Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

Кваліфікаційна робота
освітній ступінь – бакалавр

на тему: **«ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ ДО ДВОСТОРОННЬОЇ ЗАДАЧІ СЕКРЕТАРЯ»**

Виконала: студентка 4-го року навчання
Освітньої програми «Прикладна математика», 113
Радучич Олеся Сергіївна

Керівник: Щестюк Н. Ю.
кандидат фіз.-мат. наук, доцент

Рецензент _____
(прізвище та ініціали)

Кваліфікаційна робота захищена
з оцінкою _____

Секретар ЕК _____
(підпис)

«_____» _____ 20___р.

Київ – 2026

Міністерство освіти і науки України
Національний університет «Києво-Могилянська академія»
Факультет інформатики
Кафедра математики

ЗАТВЕРДЖУЮ
Зав. кафедри математики,
доцент, кандидат фіз.-мат. наук
_____*Чорней Р.К.*
(підпис)
“____”_____ 2026

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
для кваліфікаційної роботи
студентці 4-го курсу, факультету інформатики
Радучич Олесі Сергіївні

Тема: «Застосування нейронних мереж до двосторонньої задачі секретаря»
Зміст кваліфікаційної роботи:

Вступ
1. Постановка, припущення та оптимальна стратегія класичної задачі секретаря
2. Методологічні основи розв’язання задачі секретаря засобами машинного навчання
3. Реалізація двосторонньої задачі секретаря нейронними мережами в порівнянні з пороговою стратегією
Висновки
Список літератури

Дата видачі “____”_____ 2026 Керівник _____
(підпис)
Завдання отримав _____
(підпис)

Графік підготовки кваліфікаційної роботи до захисту

Графік узгоджено «_____» _____ 2026р.

№ з/п	Перелік робіт	Термін виконання етапу	Підпис наукового керівника	Дата ознайомлення наукового керівника	Примітка
1.	Отримання теми кваліфікаційної роботи.	17.09.2025			
2.	Розробка плану та структури роботи.	15.10.2025			
3.	Робота з науковою літературою, опис основних означень.	24.10.2025			
4.	Робота над текстовим оформленням теоретичної частини.	19.01.2026			
5.	Моделювання нейронної мережі.	20.02.2026			
6.	Аналіз результатів і формулювання висновків.	01.03.2026			
7.	Попередній аналіз кваліфікаційної роботи. Виправлення помилок.	25.03.2026			
8.	Попередній захист кваліфікаційної роботи.	03.04.2026			
9.	Захист кваліфікаційної роботи.	03.06.2026			

Науковий керівник _____
(ПІБ)

Виконавець кваліфікаційної роботи _____
(ПІБ)

ЗМІСТ

АНОТАЦІЯ

Задача секретаря є класичною моделлю теорії оптимальних зупинок, яка використовується для аналізу процесів прийняття рішень в умовах невизначеності. У реальних сценаріях, таких як ринок праці чи бізнес-партнерства, процес вибору є двостороннім і потребує взаємної згоди сторін, що суттєво ускладнює застосування класичних аналітичних методів. У роботі розроблено нейронну мережу на основі архітектури LSTM, навчену методом Actor-Critic, яка формує оптимальну стратегію зупинки у двосторонній задачі секретаря. Середовище моделює послідовний вибір, у якому успішний результат досягається лише за умови одночасної згоди обох сторін. Досліджено процес навчання моделі та проведено порівняльний аналіз із класичною пороговою стратегією методом Монте-Карло. LSTM-модель стабільно перевершує класичну порогову стратегію за всіма розглянутими показниками — середньою якістю обраних пар, абсолютним рангом кандидатів та швидкістю досягнення взаємної згоди.

ABSTRACT

The secretary problem is a classical model of optimal stopping theory used to analyse decision-making processes under uncertainty. In real-world scenarios such as labour markets or business partnerships, the selection process is two-sided and requires mutual agreement, which significantly limits the applicability of classical analytical methods. This thesis presents an LSTM-based neural network trained with the Actor-Critic method that learns an optimal stopping strategy for the two-sided secretary problem. The environment models a sequential selection process in which a successful outcome requires simultaneous agreement from both parties. The training dynamics are examined and a comparative analysis with the classical threshold strategy is conducted via Monte Carlo simulation. The LSTM model consistently outperforms the classical threshold strategy across all considered metrics — average pair quality, absolute candidate rank, and speed of reaching mutual agreement.

