

УДК 004.4:004.9

К. С. Сидоренко; М. С. Широкопетлева

## ГРАФОВА МОДЕЛЬ ДЛЯ ГЕНЕРАЦІЇ ПЕРСОНАЛІЗОВАНИХ НАВЧАЛЬНИХ ШЛЯХІВ У ВЕБ-СИСТЕМАХ НАВЧАННЯ

*В статті запропонована формальна графова модель для планування персоналізованих траєкторій навчання у веб-орієнтованій освіті, що ставить за мету усунути нестачу математично обґрунтованого апарату опанування навчальних планів в послідовному порядку.*

*Дорожня карта предметної галузі моделюється як когнітивно орієнтований спрямований ациклічний граф (DAG), де вершини представляють окремі концепції, а спрямовані ребра фіксують ієрархічні відносини передумов. Структурна валідність навчальних планів, зокрема дотримання ациклічності, підтримується за допомогою топологічного сортування методом Кана та алгоритму Тар'яна, який дозволяє ідентифікувати циклічні залежності через декомпозицію сильно зв'язних компонентів. Для визначення поточного стану учня система застосовує обхід графа у глибину, що дозволяє обчислити замикання предків і нащадків. Цей підхід допомагає виділити досяжну межу знань - підмножину ще не опанованих тем, для яких усі необхідні попередні умови знаходяться в множині опанованих вершин.*

*Процес стимулювання прогресу та вибір завдань для динамічного оцінювання формулюється як наближення до розв'язання комбінаторної задачі оптимізації про максимальне  $k$ -покриття. Впроваджуються дві комплементарні алгоритмічні стратегії, першою з яких є стратегія стратифікованого розміщення для первинної діагностики. Вона розділяє граф на шари та розподіляє ресурси тестування пропорційно до середнього структурного впливу кожного рівня, забезпечуючи всебічне охоплення матеріалу. Другою стратегією є жадібний алгоритм адаптивного вибору, який працює з досяжною межею знань. Він обирає завдання на основі максимізації показників структурного впливу та граничного приросту інформації, спираючись на оптимістичне припущення про успішне засвоєння тем.*

*На підставі запропонованої формальної графової моделі і відповідних алгоритмів створена програмна система, яка дозволяє будувати індивідуальні освітні маршрути, виходячи безпосередньо з формальних властивостей графа. У подальшому планується розширити модель імовірнісними методами, такими як баєсове простежування знань та динамічні баєсові мережі.*

**Ключові слова:** *графи, програмне забезпечення, веб-орієнтовані навчальні системи, динамічна траєкторія навчання, самооцінювання, топологічне сортування, сильно зв'язані компоненти, задача максимального  $k$ -покриття.*

### Вступ

Сучасна освіта зазнала фундаментального зсуву парадигми від традиційної (сфокусованої навколо викладача), до конструктивістських підходів, які надають пріоритет автономії учнів, самостійному розвитку та можливості використання віддаленого доступу, що полегшується цифровими технологіями. Хоча платформи онлайн-навчання дозволяють здійснювати освітні процеси незалежно від фізичної спільної присутності, більшість наявних систем продовжують відтворювати традиційні структури класних кімнат, вимагаючи від усіх учнів дотримуватися однакових послідовностей навчання незалежно від попередніх знань.

Відсутність адаптивних механізмів визначення послідовності навчання призводить до добре задокументованих проблем, включаючи когнітивне перевантаження та дезорієнтацію, коли навчальні матеріали не відповідають рівню готовності учня. Масштабні системи управління навчанням, такі як Moodle та Coursera, зазвичай реалізують шляхи навчання як попередньо визначені послідовності модулів або ієрархічні структури курсів. Хоча ці моделі підтримують організацію та подання контенту, вони не пропонують структур, які сприяють адаптивному навчанням. Персоналізована послідовність навчальних програм залишається

критичною та недостатньо вирішеною проблемою в веб-освіті [1].

Класичні графові методи дозволяють підтримувати механізми моделювання та обробки структур залежностей знань. Важливість такого представлення додатково посилюється роллю формувального оцінювання в сучасному адаптивному навчанні. Ефективний формувальний зворотний зв'язок потребує розуміння того, як опанування одного поняття зумовлює засвоєння пов'язаних із ним концептів [2]. Оскільки самооцінювання дає студентам можливість самостійно регулювати темп і прогрес навчання, вибір об'єктів оцінювання має враховувати структурні зв'язки між темами для отримання діагностично значущих результатів. Ця вимога стимулює розробку алгоритмічних стратегій, здатних обирати та пропонувати теми для оцінювання на основі попереднього рівня знань студента щодо представлення нового матеріалу та поточного прогресу студента в його опануванні.

