования и управления ресурсами имеет решающее значение для баланса между качеством обслуживания и операционной эффективностью. Кроме того, определена оптимальная стратегия ценообразования, которая максимизирует экономическую прибыль системы за счет управления балансом между перегрузкой и отказами клиентов.

Ключевые слова: динамическое ценообразование; марковский входной процесс; управление доступом; политика «отпусков» приборов; служба такси.

ANALYSIS OF A MULTI-SERVER SYSTEM WITH MARKOVIAN ARRIVAL PROCESS AND DYNAMIC PRICING

O. S. DUDINA[a]***, S. A. DUDIN***[a]

[a]*Belarusian State University, 4 Niezaliezhnasci Avenue, Minsk 220030, Belarus*

Corresponding author: O. S. Dudina (dudina@bsu.by)

Abstract. This paper presents a mathematical model of a multi-server queuing system which integrates a Markovian arrival process (MAP) with an adaptive admission control mechanism based on dynamic pricing and a comprehensive server vacation policy. By formulating the system dynamics as a three-dimensional continuous-time Markov chain, we derive its infinitesimal. We provide explicit formulas for key performance measures, including average numbers of active

Образец цитирования:
Дудина ОС, Дудин СА. Анализ многолинейной системы с марковским входным потоком и динамическим ценообразованием. *Журнал Белорусского государственного университета. Математика. Информатика.* 2026;2:97–109 (на англ.).
EDN: TPYPWY

For citation:
Dudina OS, Dudin SA. Analysis of a multi-server system with Markovian arrival process and dynamic pricing. *Journal of the Belarusian State University. Mathematics and Informatics.* 2026; 2:97–109.
EDN: TPYPWY

Авторы:
Ольга Сергеевна Дудина – кандидат физико-математических наук; ведущий научный сотрудник научно-исследовательской лаборатории прикладного вероятностного анализа кафедры теории вероятностей и математической статистики факультета прикладной математики и информатики.
Сергей Александрович Дудин – кандидат физико-математических наук; ведущий научный сотрудник научно-исследовательской лаборатории прикладного вероятностного анализа кафедры теории вероятностей и математической статистики факультета прикладной математики и информатики.

Authors:
Olga S. Dudina, PhD (physics and mathematics); leading researcher at the laboratory of applied probabilistic analysis, department of probability theory and mathematical statistics, faculty of applied mathematics and computer science.
dudina@bsu.by
https://orcid.org/0000-0002-6788-8783
Sergei A. Dudin, PhD (physics and mathematics); leading researcher at the laboratory of applied probabilistic analysis, department of probability theory and mathematical statistics, faculty of applied mathematics and computer science.
dudins@bsu.by
https://orcid.org/0000-0002-9439-8755

CC BY NC

and busy servers, and loss probabilities. Numerical experiments demonstrate that the synergy between dynamic pricing and resource management is crucial for balancing the quality of service and operational efficiency. Furthermore, we identify an optimal pricing strategy that maximises the system's economic profit by managing the trade-off between congestion and customer rejection.

Keywords: dynamic pricing; Markovian arrival process; admission control; server vacation policy; ride-hailing platform.

Introduction

Modern digital platforms for passenger transportation are complex dynamic ecosystems whose functioning relies heavily on the ability to effectively manage the real-time balance between the supply and demand under conditions of high uncertainty. The operational efficiency of such aggregators is directly determined by the quality of the analytical tools employed, which must adapt to the stochastic nature of incoming trip requests and the constantly shifting availability of resources on duty. In this study, the operation of a taxi service is modelled as a multi-server bufferless queuing system, where identical vehicles act as the service units. The absence of a waiting buffer in this model reflects the specific nature of mobile ride-hailing applications: if no free vehicle is available in the immediate vicinity at the moment of the request, the potential customer is typically unwilling to wait, abandons the search, or switches to a competitor's service, resulting in an immediate and permanent loss of the request for the system.

A key feature of the proposed approach is the detailed consideration of the human factor and economic levers for the demand management. Unlike classical models, this framework accounts for the variable presence of drivers on duty: they may temporarily leave the system for breaks and return to work based on the current fleet workload and the platform's operational strategy. This mechanism allows for an accurate representation of the flexible work schedules typical to modern aggregators. Simultaneously, the model incorporates an adaptive admission control mechanism through dynamic pricing. It is assumed that a potential customers decision to join the system depends on the offered fare: as the current price increases, the likelihood of the user confirming the trip request decreases. Consequently, the system does not merely distribute requests passively but actively regulates the incoming traffic intensity, balancing between profit maximisation and the risk of customer rejection due to excessive pricing.

The literature on queuing systems with dynamic pricing is already quite extensive; therefore, we do not aim to provide a comprehensive survey, citing only a few key works instead.

Paper [1] is frequently cited as the pioneering work on optimal dynamic pricing policies. The author considers the problem of maximising the expected reward in an $M/M/s$-type queue, where the service cost is adjusted to encourage or discourage arrivals. The price is selected from a finite set at each moment the number of users in the system changes. Assuming all underlying distributions are exponential, the study employs the theory of semi-Markov decision processes. It is shown that the optimal strategy is monotonic, specifically, the optimal price is a non-decreasing function of the number of users. Furthermore, a computational algorithm for determining the optimal strategy parameters (thresholds) is provided.

In work [2], a rental firm with two classes of users, contract and walk-in, is analysed. Contract users are managed via an admission policy, while walk-in users are controlled through pricing. The authors demonstrate that optimal policies are monotone with respect to the system parameters. A similar model is analysed in paper [3], which shows that if the revenue from walk-in users is sufficiently large, optimal policies are not always monotonic relative to the number of contract users. In publication [4], a general framework for investigating how optimal policies vary with system parameters is proposed within the context of event-based dynamic programming (see, for example, [5]).

The study by [6] is one of the few to consider non-constant arrival and service rates, assuming them to be bounded, periodic functions of time. Using the Markov decision processes approach, the authors show that under both infinite-horizon discounted and average reward criteria, the optimal pricing and admission control strategies are non-decreasing in the number of users for any fixed time. They propose a pointwise stationary approximation and a heuristic to improve its implementation, demonstrating its utility through numerical experiments. Additionally, paper [7] considers an inventory model with an MMPP (Markov-modulated Poisson process) demand flow under three pricing policies: static, state-dependent (based on the MMPP), and stock-dependent. Given the similarities between inventory and queuing models, hybrid queuing-inventory systems have become a prominent research subject.

