

## ВИКОРИСТАННЯ ШІ-ПЛАНУВАННЯ ДЛЯ ПОБУДОВИ ІНДИВІДУАЛЬНИХ ОСВІТНІХ ТРАЄКТОРІЙ Гришанова І.Ю., Рогушина Ю.В

### USE OF AI-PLANNING FOR DEVELOPMENT OF PERSONAL LEARNING TRAJECTORIES

. / Rogushina J.V., Grishanova I.J.

Інститут програмних систем НАН України / Institute of Software Systems, National  
Academy of Sciences of Ukraine

03187, Київ, пр. проспект Академіка Глушкова, 40 корп.5,

E-mail: i26031966@gmail.com

We propose an algorithm for service composition based on the Q-learning method. Optimization method for selection the best flow that takes into account quality criteria (QoS) is developed. This approach is approved for designing of personal learning trajectories, particularly for automated personalized selection of learning objects according to the individual information needs of students and determining the sequence of their study. For this purpose, the main components of this task are described in terms of reinforcement learning: state, actions, reward, and agent strategy.

Зараз в Україні в умовах війни посилилась актуальність персоніфікації навчання. Це обумовлено як необхідністю навчання людей різного віку та з різним досвідом, збільшенням частки дистанційного навчання через вимоги безпеки та потребою у найбільш ефективному використанні людського ресурсу та оптимізації тривалості навчання. Це викликає потребу у розробці *індивідуальних освітніх траєкторій* (ІОТ) [1], що враховує наявні знання та навички студентів, їх цілі та особисті вподобання. Традиційні підходи до планування навчання часто виявляються недостатньо гнучкими або надто складними для реалізації в умовах великих освітніх програм та дефіциту часу для їх отримання, тоді як персоніфікований підхід потребує значно більше часу кваліфікованих експертів та має враховувати велику кількість характеристик студентів, засобів та інструментів навчання та змін у навчальному середовищі.

Застосування технологій *штучного інтелекту* (ШІ) в плануванні відкриває нові перспективи для автоматизації та оптимізації процесу побудови ІОТ. Наприклад, використання алгоритмів навчання з підкріпленням [2], які здатні формувати рішення у складних і динамічних середовищах, дозволяє знаходити такі *навчальні об'єкти* (НО), на основі яких навчання конкретного студента здійснюється найбільш ефективно відповідно до обраних критеріїв (таких як час навчання, вартість ресурсів, якість отриманих навичок тощо).

Використання НО формалізується як сервіс, в якому вхідними даними є вимоги до здобувача освіти, а вихідними – компетенції, які дозволяє отримати вивчення контенту цього НО. Тоді побудова ІОТ розглядається як специфічний випадок композиції сервісів, де початковим станом є індивідуальні компетенції студента до початку навчання, а результуючим – набір компетенцій, які потрібно отримати в результаті вивчення певних навчальних курсів. При цьому набір НО та їх властивості можуть змінюватися динамічно, а метадані НО дозволяють визначити значення QoS відповідних сервісів. Особливістю підходу є адаптація *ШІ-планування* навчального процесу до потреб студента на основі алгоритму Q-learning [3] шляхом автоматичного вибору НО, які відповідають заданим цілям. Запропонований підхід може застосовуватися для формування освітніх траєкторій у формальній, неформальній та інформальній освіті.

Компоненти задачі машинного навчання (Machine Learning – ML) інтерпретуються наступним чином: *середовище* – набір навчальних модулів, кожен із яких визначається входами (наявні знання/компетенції) та виходами (нові знання/компетенції); *агент* – алгоритм, який обирає послідовність вивчення НО для визначеної цілі; *стан* – множина компетенцій, якими володіє студент у певний момент

часу; дії – вибір НО з набору доступних; *ціль* – отримання певного набору компетенцій. Застосування МЛ дозволяє визначити послідовність НО, яка дозволяє перейти з початкового стану (попередніх компетенцій) у цільовий стан (задані компетенції) із максимальною ефективністю відповідно до параметрів QoS. Особливість виконання задачі полягає в тому, що це послідовне виконання дій без можливостей розпаралелювання.

*Представлення навчального середовища.* Кожен НО представлено як сервіс, що має входи – попередні знання (компетенції), необхідні для проходження модуля, виходи – знання (компетенції), які отримуються після завершення модуля і QoS: тривалість навчання, вартість, складність. Маршрут навчання – це граф, де вузли відповідають станам компетенцій студента, а ребра — діям з вивчення НО [4].

*Застосування Q-learning.* Для вибору оптимального шляху використовується алгоритм Q-learning, який дозволяє знаходити оптимальну стратегію дій агента. Під час навчання агент обирає сервіси, які відповідають доступним кодам, та оцінює винагороду для кожного обраного сервісу з урахуванням QoS-параметрів, таких як тривалість курсу, вартість і складність вивчення. Навчання агента відбувається з використанням  $\epsilon$ -жадібною стратегії для балансування дослідження та експлуатації, де обчислення винагороди відбувається на основі QoS. Оптимізація для запобігання тупиковим станам полягає у перевірках та виключенні надмірних або непотрібних дій. Навчання завершується, коли досягнуто цільового стану або виконано задану кількість ітерацій. За результатами отриманої множини можливих планів (маршрутів) виконується визначення оптимального за параметрами QoS. На рис.1 наведено приклад побудови ІОТ для студента, що має базові знання з математики та хоче вивчати машинне навчання, використовуючи НО з розділів вищої математики (теорія ймовірностей та лінійна алгебра), теорії програмування і практичного використання Python, для якого необхідні знання з теорії програмування. У графі ІОТ вершини – стани, а ребра – дії (вибір НО).