**Метою цієї роботи** є формування моделі структурування та планування персоналізованих траєкторій навчання у веб-системах. Також необхідно визначити структуру даних, придатну для представлення дорожньої карти навчання з урахуванням обмежень предметної області. На динамічному рівні персоналізована траєкторія навчання визначається як специфічний для конкретного студента шлях, похідний від дорожньої карти, що формується через структуроване оцінювання.

Для досягнення поставленої мети було сформульовано такі завдання:

1. Формально визначити дорожню карту навчання як модель предметної області та охарактеризувати її структурні властивості.
2. Визначити та застосувати графові алгоритми для вирішення таких завдань, як структурної валідації та організації зберігання даних.
3. Запропонувати стратегії створення оцінювання за наявності заданих ресурсних обмежень для підтримки самонавчання студентів.
4. Продемонструвати інтеграцію запропонованих компонентів у межах цілісної архітектури системи.

## Огляд літератури

Дослідження персоналізованого навчання у вебсистемах можна розділити на три основні напрями: графове представлення знань, планування навчальних програм на основі оцінювання та адаптивні механізми зі зворотним зв'язком. Кожен із цих напрямів пропонує часткові рішення проблеми планування персоналізованих траєкторій.

Згідно з [3], графові методи застосовуються в системах електронного навчання для генерації персоналізованих шляхів. Запропонований авторами алгоритм генерації навчальних траєкторій (LPG) використовує карти понять, представлені у вигляді графів, кластеризує студентів на основі їхньої успішності та застосовує топологічне сортування для створення спрощених послідовностей, що узгоджуються з пререквізитами [3]. Ця робота доводить доцільність застосування методів теорії графів для структурування навчальних планів і встановлює топологічне впорядкування як принципову основу для дотримання послідовності вивчення тем. Проте запропонований алгоритм функціонує як ізольована процедура розрахунку шляху. Як наслідок, отриманий результат є статичним і не залежить від поточного рівня знань конкретного студента.

Підходи, що базуються на оцінюванні, вирішують проблему врахування поточного рівня знань студента, але зазвичай працюють без формально визначених структур залежностей. Існують праці, що пропонують використання генетичного алгоритму для генерації персоналізованих шляхів навчання, де попереднє тестування (pre-test) використовується для виявлення неправильних відповідей, а функція придатності, яка використовується для балансування між складністю матеріалу та зв'язаністю концептів, застосовується для побудови індивідуальних траєкторій [1]. Експериментальні результати демонструють, що структурування навчального плану за результатами оцінювання значно підвищує ефективність навчання порівняно з режимами вільного перегляду навчального матеріалу.

Низка робіт підкреслює важливість динамічної генерації завдань на основі рівня знань студента. Такі підходи підтримують навчання через набори практичних завдань різного рівня складності, що коригується відповідно до результатів на попередніх етапах. У цій парадигмі студенти, які демонструють високі результати, отримують складніші завдання для подальшого розвитку навичок, тоді як тим, хто має труднощі, пропонуються простіші справи для підвищення ймовірності опанування базового теоретичного змісту. Як зазначається у [4], під час генерації наборів практичних завдань слід враховувати індивідуальність кожного студента та його специфічний прогрес, гарантуючи, що оцінювання відповідає персональному рівню готовності та траєкторії розвитку.

Формування адаптивного зворотного зв'язку представляє третій напрям досліджень, де оцінювання пов'язане зі структурованою організацією знань, а не з формальними графами залежностей. В [2] описано розроблений інструмент самооцінювання, у якому методи семантичної близькості на основі NLP використовуються для вибору пояснювальних фрагментів із довідкових текстів у відповідь на відповіді студента. Інструмент включає персоналізацію через впорядкування питань за складністю та відстеження історії використання. Хоча ця робота суттєво покращує якість і автоматизацію зворотного зв'язку, структура знань, на основі якої здійснюється оцінювання, є ієрархією тем, а не формально визначеним графом пререквізитів. Вибір елементів оцінювання зумовлений семантичною пов'язаністю зі змістом запитання, а не структурним охопленням зв'язків між темами. Таким чином, зв'язок між результатами оцінювання та позицією студента у формально визначеній структурі знань залишається невстановленим.