Various aspects of the dynamic pricing in ride-hailing platforms are discussed in works such as [8–14]. Among these works, only paper [14] deals with correlated arrival flow of customers. This paper proposes a novel predictive dynamic pricing mechanism, where decision moments coincide with the transition epochs of the MAP's

underlying process. The authors analyse a multi-server retrial queuing model, where customers' decisions to join, balk, or postpone the service (enter an orbit) depend on the current price. The system's behaviour is described by a multidimensional Markov chain with state-inhomogeneous transitions. The main disadvantage of paper [14] is the assumption that the number of available cars is constant and does not change in time. In modern ride-hailing systems like Uber, the number of active cars is never static and fluctuates in real-time. The primary engine behind this is dynamic pricing. Because drivers operate as independent contractors, they have the freedom to choose when to log on or off. When the demand spikes during heavy rain, major events, or rush hour, the system automatically raises fares to attract more drivers start work. Drivers are naturally more motivated to work during these peak hours because the earning potential is significantly higher. This creates a self-regulating cycle: as prices go up, the supply of active cars increases to meet the demand. Once the rush subsides and the extra incentives disappear, many drivers head home, and the number of active vehicles drops. In this paper, we eliminate this disadvantage by considering a queuing model with variable number of active servers.

Our goal is to develop a comprehensive analytical model that allows calculating the key performance measures of the service. These include the probability of service denial due to a lack of available cars on the line and the average number of active drivers on duty. The model integrates a dynamic pricing mechanism that adjusts the flow of incoming requests based on vehicle availability. This not only optimises the aggregator's total profit but also helps in balancing the overall workload of the fleet. The results of our paper can be applied to determine the optimal pricing policy.

Beyond the passenger transportation sector, this mathematical framework can be effectively adapted to describe other shared-resource systems. Specifically, it is well-suited for cloud computing environments, where virtual servers are switched to energy-saving modes during periods of low demand. Furthermore, the model is highly relevant for next-generation wireless networks, where base stations dynamically adjust the number of active communication channels, and for modern call centres, where operators join the service or take breaks based on the current density of incoming calls.

Mathematical model

We consider the operation of a taxi service as a multi-server bufferless queuing system with N identical vehicles (servers). The structure of the system is illustrated in fig. 1.

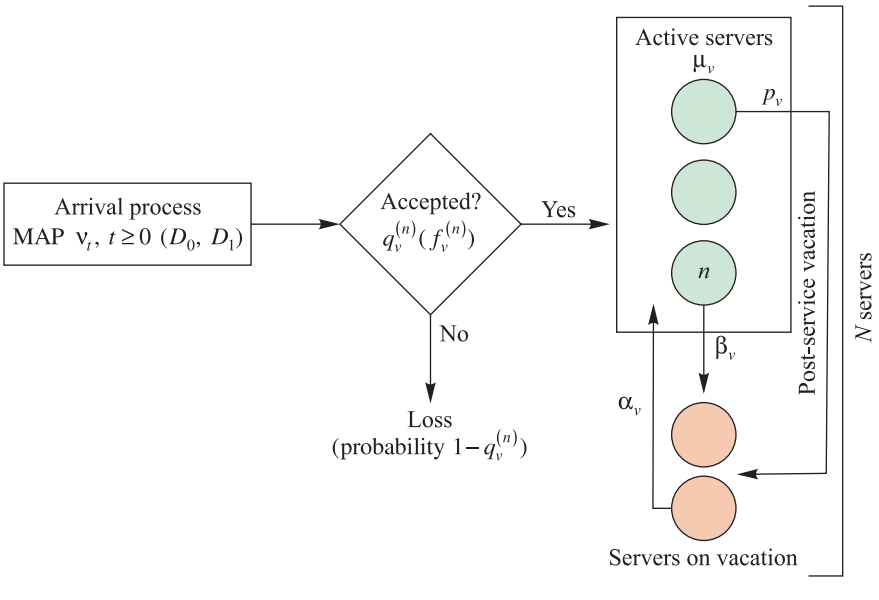


Fig. 1. The structure of the queue

The flow of incoming trip requests is modelled by a Markovian arrival process (MAP) defined by the matrix pair (D_0, D_1) of the dimension $W \times W$. This process is controlled by an irreducible underlying Markov chain ν_t, $t \geq 0$, with the state space $\{1, 2, \ldots, W\}$, which represents the current demand state (e. g., peak hours, weather conditions, or traffic density). D_0 contains the intensities of transitions that do not result in an arrival, and D_1 contains the intensities of transitions accompanied by a request generation. The fundamental arrival rate λ of trip requests is calculated using the formula

$$\lambda = \boldsymbol{\theta} D_1 \boldsymbol{e},$$

where $\boldsymbol{\theta}$ is the stationary state vector of the underlying Markov chain with generator $D = D_0 + D_1$, satisfying the equations

$$\boldsymbol{\theta}\left(D_0 + D_1\right) = \mathbf{0}, \ \boldsymbol{\theta}\boldsymbol{e} = 1,$$

where $\mathbf{0}$ is a row vector of zeros, and $\boldsymbol{e}$ is a column vector of ones. A more detailed description of the MAP can be found in paper [15].

The average arrival rate λ_v when the process ν_t is in the state v, $v = \overline{1, W}$, is defined as $\lambda_v = \left(D_1\right)_v \boldsymbol{e}$, where $\left(D_1\right)_v$ is the v^{th} row of the matrix D_1. We assume that the states of the MAP are numbered in the increasing order of intensities λ_v, i. e., $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_W$.

To manage the fleet workload and maximise the efficiency, the platform implements dynamic pricing (surge pricing). If a request arrives when the city is in demand state $v \in \{1, 2, \ldots, W\}$ and there are $n \in \{1, 2, \ldots, N\}$ active drivers on the line, the request is accepted for service with the probability $q_v^{(n)}$. This probability depends on the pricing coefficient $f_v^{(n)}$. Otherwise, with the probability $1 - q_v^{(n)}$ the request rejects the price and leaves the system. As the surge multiplier $f_v^{(n)}$ increases, the acceptance probability $q_v^{(n)}$ decreases, reflecting the customer's decision to reject the trip due to high cost. If all active vehicles are already occupied, the request is lost.

The number of available vehicles is dynamic, reflecting the behaviour of drivers who may switch between being on-duty and taking a vacation state.

1. Post-service vacation: upon completing a trip, a driver enters a vacation state with the probability p_v, $v = \overline{1, W}$. Otherwise, with the probability $1 - p_v$, they remain active and ready for the next ping.

