



DOI 10.28925/2663-4023.2026.35.1332

УДК 004.89:656.1:004.738.5

Андрух Анна Олександрівна

студентка магістратури

Державний торговельно-економічний університет, Київ

ORCID: 0009-0007-9041-1258

a.andrukh_fit_4m_25_m_z@knute.edu.ua

Тищенко Ігор Анатолійович

старший викладач

Державний торговельно-економічний університет, Київ

ORCID: 0000-0003-1813-0797

i.tyshchenko@knute.edu.ua

МОДЕЛЬ ТА АЛГОРИТМ АДАПТИВНОГО ІНТЕЛЕКТУАЛЬНОГО АГЕНТА ДЛЯ ОПТИМІЗАЦІЇ МІСЬКОЇ ДОСТАВКИ У ВЕБОРІЄНТОВАНОМУ ДИНАМІЧНОМУ СЕРЕДОВИЩІ

Анотація. У статті досліджено математичну модель та алгоритм адаптивного інтелектуального агента для автоматизованого вибору маршруту міської доставки в умовах змінної дорожньої ситуації. Міську логістичну систему подано як веборієнтоване середовище, у якому дані про замовлення, поточне завантаження доріг та інциденти надходять через REST API до серверного модуля агента, реалізація якого може виконуватися мовою Python. Задачу вибору маршруту формалізовано як нестационарний марковський процес прийняття рішень. Функція винагороди враховує тривалість поїздки, факт запізнення, дорожній інцидент і витрати на зміну маршруту. Для навчання застосовано Q-learning, доповнений віконним детектором погіршення середньої винагороди, тимчасовим підвищенням коефіцієнтів дослідження та навчання і частковим зменшенням накопичених Q-оцінок. Проведено імітаційне дослідження, яке охоплювало 300 циклів прийняття рішень, три послідовні фази зміни дорожньої ситуації та 200 незалежних прогонів. Порівняно фіксовану політику, стандартний Q-learning та запропонований адаптивний варіант. Адаптивний агент забезпечив 80,0 % своєчасних доставок, середній час маршруту 24,55 хв та інтегральний показник адаптивності 0,707. Для стандартного Q-learning ці значення становили відповідно 76,8 %, 25,22 хв і 0,670, а для фіксованої політики – 50,5 %, 29,41 хв і 0,426. Середній час стабілізації політики після зміни середовища скоротився з 36,9 до 25,6 циклу. Результати підтвердили, що виявлення зміни середовища та кероване часткове забування попереднього досвіду підвищують стійкість агента до зміни трафіку. Запропоновані модель і алгоритм можуть слугувати основою для подальшої реалізації вебсистеми підтримки прийняття рішень у міській логістиці.

Ключові слова: інтелектуальний агент; міська логістика; динамічна маршрутизація; веборієнтована система; Q-learning; марковський процес прийняття рішень; імітаційне моделювання.

ВСТУП

Постановка проблеми. Міська доставка відбувається в умовах постійної зміни інтенсивності руху, тривалості проїзду окремими ділянками, доступності доріг і кількості активних замовлень. Статичний маршрут, побудований до виїзду автомобіля, може втратити ефективність уже під час виконання доставки. Огляд стохастичних динамічних задач маршрутизації засвідчує, що для міської логістики характерні невизначені запити клієнтів, змінний час руху, обмеження часових вікон і необхідність приймати рішення в режимі реального часу [1]. Тому результатом оптимізації має бути не один незмінний маршрут, а політика вибору дій, здатна оновлюватися після зміни середовища.

Вебтехнології створюють інфраструктурну основу для такого оновлення. Замовлення можуть надходити через вебінтерфейс або мобільний застосунок, інформація про трафік – через зовнішній API, а рішення агента – передаватися до диспетчерської панелі у форматі JSON. У самоадаптивних системах подібний цикл передбачає моніторинг параметрів, аналіз відхилень, вибір нової конфігурації та її виконання [2], [3]. Однак використання постійних параметрів навчання може призвести до повільної



реакції агента після різкої зміни дорожніх умов. Отже, наукове завдання полягає в поєднанні моделі маршрутизації, алгоритму навчання з підкріпленням і механізму виявлення зміни середовища.