ВСТУП

Задача секретаря є класичною моделлю теорії оптимальних зупинок, яка використовується для аналізу процесів прийняття рішень в умовах невизначеності. У класичному формулюванні вона полягає у виборі найкращого кандидата із послідовності варіантів, що з'являються випадково, за умови миттєвого прийняття рішення без можливості повернення до попередніх варіантів. Завдяки своїй універсальності модель застосовується в економіці, теорії ігор та управлінні персоналом. Більшість традиційних досліджень розглядають односторонній процес вибору, проте у реальних сценаріях, таких як ринок праці чи бізнес-партнерства, процес є двостороннім і потребує взаємної згоди сторін. Стохастичні узагальнення задачі секретаря та їх теоретичні властивості досліджено, зокрема, у роботі [12], де розглянуто імовірнісні аспекти різних модифікацій класичної постановки. Зі збільшенням складності таких сценаріїв класичні аналітичні методи стикаються з проблемою високої обчислювальної складності. Навчання з підкріпленням (Reinforcement Learning) у поєднанні з рекурентними нейронними мережами (LSTM) відкриває нові можливості для пошуку оптимальних стратегій у стохастичних середовищах, де агент навчається через взаємодію та максимізацію очікуваної винагороди.

Метою даної роботи є розробка та аналіз моделей прийняття рішень у односторонній та двосторонній задачах секретаря на основі алгоритмів навчання з підкріпленням для знаходження ефективних стратегій вибору. Для досягнення поставленої мети було визначено низку наукових завдань: проаналізувати теоретичні підходи до задач оптимальної зупинки; обґрунтувати застосування навчання з підкріпленням для пошуку оптимальних стратегій у динамічних середовищах; побудувати модель для двосторонньої задачі та провести порівняльний аналіз отриманих результатів із класичними аналітичними методами.

Об'єктом дослідження є процес прийняття рішень у задачах оптимальної зупинки в умовах стохастичного надходження інформації. Предметом дослідження виступають стратегії та алгоритми навчання з підкріпленням на основі архітектури LSTM. Методологічну основу роботи складають методи теорії ймовірностей, теорії ігор, машинного навчання та імітаційного моделювання. Наукова новизна результатів полягає у застосуванні рекурентних нейронних мереж до задачі двостороннього некооперативного вибору, що дозволяє зна-

ходити ефективні рішення там, де аналітичне розв’язання є надто складним. Практичне значення роботи полягає у можливості використання розроблених моделей для автоматизації відбору в сучасних інформаційних системах.

Робота складається зі вступу, трьох розділів та висновків. У першому розділі проведено аналіз класичної постановки задачі секретаря, визначено умови знаходження оптимальної стратегії та розглянуто обмеження аналітичних методів розв’язання задач оптимальної зупинки, що ускладнюють їх ефективне застосування в практичних сценаріях. Другий розділ присвячений методологічним та теоретичним основам розв’язання задачі: обґрунтовано доцільність використання навчання з підкріпленням, наведено порівняльний огляд нейромережевих архітектур, за результатами якого для моделювання послідовних рішень обрано структуру LSTM, викладено математичний апарат Reinforcement Learning, описано метод Actor-Critic, а також детально розглянуто архітектуру спроєктованої рекурентної нейронної мережі та процес її навчання. У третьому розділі наведено порівняльний аналіз порогових стратегій і спроєктованої нейронної моделі в розв’язанні двосторонньої задачі оптимальної зупинки на основі статистичного моделювання Монте-Карло та виконано оцінювання ефективності запропонованого підходу.

Робота пройшла апробацію на XIV Всеукраїнській науковій конференції молодих математиків, що відбулася 7–8 травня 2026 року в Українському державному університеті імені Михайла Драгоманова [11]. Матеріали конференції опубліковано у відкритому доступі та доступні за посиланням.

РОЗДІЛ 1. ПОСТАНОВКА, ПРИПУЩЕННЯ ТА ОПТИМАЛЬНА СТРАТЕГІЯ КЛАСИЧНОЇ ЗАДАЧІ СЕКРЕТАРЯ

1.1. Постановка задачі в класичному варіанті

Класична задача секретаря описує ситуацію, у якій необхідно вибрати найкращий об'єкт із послідовності альтернатив, що з'являються одна за одною у випадковому порядку. Після розгляду кожної альтернативи потрібно одразу прийняти рішення: прийняти її або відхилити назавжди. Повернутися до вже відхилених варіантів неможливо. Метою є максимізація ймовірності вибору найкращого варіанта.