2. Idle-to-vacation transition: an active but idle driver (waiting for an order) may go offline with the intensity β_v, $v = \overline{1, W}$. This mechanism models drivers logging off during low-demand periods.

3. Recovery: a driver currently on vacation returns to the active fleet with the intensity α_v, $v = \overline{1, W}$.

The trip duration (service time) follows an exponential distribution with rate μ_v, which accounts for the average travel speed in the current demand state v.

Process of system states

The behaviour of the system is described by a regular irreducible continuous-time Markov chain

$$\xi_t = \{n_t, m_t, \nu_t\}, \ t \ge 0,$$

defined on the state space

$$\mathcal{S} = \left\{\left(n_t, m_t, \nu_t\right) : 0 \le n_t \le N, \ 0 \le m_t \le n_t, \ 1 \le \nu_t \le W\right\}, \tag{1}$$

where at the time t, $t \ge 0$, n_t is the number of available active servers (active drivers); m_t is the number of busy servers (occupied vehicles); ν_t is the state of the underlying process of the MAP (the current demand state of the city).

By enumerating the state space (1) lexicographically, we group individual states into macro-states $\left(n, m\right)$, where the first two components are fixed. The set of states characterised by a fixed value n of the first component is defined as level n.

Theorem 1. *The infinitesimal generator Q of the process ξ_t, $t \ge 0$, has a block-tridiagonal structure*

$$Q = \begin{pmatrix} Q_{0,0} & Q_{0,1} & O & \ldots & O & O \\ Q_{1,0} & Q_{1,1} & Q_{1,2} & \ldots & O & O \\ O & Q_{2,1} & Q_{2,2} & \ldots & O & O \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ O & O & O & \ldots & Q_{N,N-1} & Q_{N,N} \end{pmatrix},$$

where non-zero blocks $Q_{n,k}$, $|n-k| \le 1$, $n, k = \overline{0, N}$, are defined by the following expressions.

1. Server recovery block ($k = n+1$):

$$Q_{n,n+1} = \left(N - n\right)\hat{I}_n \otimes \mathcal{A}, \ n = \overline{0, N-1},$$

where

$$\mathcal{A} = \operatorname{diag}\{\alpha_1, \ldots, \alpha_W\},$$

$$\hat{I}_n = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & 0 \end{pmatrix}_{(n+1)\times(n+2)}.$$

2. Server vacation block ($k = n-1$):

$$Q_{n,n-1} = \Delta_n^- \otimes \mathcal{B} + \Omega_n^- \otimes \mathcal{MP},\ n = \overline{1,\ N},$$

where

$$\mathcal{B} = \operatorname{diag}\{\beta_v,\ v = \overline{1,\ W}\},\ \mathcal{M} = \operatorname{diag}\{\mu_v,\ v = \overline{1,\ W}\},\ \mathcal{P} = \operatorname{diag}\{p_v,\ v = \overline{1,\ W}\},$$

$$\Delta_n^- = \begin{pmatrix} n & 0 & \ldots & 0 \\ 0 & n-1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \\ 0 & 0 & \ldots & 0 \end{pmatrix}_{(n+1)\times n}, \quad \Omega_n^- = \begin{pmatrix} 0 & 0 & \ldots & 0 \\ 1 & 0 & \ldots & 0 \\ 0 & 2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & n \end{pmatrix}_{(n+1)\times n}.$$

3. Diagonal block ($k = n$):

$$Q_{0,0} = D_0 + D_1 - N\mathcal{A},$$

$$Q_{n,n} = I_{n+1} \otimes D_0 + \tilde{I}_n \otimes (I_W - \mathcal{Q}_n) D_1 + (I_{n+1} - \tilde{I}_n) \otimes D_1 +$$

$$+ \bar{\Omega}_n^- \otimes \mathcal{M}(I_W - \mathcal{P}) + \Omega_n^+ \otimes \mathcal{Q}_n D_1 -$$

$$- C_n \otimes \mathcal{M} - (N-n) I_{n+1} \otimes \mathcal{A} - B_n \otimes \mathcal{B},\ n = \overline{1,\ N},$$

where

$$\mathcal{Q}_n = \operatorname{diag}\{q_v^{(n)},\ v = \overline{1,\ W}\},$$

$$\tilde{I}_n = \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 0 \end{pmatrix}_{(n+1)\times(n+1)}, \quad C_n = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & n \end{pmatrix}_{(n+1)\times(n+1)},$$

$$B_n = \begin{pmatrix} n & 0 & 0 & \ldots & 0 & 0 \\ 0 & n-1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 0 \end{pmatrix}_{(n+1)\times(n+1)}, \quad \Omega_n^+ = \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ 0 & 0 & 0 & \ldots & 0 & 0 \end{pmatrix}_{(n+1)\times(n+1)},$$

$$\bar{\Omega}_n^- = \begin{pmatrix} 0 & 0 & \ldots & 0 & 0 \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 2 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & n & 0 \end{pmatrix}_{(n+1)\times(n+1)}.$$

Here the notation $\operatorname{diag}\{\ldots\}$ *denotes a diagonal matrix with the diagonal elements listed in brackets.*

P r o o f. The block-tridiagonal structure of the infinitesimal generator Q arises from the fact that in an infinitesimal time interval, the number of available active servers n_t can change by at most one unit.

The off-diagonal blocks are derived as follows.

1. Server recovery ($n \to n+1$): with $(N-n)$ servers currently on vacation, each returning to the active state at the rate α_v, the transition rate is $(N-n)\alpha_v$. Since the number of busy servers m remains constant during this transition, the expansion matrix $\hat{I}_n$ is applied. Thus $Q_{n,\,n+1} = (N-n)\hat{I}_n \otimes \mathcal{A}$.

2. Server vacation ($n \to n-1$). This transition occurs in two cases:

• idle server vacation: an active but idle server moves to the vacation state with the intensity β_v. With $(n-m)$ idle servers, this is captured by the term $\Delta_n^- \otimes \mathcal{B}$;

• post-service vacation: a busy server completes the service (at the rate μ_v) and enters a vacation state with the probability p_v. This is captured by $\Omega_n^- \otimes \mathcal{MP}$.

The resulting block is $Q_{n,\,n-1} = \Delta_n^- \otimes \mathcal{B} + \Omega_n^- \otimes \mathcal{MP},\ n = \overline{1,\ N}$.

The diagonal entries of the block matrices $Q_{n,\,n}$ are negative, and their absolute values represent the total exit rates of the process ξ_t, $t \geq 0$, from the corresponding states. A transition out of the current state may occur due to the following events.