З іншого боку, сучасні системи штучного інтелекту, зокрема алгоритми відстеження знань (knowledge tracing), рекомендаційні системи та великі мовні моделі (LLM), забезпечують потужні інструменти для мікроадаптації контенту, динамічної інтерпретації помилок та генерації персоналізованих пояснень. Прикладами такої мікроадаптації є використання методів обробки природної мови (NLP) для аналізу семантичної близькості та динамічного вибору пояснювальних фрагментів у відповідь на помилки студента, а також інтелектуальна генерація контекстно-залежних діалогів у системах вивчення мов. Проте ці AI-моделі переважно функціонують на семантичному або імовірнісному рівнях. Без чіткого математичного фундаменту предметної області такі AI-моделі здатні генерувати педагогічно некоректні освітні траєкторії, порушуючи вимоги пререквізитів. Тому запропонований у цій роботі алгоритмічно-графовий підхід не конкурує із сучасними AI-засобами, а доповнює їх. Він виступає детермінованим структурним каркасом (макрорівнем), який гарантує топологічну валідність ієрархії знань, на базі якої AI-моделі можуть безпечно та ефективно реалізовувати мікроадаптацію та імовірнісне моделювання знань без ризику навігаційної неоднозначності

Крім того, адаптивні системи у вивченні мов та в інших галузях продемонстрували, що персоналізовані підходи стабільно перевершують уніфіковані моделі навчання за результатами засвоєння матеріалу, що підтверджується дослідженнями в різних освітніх контекстах, включаючи системи навчання англійської розмовної мови на основі ШІ [5].

Наявна література демонструє цінність адаптивних систем навчання. Проте в наявних роботах бракує уніфікованої математичної моделі, яка б не лише забезпечувала детерміноване планування персоналізованих навчальних траєкторій, але й слугувала б структурно надійним фундаментом для безпечної інтеграції сучасних імовірнісних AI-методів та рекомендаційних систем у межах веб-систем. Ця стаття спрямована на усунення зазначеної прогалини шляхом впровадження цілісної моделі, що інтегрує графові алгоритми в архітектуру системи.

### **Модель предметної області**

Дослідження моделювання предметних областей у навчальних системах виділяють два основні підходи до організації знань. Педагогічно орієнтовані моделі представляють Наукові праці ВНТУ, 2026, № 1

структуру предметної області у вигляді дерев, де теми ієрархічно декомпонуються на підтеми. Такий підхід забезпечує чіткий глобальний огляд структури курсу, є простим у реалізації та візуалізації. Проте представлення у вигляді дерева нав'язує єдину декомпозицію «батько-нащадок», що обмежує концептуальну гнучкість, може призводити до надмірно глибоких ієрархій і не здатне відобразити випадки, коли поняття залежить від декількох незалежних пререквізитів.

Когнітивно орієнтовані моделі долають це обмеження, представляючи знання як графи взаємопов'язаних концептів, що дозволяє явно фіксувати відношення залежності, які виходять за межі суворої ієрархічної декомпозиції [6]. Такі представлення підтримують ширші можливості для структурного аналізу та дозволяють безпосередньо виражати концептуальні залежності. Основним ризиком таких моделей, за відсутності структурних обмежень, є навігаційна неоднозначність та потенційна поява циклічних або некоректно визначених ланцюжків залежностей, що робить послідовність навчання невизначеною [6]. Запропонована модель узгоджується з когнітивно орієнтованими представленнями, де відношення пререквізитів моделюються як орієнтовані ребра.

Дорожня карта навчання представлена як математична структура над скінченною множиною навчальних концептів. Дорожня карта визначається як орієнтований граф  $G = (V, E)$  [7], де кожна вершина  $v \in V$  відповідає навчальній темі, а кожне орієнтоване ребро  $(u, v) \in E$  кодує відношення пререквізиту, вказуючи на те, що концепт  $u$  має бути засвоєний перед тим, як можна буде продуктивно приступити до вивчення концепту  $v$ . Вершини можуть містити семантичні метадані, такі як назва теми, опис та навчальні ресурси, що не впливає на структурну семантику відношення залежності.