Класична інтерпретація цієї задачі пов'язана з процесом відбору кандидата на роботу. Припустимо, що роботодавець має заповнити одну вакансію та проводить співбесіди з певною кількістю претендентів. Кандидати приходять на співбесіду послідовно у випадковому порядку. Після кожної співбесіди роботодавець повинен вирішити, чи наймати цього кандидата, чи відхилити його та продовжити пошук, оскільки повернення до відхиленого кандидата неможливе. Таким чином, основна проблема полягає у визначенні оптимального моменту зупинки, який дозволить із найбільшою ймовірністю обрати найкращого кандидата серед усіх.

Важливою особливістю задачі є обмеженість доступної інформації. Роботодавець не знає абсолютної якості кандидатів і не може порівняти їх із тими, кого ще не було розглянуто. Натомість він може оцінювати кожного нового кандидата лише відносно тих, які вже пройшли співбесіду. Іншими словами, відомий лише його відносний ранг серед уже розглянутих кандидатів, але невідомо, яке місце він займає серед усіх претендентів.

Формально постановка класичної задачі секретаря базується на таких припущеннях:

1. Існує одна вакансія, яку необхідно заповнити
2. Є n кандидатів на цю вакансію, причому значення n відоме заздалегідь
3. Якщо всіх кандидатів оцінити одночасно, їх можна однозначно впорядкувати від найкращого до найгіршого

4. Кандидати проходять співбесіду послідовно у випадковому порядку, причому кожна можлива послідовність появи кандидатів є однаково ймовірною

5. Після співбесіди роботодавець повинен одразу прийняти рішення — найняти кандидата або відхилити його, і це рішення є остаточним

6. Рішення про прийняття кандидата може базуватися лише на відносному ранзі цього кандидата серед тих кандидатів, які вже були розглянуті

7. Метою є максимізація ймовірності вибору абсолютно найкращого кандидата серед усіх претендентів

1.2. Оптимальна стратегія зупинки

Нехай n — загальна кількість кандидатів, j — номер кандидата у послідовності, а r — кількість перших кандидатів, які відхиляються на початковому етапі спостереження. Класична оптимальна стратегія складається з двох етапів. На першому етапі, який називають етапом спостереження, перші r кандидатів автоматично відхиляються, але їхні якості використовуються для формування орієнтиру. На другому етапі, починаючи з кандидата $r+1$, приймається перший кандидат, який є кращим за всіх попередніх. Якщо до кандидата $n-1$ такий варіант не з'являється, тоді обирається останній кандидат.

Ймовірність того, що така стратегія дозволить вибрати найкращого кандидата, позначимо $Q(r)$. Вона визначається формулою

$$Q(r) = \frac{1}{n}\left(1 + \frac{r}{r+1} + \frac{r}{r+2} + \cdots + \frac{r}{n-1}\right)$$

для будь-якого $r = 1, 2, \ldots, n-1$. Цей вираз можна переписати у компактнішій формі через гармонічні числа:

$$Q(r) = \frac{r}{n}\sum_{i=r}^{n-1}\frac{1}{i},$$

або

$$Q(r) = \frac{r}{n}\left(H_{n-1} - H_{r-1}\right),$$

де

$$H_k = \sum_{i=1}^{k} \frac{1}{i}$$

— гармонічний ряд.

Оптимальне значення r визначається з умови максимізації функції $Q(r)$. Для великої кількості кандидатів використовується асимптотичне наближення гармонічних чисел

$$H_n \approx \ln n + \gamma,$$

де γ — стала Ейлера.

Після підстановки цього наближення отримується оптимальне значення

$$r^* \approx \frac{n}{e},$$

де $e \approx 2.71828$ — основа натурального логарифма. Отже, оптимальна стратегія полягає у відхиленні приблизно

$$\frac{n}{e} \approx 0.37n$$

перших кандидатів, після чого потрібно вибрати першого кандидата, який перевищує за якістю всіх попередніх. Цей результат отримав назву правило 37%. При використанні цієї стратегії ймовірність вибрати найкращого кандидата наближається до

$$\frac{1}{e} \approx 0.37.$$

Це значно більше, ніж випадковий вибір, де ймовірність успіху становить лише $1/n$.