1. Change in the state of the underlying process ν_t: the exit rates are determined by the absolute values of the diagonal entries of the matrix $I_{n+1} \otimes D_0$ for $n = \overline{0,\ N}$. Note that the process ν_t may transit to the same state upon a customer arrival if the customer is either: rejected and leaves the system (is lost), or lost due to the lack of available active servers. In this case, the change does not occur. The intensities of these events are given by the diagonal entries of the matrices $\tilde{I}_n \otimes (I_W - \mathcal{Q}_n)D_1$ and $(I_{n+1} - \tilde{I}_n) \otimes D_1$, respectively.

2. Service completion: the corresponding transition rates are determined by the diagonal elements of the matrix $C_n \otimes \mathcal{M}$.

3. Server activation: one of the $(N-n)$ servers currently on vacation returns to the active state. The transition rates are given by the elements of the matrix $(N-n)I_{n+1} \otimes \mathcal{A}$.

4. Transition to the vacation state: one of n idle active servers moves to the vacation state. The intensities of these transitions are defined by the elements of the matrix $B_n \otimes \mathcal{B}$.

The non-negative off-diagonal entries of the diagonal blocks $Q_{n,\,n}$ represent the transition rates of the Markov chain ξ_t, $t \geq 0$, that do not result in a change in the number of active servers n. These transitions occur in the following cases.

1. Transitions of the underlying process ν_t:

(i) the process ν_t changes its state without generating a customer;

(ii) a customer arrival occurs, but the customer is lost due to rejection by the admission control mechanism or the absence of available active servers;

(iii) an arriving customer is accepted for service.

The intensities are determined by off-diagonal entries of the matrix $I_{n+1} \otimes D_0$ for case (i), the entries of the matrix $\tilde{I}_n \otimes (I_W - \mathcal{Q}_n)D_1 + (I_{n+1} - \tilde{I}_n) \otimes D_1$ for case (ii) and the entries of the matrix $\Omega_n^+ \otimes \mathcal{Q}_n D_1$ for case (iii).

2. Service completion: a server finishes the service but remains in the active state. The intensities are provided by the off-diagonal elements of the matrix $\bar{\Omega}_n^- \otimes \mathcal{M}(I_W - \mathcal{P})$.

Combining these parts, we obtain the complete expression for the diagonal block for $Q_{n,\,n}$. The proof is complete.

Once the block-tridiagonal structure of the infinitesimal generator Q is established (see theorem 1), we proceed to determine the stationary probability distribution of the process ξ_t, $t \geq 0$.

Let $\boldsymbol{\pi}(n,\ m)$ be the row vector of stationary probabilities:

$$\boldsymbol{\pi}(n,\ m) = \big(\pi(n,\ m,\ 1),\ \pi(n,\ m,\ 2),\ \ldots,\ \pi(n,\ m,\ W)\big),$$

where $0 \leq n \leq N$, $0 \leq m \leq n$. Each component $\pi(n,\ m,\ \nu)$, $1 \leq \nu \leq W$, represents the steady-state probability that there are n active servers, m busy servers, and the underlying process ν_t is in the state ν. Since the Markov chain ξ_t, $t \geq 0$, is irreducible, regular, and has a finite state space, the stationary distribution exists for any system parameters.

Subvectors $\boldsymbol{\pi}_n$ corresponded to the state space level, where n active servers are present are structured as follows:

$$\boldsymbol{\pi}_n = \big(\boldsymbol{\pi}(n,\ 0),\ \boldsymbol{\pi}(n,\ 1),\ \ldots,\ \boldsymbol{\pi}(n,\ n)\big),\ n = \overline{0,\ N}.$$

Let $\boldsymbol{\pi}$ be the stationary probability vector, partitioned according to the number of active servers n as follows:

$$\boldsymbol{\pi} = (\boldsymbol{\pi}_0,\ \boldsymbol{\pi}_1,\ \ldots,\ \boldsymbol{\pi}_N).$$

The vector $\boldsymbol{\pi}$ is the unique solution to the system of equilibrium equations

$$\boldsymbol{\pi} Q = \mathbf{0},\ \boldsymbol{\pi}\boldsymbol{e} = 1. \tag{2}$$

The block-tridiagonal structure of the generator Q facilitates the use of numerically stable methods, such as the stable algorithm from paper [16], to solve system (2).

Performance measures

Once the stationary probability vector $\boldsymbol{\pi}_n$, $n = \overline{0,\ N}$, is obtained, the following performance measures can be calculated:

- the average number of active servers (active drivers)

$$E_{\text{act}} = \sum_{n=1}^{N} n\boldsymbol{\pi}_n\boldsymbol{e};$$

- the average number of busy servers (occupied vehicles)

$$E_{\text{busy}} = \sum_{n=0}^{N}\sum_{m=1}^{n} m\boldsymbol{\pi}(n,\ m)\boldsymbol{e};$$

- the average intensity of the output flow of requests (average trip completion rate)

$$\lambda_{\text{out}} = \sum_{n=1}^{N}\sum_{m=1}^{n} m\boldsymbol{\pi}(n,\ m)\mathcal{M}\boldsymbol{e};$$

- the probability that an arriving customer is rejected due to all active servers being busy upon the arrival («no cars are available» probability)

$$P_{\text{arr}-\text{loss}} = \frac{1}{\lambda}\sum_{n=0}^{N}\boldsymbol{\pi}(n,\ n)D_1\boldsymbol{e};$$

- the probability that an arriving customer decides not to join the system despite the presence of idle active servers because the current surge price is too high (surge price rejection probability)

$$P_{\text{join}-\text{loss}} = \frac{1}{\lambda}\sum_{n=1}^{N}\sum_{m=0}^{n-1}\boldsymbol{\pi}(n,\ m)(I-\mathcal{Q}_n)D_1\boldsymbol{e};$$

- the total loss probability: the probability that an arriving customer is rejected due to either all active servers being busy or the dynamic pricing mechanism

$$P_{\text{loss}} = \frac{1}{\lambda}\left[\sum_{n=1}^{N}\left(\sum_{m=0}^{n-1}\boldsymbol{\pi}(n,\ m)(I-\mathcal{Q}_n)D_1 + \boldsymbol{\pi}(n,\ n)D_1\right)\boldsymbol{e} + \boldsymbol{\pi}(0,\ 0)D_1\boldsymbol{e}\right] = P_{\text{arr}-\text{loss}} + P_{\text{join}-\text{loss}}$$

or

$$P_{\text{loss}} = 1 - \frac{\lambda_{\text{out}}}{\lambda}.$$

The two distinct formulas obtained for the loss probability P_{loss} can be used to verify the correctness of the calculated stationary state probabilities;