Аналіз останніх досліджень і публікацій. Машинне навчання у самоадаптивних системах застосовується для оновлення правил керування, прогнозування станів та автоматичного вибору конфігурацій [4]. Водночас невизначеність стосується не лише вхідних параметрів, а й переходів між станами та наслідків прийнятих дій [5]. Для динамічних задач маршрутизації навчання з підкріпленням є доцільним, оскільки оптимізація має враховувати майбутні стохастичні події, а її результатом виступає політика послідовних рішень [6].

У сучасних дослідженнях динамічна маршрутизація формалізується як MDP. В. Пан і С. Лю запропонували модель глибокого навчання з підкріпленням для невизначеної задачі маршрутизації [7]. Ч. Чжоу та співавтори розглянули роботу кур'єра за стохастичного надходження нових запитів [8]. А. Коноваленко і Л. Хваттум показали, що якість рішення істотно залежить від складу простору станів, зокрема від урахування часових обмежень, завантаження та характеристик замовлень [9]. Для статті обрано табличний Q-learning, оскільки він забезпечує прозорий математичний опис і придатний для компактного імітаційного експерименту [10].

Окреме значення має виявлення концептуального дрейфу, тобто зміни статистичних властивостей даних і залежностей, на основі яких агент сформував політику. Методи виявлення, локалізації та пояснення дрейфу систематизовано в працях Ф. Гіндера, В. Ваке та Б. Гаммера [11], [12]. Онлайн Q-learning дозволяє перебудовувати рішення під час функціонування системи [13], але потребує керування швидкістю навчання та дослідження. У вебзастосунках навчання з підкріпленням уже використовується для адаптивного балансування навантаження [14], що підтверджує можливість інтеграції подібного модуля до веборієнтованої логістичної системи.

Мета статті. Метою роботи є дослідження математичної моделі та алгоритму адаптивного інтелектуального агента для вибору маршруту міської доставки у веборієнтованому динамічному середовищі, а також оцінювання його ефективності за результатами імітаційного експерименту. Для досягнення мети формалізовано стан логістичного середовища, функцію винагороди та правило оновлення Q-оцінок; визначено механізм виявлення зміни; змодельовано три фази дорожньої ситуації та проведено порівняння з фіксованою політикою і стандартним Q-learning.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

Веборієнтована архітектура системи включає клієнтський інтерфейс диспетчера, серверний REST API, базу даних замовлень, модуль отримання дорожніх показників і модуль інтелектуального агента. Вебклієнт передає координати пунктів доставки та часові обмеження. Сервер нормує отримані значення й формує стан середовища. Агент обирає один із доступних маршрутів, а результат повертається до вебпанелі. Для майбутньої програмної реалізації можуть використовуватися Python, FastAPI або Flask, бібліотеки NumPy і pandas, а для відображення маршруту – JavaScript та картографічний вебсервіс. У цій статті програмний застосунок не створювався: перевірка алгоритму виконана засобами імітаційного моделювання.

Динамічне логістичне середовище представлено як нестационарний марковський процес прийняття рішень:

$$M_t = \langle S, A, P_t, R_t, \gamma \rangle, \quad (1)$$

де S – множина станів; A – множина допустимих маршрутних дій; P_t – залежна від часу функція переходів; R_t – функція винагороди; $\gamma \in [0; 1]$ – коефіцієнт дисконтування. Залежність P_t і R_t від часу відображає зміну дорожнього трафіку, виникнення інцидентів і перерозподіл переваг між маршрутами.

У загальному випадку стан агента задається вектором:

$$s_t = (v_t, O_t, z_t, b_t), \quad (2)$$

де v_t – поточне положення транспортного засобу; O_t – множина активних замовлень; z_t – дискретизований рівень дорожнього навантаження; b_t – залишок допустимого часу. У моделі статті розглянуто одну партію доставки та три альтернативні коридори A, B і C, тому простір станів скорочено до рівня трафіку $z_t \in \{\text{низький, середній, високий}\}$, а множина дій визначена як $A = \{aA, aB, aC\}$.