1.3. Постановка двостороннього варіанту задачі секретаря

Подальшим розвитком класичної моделі є двостороння задача секретаря (*Two-sided Secretary Problem*). У цій постановці рішення приймається двома сторонами одночасно, наприклад роботодавцем і кандидатом або двома осо-

бами, що шукають партнера. Кожна сторона послідовно зустрічає потенційних партнерів і повинна прийняти рішення про прийняття або відхилення, не маючи інформації про майбутні варіанти. На відміну від класичної моделі, успішний результат у цьому випадку можливий лише за умови взаємної згоди обох сторін.

Стратегія у двосторонній задачі також ґрунтується на принципі початкового етапу спостереження. Кожна сторона спочатку відхиляє певну частку варіантів, формуючи еталон якості, після чого погоджується на перший варіант, який перевищує цей еталон і водночас приймає іншу сторону. Аналогічно до класичної моделі, у першій фазі часто використовується правило приблизно 37% спостереження, що дозволяє отримати достатню інформацію про розподіл якостей потенційних партнерів. Після цього сторони переходять до етапу вибору, на якому приймається перший взаємно прийнятний варіант.

Таким чином, класична задача секретаря формує математичну основу для аналізу оптимальних стратегій вибору у послідовних процесах прийняття рішень. Її двостороннє узагальнення, у свою чергу, дозволяє моделювати ситуації взаємного вибору, характерні для ринку праці, моделей пошуку партнера та інших соціально-економічних процесів.

1.4. Обмеження класичної задачі секретаря у практичних застосуваннях

Незважаючи на елегантність математичного розв’язку класичної задачі секретаря та наявність чіткої оптимальної стратегії (правила 37%), її безпосереднє застосування у практичних задачах часто є неможливим. Причина полягає в тому, що класична модель базується на низці спрощених припущень, які рідко виконуються у реальних економічних, соціальних або інженерних системах.

Наявність лише відносних рангів. Класична постановка задачі передбачає, що особа, яка приймає рішення, може оцінювати кандидата лише відносно тих, які вже були розглянуті. Тобто використовується лише інформація про відносні ранги, тоді як абсолютні значення якості невідомі. У реальних задачах ситуація суттєво відрізняється. Наприклад, під час відбору працівників або вибору технічного рішення часто використовуються кількісні показники: результати тестів, досвід роботи, фінансові показники або технічні характеристики. У таких ви-

падках особа, що приймає рішення, оперує кардинальними оцінками, а не лише порядковими порівняннями.

Оптимізація ймовірності замість корисності. Класична стратегія спрямована виключно на максимізацію ймовірності вибору найкращого кандидата. Проте в практичних задачах рішення часто приймається з урахуванням очікуваної корисності або економічної вигоди. Наприклад, якщо різниця між першим і другим кандидатом є незначною, але продовження пошуку пов'язане з додатковими витратами часу або ресурсів, раціональніше може бути обрати не абсолютно найкращого кандидата, а достатньо хорошого. Таким чином, правило 37% не враховує витрати на пошук і тому може бути економічно неоптимальним.

Наявність апріорної інформації. У класичній задачі передбачається, що до початку процесу вибору особа, яка приймає рішення, не має жодної інформації про розподіл якості кандидатів. У реальних умовах це припущення зазвичай не виконується. Досвідчені менеджери, інвестори або інженери мають певні попередні знання, сформовані на основі попереднього досвіду, статистики або експертних оцінок. Ці знання дозволяють формувати очікування щодо якості кандидатів і адаптувати стратегію пошуку в процесі прийняття рішень. Класична стратегія не дозволяє враховувати такі апріорні знання.

Невідоме число альтернатив. Ще одним обмеженням є припущення про те, що загальна кількість кандидатів n відома заздалегідь. У реальних системах ця інформація часто є невизначеною або змінною. Наприклад, під час пошуку інвестиційних можливостей або технологічних рішень кількість потенційних альтернатив може змінюватися з часом.