- the average intensity of active servers going into a vacation state (average driver log-off intensity)

$$R_{\text{vac}} = \sum_{n=1}^{N}\sum_{m=0}^{n}\boldsymbol{\pi}(n,\ m)\left[(n-m)\mathcal{B} + m\mathcal{M}\mathcal{P}\right]\boldsymbol{e};$$

- the average activation intensity of servers from the vacation state (average driver log-on intensity)

$$R_{\text{act}} = \sum_{n=0}^{N-1}\sum_{m=0}^{n}(N-n)\boldsymbol{\pi}(n,\ m)\mathcal{A}\boldsymbol{e}.$$

Remark. Since the system operates in a steady state, the following relation holds:

$$R_{\text{vac}} = R_{\text{act}}.$$

Numerical examples

This section evaluates the performance of the proposed model within a real-world taxi aggregator scenario. We analyse how dynamic pricing strategies (surge pricing) and driver work-rest cycles impact the platform's service quality and overall profitability.

We consider a regional fleet of $N = 150$ vehicles. The incoming demand for rides is modelled as a MAP with the following fundamental matrices:

$$D_0 = \begin{pmatrix} -3.082\,03 & 0.030\,820\,3 & 0.015\,410\,1 \\ 0.154\,101 & -7.705\,07 & 0.154\,101 \\ 0.154\,101 & 0.077\,050\,7 & -15.410\,1 \end{pmatrix}, \; D_1 = \begin{pmatrix} 3.035\,8 & 0 & 0 \\ 0 & 7.396\,87 & 0 \\ 0 & 0 & 15.179 \end{pmatrix}.$$

The MAP has the fundamental arrival rate $\lambda = 5$, $c_{\text{cor}} = 0.207\,8$ and $c_{\text{var}} = 1.742\,1$ and represents three distinct city states.

1. State 1 (low demand): $\lambda_1 = 3.035\,8$ requests per minute (e. g., late night or early morning).
2. State 2 (moderate demand): $\lambda_2 = 7.396\,87$ requests per minute (e. g., standard business hours).
3. State 3 (peak demand): $\lambda_3 = 15.179$ requests per minute (e. g., rush hours or bad weather).

The average trip durations are 10 min $(\mu_1 = 0.1)$ in low traffic, 12.5 min $(\mu_2 = 0.08)$ in standard city traffic and 16.7 min $(\mu_3 = 0.06)$ during peak hours due to congestion.

The recovery intensity regulates the rate at which drivers rejoin the active car fleet from breaks, varying from 200 min $(\alpha_1 = 0.005)$ during low demand, 125 min $(\alpha_2 = 0.008)$ in moderate periods and 83.33 min $(\alpha_3 = 0.012)$ during peak hours to maintain system stability.

The idle vacation intensity represents the rate at which active but idle drivers (those waiting for an order) choose to log off, varying from 200 min $(\beta_1 = 0.005)$ during low demand, 333 min $(\beta_2 = 0.003)$ in moderate periods and to 1000 min $(\beta_3 = 0.001)$ during peak hours as they prefer to stay online when the likelihood of receiving a request is higher.

The post-service vacation probability defines the likelihood of a driver logging off immediately after a trip completion, which decreases from $p_1 = 0.07$ (7 %) during low demand, $p_2 = 0.05$ (5 %) in moderate periods to $p_3 = 0.02$ (2 %) in peak periods as higher earning potential and surge pricing incentivise drivers to stay online.

We assume that the system implements the dynamic pricing based on the load coefficient $R_v^{(n)} = \dfrac{\lambda_v}{n\mu_v}$, $n = \overline{0,\ N}$. The pricing coefficients $f_v^{(n)}$ are defined by formula

$$f_v^{(n)} = \begin{cases} f_1, \text{ if } R_v^{(n)} < 0.7, \\ f_2, \text{ if } 0.7 \le R_v^{(n)} < 1.5, \\ f_3, \text{ if } R_v^{(n)} \ge 1.5. \end{cases}$$

The corresponding acceptance probabilities $q_v^{(n)}$ are calculated as follows:

$$q_v^{(n)} = \begin{cases} 0.98, \text{ if } f_v^{(n)} = f_1, \\ \min\left\{0.98,\ \dfrac{0.5}{\max\left\{0.7 f_v^{(n)}, 1\right\}} + \dfrac{0.5}{\left(f_v^{(n)}\right)^2}\right\} \text{ otherwise.} \end{cases}$$

We set $f_1 = 1$ and vary the pricing coefficients f_2 and f_3 within the intervals $[1,\ 3]$ and $[f_2,\ 3]$, respectively, with a step of 0.1. These coefficients directly determine the request acceptance probability $q_v^{(n)}$, where higher prices lead to a lower probability of customer admission to the system. Then we investigate the influence of pricing coefficients f_2 and f_3 on the key system performance metrics.

Figure 2 illustrates the dependence of the average number of active servers, E_{act}, on the pricing coefficients f_2 and f_3. It can be observed that E_{act} depends only slightly on these parameters. In this specific case, E_{act} increases as f_2 and f_3 rise, reaching its maximum when the pricing coefficients are at their peak. However, this trend may change if different values for the intensities β_v and probabilities p_v, $v = 1,\ 2,\ 3$, are selected.

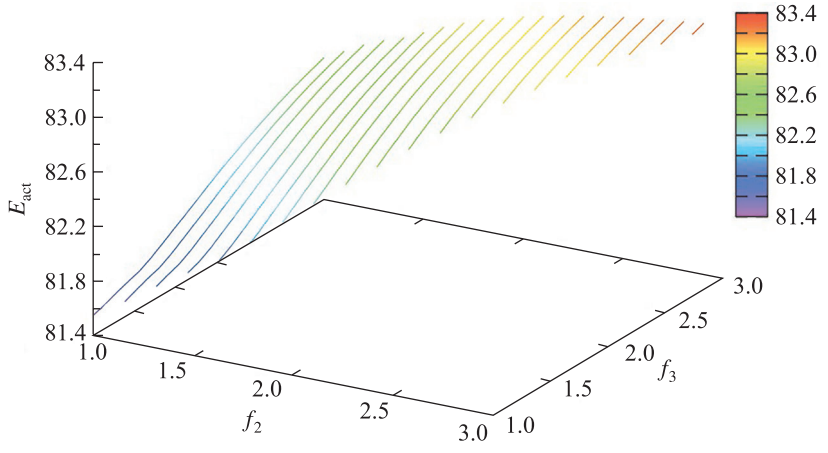


Fig. 2. The dependence of E_{act} on f_2 and f_3

Conversely, fig. 3 depicts the impact of the pricing coefficients on the average number of busy servers E_{busy}. The plot shows a clear downward trend, where E_{busy} decreases as f_2 and f_3 grow. This result confirms that the adaptive admission control mechanism effectively regulates the number of busy servers by discouraging customer arrivals through increased pricing during periods of high demand, thus preventing system congestion.

