

С. Г. Ковальов, канд. пед. наук

Центральноукраїнський національний технічний університет, м. Кропивницький, Україна

e-mail: kovalyovserggr@ukr.net

Оптимізація часу виробництва за допомогою методу навчання з підкріпленням як частинний випадок підвищення ефективності автоматизованих виробничих ліній

У статті розглянуто застосування методу навчання з підкріпленням для оптимізації часу роботи автоматизованої виробничої лінії. Лінія моделюється у вигляді графу, де вершини представляють обладнання, здатне виконувати одну або кілька виробничих операцій. Такий підхід дозволяє не лише будувати точну модель середовища функціонування агента, але й реалізовувати комп'ютерну симуляцію, що слугує основою для аналізу ефективності методів навчання з підкріпленням. Завдяки симуляціям оцінюється можливість інтеграції цих методів у реальні виробничі лінії з метою підвищення їхньої ефективності, оптимізації використання ресурсів та забезпечення надійності функціонування.

виробнича лінія, оптимізація ефективності використання, штучний інтелект, час виробництва, моделювання лінії як системи, графи станів

Постановка проблеми. Підвищення ефективності використання сучасного виробництва в цілому та його виробничих ліній зокрема, передбачає оптимізацію таких складових ефективності як: час виробництва одиниці продукції, затрати ресурсів на виробництво, обсяг виробництва (кількість одиниць продукції, які виготовляються за одиницю часу), коефіцієнт використання обладнання (частка часу коли обладнання використовується на виробництво), надійність виробництва. Надійність має власні складові зокрема: довговічність (тривалість експлуатації елемента без поломок), середній час між поломками (середній час між випадковими поломками), час відновлення після поломки, збережувальність (легкість та швидкість обслуговування обладнання).

Кожна конкретна задача по оптимізації ефективності та надійності для конкретної виробничої лінії має на меті оптимізацію відповідних складових, що свідчить про необхідність формування єдиного підходу для оптимізації кожного з цих компонентів. Зважаючи на це можна розробити універсальні методи, які забезпечать досягнення високої ефективності у різних виробничих середовищах, незалежно від специфіки їх функціонування.

Задачі по визначенню універсальних методів для оптимізації часу виробництва як складової ефективності виробничих ліній є складними і потребують ретельного моделювання виробничих ліній. Застосування методів штучного інтелекту (ШІ) до зазначених моделей є особливо актуальним, оскільки вони представляють собою потужні інструменти, здатні ефективно вирішувати задачі такого характеру. Методи ШІ мають ознаки універсальності та адаптивності, що дозволяє створювати загальні підходи для оптимізації різних виробничих ліній, враховуючи їхні унікальні характеристики та потреби. Це робить штучний інтелект особливо цінним у контексті досягнення високої ефективності виробництва.

Аналіз останніх досліджень і публікацій. Дослідження у сфері оптимізації виробничих процесів за допомогою методів навчання з підкріпленням стали актуальними завдяки зростаючій складності автоматизованих виробничих ліній. Останні роботи у цьому напрямку зосереджуються на моделюванні виробничих систем

як графів, де вершини відповідають вузлам обладнання, а ребра визначають маршрути переходу між операціями. Використання агентів на основі глибокого навчання, зокрема Deep Q-Network (DQN), демонструє значний потенціал для оптимізації маршрутів, мінімізації простоїв та зменшення часу виробництва. Подібні дослідження вже показали свою ефективність у галузях, де необхідна інтеграція складного обладнання з непередбачуваними умовами роботи [1,2,5].

Актуальність таких досліджень визначається сучасними викликами у сфері промислової автоматизації. Зі збільшенням вимог до ефективності та надійності виробництва, традиційні підходи оптимізації часто не відповідають потребам динамічних виробничих середовищ. Методи навчання з підкріпленням дозволяють створювати адаптивні системи, які можуть самостійно вдосконалювати свої стратегії у процесі роботи. Це відкриває нові перспективи для автоматизації, де виробничі лінії не лише виконують завдання, але й адаптуються до змінних умов і складних сценаріїв [3,6,7].