Винагорода після виконання маршрутної дії визначається за формулою:

$$r_t = 1 - \frac{T_t}{35} - 0,65 \cdot I(T_t > 30) - 0,15 \cdot I_{inc} - 0,03 \cdot c_t, \quad (3)$$

де T_t – тривалість маршруту, хв; $I(T_t > 30)$ – індикатор запізнення; I_{inc} – індикатор дорожнього інциденту; c_t – ознака перемикавання маршруту. Така функція стимулює агента обирати короткі та надійні маршрути й одночасно обмежує надмірне перемикавання.

Оновлення Q-оцінки виконується стандартним правилом:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t [r_t + \gamma \max_a Q_t(s_{t+1}, a) - Q_t(s_t, a_t)]. \quad (4)$$

Вибір дії здійснюється за ϵ -жадібною стратегією. У стандартному варіанті ϵ поступово зменшується протягом усього експерименту, тому після зміни середовища агент рідко перевіряє альтернативні маршрути. В адаптивному варіанті аналізуються два послідовні вікна по $h = 12$ винагород. Показник зміни обчислюється так:

$$D_t = \bar{r}_{t-2h+1:t-h} - \bar{r}_{t-h+1:t}. \quad (5)$$

Якщо $D_t > \delta$, де $\delta = 0,10$, фіксується погіршення середовища. Після цього ϵ тимчасово збільшується до 0,85, коефіцієнт навчання α – до 0,45, а накопичені Q-оцінки множаться на 0,45. Часткове забування не руйнує весь попередній досвід, але зменшує вплив політики, сформованої за застарілих умов.

Алгоритм роботи агента передбачає отримання вебданих про замовлення і трафік, формування дискретного стану, ϵ -жадібний вибір маршруту, обчислення винагороди та оновлення Q-оцінки. Після кожного циклу порівнюються два вікна винагород; у разі погіршення активується прискорена адаптація, а оновлений маршрут передається до вебпанелі.

МЕТОДИКА ДОСЛІДЖЕННЯ

Експеримент охоплював 300 циклів і три фази по 100 рішень. Рівень трафіку мав базовий час маршруту на 0,90; 1,00 або 1,15, після чого додавалося нормально розподілене відхилення ($\sigma = 1,7$ хв). Інцидент збільшував час на 7–15 хв. Доставка вважалася своєчасною за $T \leq 30$ хв. Виконано 200 незалежних прогонів. Параметри трьох фаз імітаційного експерименту наведено в табл. 1.

Таблиця 1

Параметри фаз імітаційного експерименту

Фаза	Переважний маршрут	Час А/В/С, хв; інциденти, %	Характер зміни
1	А	18 / 23 / 27; 4 / 6 / 8	Стабільна перевага А
2	В	36 / 20 / 25; 20 / 5 / 8	Перевантаження А
3	С	29 / 34 / 21; 10 / 18 / 5	Інциденти на В

Порівнювалися три підходи: фіксована політика, яка в усіх фазах використовувала маршрут А; стандартний Q-learning із безперервним зменшенням ϵ від 0,75 до 0,025 та сталим $\alpha = 0,08$; адаптивний Q-learning із детектором зміни та частковим забуванням. В обох навчальних алгоритмах $\gamma = 0,90$. Час адаптації визначався як кількість циклів після зміни фази до першого вікна з десяти рішень, у якому не менше семи разів обирався новий найкращий маршрут. Узагальнені показники порівняння алгоритмів наведено в табл. 2.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Таблиця 2

Узагальнені результати імітаційного дослідження

Алгоритм	Своєчасні доставки, %	Середній час, хв	Час адаптації, цикли	ІА
Фіксована політика	50,5	29,41	–	0,426
Стандартний Q-learning	76,8	25,22	36,9	0,670
Адаптивний Q-learning	80,0	24,55	25,6	0,707