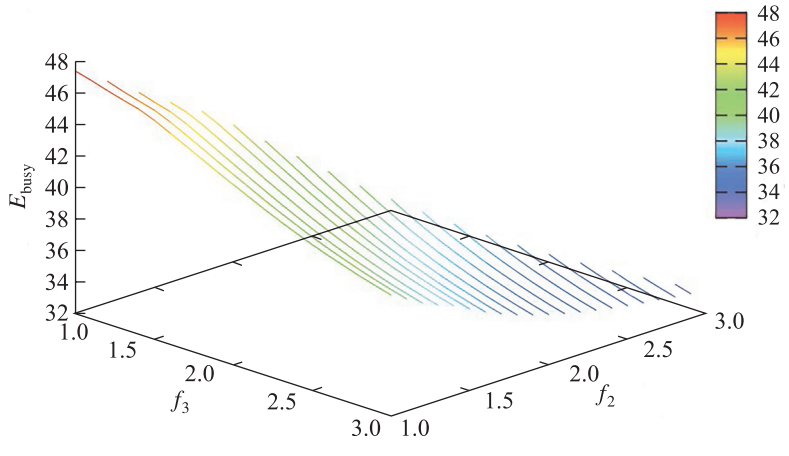


Fig. 3. The dependence of E_{busy} on f_2 and f_3

Figure 4 illustrates the dependence of the probability that an arriving customer is rejected due to all active servers being busy, denoted as $P_{arr-loss}$, on the pricing coefficients f_2 and f_3. The plot reveals a consistent decrease in $P_{arr-loss}$ as the values of f_2 and f_3 increase. This downward trend is explained by the fact that higher prices lead to a lower acceptance probability $q_v^{(n)}$, which reduces the overall number of requests admitted to the system. As a result, the active servers are less likely to be fully occupied, decreasing the probability that a new arrival finds no free server available.

Figure 5 shows the dependence of $P_{join-loss}$, the probability that a customer decides not to join the system due to the dynamic pricing mechanism despite the presence of idle servers, on the coefficients f_2 and f_3. This probability exhibits a significant upward trend as the pricing coefficients grow. Since the admission control mechanism intentionally reduces the acceptance probability $q_v^{(n)}$ as the price increases to manage congestion, the likelihood of a customer being rejected by the pricing rule rises accordingly. This behaviour confirms the effective operation of the adaptive admission control, which uses higher prices to discourage arrivals during high-load periods.

Figure 6 demonstrates the dependence of the average intensity of the output flow of requests, denoted as λ_{out}, on the pricing coefficients f_2 and f_3. The plot reveals that λ_{out} decreases as f_2 and f_3 rise. This is a direct consequence of the adaptive admission control mechanism: higher pricing reduces the acceptance probability $q_v^{(n)}$, which results in a smaller number of requests being admitted and subsequently processed by the system per unit of time.