Крім того, дослідження в цій галузі сприяють розвитку індустрії 4.0, яка орієнтована на впровадження інтелектуальних технологій у виробництво. Використання навчання з підкріпленням у поєднанні зі штучним інтелектом та машинним навчанням дозволяє інтегрувати інформаційні та фізичні системи, створюючи «розумні» виробничі процеси. Такі дослідження не лише забезпечують науковий прогрес, але й мають значний вплив на економічний розвиток, сприяючи оптимальному використанню ресурсів, зменшенню витрат і підвищенню продуктивності [8,9].

Постановка завдання. Метою даної роботи є розробка ефективних підходів до оптимізації часу виробництва як складової ефективності виробничих ліній за допомогою методів навчання з підкріпленням.

Для реалізації мети розв'язувалися наступні завдання:

1. Дослідження підходу оптимізації виробничих процесів для зменшення часу виконання операцій та мінімізації простоїв.
2. Моделювання впровадження стратегій для підвищення надійності та довговічності виробничих вузлів.
3. Розвиток нових методологій та технологій для управління та оптимізації виробничих ліній на основі сучасних досягнень в галузі ШІ.

Викладення основного матеріалу. Використання методу навчання з підкріпленням (НП) є особливо перспективним підходом для реалізації завдань по оптимізації ефективності використання виробничих ліній. Метод НП може бути реалізований у вигляді комп'ютерної програми, що інтегрована у виробничу лінію.

Метод НП передбачає функціонування програми-агента (далі агента) в межах певного середовища. Взаємодія агента з середовищем відбувається циклічно. Схематично один життєвий цикл взаємодії агента з середовищем показано на рис. 1.

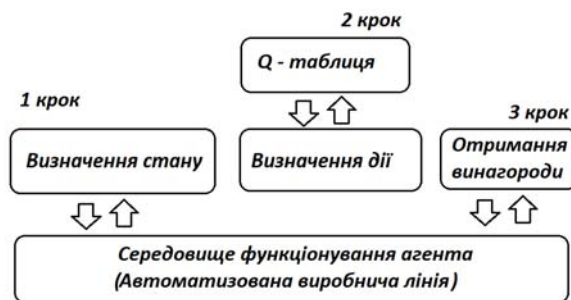


Рисунок 1 – Життєвий цикл агента в середовищі

Джерело: розроблено автором

На першому кроці циклу (рис. 1) агент взаємодіє із середовищем і визначає свій стан. У задачі оптимізації ефективності використання автоматизованої виробничої лінії середовищем є сама лінія, а її стан визначається як стан агента.

Другий крок передбачає визначення множини дій, які агент може виконати в конкретному стані. Це реалізується через взаємодію з Q-таблицею або нейронною мережею (НМ), що виконує функції Q-таблиці. Цей підхід відомий як глибоке навчання з підкріпленням (Deep Q-learning, DQN), коли функції Q-таблиці реалізовано через відповідну НМ.

На основі програми-політики (далі — політика) агент обирає із Q-таблиці одну дію і виконує її.

На третьому кроці циклу агент отримує винагороду від середовища за виконану дію. Тимчасові винагороди, отримані після кожного циклу, накопичуються та зазвичай сумуються. Ця сума розглядається як довготривала винагорода за діяльність агента в середовищі. Саме довготривала винагорода використовується в алгоритмі навчання з підкріпленням для корекції параметрів Q-таблиці та реалізації оптимізації функціонування агента в середовищі.

Для розширення можливостей оптимізації методом навчання з підкріплення (НП) та представлення автоматизованої виробничої лінії як середовища, необхідно виконати коректне моделювання лінії, адаптоване для функціонування агента.

Розглянемо модель автоматизованої виробничої лінії, представлену у вигляді графа. У такому підході вершини графа відповідають обладнанню лінії або її вузлам, які можуть виконувати одну чи декілька виробничих операцій. Кожна вершина графа характеризується низкою параметрів, що відображають специфіку задачі оптимізації ефективності виробничої лінії і у випадку оптимізації часу виробництва це: середній час виконання операції, час простою обладнання, час налаштування, середній час очікування між операціями, а також номери виробничих операцій відповідно до їхньої послідовності у виробничому циклі. Для надійності вузлів враховуються довговічність, середній час між поломками, час відновлення та збережуваність.

Для перевірки запропонованої ідеї було проведено комп'ютерну симуляцію оптимізації часу виробництва на основі алгоритму DQN. Для реалізації комп'ютерної симуляції використано мову програмування Python та її бібліотеки numpy, tensorflow, matplotlib.pyplot.