РОЗДІЛ 2. МЕТОДОЛОГІЧНІ ОСНОВИ РОЗВ'ЯЗАННЯ ЗАДАЧІ СЕКРЕТАРЯ ЗАСОБАМИ МАШИННОГО НАВЧАННЯ

2.1. Застосування навчання з підкріпленням у задачі секретаря

Наведені обмеження класичної задачі секретаря зумовлюють необхідність використання більш адаптивних методів прийняття рішень. Одним із перспективних підходів до розв'язання таких задач є застосування методів машинного навчання, зокрема навчання з підкріпленням (Reinforcement Learning, RL). Ця парадигма розроблена спеціально для розв'язання задач послідовного прийняття рішень у невизначених та динамічних середовищах. На відміну від аналітичних методів, які вимагають чітко визначеної математичної моделі процесу, RL дозволяє агенту самостійно навчатися оптимальній поведінці через взаємодію із середовищем.

У рамках цього підходу система розглядається як взаємодія між агентом та середовищем. Агент послідовно спостерігає стан середовища S_t, обирає дію A_t та отримує винагороду R_t, яка відображає якість прийнятого рішення. Метою навчання є формування такої політики прийняття рішень, яка максимізує очікувану сумарну винагороду.

У задачі секретаря агентом виступає система прийняття рішень, а середовищем — послідовність кандидатів. На кожному кроці агент отримує інформацію про поточного кандидата та повинен обрати одну з двох можливих дій:

- прийняти кандидата та завершити процес відбору;
- відхилити кандидата та перейти до наступного.

На відміну від класичної моделі, де метою є лише максимізація ймовірності вибору найкращого кандидата, підхід RL дозволяє використовувати більш гнучкі функції винагороди. Наприклад, можна оптимізувати середню якість обраного кандидата, враховувати витрати на пошук або балансувати між швидкістю прийняття рішення та його якістю.

Ще однією важливою перевагою навчання з підкріпленням є його здатність адаптуватися до невідомих розподілів даних. Агент не потребує попереднього знання кількості кандидатів або їх статистичних характеристик. Натомість він поступово вивчає закономірності середовища на основі великої кіль-

кості епізодів взаємодії. Це дозволяє моделі автоматично формувати ефективні стратегії зупинки навіть у складних і динамічних умовах.

Крім того, RL природно враховує довгострокові наслідки прийнятих рішень. Агент повинен балансувати між негайною вигодою від прийняття поточного кандидата та потенційною можливістю зустріти кращого кандидата в майбутньому. Саме така дилема лежить в основі задач оптимальної зупинки, що робить навчання з підкріпленням одним із найбільш релевантних інструментів для їх розв'язання.

2.2. Вибір архітектури LSTM для моделювання процесу прийняття рішень

Ефективність алгоритмів навчання з підкріпленням значною мірою залежить від вибору архітектури нейронної мережі, яка використовується для апроксимації політики або функції цінності. У задачі секретаря процес прийняття рішень має послідовний характер: кандидати надходять один за одним, і рішення щодо поточного кандидата залежить від інформації про попередні спостереження. Це вимагає моделі, здатної враховувати історичний контекст та довгострокові залежності.

Одним із найпростіших варіантів є використання повнозв'язних нейронних мереж (Multilayer Perceptron, MLP). Такі моделі отримують на вхід вектор ознак поточного стану та видають рішення без внутрішнього механізму пам'яті. У контексті задачі секретаря це означає, що для врахування історії необхідно явно формувати додаткові ознаки (наприклад, максимальне значення серед попередніх кандидатів або кількість вже переглянутих елементів). Однак подібний підхід не дозволяє моделі автоматично виявляти складні часові залежності та обмежує її здатність до узагальнення.

Більш природним підходом для роботи з послідовними даними є рекурентні нейронні мережі (Recurrent Neural Networks, RNN), які передають прихований стан між кроками послідовності. Це дозволяє моделі зберігати інформацію про попередні спостереження та використовувати її при прийнятті рішень. Водночас класичні RNN мають суттєві обмеження, зокрема проблему зникаючого градієнта, що ускладнює навчання на довгих послідовностях і призводить до втрати інформації про віддалені у часі події.

Модифікацією RNN є архітектура Long Short-Term Memory (LSTM), яка була спеціально розроблена для подолання зазначених проблем. Вона містить внутрішній стан пам'яті та набір керуючих механізмів (гейтів), що дозволяють вибірково зберігати або забувати інформацію. Завдяки цьому LSTM ефективно моделює довгострокові залежності та забезпечує стабільніше навчання порівняно зі звичайними RNN.

Альтернативою є архітектура Gated Recurrent Unit (GRU), яка є спрощеною версією LSTM і має меншу кількість параметрів. GRU демонструє хорошу продуктивність на багатьох задачах, однак її спрощена структура обмежує гнучкість у керуванні внутрішнім станом пам'яті. У задачах, де важливо точно моделювати складні та довготривалі залежності між елементами послідовності, це може призводити до зниження якості рішень.