Для того щоб дорожня карта представляла логічно несуперечливу послідовність навчання, граф  $G$  повинен задовольняти обмеженню ациклічності:  $G$  не може містити орієнтованих циклів. Наявність циклу в  $G$  означала б, що певна тема через ланцюжок залежностей одночасно є пререквізитом самої себе, що є семантично некоректним. Відповідно, структурна валідність  $G$  визначається як відсутність орієнтованих циклів, і ця умова має виконуватися кожного разу, коли  $G$  фіксується у постійному сховищі даних.

Спрямований ациклічний граф (DAG) гарантує існування принаймні одного топологічного сортування  $V$  – лінійного впорядкування вершин, такого, що для кожного ребра  $(u, v) \in E$  вершина  $u$  передре  $v$  у послідовності [8]. Ця властивість гарантує, що валідні траєкторії навчання можуть бути обчислені детерміновано, виконання пререквізитів може бути перевірено алгоритмічно для будь-якого заданого стану студента, а досяжна множина ще не засвоєних тем може бути ідентифікована за допомогою ефективного обходу графа. Крім того, структура DAG підтримує обчислення замикання предків та нащадків, що є необхідним для механізму вибору об'єктів оцінювання, описаного в наступних розділах.

Дорожня карта  $G$  не розглядається як статичний артефакт. Теми та відношення пререквізитів можуть додаватися або видалятися з часом відповідно до еволюції знань у предметній області або змін у методиках викладання. Це породжує постійну необхідність валідації: щоразу, коли  $G$  змінюється, його структурна валідність має бути повторно оцінена перед збереженням оновленого стану. Невалідні стани мають виявлятися детерміновано і не повинні потрапляти до сховища, що вимагає повторної перевірки обмеження ациклічності перед кожною транзакцією запису оновленої структури. Алгоритмічна реалізація цієї процедури валідації розглядається в наступному розділі.

Студент, який взаємодіє із системою, у будь-який момент часу характеризується множиною засвоєння  $M \subseteq V$ , що представляє теми, які студент успішно опанував. Доповнення  $V \setminus M$  представляє незасвоєні теми. Серед незасвоєних тем виділяється підмножина, яка є досяжною за поточного рівня знань – зокрема ті теми, чиє замикання пререквізитів повністю міститься в  $M$ . Ця досяжна множина становить простір тем, на основі яких може бути побудована персоналізована траєкторія навчання.

## Графова обробка

Обробка дорожньої карти навчання спирається на набір графових алгоритмів, які систематично застосовуються в системі для структурної валідації, аналізу залежностей та обчислення поточного стану студента. Формально визначимо ці алгоритми, а також надамо обґрунтування та обмеження їх використання в межах запропонованої моделі.

Центральною структурною вимогою до дорожньої карти навчання є ациклічність. Алгоритмічна перевірка цієї властивості еквівалентна визначенню того, чи існує для графа  $G = (V, E)$  топологічне сортування [9]. Топологічне сортування – це бієкція  $\sigma : V \rightarrow \{1, 2, \dots, |V|\}$ , така, що для кожного ребра  $(u, v) \in E$  виконується умова  $\sigma(u) < \sigma(v)$ . Згідно з фундаментальною теоремою теорії графів, топологічне сортування орієнтованого графа існує тоді й лише тоді, коли граф не містить орієнтованих циклів. Ця еквівалентність означає, що будь-який алгоритм обчислення топологічного сортування одночасно слугує процедурою виявлення циклів.

Алгоритм Кана обчислює топологічне сортування ітеративно. Він ініціалізує чергу  $Q$  усіма вершинами, напівступінь входу яких дорівнює нулю (тобто вершинами, що не мають пререквізитів). На кожному кроці вершина  $u$  вилучається з  $Q$ , додається до вихідної послідовності  $\sigma$ , а для кожного наступника  $v$  вершини  $u$  напівступінь входу зменшується на одиницю. Якщо в результаті напівступінь входу  $v$  стає рівним нулю,  $v$  додається до  $Q$ . Алгоритм завершується, коли  $Q$  порожня. Якщо довжина  $\sigma$  дорівнює  $|V|$ , граф є ациклічним, а  $\sigma$  – валідним топологічним сортуванням [10]. Якщо  $|\sigma| < |V|$ , вершини, відсутні в  $\sigma$ , беруть участь принаймні в одному орієнтованому циклі, і модифікація зв'язків, що призвела до появи циклу, має бути відхилена до моменту збереження стану графа у постійному сховищі. Час виконання алгоритму можна визначити як  $O(|V| + |E|)$ .