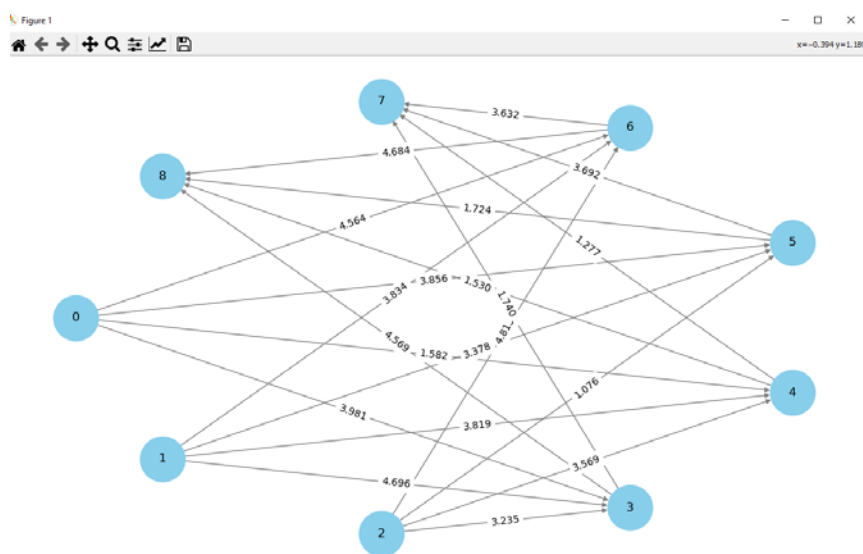


Рисунок 2 – Граф автоматизованої лінії для комп'ютерної симуляції оптимізації часу виробництва

Джерело: розроблено автором

Реалізація комп'ютерної симуляції оптимізації часу виробництва передбачала наступні етапи: підготовка вхідних даних, реалізація симуляції середовища, побудова агента DQN, реалізація алгоритму навчання, оцінка ефективності. Повний код симуляції є об'ємним і тому його наведення в рамках публікації не є доречним.

Етап підготовки вхідних даних передбачав симуляцію лінії з 8 виробничих вузлів відповідно: 3 вузли для 1-ї операції, 4 вузли для 2-ї операції та 2 вузли для 3-ї операції. В результаті середовище автоматизованої виробничої лінії представлено графом з рис. 3.

Фрагмент коду опису типів об'єктів для представлення виробничих операцій та верши графа:

```
# Клас для опису вузлів графа
class Node:
    def __init__(self, operations, longevity_mean, mtbf_mean, recovery_mean,
reliability_mean):
        self.operations = operations # Масив об'єктів Operation
        self.longevity_mean = longevity_mean # Середнє значення довговічності
        self.mtbf_mean = mtbf_mean # Середній час між поломками (Mean
Time Between Failures)
        self.recovery_mean = recovery_mean # Час відновлення
        self.reliability_mean = reliability_mean # Збережуваність

    # Метод для обчислення поточних параметрів надійності вузла
    def compute_reliability_params(self, current_time):
        longevity = np.random.lognormal(mean=np.log(self.longevity_mean),
sigma=0.5)
        mtbf = np.random.normal(loc=self.mtbf_mean, scale=5)
        recovery = np.random.normal(loc=self.recovery_mean, scale=1)
        reliability = np.random.normal(loc=self.reliability_mean, scale=0.02)

        print(f"Time={current_time}: Longevity={longevity:.2f}, MTBF={mtbf:.2f}, "
f"Recovery={recovery:.2f}, Reliability={reliability:.2f}")

        return {
            "longevity": longevity,
            "mtbf": mtbf,
            "recovery": recovery,
            "reliability": reliability
        }
```

Джерело: розроблено автором

Симуляції середовища передбачає створення відповідного класу Production Environment. Цей клас передбачає список дій, станів, винагород та покарань агента.

Дії агента:

1. Виконання операції за нормальних умов:
 - Операція виконується у вузлі, стан якого "working".
 - Час виконання операції скорочується (знижується на 10%).
2. Зупинка вузла для планового ТО:
 - Вузол переходить у стан "planned_maintenance".
 - Дія спрямована на обслуговування вузла, після якого він повертається в робочий стан.
3. Зупинка вузла в результаті аварії (ремонт):
 - Вузол переходить у стан "repairing".
 - Аварія вимагає часу на ремонт перед поверненням у робочий стан.
4. Зупинка вузла для простою:
 - Вузол переходить у стан "idle".
 - Причиною простою є, наприклад, очікування матеріалів.
5. Ігнорування планового ТО (продовження роботи):

- Вузол залишається в стані "working", навіть якщо настав час ТО.
- Це може підвищити ймовірність аварії або передаварійного стану.