Інтегральний показник адаптивності обчислено як зважену згортку частки своєчасних доставок SR , нормованого часу T , частоти інцидентів p_{inc} та часу адаптації T_{ad} :

$$I_A = 0,45SR + 0,25(1 - \frac{T}{40}) + 0,15(1 - p_{inc}) + 0,15(1 - \frac{T_{ad}}{100}). \quad (6)$$

Для фіксованої політики останній компонент прийнято рівним нулю, оскільки зміна маршруту не передбачалася. Результати показують, що стандартний Q-learning підвищив частку своєчасних доставок на 26,3 відсоткового пункту порівняно з фіксованою політикою. Адаптивний алгоритм забезпечив додаткове зростання на 3,3 відсоткового пункту, скоротив середній час доставки на 0,67 хв, підвищив середню винагороду з 0,116 до 0,157 і зменшив частоту інцидентів з 8,11 до 7,52 %. Найбільша різниця зафіксована у третій фазі: частка своєчасних доставок становила 71,3 % для адаптивного агента проти 61,6 % для стандартного Q-learning. Середній час стабілізації після зміни умов зменшився на 30,6 %. Частку своєчасних доставок у трьох фазах імітаційного експерименту подано на рис. 1.

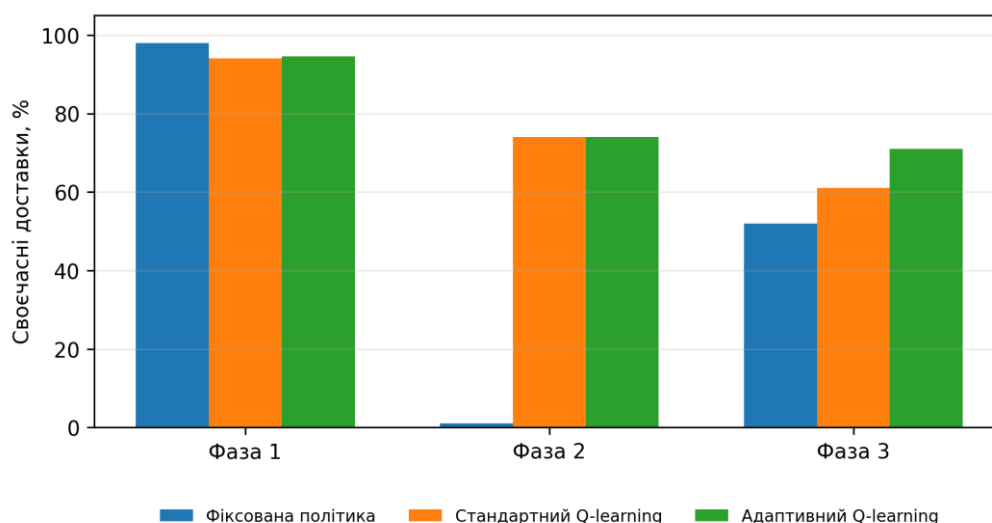


Рис. 1. Частка своєчасних доставок у трьох фазах імітаційного експерименту

У першій фазі всі підходи показали високий результат, оскільки фіксована політика відповідала найкращому маршруту А. Після перевантаження цього маршруту її ефективність різко знизилася. Обидва навчальні алгоритми переорієнтувалися на маршрут В, однак адаптивний механізм не дав значної переваги у другій фазі, оскільки зміна була виявлена після накопичення двох вікон спостережень. У третій фазі кероване забування Q-оцінок прискорило перехід до маршруту С, що пояснює найбільшу різницю між алгоритмами. Отже, перевага адаптивного підходу проявляється не у стабільному середовищі, а після повторних змін, коли ранні оцінки починають суперечити актуальним умовам.