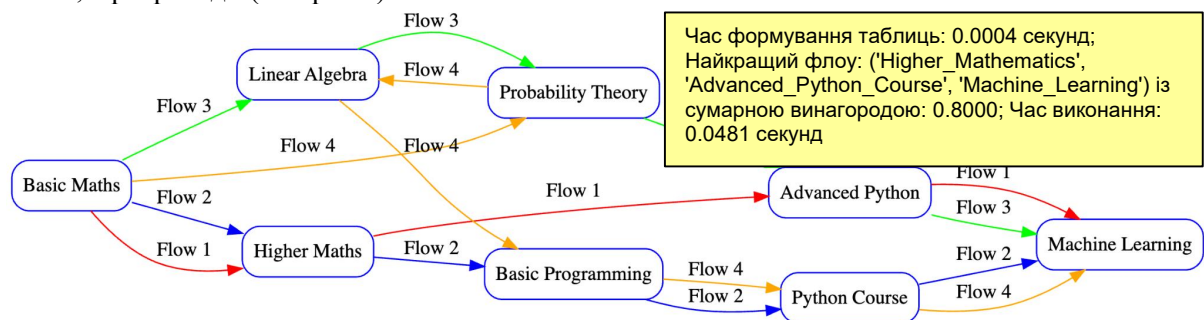


Рис.1. Маршрути отримання компетенцій та результати роботи алгоритма.

Основні відмінності запропонованого методу від базового Q-learning: модель орієнтована на навчальні потреби, а не на класичні завдання з дискретними станами, введені глобальні ваги важливості параметрів QoS; додано механізми автоматичного налаштування параметрів навчання; реалізовано перевірку на доцільність стану, щоб уникати маршрутів, які не ведуть до цілі; додано перевірку зациклювання – алгоритм визначає й припиняє повторювані дії (контроль зациклювання реалізований за допомогою обмеження кількості кроків в одному епізоді і перевірку прогресу); розширено вибір стратегій зниження  $\epsilon$  (лінійного або експоненційного) для гнучкості навчання і оновлено формулу розрахунку винагород з урахуванням глобальних ваг QoS.

Запропонований підхід є кроком у напрямку використання сучасних методів ШІ для вирішення задач автоматизації навчального процесу та персоналізації освіти.

В результаті виконаних досліджень запропоновано алгоритм ШІ-планування, який забезпечує створення композиції сервісів на основі Q-learning та його оптимізацію для вибору найкращого маршруту з урахуванням критеріїв якості QoS. Цей підхід апробовано для проектування індивідуальних освітніх траєкторій, а саме – для автоматизованого персоналізованого підбору навчальних об'єктів відповідно до

індивідуальних інформаційних потреб студентів та визначення послідовності їх вивчення.

1. J. V. Rogushina, A. Y. Gladun, O. V. Anishchenko, S. M. Pryima, (2024) Semantic Support of Personal Learning Trajectory Development, in: UkrPROG 2024 International Scientific and Practical Programming Conference, CEUR Vol-3806, p.487-505.
2. A.G. Barto, R. S. Sutton, (2018). Reinforcement Learning: An Introduction, The MIT Press: Cambridge, Massachusetts, 2018.
3. B. Jang, M. Kim, G. Harerimana, J. W. Kim, (2019). Q-learning algorithms: A comprehensive classification and applications. IEEE access, 7, 133653-133667.
4. Гришанова І.Ю., Рогушина Ю.В., (2024) Використання методу Q-learning для побудови та оптимізації маршрутів композитного вебсервісу. Проблеми програмування, № 4.

## **ПЕРСПЕКТИВИ ВПРОВАДЖЕННЯ ІНДЕКСУ НА БАЗІ СУФІКСНОГО ДЕРЕВА ДЛЯ ПОШУКУ ПІДРЯДКІВ У СУБД ВЕЛИКОГО РОЗМІРУ Зважій Д.В**

### **PROSPECTS OF IMPLEMENTING A SUFFIX TREE-BASED INDEX FOR SUBSTRINGS SEARCHING IN LARGE DMBS Zvazhii D. V.**

*This study explores the advantages and disadvantages of implementing a suffix tree-based index to optimize substring search operations in a DBMS when working with large datasets. Experimental evaluations of the time complexity for substring search operations using suffix trees and database management systems like Elasticsearch, PostgreSQL, MySQL, and ClickHouse have been conducted. Based on the obtained results, the hypothesis regarding the potential efficiency of implementing a suffix tree-based index to optimize substring search operations in a DBMS is confirmed.*

#### **Вступ:**

Сучасні інформаційні системи стикаються зі стрімким зростанням обсягів даних, що створює нові виклики для їх зберігання та обробки. Ефективний доступ до даних, особливо під час виконання пошукових операцій, стає критично важливим для підтримки продуктивності ІТ-систем. Реляційні та нереляційні системи керування базами даних (СУБД), такі як *PostgreSQL*, *MySQL*, *ClickHouse*, *Elasticsearch*, пропонують різноманітні інструменти для швидкого пошуку та індексації. Проте індекси, що зазвичай використовуються, зокрема на основі В-дерев [1] або інвертованих індексів [2], не завжди оптимально підходять для пошуку підрядків у великих текстових масивах. Це часто призводить до збільшення часу обробки запитів та обмеження масштабованості системи.

Суфіксне дерево — це структура даних [3], яка може значно скоротити час виконання пошуку підрядків, роблячи її потенційно ефективним рішенням для покращення роботи СУБД з великими текстовими обсягами. У нашій роботі ми прагнули експериментально підтвердити гіпотезу, що впровадження суфіксного дерева як індексу для сучасних СУБД дозволить підвищити швидкість виконання операцій пошуку підрядків.

#### **Методологія:**