Стан вузлів:

1. Робочий стан ("working"):

- Вузол працює та виконує операції без проблем.

2. Підходить час планового ТО ("planned_maintenance"):

- Вузол перебуває у стані технічного обслуговування.

3. Передаварійний стан ("pre_failure"):

- У вузлі спостерігається підвищений ризик аварії (можливий перехід в "failure").

4. Аварія ("failure"):

- Вузол зупинений через поломку та потребує ремонту.

5. Ремонт ("repairing"):

- Вузол ремонтується після аварії.

6. Простій ("idle"):

- Вузол простоє, очікуючи на матеріали або через зупинку попередніх вузлів у графі.

Винагорода та покарання:

1. Виконання операції за нормальних умов:

- *Винагорода*: Негативна величина часу виконання операції, наприклад, - *current_time*. Чим коротший час, тим більше значення винагорода (враховується оптимізація часу).

2. Планове ТО:

- *Штраф*: -3 за час, витрачений на обслуговування вузла.

3. Аварія (ремонт):

- *Штраф*: -10 за аварію.

- Витрачається додатковий час на ремонт (збільшення часу виконання операцій).

4. Простій:

- *Штраф*: -2 за кожен крок у стані "idle".

5. Ігнорування планового ТО:

- Прямої винагороди чи штрафу немає, але підвищується ймовірність:

- Передаварійного стану ("pre_failure").

- Аварії ("failure") із подальшими штрафами.

6. Передаварійний стан:

- *Штраф*: -5, якщо вузол переходить у стан "failure".

- *Винагорода*: Відсутня, якщо вузол успішно повертається до "working".

Етап створення агента передбачав використання Q-функції, аналітичне представлення якої має наступний вигляд:

$$Q(s_t, a_t) = r_t + \gamma \max Q(s_{t+1}, a') \quad (1)$$

де: r_t - винагорода, отримана на поточному кроці,

γ - фактор дисконтування, який визначає вагу майбутніх нагород ($0 \leq \gamma \leq 10$),

$\max Q(s_{t+1}, a')$ - максимальне значення Q-функції для наступного стану s_{t+1} , враховуючи всі можливі дії a' .

Для коригування ваг нейронної мережі під час навчання застосовувалась функція-втрат, яка має наступний вигляд:

$$L = E((Q_{target} - Q(s_t, a_t))^2) \quad (2)$$

де: L – функція втрат

E – математичне сподівання значень функції втрат,

Q_{target} - цільове значення Q-функції,
 $Q(s_t, a_t)$ - поточне передбачення Q-функції.

На сам кінець критерій ефективності функціонування агента, а відповідно і реалізації оптимізації часу виробництва розраховувався за формулою:

$$E = \frac{T_{\text{base}} - T_{\text{DQN}}}{T_{\text{base}}} 100\% \quad (3)$$

Де E – критерій ефективності,

T_{base} – загальний час виробництва без оптимізації використовується як базова точка відліку, щоб обчислити відсоткове поліпшення,

T_{base} – час виробництва з оптимізацією.

В результаті проведення комп'ютерної симуляції згідно критерію (3) було отримано наступну залежність для помилки функціонування нейронної мережі в межах алгоритму агента, для DQN навчання:

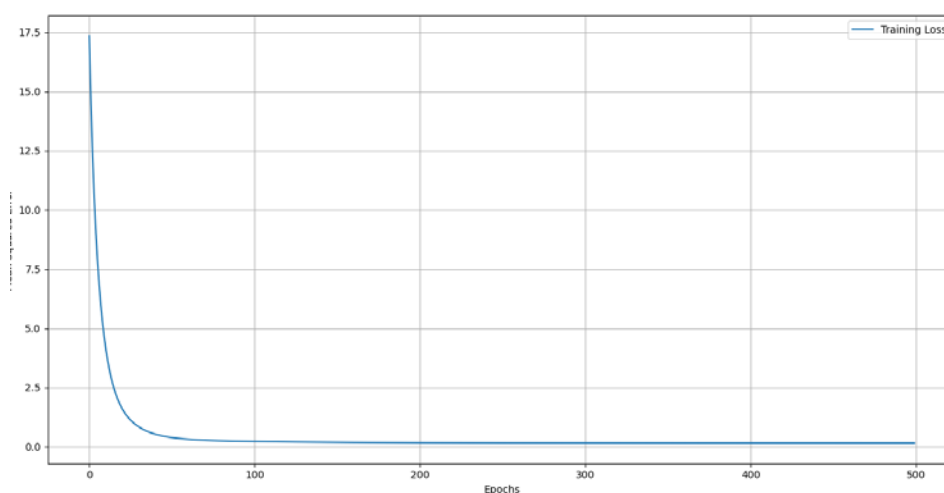


Рисунок 4 – графік мінімізації помилки для обчислення Q-значень

Джерело: розроблено автором

Отримана залежність вказує на позитивний ефект застосування алгоритму НП для оптимізації ефективності використання автоматизованої виробничої лінії в контексті оптимізації її часу виробництва.

Висновки

1. У висновках статті було підкреслено, що запропонований підхід до оптимізації виробничих процесів виявився ефективним для досягнення поставлених цілей. Дослідження підтвердило, що застосування сучасних методів навчання з підкріпленням дозволяє значно зменшити час виконання операцій і мінімізувати простій, що забезпечує суттєве підвищення продуктивності автоматизованих виробничих ліній.

2. Завдяки моделюванню стратегій оптимізації вдалося створити умови для підвищення надійності та довговічності окремих виробничих вузлів. Отримані результати демонструють, що інтеграція інтелектуальних алгоритмів у виробниче середовище дозволяє забезпечити стійкість систем до змінних умов та складних сценаріїв функціонування.

3. Загалом, проведена робота зробила вагомий внесок у розвиток методологій та технологій управління виробничими лініями, використовуючи досягнення в галузі штучного інтелекту. Отримані результати не лише відкривають перспективи подальших досліджень, але й сприяють розробці більш адаптивних, ефективних та надійних систем автоматизації виробництва.