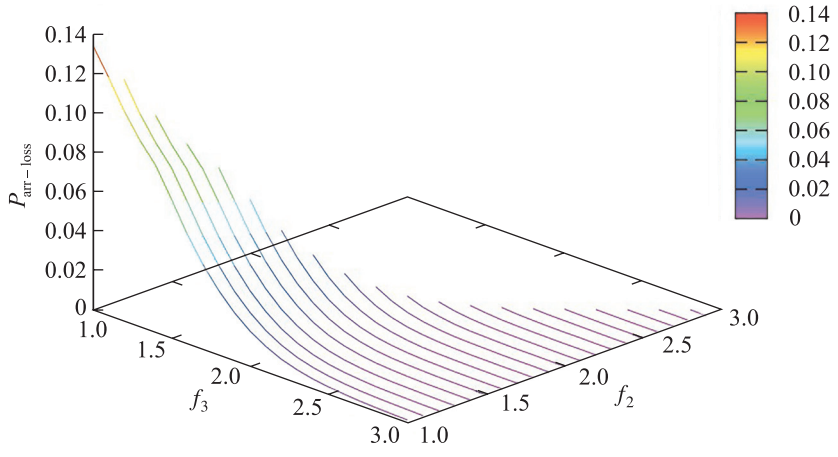


Fig. 4. The dependence of $P_{\text{arr}-\text{loss}}$ on f_2 and f_3

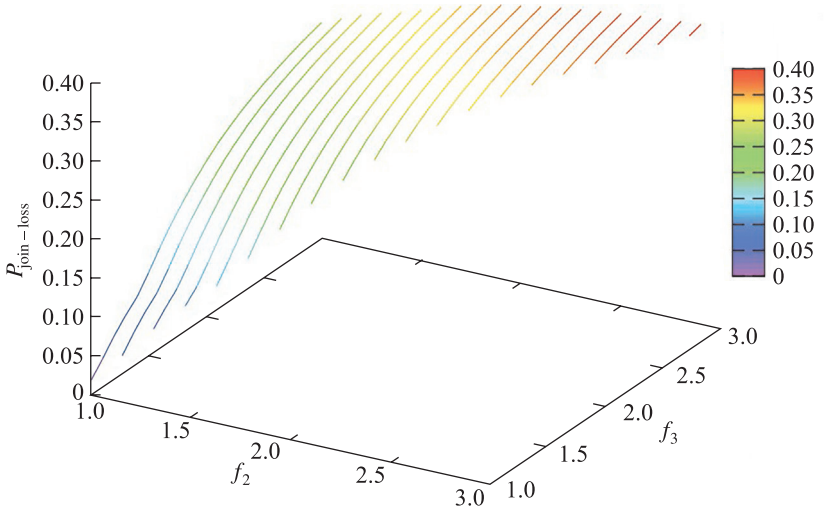


Fig. 5. The dependence of $P_{\text{join}-\text{loss}}$ on f_2 and f_3

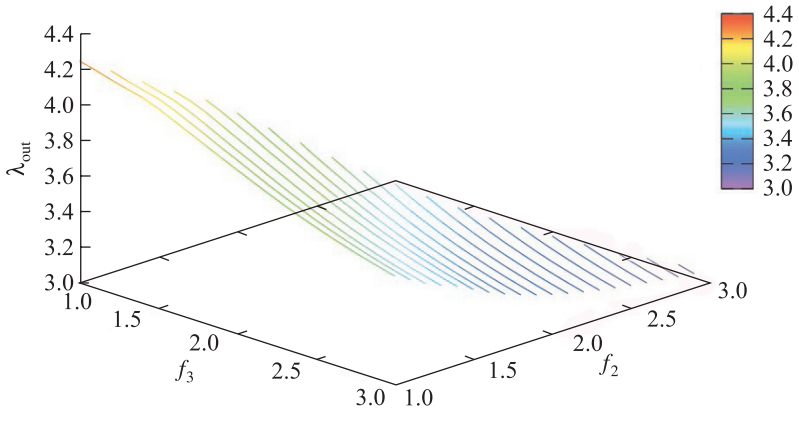


Fig. 6. The dependence of λ_{out} on f_2 and f_3

The total loss probability as a function of the pricing coefficients is depicted in fig. 7. It is observed that P_{loss} increases significantly as f_2 and f_3 grow. Although higher prices lead to a decrease in the probability of rejection due to all servers being busy ($P_{\text{arr}-\text{loss}}$), the sharp rise in rejections caused directly by the pricing mechanism ($P_{\text{join}-\text{loss}}$) dominates the overall behaviour. This leads to an increase in the total probability that an arriving customer is rejected, either due to the system congestion or the dynamic pricing strategy.

Figure 8 illustrates the average intensity of active servers transitioning into the vacation state R_{vac}, in relation to f_2 and f_3. This intensity also slightly depends on f_2 and f_3. In this case, the intensity decreases as the pricing coefficients increase. However, a different choice of intensities and probabilities may alter this trend.

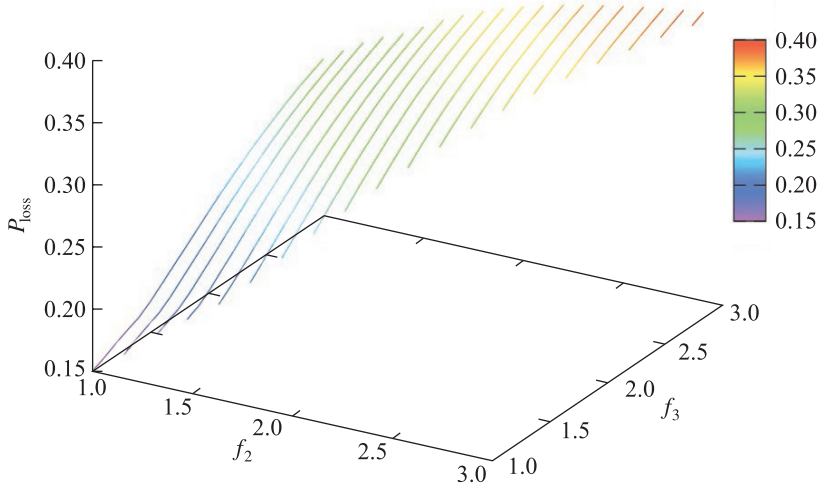


Fig. 7. The dependence of P_{loss} on f_2 and f_3

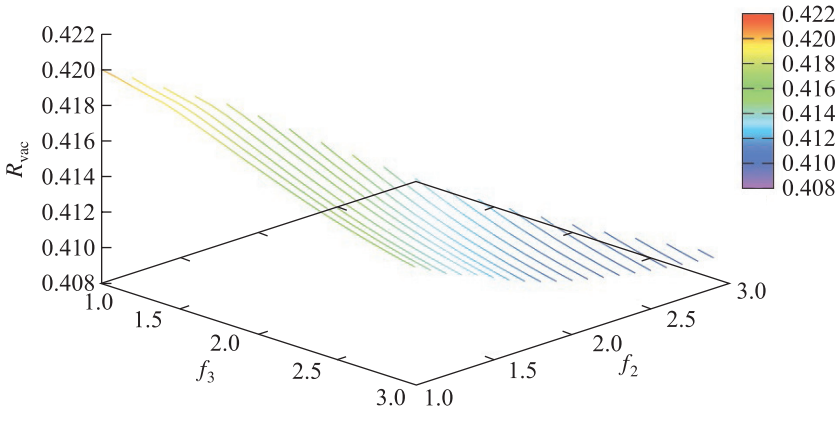


Fig. 8. The dependence of R_{vac} on f_2 and f_3

To evaluate the system's economic efficiency, we define the expected system profit per unit time as follows:

$$J = a\sum_{n=1}^{N}\sum_{m=0}^{n-1}\boldsymbol{\pi}(n,\, m)\mathcal{Q}_n D_1 \operatorname{diag}\left\{f_v^{(n)},\, n=\overline{1,\, W}\right\}\boldsymbol{e} - b_1\lambda P_{\text{arr}-\text{loss}} - b_2\lambda P_{\text{join}-\text{loss}},$$

where a is the revenue earned per successfully served customer excluding price coefficients; b_1 and b_2 are the penalty cost per rejected customer due to either all active servers being busy or due to the lack of available active servers.

The first term, $a\sum_{n=1}^{N}\sum_{m=0}^{n-1}\boldsymbol{\pi}(n,\, m)\mathcal{Q}_n D_1 \operatorname{diag}\left\{f_v^{(n)},\, n=\overline{1,\, W}\right\}\boldsymbol{e}$, represents the expected revenue earned by the system per unit time while the second term, $b_1\lambda P_{\text{arr}-\text{loss}} + b_2\lambda P_{\text{join}-\text{loss}}$, quantifies the economic impact of service denials. The optimisation problem is to find the optimal vector of controllable parameters $f_v^{(n)}$, $n=\overline{1,\, W}$, that maximises the objective function J.

Set $a=1$, $b_1=2$ and $b_2=0.5$.

Figure 9 illustrates the behaviour of the expected system profit J as a function of the pricing coefficients f_2 and f_3.