Альтернативний підхід використовує пошук у глибину (DFS). Під час обходу графа  $G$  у глибину кожна вершина проходить через три стани: невідвіdana, у процесі обробки та завершена. Вершина маркується як «у процесі», коли починається її обхід, і як «завершена», коли всі досяжні з неї вершини повністю оброблені. Орієнтований цикл присутній тоді й лише тоді, коли під час обходу виявляється ребро, що веде до вершини, яка наразі маркована як «у процесі» (це вказує на зворотне ребро, що замикає цикл). Цей підхід також працює за час  $O(|V| + |E|)$  і дозволяє отримати топологічне сортування шляхом додавання кожної вершини до вихідної послідовності після завершення її обробки з подальшим реверсом отриманої послідовності [10].

Коли цикл виявлено, системі потрібен не лише бінарний вердикт про валідність, а й діагностична інформація для ідентифікації залучених вершин [11]. Це завдання вирішується через декомпозицію на компоненти сильної зв'язності. Компонента сильної зв'язності орієнтованого графа – це максимальна підмножина  $S \subseteq V$ , така, що для кожної пари вершин  $u, v \in S$  існує орієнтований шлях від  $u$  до  $v$  і від  $v$  до  $u$ . У валідному DAG кожна компонента сильної зв'язності містить рівно одну вершину. Будь-яка компонента розміром більше одиниці вказує на набір взаємозалежних концептів, чий відношення пререквізитів утворюють цикл. Алгоритм Тар'яна обчислює всі компоненти сильної зв'язності за час  $O(|V| + |E|)$  за допомогою одного обходу в глибину, доповненого стеком та значенням слабкого зв'язку, який присвоюється кожній вершині [7]. Значення слабкого зв'язку вершини – це найменший індекс виявлення, досяжний з  $u$  через поточне дерево пошуку в глибину. Коли значення слабкого зв'язку  $u$  дорівнює його власному індексу виявлення після завершення,  $u$  є коренем сильно зв'язного компонента, а вершини в стеку над  $u$  утворюють цей компонент. Компоненти, розміром більше 1, є маркерами точних місць порушення циклічних залежностей, про що має повідомляти система.

Для структурно валідної дорожньої карти  $G = (V, E)$  визначено дві операції замикання над її множиною вершин, які обчислюються шляхом обходу орієнтованого графа. Ці операції характеризують повний контекст залежностей будь-якої вершини і використовуються в

системі для контролю прогресу, оцінки рівня знань студента та вибору завдань.

Замикання предків вершини  $v$ , що позначається  $\text{Anc}(v)$ , – це множина всіх вершин  $u \in V$ , для яких у  $G$  існує орієнтований шлях від  $u$  до  $v$ . Формально це можна представити як:  $\text{Anc}(v) = \{u \in V : u \neq v \wedge \exists \text{ path } u \rightarrow^* v \text{ in } G\}$ . Замикання предків представляє повний набір пререквізитів  $v$  – як прямих, так і транзитивних.  $\text{Anc}(v)$  обчислюється шляхом пошуку в глибину на транспонованому графі.

Замикання нащадків вершини  $v$ , що позначається  $\text{Desc}(v)$ , – це множина всіх вершин  $w \in V$ , для яких у  $G$  існує орієнтований шлях від  $v$  до  $w$ . Формально це можна представити як:  $\text{Desc}(v) = \{w \in V : w \neq v \wedge \exists \text{ path } v \rightarrow^* w \text{ in } G\}$ . Замикання нащадків представляє всі теми, чия майбутня доступність залежить від опанування  $v$ . Воно обчислюється шляхом пошуку в глибину в  $G$ , починаючи з  $v$ .

Обидва замикання можуть бути попередньо обчислені та кешовані для всіх вершин за час  $O(|V| \cdot (|V| + |E|))$  шляхом виконання незалежних обходів для кожної вершини. Для дорожніх карт того масштабу, що зазвичай використовується в навчальних системах (де  $|V|$  становить від десятків до сотень вершин), таке попереднє обчислення є доцільним і знижує вартість повторних запитів до  $O(1)$ .

Операція замикання предків забезпечує обчислювальну основу для визначення тем, доступних студенту в будь-який момент його навчання. Для заданої множини засвоєння  $M \subseteq V$  досяжна множина визначається як  $F(M) = \{v \in V \setminus M : \text{Anc}(v) \subseteq M\}$ , тобто множина незасвоєних тем, повний набір пререквізитів яких задовольняється поточним рівнем знань.