Список літератури

1. Neves M., Vieira M., Neto P. A study on a Q-Learning algorithm application to a manufacturing assembly problem. *Journal of Manufacturing Systems*. Issue 59. 2021. P. 426–440.
2. Аулін В.В., Гриньків А.В., Лисенко С.В., Голуб Д.В. Синергетика підвищення надійності машин використанням моделей марківських процесів. Перспективи і тенденції розвитку конструкцій та технічного сервісу с/х машин і знарядь: зб. матеріалів доп. учасн. V Всеукраїнської науково-практичної конф. Житомир: Житомирський агротехнічний коледж, 2019. С. 242-245.
3. Аулін В. В., Гриньків А. В., Головатий А. О., Лисенко С. В., Голуб Д. В., Кузик О.В., Тихий А. А. Методологічні основи проектування та функціонування інтелектуальних транспортних і виробничих систем: монографія під заг. ред. д.т.н., проф. Ауліна В.В. Кропивницький: Видавець Лисенко В.Ф., 2020. 428с.
4. Zhao, M.; Lu, H.; Yang, S.; Guo, F. The Experience-Memory Q-Learning Algorithm for Robot Path Planning in Unknown Environment. *IEEE Access* 2020, 8, 47824–47844.
5. Palacio, J.C.; Jiménez, Y.M.; Schietgat, L.; Van Doninck, B.; Nowé, A. A Q-Learning algorithm for flexible job shop scheduling in a real-world manufacturing scenario. *Procedia CIRP* 2022, 106, 227–232.
6. Ha D. Reinforcement learning for improving agent design. *Artificial life*. 2019. Т. 25, № 4. С. 352–365. URL: https://doi.org/10.1162/artl_a_00301.
7. Han R., Chen K., Tan C. Curiosity-driven recommendation strategy for adaptive learning via deep reinforcement learning. *British journal of mathematical and statistical psychology*. 2020. Т. 73, № 3. С. 522–540. URL: <https://doi.org/10.1111/bmsp.12199>.
8. Inverse reinforcement learning-based time-dependent A* planner for human-aware robot navigation with local vision / S. Sun та ін. *Advanced robotics*. 2020. Т. 34, № 13. С. 888–901. URL: <https://doi.org/10.1080/01691864.2020.1753569>.
9. L. A. P., Fu M. C. Risk-Sensitive reinforcement learning via policy gradient search. *Foundations and trends® in machine learning*. 2022. Т. 15, № 5. С. 537–693. URL: <https://doi.org/10.1561/22000000091>.
10. Reinforcement learning and adaptive optimization of a class of Markov jump systems with completely unknown dynamic information / S. He та ін. *Neural computing and applications*. 2019. Т. 32, № 18. С. 14311–14320. URL: <https://doi.org/10.1007/s00521-019-04180-2>.
11. Reinforcement learning / B. L. Moore та ін. *Anesthesia & analgesia*. 2011. Т. 112, № 2. С. 360–367. URL: <https://doi.org/10.1213/ane.0b013e31820334a7>.
12. Reinforcement learning for logistics and supply chain management: methodologies, state of the art, and future opportunities / Y. Yan та ін. *Transportation research part E: logistics and transportation review*. 2022. Т. 162. С. 102712. URL: <https://doi.org/10.1016/j.tre.2022.102712>.
13. Dynamic handoff policy for RAN slicing by exploiting deep reinforcement learning / Y. Wu та ін. *EURASIP journal on wireless communications and networking*. 2021. Т. 2021, № 1. URL: <https://doi.org/10.1186/s13638-021-01939-x>.
14. Deep deterministic policy gradient for navigation of mobile robots in simulated environments / J. C. Jesus та ін. 2019 19th international conference on advanced robotics (ICAR), м. Belo Horizonte, Brazil, 2–6 груд. 2019 р. 2019. URL: <https://doi.org/10.1109/icar46387.2019.8981638>.
15. Ковальов С.Г. Ковальов Ю.Г. Особливості реалізації моделі штучної нейронної мережі апаратними засобами. «Наука і технікасьогодні» (Серія «Педагогіка», Серія «Право», Серія «Економіка», Серія «Фізико-математичні науки», Серія «Техніка»): журнал. 2024. №6(34) 2024. С. 1131. URL: DOI: [https://doi.org/10.52058/2786-6025-2024-6\(34\)](https://doi.org/10.52058/2786-6025-2024-6(34)).

References

1. Neves, M., Vieira, M., & Neto, P. (2021). A study on a Q-Learning algorithm application to a manufacturing assembly problem. *Journal of Manufacturing Systems*, 59, 426–440.
2. Aulin, V. V., Hryniv, A. V., Lysenko, S. V., & Holub, D. V. (2019). Synergetics of improving machine reliability using Markov process models. In *Proceedings of the V All-Ukrainian Scientific-Practical Conference "Perspectives and Trends in the Development of Structures and Technical Service of Agricultural Machines and Tools"* (pp. 242–245). Zhytomyr Agricultural Technical College [in Ukrainian].
3. Aulin, V. V., Hryniv, A. V., Holovaty, A. O., Lysenko, S. V., Holub, D. V., Kuzyk, O. V., & Tykhyi, A. A. (2020). Methodological foundations of design and operation of intelligent transportation and manufacturing systems. Lysenko V.F. [in Ukrainian].
4. Zhao, M., Lu, H., Yang, S., & Guo, F. (2020). The experience-memory Q-learning algorithm for robot path planning in unknown environment. *IEEE Access*, 8, 47824–47844. doi.org/10.1109/ACCESS.2020.2978978.
5. Palacio, J. C., Jiménez, Y. M., Schietgat, L., Van Doninck, B., & Nowé, A. (2022). A Q-learning algorithm for flexible job shop scheduling in a real-world manufacturing scenario. *Procedia CIRP*, 106, 227–232.
6. Ha, D. (2019). Reinforcement learning for improving agent design. *Artificial Life*, 25(4), 352–365. https://doi.org/10.1162/artl_a_00301.