Сучасні архітектури, такі як Transformer, використовують механізм self-attention для одночасного врахування взаємозв'язків між усіма елементами послідовності. Це дозволяє ефективно моделювати глобальні залежності без рекурентної структури. Однак Transformer має квадратичну обчислювальну складність $O(n^2)$ від довжини послідовності, що робить його менш ефективним у задачах онлайн-прийняття рішень, де дані надходять послідовно і необхідна низька затримка обчислень. Крім того, такі моделі зазвичай потребують значно більших обсягів даних та обчислювальних ресурсів для навчання.

У контексті навчання з підкріпленням вибір архітектури також впливає на стабільність і швидкість навчання агента. Наприклад, у підходах Deep Q-Learning (DQN) або Actor-Critic нейронна мережа використовується для апроксимації функції цінності або політики. Для задачі секретаря важливо, щоб модель могла не лише оцінювати поточний стан, але й враховувати історію взаємодій із середовищем, що додатково підсилює вимоги до наявності механізму пам'яті.

У порівнянні з наведеними архітектурами, LSTM забезпечує оптимальний баланс між здатністю моделювати послідовності, стабільністю навчання та обчислювальною ефективністю. Вона природно інтегрується у рекурентні RL-агенти та дозволяє формувати внутрішнє представлення історії кандидатів, яке включає інформацію про попередні оцінки, динаміку їх змін та відносну якість поточного спостереження.

Таким чином, хоча MLP, RNN, GRU та Transformer можуть бути застосо-

вані для розв'язання задачі секретаря, кожна з цих архітектур має певні обмеження. LSTM у свою чергу поєднує здатність до моделювання довгострокових залежностей, стійкість до проблеми зникаючого градієнта та прийнятну обчислювальну складність, що робить її найбільш доцільним вибором для даної задачі.

2.3. Основні визначення навчання з підкріпленням

Навчання з підкріпленням (Reinforcement Learning, RL) є одним із напрямів машинного навчання, у якому агент навчається приймати оптимальні рішення шляхом взаємодії з середовищем. Основна ідея RL полягає в тому, що агент виконує певні дії, отримує за них винагороду або штраф та поступово коригує свою поведінку таким чином, щоб максимізувати сумарну винагороду у довгостроковій перспективі. На відміну від класичного навчання з учителем, у RL агент не отримує правильних відповідей наперед, а навчається на основі власного досвіду взаємодії із середовищем.

Процес навчання з підкріпленням описується через взаємодію двох компонентів: агента та середовища. У момент часу t агент перебуває у стані s_t, виконує дію a_t, після чого середовище переходить у новий стан s_{t+1} та повертає винагороду r_{t+1}. Головною метою агента є вибір такої послідовності дій, яка дозволить максимізувати накопичену винагороду.

Формально задача навчання з підкріпленням описується за допомогою марковського процесу прийняття рішень (Markov Decision Process, MDP). MDP визначається четвіркою:

$$(S, A, P, R),$$

де:

- S — множина можливих станів середовища;
- A — множина можливих дій агента;
- $P(s'|s, a)$ — ймовірність переходу із стану s у стан s' після виконання дії a;
- $R(s, a)$ — функція винагороди, яка визначає миттєву винагороду агента.

Марковська властивість означає, що наступний стан залежить лише від поточного стану та виконаної дії, а не від усієї попередньої історії взаємодій. Це дозволяє спростити математичний опис задачі та побудову алгоритмів навчання.

Середовище у задачах навчання з підкріпленням може бути детермінованим або стохастичним, повністю чи частково спостережуваним, а також стаціонарним або динамічним. У детермінованому середовищі однакові дії в однакових станах завжди приводять до однакових результатів, тоді як у стохастичному середовищі результат дії може змінюватися випадковим чином. У повністю спостережуваному середовищі агент отримує повну інформацію про поточний стан системи, тоді як у частково спостережуваному середовищі агент має доступ лише до частини інформації про стан середовища.

У деяких задачах агент може не мати повної інформації про стан середовища. У такому випадку замість повного стану s_t агент отримує спостереження o_t, яке містить лише часткову інформацію про середовище. Такі задачі описуються частково спостережуваними марковськими процесами прийняття рішень (Partially Observable Markov Decision Process, POMDP). У задачі секретаря агент отримує лише інформацію про поточного кандидата та попередні спостереження, не маючи доступу до повної інформації про всіх кандидатів, що робить задачу частково спостережуваною.