Досяжна множина розділяє множину незасвоєних тем на доступні та недоступні підмножини, де недоступність зумовлена наявністю принаймні одного незадоволеного пререквізиту.  $F(M)$  утворює множину кандидатів, на основі якої здійснюється побудова персоналізованої траєкторії та вибір завдань для оцінювання.

### Механізм вибору тем оцінювання

Компонент оцінювання системи базується безпосередньо на графовому представленні дорожньої карти навчання, описаному вище. Замість того, щоб розглядати оцінювання як незалежне завдання вибору контенту (наприклад, рівномірну вибірку тем), воно формулюється як задача структурного вибору на основі DAG. Вузли відповідають навчальним темам, значущість яких визначається їхнім положенням у структурі пререквізитів, а орієнтовані ребра кодують обмеження залежностей. Метою оцінювання є не вичерпна перевірка знань, а вибір щонайбільше  $k$  тем, перевірка яких забезпечує максимальний логічний вплив на граф залежностей.

Механізм визначає два сценарії, що потребують різних алгоритмічних стратегій: перший, коли інформація про попереднє засвоєння відсутня (початкове тестування для визначення рівня), та другий, коли відома непорожня множина засвоєння  $M$  (адаптивне оцінювання сесії). Вибір стратегії залежить від того, чи  $M = \emptyset$  or  $M \neq \emptyset$  на момент початку оцінювання.

Обидві стратегії мають спільну фазу попередньої обробки, під час якої обчислюються структурні метадані для кожної вершини. За допомогою пошуку в глибину для кожної вершини  $v \in V$  визначаються повні замикання предків та нащадків. Кожній вершині присвоюється топологічна глибина як довжина найбільшого ланцюжка пререквізитів, що веде до неї. На основі результатів збору метаданих кожній вершині призначається показник структурного впливу з урахуванням властивостей графа.

Коли доступна непорожня множина засвоєння  $M$ , обчислюється досяжна множина  $F(M)$ . Метою оцінювання в цьому випадку є покриття специфічної незасвоєної області графа, прилеглої до поточного рівня знань студента.

Вибір тем оцінювання формулюється як зведення до задачі про максимальне  $k$ -покриття – класичної NP-складної задачі комбінаторної оптимізації, метою якої є вибір  $k$  множин із колекції таким чином, щоб потужність їхнього об'єднання була максимальною [12]. Кожна вершина  $v \in F(M)$  розглядається як така, що «покриває» певну множину вершин. Цільова

функція оптимізації:

$$\max_{S \subseteq F(M), |S| < k} \bigcup_{v \in S}^m (Anc(v) \cup \{v\} \cup Desc(v)) \quad (1)$$

Вибір вершини реалізований із застосуванням жадібного алгоритму (greedy algorithms). На кожній ітерації набір доступних кандидатів обмежений вершинами у  $F(M)$ . Для кожного кандидата  $v$  розраховується граничний виграш від вибору цього вузла на основі попередньо обчисленого показника структурного впливу та відсотка потенційно покритих вузлів відносно поточного. Вершина з найвищим показником додається до  $S$ , а її об'єднання замикань – до множини покритих вершин. Множина  $M$  оновлюється шляхом додавання обраної вершини на основі «оптимістичного припущення про засвоєння»: обрана вершина розглядається як засвоєна для цілей розширення множини досяжності. Це дозволяє її прямим нащадкам стати доступними для вибору на наступних ітераціях без очікування підтверджених результатів тестування. Це припущення є необхідним архітектурним рішенням, яке дозволяє алгоритму просуватися по графу в межах однієї сесії, і воно розглядається як структурна евристика, а не семантична гарантія знань. Процедура повторюється, доки  $|S| = k$  або на межі досяжності не залишаться кандидатів із позитивним граничним приростом.

Коли певний підграф виявляється покритим раніше обраною вершиною, наступні вибори природним чином зміщуються до непокритих областей дорожньої карти, що покращує глобальне структурне покриття незасвоєного підграфа в межах виділеного ліміту.

Для адаптивної стратегії фаза вибору потребує часу  $O(k \cdot |V|)$ , оскільки кожна з  $k$  ітерацій оцінює не більше  $|V|$  вершин-кандидатів з обчисленням граничного коефіцієнта приросту, обмеженим розміром замикання.