7. Han, R., Chen, K., & Tan, C. (2020). Curiosity-driven recommendation strategy for adaptive learning via deep reinforcement learning. *British Journal of Mathematical and Statistical Psychology*, 73(3), 522–540. <https://doi.org/10.1111/bmsp.12199>.
8. Sun, S., et al. (2020). Inverse reinforcement learning-based time-dependent A* planner for human-aware robot navigation with local vision. *Advanced Robotics*, 34(13), 888–901. doi.org/10.1080/01691864.2020.1753569.
9. L. A. P., & Fu, M. C. (2022). Risk-sensitive reinforcement learning via policy gradient search. *Foundations and Trends® in Machine Learning*, 15(5), 537–693. <https://doi.org/10.1561/22000000091>.
10. He, S., et al. (2019). Reinforcement learning and adaptive optimization of a class of Markov jump systems with completely unknown dynamic information. *Neural Computing and Applications*, 32(18), 14311–14320. <https://doi.org/10.1007/s00521-019-04180-2>.
11. Moore, B. L., et al. (2011). Reinforcement learning. *Anesthesia & Analgesia*, 112(2), 360–367. <https://doi.org/10.1213/ane.0b013e31820334a7>
12. Yan, Y., et al. (2022). Reinforcement learning for logistics and supply chain management: Methodologies, state of the art, and future opportunities. *Transportation Research Part E: Logistics and Transportation Review*, 162, 102712. <https://doi.org/10.1016/j.tre.2022.102712>.
13. Wu, Y., et al. (2021). Dynamic handoff policy for RAN slicing by exploiting deep reinforcement learning. *EURASIP Journal on Wireless Communications and Networking*, 2021(1). doi.org/10.1186/s13638-021-01939-x.
14. Jesus, J. C., et al. (2019). Deep deterministic policy gradient for navigation of mobile robots in simulated environments. In *Proceedings of the 2019 19th International Conference on Advanced Robotics (ICAR)*, Belo Horizonte, Brazil. IEEE. <https://doi.org/10.1109/icar46387.2019.8981638>.
15. Kovalev, S. G., & Kovalev, Yu. G. (2024). Features of implementing an artificial neural network model using hardware. *Nauka i Tekhnika Sohodni*, (6(34)), 11–31. [doi.org/10.52058/2786-6025-2024-6\(34\)](https://doi.org/10.52058/2786-6025-2024-6(34)) [in Ukrainian].

Serhii Kovalov, PhD ped. sci.

Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine

Optimization of Production Time using the Reinforcement Learning Method as a Particular Case Of Improving the Efficiency of Automated Production Lines

This article examines the application of reinforcement learning methods to optimize the production time of an automated production line modeled as a graph. In this graph representation, nodes correspond to pieces of equipment capable of performing one or multiple manufacturing operations. Such an approach not only creates a precise model of the agent's operating environment but also enables the implementation of computer simulations. These simulations serve as a critical foundation for assessing the potential effectiveness of reinforcement learning methods in optimizing real-world production lines. By adopting this approach, the study explores opportunities to improve efficiency, optimize resource utilization, and enhance the reliability of production systems.

A key focus of the article is the detailed investigation of the stages involved in the computer simulation of production time optimization. The simulation process consisted of several integral stages: preparation of input data, design and implementation of the simulation environment, construction of a Deep Q-Network (DQN) agent, execution of the learning algorithm, and evaluation of optimization efficiency. These stages are thoroughly analyzed, demonstrating the systematic approach required to integrate reinforcement learning into manufacturing processes.

The research also emphasizes the advantages of modeling the production line as a graph, highlighting how it enables the simulation of dynamic and complex production environments. This graph-based framework provides the agent with a structured understanding of equipment connectivity and operational constraints, allowing it to develop effective decision-making policies. Through iterative interactions with the environment, the DQN agent identifies optimal production sequences, minimizes downtime, and enhances throughput.

Furthermore, the article explores the practical implications of integrating reinforcement learning into industrial applications. Computer simulations not only validate the feasibility of these methods but also provide insights into their scalability and adaptability to diverse manufacturing scenarios. The findings underscore the potential of reinforcement learning to transform automated production lines into more intelligent, adaptive, and resilient systems.

By addressing both theoretical and practical aspects, the study lays the groundwork for future research in applying artificial intelligence to industrial automation. This comprehensive approach enables stakeholders to better understand the value of advanced learning algorithms in boosting operational efficiency and ensuring sustainable growth in automated production.

production lines, optimization of utilization efficiency, artificial intelligence, production time, modeling the line as a system, state graphs

Одержано (Received) 14.03.2025

Прорецензовано (Reviewed) 18.03.2025

Прийнято до друку (Approved) 21.03.2025