Імітаційний характер дослідження означає, що наведені результати характеризують запропоновану модель за заданих розподілів, а не роботу реального транспортного підприємства. Водночас структура експерименту дозволяє відтворити розрахунок у Python і надалі замінити синтетичні параметри фактичними даними вебсервісів трафіку, історією замовлень і GPS-треками. Це створює основу для подальшої перевірки MDP-моделі, Q-learning, механізму адаптації та програмної архітектури системи на реальних даних.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Досліджено модель адаптивного інтелектуального агента для автоматизованого вибору маршруту міської доставки. Логістичне середовище формалізовано як нестационарний MDP, а вебтехнології розглянуто як засіб отримання даних і передавання рішення між модулем агента та диспетчерським інтерфейсом. Алгоритм поєднує Q-learning, віконне виявлення погіршення винагороди, тимчасове збільшення інтенсивності дослідження та часткове забування накопичених оцінок. За результатами 200 прогонів і 300 циклів адаптивний агент забезпечив 80,0 % своєчасних доставок та інтегральний показник 0,707. Порівняно зі стандартним Q-learning час адаптації скоротився з 36,9 до 25,6 циклу, а у третій фазі частка своєчасних доставок зросла на 9,7 відсоткового пункту. Наукова новизна полягає у включенні простого детектора зміни до прозорої Q-learning-моделі вибору логістичного маршруту. Практичне



значення результатів полягає у можливості використання моделі як основи серверного Python-модуля вебсистеми підтримки рішень. Подальша робота має передбачати перехід від трьох кандидатних маршрутів до повного дорожнього графа, урахування кількох транспортних засобів і перевірку на реальних даних трафіку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mardešić, N., Erdelić, T., Carić, T., & Đurasević, M. (2024). Review of stochastic dynamic vehicle routing in the evolving urban logistics environment. *Mathematics*, 12(1), Article 28. <https://doi.org/10.3390/math12010028>
2. Aguayo, O., Sepúlveda, S., & Mazo, R. (2024). Variability management in self-adaptive systems through deep learning: A dynamic software product line approach. *Electronics*, 13(5), Article 905. <https://doi.org/10.3390/electronics13050905>
3. Alfonso, I., Garcés, K., Castro, H., & Cabot, J. (2021). Self-adaptive architectures in IoT systems: A systematic literature review. *Journal of Internet Services and Applications*, 12, Article 14. <https://doi.org/10.1186/s13174-021-00145-8>
4. Gheibi, O., Weyns, D., & Quin, F. (2021). Applying machine learning in self-adaptive systems: A systematic literature review. *ACM Transactions on Autonomous and Adaptive Systems*, 15(3), Article 9, 1–37. <https://doi.org/10.1145/3469440>
5. Mahdavi-Hezavehi, S. M., Weyns, D., Avgeriou, P., Calinescu, R., Mirandola, R., & Perez-Palacin, D. (2021). Uncertainty in self-adaptive systems: A research community perspective. *ACM Transactions on Autonomous and Adaptive Systems*, 15(4), Article 10, 1–36. <https://doi.org/10.1145/3487921>
6. Hildebrandt, F. D., Thomas, B. W., & Ulmer, M. W. (2023). Opportunities for reinforcement learning in stochastic dynamic vehicle routing. *Computers & Operations Research*, 150, Article 106071. <https://doi.org/10.1016/j.cor.2022.106071>
7. Pan, W., & Liu, S. Q. (2023). Deep reinforcement learning for the dynamic and uncertain vehicle routing problem. *Applied Intelligence*, 53, 405–422. <https://doi.org/10.1007/s10489-022-03456-w>
8. Zhou, C., Ma, J., Douge, L., Chew, E. P., & Lee, L. H. (2023). Reinforcement learning-based approach for dynamic vehicle routing problem with stochastic demand. *Computers & Industrial Engineering*, 182, Article 109443. <https://doi.org/10.1016/j.cie.2023.109443>
9. Konovalenko, A., & Hvattum, L. M. (2024). Optimizing a dynamic vehicle routing problem with deep reinforcement learning: Analyzing state-space components. *Logistics*, 8(4), Article 96. <https://doi.org/10.3390/logistics8040096>
10. Shakya, A. K., Pillai, G., & Chakrabarty, S. (2023). Reinforcement learning algorithms: A brief survey. *Expert Systems with Applications*, 231, Article 120495. <https://doi.org/10.1016/j.eswa.2023.120495>
11. Hinder, F., Vaquet, V., & Hammer, B. (2024a). One or two things we know about concept drift—A survey on monitoring in evolving environments. Part A: Detecting concept drift. *Frontiers in Artificial Intelligence*, 7, Article 1330257. <https://doi.org/10.3389/frai.2024.1330257>
12. Hinder, F., Vaquet, V., & Hammer, B. (2024b). One or two things we know about concept drift—A survey on monitoring in evolving environments. Part B: Locating and explaining concept drift. *Frontiers in Artificial Intelligence*, 7, Article 1330258. <https://doi.org/10.3389/frai.2024.1330258>
13. Metzger, A., Quinton, C., Mann, Z. Á., Baresi, L., & Pohl, K. (2024). Realizing self-adaptive systems via online reinforcement learning and feature-model-guided exploration. *Computing*, 106(4), 1251–1272. <https://doi.org/10.1007/s00607-022-01052-x>
14. Al-Shaikh, R. Z., Al-Nayar, M. M. J., & Hasan, A. M. (2025). Reinforcement learning algorithms for adaptive load-balancing for web applications. *South African Computer Journal*, 37(2), 121–145. <https://doi.org/10.18489/sacj.v37i2.20753>