Одним із центральних понять RL є політика агента. Політика визначає стратегію вибору дій у кожному стані середовища. Формально політика задається як:

$$\pi(a|s) = P(A_t = a \mid S_t = s),$$

де $\pi(a|s)$ — ймовірність вибору дії a у стані s. Політика може бути детермінованою або стохастичною. У детермінованому випадку для кожного стану існує лише одна дія, тоді як у стохастичному випадку агент обирає дії згідно з певним розподілом ймовірностей.

Для оцінювання якості політики використовується поняття сумарної винагороди або return. Дисконтована сумарна винагорода визначається як:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots$$

або у компактному вигляді:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1},$$

де $\gamma \in [0, 1]$ — коефіцієнт дисконтування.

Коефіцієнт дисконтування визначає важливість майбутніх винагород. Якщо γ близький до нуля, агент враховує переважно миттєві винагороди. Якщо ж γ наближається до одиниці, агент орієнтується на довгострокову перспективу та враховує майбутні наслідки своїх дій.

Основною метою навчання з підкріпленням є знаходження оптимальної політики π^*, яка максимізує математичне сподівання сумарної винагороди:

$$\pi^* = \arg\max_{\pi} \mathbb{E}[G_t].$$

Для оцінки якості станів та дій використовуються функції цінності. Функція цінності стану $V^\pi(s)$ визначає очікувану сумарну винагороду, яку агент отримає, перебуваючи у стані s та дотримуючись політики π:

$$V^\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s].$$

Функція цінності дії $Q^\pi(s, a)$ оцінює очікувану винагороду від виконання дії a у стані s:

$$Q^\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a].$$

Ключовою математичною основою RL є рівняння Беллмана, яке встановлює рекурсивний зв'язок між поточним значенням функції цінності та майбутніми винагородами. Для функції цінності стану рівняння Беллмана має вигляд:

$$V^\pi(s) = \sum_{a \in A} \pi(a|s) \sum_{s', r} P(s', r|s, a) \left[r + \gamma V^\pi(s') \right].$$

Це рівняння показує, що цінність поточного стану складається з миттєвої винагороди та дисконтованої цінності наступного стану.

Для оптимальної політики використовується рівняння оптимальності Беллмана:

$$V^*(s) = \max_a \sum_{s',r} P(s', r|s, a) \left[r + \gamma V^*(s') \right].$$

Таким чином, навчання з підкріпленням є ефективним підходом до розв'язання задач послідовного прийняття рішень. Використання RL у задачі секретаря дозволяє агенту поступово формувати оптимальну стратегію вибору моменту зупинки на основі накопиченого досвіду взаємодії із середовищем.

2.4. Метод Actor-Critic

Метод Actor-Critic (AC) є одним із найбільш ефективних і поширених підходів у задачах навчання з підкріпленням. Його популярність зумовлена поєднанням переваг методів градієнта політики (Policy Gradient Methods) та методів оцінювання функції цінності (Value Function Methods). Завдяки такій комбінації алгоритм забезпечує більш стабільне навчання агента та дозволяє ефективно працювати як із дискретними, так і з неперервними просторами дій.

Архітектура Actor-Critic складається з двох основних компонентів: актора (Actor) та критика (Critic). Актор відповідає за формування політики поведінки агента, тобто визначає, яку дію необхідно виконати в певному стані середовища. Критик, у свою чергу, оцінює якість дій, виконаних актором, та визначає, наскільки отриманий результат відповідає очікуванням. Такий підхід дозволяє агенту поступово вдосконалювати власну стратегію прийняття рішень на основі накопиченого досвіду.

На відміну від класичних алгоритмів, де політика формується неявно через максимізацію функції цінності, у методі Actor-Critic політика задається безпосередньо у вигляді параметризованої функції:

$$\pi_\theta(a|s),$$

де s — поточний стан середовища, a — дія агента, а θ — параметри політики. Основною метою актора є максимізація очікуваної сумарної винагороди, яку агент отримує під час взаємодії із середовищем.