У випадку, коли початкова інформація про знання відсутня, досяжна множина  $F(\emptyset)$  містить лише кореневі вершини (джерела). Суто жадібна стратегія покриття, що діє з цієї межі, не може досягти вузлів глибокої конвергенції за малого бюджету, оскільки множина розширюється лише на один шар за раз за наявності оптимістичного припущення про засвоєння. Для дорожньої карти з глибиною  $d$  та бюджетом  $k \ll d$  застосований підхід систематично концентрує вибір у найбільш поверхневих шарах, залишаючи глибокі шари повністю неоціненими.

Діагностична мета вхідного тестування вимагає, щоб вибір був розподілений за всіма рівнями топологічної глибини дорожньої карти. Тест, який перевіряє лише поверхневі шари, не може визначити наявність у студента поглиблених знань; тест, що перевіряє лише глибокі шари, не може діагностувати фундаментальні прогалини. Отже, алгоритм має гарантувати широту покриття по всій глибині графа, чого жадібний критерій не забезпечує структурно.

Це зумовлює використання стратифікованої стратегії вибору. Множина вершин  $V$  розбивається на топологічні  $L_0, L_1, \dots, L_m$  де  $L_i = \{v \in V : \text{Depth}(v) = i\}$ . Бюджет оцінювання  $k$  розподіляється між шарами пропорційно до середнього структурного впливу кожного шару.

Обидві стратегії не є альтернативними, а доповнюють одна одну, працюючи на різних фазах життєвого циклу навчання. Стратифікована стратегія слугує для студентів без історії знань, забезпечуючи діагностичну широту на всіх рівнях графа залежностей. З іншого боку, жадібна адаптивна стратегія застосовується у випадку відомої множини засвоєння, максимізуючи структурне покриття конкретної незасвоєної області, прилеглої до поточної межі знань.

## Висновки

Ця робота розглядає рішення проблеми в онлайн-освіті, а саме – брак формально обґрунтованих механізмів персоналізованого планування навчальних програм. Хоча наявні платформи пропонують адаптивні функції, вони часто спираються на евристики, нав'язуючи траєкторії, які не повною мірою враховують індивідуальні пререквізити студента. Для вирішення цієї проблеми у статті запропоновано формальну модель, у якій графові структури та алгоритми інтегровані безпосередньо в архітектуру навчальної системи, а персоналізація випливає з математичних властивостей самої моделі предметної області.

Усі поставлені завдання було вирішено в межах єдиної концептуальної структури. Дорожня карта навчання була формально визначена як спрямований ациклічний граф для забезпечення цілісності пререквізитів та логічної послідовності навчання. На системному рівні цілісність під час виконання та продуктивність досягаються завдяки: алгоритму виявлення циклів на основі пошуку в глибину для структурної валідації та алгоритму динамічного топологічного сортування для відстеження прогресу студента, що сприяє інкрементному оновленню позиції студента в DAG і усуває потребу в повному перерахунку. На етапі перебудови траєкторії було впроваджено механізм вибору завдань оцінювання з двома стратегіями: стратифікованим розміщенням, яке розділяє граф на  $k$  шарів на основі найдовшої довжини шляху від вихідних вузлів для встановлення широкого початкового оцінювання, та жадібним алгоритмом адаптивного вибору, який на кожному кроці обирає завдання, що максимізує очікуваний приріст інформації щодо засвоєння студентом невирішених пререквізитів. Ці стратегії гарантують діагностично значуще охоплення матеріалу протягом усього життєвого циклу навчання. Усі компоненти були інтегровані в цілісну архітектуру системи, що демонструє, як персоналізовані траєкторії навчання можуть бути виведені безпосередньо з внутрішніх властивостей структурованої моделі предметної області. Впровадження такого детермінованого підходу має ключову перевагу перед суто імовірнісними AI-системами, оскільки повністю виключає можливість порушення логіки навчальних пререквізитів під час генерації шляху. Запропонована топологічна модель утворює надійний алгоритмічний каркас, що доповнює сучасні засоби штучного інтелекту: вона бере на себе безпомилкову макронавігацію дорожньою картою, залишаючи великим мовним моделям та рекомендаційним системам задачі мікроадаптації контенту. Ефективність моделі залежить від точності побудови графа предметної області та може зіткнутися з проблемами масштабованості у великих мережах знань.

Подальші дослідження будуть спрямовані на емпіричну валідацію запропонованої моделі в реальних навчальних середовищах та розширення її адаптивних можливостей через інтеграцію з імовірнісними моделями навчання, включаючи басівське відстеження знань та динамічні басівські мережі.