**Anna Andrukh**

Master's Student

State University of Trade and Economics, Kyiv

ORCID: 0009-0007-9041-1258

*a.andrukh_fit_4m_25_m_z@knute.edu.ua***Ihor Tyshchenko**

Senior Lecturer

State University of Trade and Economics, Kyiv

ORCID: 0000-0003-1813-0797

*i.tyshchenko@knute.edu.ua***MODEL AND ALGORITHM OF AN ADAPTIVE INTELLIGENT AGENT FOR URBAN DELIVERY OPTIMIZATION IN A WEB-ORIENTED DYNAMIC ENVIRONMENT**

Abstract. The article investigates a mathematical model and an adaptive intelligent-agent algorithm for automated route selection in urban delivery under changing traffic conditions. The urban logistics system is represented as a web-oriented environment in which order data, current road congestion and incident information are transmitted through REST APIs to a server-side agent module that can be implemented in Python. The route-selection problem is formalized as a non-stationary Markov decision process. The reward function accounts for travel duration, late delivery, road incidents and route-switching costs. Q-learning is enhanced by a window-based detector of declining mean reward, temporary increases in exploration and learning rates, and partial discounting of accumulated Q-values. A simulation study covered 300 decision cycles, three consecutive phases of changing road conditions and 200 independent runs. A fixed policy, standard Q-learning and the proposed adaptive variant were compared. The adaptive agent achieved an on-time delivery rate of 80.0%, an average travel time of 24.55 minutes and an integral adaptability index of 0.707. The corresponding values were 76.8%, 25.22 minutes and 0.670 for standard Q-learning, and 50.5%, 29.41 minutes and 0.426 for the fixed policy. The mean policy stabilization time after an environmental change decreased from 36.9 to 25.6 cycles. The results show that change detection and controlled partial forgetting improve the agent's robustness to traffic drift. The proposed model and algorithm can serve as a basis for a web-based decision-support system for urban logistics. The comparison further demonstrates that the proposed mechanism is most effective after repeated environmental changes, when previously learned action values no longer correspond to current traffic conditions.