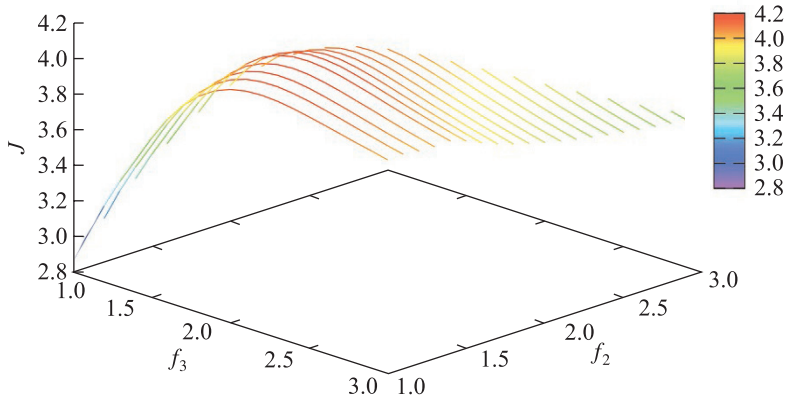


Fig. 9. The dependence of the cost function J on f_2 and f_3

The 3D surface plot in fig. 9 demonstrates a clear non-monotonic relationship, indicating the existence of an optimal pricing strategy. Initially, as f_2 and f_3 increase, the profit J rises due to the effective congestion control and balanced request acceptance. However, beyond the optimal point, further increases in pricing lead to a significant rise in the probability $P_{\text{join}-\text{loss}}$, which negatively impacts the total revenue. According to the numerical analysis, the maximum profit $J^* = 4.1816$ is achieved at the optimal values $f_2^* = 1.4$ and $f_3^* = 2.1$.

Conclusions

In this paper, we have developed a comprehensive mathematical model of a multi-server queuing system with a MAP, adaptive admission control and a complex server vacation policy. The qualitative analysis of the proposed model reveals that the synergy between dynamic pricing and server management is essential for maintaining a balance between quality of service and operational efficiency. The adaptive admission control mechanism, which regulates a request acceptance based on the current load, proves to be a highly effective tool for the congestion management. It ensures system stability by discouraging new arrivals during peak periods, thereby preventing the exhaustion of active resources. Simultaneously, the inclusion of both post-service and idle server vacation policies allows the system to significantly reduce the energy consumption and operational expenditures during periods of low traffic intensity.

Our investigation into the system's economic efficiency further suggests that the pricing coefficients must be carefully optimised to maximise total profit. While increased pricing reduces the probability of loss due to busy servers, it simultaneously increases the likelihood of requests being rejected by the admission control mechanism. The existence of an optimal pricing strategy indicates that resource providers in cloud data centres and next-generation communication networks can fine-tune these parameters to achieve the best economic outcome.

The findings of this paper can be extended to the case of a phase-type service time distribution (see [17]).

References

1. Low DW. Optimal dynamic pricing policies for an *M*/*M*/*s* queue. *Operations Research*. 1974;22(3):545–561. DOI: 10.1287/opre.22.3.545.

2. Gans N, Savin S. Pricing and capacity rationing for rentals with uncertain durations. *Management Science*. 2007;53(3):390–407. DOI: 10.1287/mnsc.1060.0651.

3. Son J-D. Optimal admission and pricing control problem with deterministic service times and sideline profit. *Queueing Systems*. 2008;60(1–2):71–85. DOI: 10.1007/s11134-008-9087-4.

4. Çil EB, Örmeci EL, Karaesmen F. Effects of system parameters on the optimal policy structure in a class of queueing control problems. *Queueing Systems*. 2009;61(4):273–304. DOI: 10.1007/s11134-009-9109-x.

5. Koole G. Structural results for the control of queueing systems using event-based dynamic programming. *Queueing Systems*. 1998;30(3–4):323–339. DOI: 10.1023/A:1019177307418.

6. Yoon S, Lewis ME. Optimal pricing and admission control in a queueing system with periodically varying parameters. *Queueing Systems*. 2004;47(3):177–199. DOI: 10.1023/B:QUES.0000035313.20223.3f.

7. Gayon J-P, Talay-Değirmenci I, Karaesmen F, Örmeci EL. Optimal pricing and production policies of a make-to-stock system with fluctuating demand. *Probability in the Engineering and Informational Sciences*. 2009;23(2):205–230. DOI: 10.1017/S026996480900014X.

8. Alwan AA, Ata B, Zhou Y. A queueing model of dynamic pricing and dispatch control for ride-hailing systems incorporating travel times. *Queueing Systems*. 2024;106(1–2):1–66. DOI: 10.1007/s11134-023-09901-y.

9. Castillo JC, Knoepfle D, Weyl EG. Matching and pricing in ride hailing: wild goose chases and how to solve them. *Management Science*. 2024;71(5):4377–4395. DOI: 10.1287/mnsc.2022.00096.

10. Chen Q, Lei Y, Jasin S. Real-time spatial-intertemporal pricing and relocation in a ride-hailing network: near-optimal policies and the value of dynamic pricing. *Operations Research*. 2024;72(5):2097–2118. DOI: 10.1287/opre.2022.2425.

11. Guo D, Fan Z-P, Liu Y. The strategic analysis of service mode selection for a ride-hailing platform. *International Transactions in Operational Research*. 2024;31(5):3135–3172. DOI: 10.1111/itor.13271.

12. Tripathy M, Bai J, Heese HS. Driver collusion in ride-hailing platforms. *Decision Sciences*. 2023;54(4):434–446. DOI: 10.1111/deci.12561.

13. Wang C, Wang J, Zhang Y, Malenje JO, Han Y. Optimizing taxi-pooling operations to enhance efficiency and revenue: a queuing model approach. *Mathematics*. 2024;12(20):3210. DOI: 10.3390/math12203210.

14. Dudin A, Dudin S, Dudina O. Analysis of a retrial queueing system suitable for modeling operation of ride-hailing platforms with the dynamic service pricing. *Axioms*. 2025;14(9):714. DOI: 10.3390/axioms14090714.

15. Dudin AN. Admission control in a parcel delivery system with account of the finite capacity of the warehouse at the pick-up point. *Journal of the Belarusian State University. Mathematics and Informatics*. 2025;1:68–78. EDN: ZIAXUB.

16. Dudin SA, Dudina OS. Call center operation model as a *MAP*/*PH*/*N*/*R* – *N* system with impatient customers. *Problems of Information Transmission*. 2011;47(4):364–377. DOI: 10.1134/S0032946011040053.

17. Dudina OS. Analytical modelling of systems with a ticket queue. *Journal of the Belarusian State University. Mathematics and Informatics*. 2024;2:40–53. Russian. EDN: NSXUQQ.

Received 01.04.2026 / revised 08.06.2026 / accepted 09.06.2026.