## СПИСОК ЛІТЕРАТУРИ

1. Chen C.-M. Intelligent web-based learning system with personalized learning path guidance. *Computers & Education*. 2008. Vol. 51, №2. P. 787–814. <https://doi.org/10.1016/j.compedu.2007.08.004> (date of access: 21.01.2026).
2. Automated Formative Feedback for Algorithm and Data Structure Self-Assessment / L. Araujo et al. *Electronics*. 2025. Vol. 14, №5. P. 1034. <https://doi.org/10.3390/electronics14051034> (date of access: 18.01.2026).
3. Zheng Q. H., Huang X. J., Zhang C. A Learning Navigation Path Algorithm based on knowledge map in E-Learning. *2013 Second International Conference on e-Learning and e-Technologies in Education (ICEEE), Lodz, Poland, 23–25 September 2013*. 2013. <https://doi.org/10.1109/icelete.2013.6644370> (date of access: 07.02.2026).
4. Kyrychenko I., Malikin D. Research of Methods for Practical Educational Tasks Generation Based on Various Difficulty Levels. *6th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2022), May 12–13, 2022, Gliwice, Poland*. CEUR Workshop Proceedings 3171, Volume I: Main, 2022. P. 1030–1042. <https://ceur-ws.org/Vol-3171/paper74.pdf>.
5. Lili Qin. Adaptive System of English-Speaking Learning Based on Artificial Intelligence. *Journal of Electrical Systems*. 2024. Vol. 20, №6s. P. 267–275. <https://doi.org/10.52783/jes.2637> (date of access: 18.03.2026).
6. Fiqri M., Nurjanah D. Graph-based domain model for adaptive learning path recommendation. *2017 IEEE Global Engineering Education Conference (EDUCON), Athens, Greece, 25–28 April 2017*. 2017. <https://doi.org/10.1109/educon.2017.7942875> (date of access: 08.02.2026).
7. Slota G., Rajamanickam S., Madduri K. BFS and Coloring-Based Parallel Algorithms for Strongly Connected Components and Related Problems. *Proceedings of the International Parallel and Distributed Processing Symposium, IPDPS*. 2014. P. 550–559. <https://doi.org/10.1109/IPDPS.2014.64>.
8. Diestel R. *Graph Theory*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2017. <https://doi.org/10.1007/978-3-662-53622-3> (date of access: 23.01.2026).
9. Iozzi F., Trevisan L. Topological Sort of a Directed Acyclic Graph. *Computer Science II, Handout TS, Bocconi University*. Milan, Italy, Feb. 16, 2023. <https://lucatrevisan.github.io/30540/notes-dag.pdf> (date of access: 17.02.2026).
10. De Melo D. Topological Sorting, CS 560 Course Notes. *Computer Science Department, Illinois Institute of Technology, Chicago, IL, USA*. [http://www.cs.iit.edu/~cs560/fall\\_2012/Research\\_Paper\\_Topological\\_sorting/De%20Melo%20Bianca%20-%20Topological%20Sorting.pdf](http://www.cs.iit.edu/~cs560/fall_2012/Research_Paper_Topological_sorting/De%20Melo%20Bianca%20-%20Topological%20Sorting.pdf) (date of access: 21.01.2026).

11. Sydorenko K., Shirokopetleva M. Detecting and resolving cycles in curriculum design: a graph-theoretic approach with NoSQL storage models. *Сучасні інформаційні технології та системи штучного інтелекту: матеріали 1-ї Міжнародної науково-практичної конференції*. Ч. 2, Харків-Яремче, 19-22 травня 2025 року. Х.: ХНУРЕ, 2025. С. 14–15

12. Fixed Set Search Applied to the Max-Cut Problem / I. Šević et al. *IEEE 8th Energy Conference, ENERGYCON*. 2024. P. 1–6. <https://doi.org/10.1109/ENERGYCON58629.2024.10488777>.

Стаття надійшла до редакції 13.03.2026.

Стаття пройшла рецензування 26.03.2026.

**Сидоренко Катерина Сергіївна** – здобувач вищої освіти.

**Широкопетлева Марія Сергіївна** – старший викладач кафедри програмної інженерії,  
e-mail: [marija.shirokopetleva@nure.ua](mailto:marija.shirokopetleva@nure.ua).

Харківський національний університет радіоелектроніки.