Keywords: intelligent agent; urban logistics; dynamic routing; web-oriented system; Q-learning; Markov decision process; simulation modelling.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Mardešić, N., Erdelić, T., Carić, T., & Đurasević, M. (2024). Review of stochastic dynamic vehicle routing in the evolving urban logistics environment. *Mathematics*, 12(1), Article 28. <https://doi.org/10.3390/math12010028>
2. Aguayo, O., Sepúlveda, S., & Mazo, R. (2024). Variability management in self-adaptive systems through deep learning: A dynamic software product line approach. *Electronics*, 13(5), Article 905. <https://doi.org/10.3390/electronics13050905>
3. Alfonso, I., Garcés, K., Castro, H., & Cabot, J. (2021). Self-adaptive architectures in IoT systems: A systematic literature review. *Journal of Internet Services and Applications*, 12, Article 14. <https://doi.org/10.1186/s13174-021-00145-8>
4. Gheibi, O., Weyns, D., & Quin, F. (2021). Applying machine learning in self-adaptive systems: A systematic literature review. *ACM Transactions on Autonomous and Adaptive Systems*, 15(3), Article 9, 1–37. <https://doi.org/10.1145/3469440>
5. Mahdavi-Hezavehi, S. M., Weyns, D., Avgeriou, P., Calinescu, R., Mirandola, R., & Perez-Palacin, D. (2021). Uncertainty in self-adaptive systems: A research community perspective. *ACM Transactions on Autonomous and Adaptive Systems*, 15(4), Article 10, 1–36. <https://doi.org/10.1145/3487921>



6. Hildebrandt, F. D., Thomas, B. W., & Ulmer, M. W. (2023). Opportunities for reinforcement learning in stochastic dynamic vehicle routing. *Computers & Operations Research*, 150, Article 106071. <https://doi.org/10.1016/j.cor.2022.106071>
7. Pan, W., & Liu, S. Q. (2023). Deep reinforcement learning for the dynamic and uncertain vehicle routing problem. *Applied Intelligence*, 53, 405–422. <https://doi.org/10.1007/s10489-022-03456-w>
8. Zhou, C., Ma, J., Douge, L., Chew, E. P., & Lee, L. H. (2023). Reinforcement learning-based approach for dynamic vehicle routing problem with stochastic demand. *Computers & Industrial Engineering*, 182, Article 109443. <https://doi.org/10.1016/j.cie.2023.109443>
9. Konovalenko, A., & Hvattum, L. M. (2024). Optimizing a dynamic vehicle routing problem with deep reinforcement learning: Analyzing state-space components. *Logistics*, 8(4), Article 96. <https://doi.org/10.3390/logistics8040096>
10. Shakya, A. K., Pillai, G., & Chakrabarty, S. (2023). Reinforcement learning algorithms: A brief survey. *Expert Systems with Applications*, 231, Article 120495. <https://doi.org/10.1016/j.eswa.2023.120495>
11. Hinder, F., Vaquet, V., & Hammer, B. (2024a). One or two things we know about concept drift—A survey on monitoring in evolving environments. Part A: Detecting concept drift. *Frontiers in Artificial Intelligence*, 7, Article 1330257. <https://doi.org/10.3389/frai.2024.1330257>
12. Hinder, F., Vaquet, V., & Hammer, B. (2024b). One or two things we know about concept drift—A survey on monitoring in evolving environments. Part B: Locating and explaining concept drift. *Frontiers in Artificial Intelligence*, 7, Article 1330258. <https://doi.org/10.3389/frai.2024.1330258>
13. Metzger, A., Quinton, C., Mann, Z. Á., Baresi, L., & Pohl, K. (2024). Realizing self-adaptive systems via online reinforcement learning and feature-model-guided exploration. *Computing*, 106(4), 1251–1272. <https://doi.org/10.1007/s00607-022-01052-x>
14. Al-Shaikh, R. Z., Al-Nayar, M. M. J., & Hasan, A. M. (2025). Reinforcement learning algorithms for adaptive load-balancing for web applications. *South African Computer Journal*, 37(2), 121–145. <https://doi.org/10.18489/sacj.v37i2.20753>

Отримано редакцією журналу / Received: 14.06.26

Прорецензовано / Revised: 22.06.26

Схвалено до друку / Accepted: 26.10.26



This work is licensed under Creative Commons Attribution-noncommercial-sharealike 4.0 International License.