Для оптимізації політики використовується метод градієнта політики (Policy Gradient), який дозволяє безпосередньо змінювати параметри політики у напрямку збільшення очікуваної винагороди. Градієнт функції цілі визначається

виразом:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta} \left[\nabla_\theta \log \pi_\theta(a|s) \cdot A(s,a) \right],$$

де $A(s,a)$ — функція переваги (Advantage Function), яка показує, наскільки виконана дія є кращою або гіршою від очікуваного результату. Використання функції переваги дозволяє суттєво знизити дисперсію оцінки градієнта та зробити процес навчання більш стабільним.

Критик у методі Actor-Critic виконує задачу оцінювання функції цінності стану:

$$V^\pi(s) = \mathbb{E}_\pi[G_t | S_t = s],$$

де G_t — сумарна дисконтована винагорода. Критик оцінює, наскільки поточний стан є перспективним з точки зору майбутніх винагород, та передає цю інформацію акторові для оновлення політики.

У більшості сучасних реалізацій критик навчається за допомогою методів часової різниці (Temporal Difference Learning, TD-learning). Основний сигнал навчання формується через TD-помилку:

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t),$$

де r_{t+1} — отримана винагорода, а γ — коефіцієнт дисконтування. TD-помилка показує різницю між реально отриманою винагородою та прогнозом критика. Якщо значення помилки є додатним, це означає, що результат перевищив очікування, тому ймовірність вибору відповідної дії повинна збільшитися. Якщо ж помилка є від'ємною, політика коригується у протилежному напрямку.

Параметри актора оновлюються за правилом:

$$\theta_{t+1} = \theta_t + \alpha \nabla_\theta \log \pi_\theta(a_t|s_t) \delta_t,$$

де α — швидкість навчання. Таким чином, критик виступає механізмом оцінювання якості дій актора та допомагає спрямовувати процес навчання у правильному напрямку.

У сучасних реалізаціях Actor-Critic обидва компоненти найчастіше реалізуються за допомогою нейронних мереж. Актор формує ймовірності вибору дій, тоді як критик прогнозує значення функції цінності. Часто використову-

ється спільна нейронна мережа із загальними прихованими шарами та двома окремими виходами — для політики та функції цінності. Такий підхід дозволяє ефективніше використовувати вивчені ознаки стану та покращує якість навчання моделі.

Однією з важливих особливостей методу Actor-Critic є можливість навчання стохастичної політики, у якій агент формує розподіл імовірностей над можливими діями замість вибору єдиної фіксованої дії для кожного стану середовища. Такий підхід дозволяє забезпечити баланс між дослідженням нових стратегій (exploration) та використанням уже накопиченого досвіду (exploitation), що є особливо важливим у задачах послідовного прийняття рішень та в умовах невизначеності середовища.

Для покращення процесу дослідження простору дій у багатьох реалізаціях використовується ентропійна регуляризація. Вона стимулює агента підтримувати вищу невизначеність політики та сприяє ефективнішому пошуку оптимальної стратегії. Загальна функція втрат у методі Actor-Critic зазвичай містить три складові: втрати актора, втрати критика та ентропійну регуляризацію:

$$L = L_{actor} + c_v L_{critic} + c_e L_{entropy},$$

де c_v та c_e — вагові коефіцієнти, що балансують між оновленням політики, навчанням критика та стимулюванням дослідження.

Таким чином, метод Actor-Critic є ефективним підходом для задач послідовного прийняття рішень та оптимальної зупинки, у яких агент повинен формувати стратегію вибору дій на основі послідовності спостережень. У задачі секретаря даний підхід дозволяє агенту поступово навчатися визначати оптимальний момент зупинки та приймати рішення на основі накопиченої інформації про попередньо переглянутих кандидатів.

2.5. Огляд архітектури спроєктованої рекурентної нейронної мережі

Для розв'язання задачі секретаря в умовах послідовного прийняття рішень розроблено рекурентну нейронну мережу, яка реалізує механізм послідовного прийняття рішень для задачі оптимальної зупинки. Модель функціонує в покроковому режимі без доступу до майбутніх спостережень, що відповідає

природі задачі та забезпечує коректне моделювання процесу послідовного вибору.

Загальна архітектура мережі, представлена на рис. 2.1, складається з трьох логічно пов'язаних компонентів: вхідного шару, що формує представлення поточного стану середовища; рекурентного блоку LSTM, який накопичує інформацію про всю історію спостережень; та вихідного шару з логікою прийняття рішення, що перетворює внутрішнє представлення мережі на бінарну дію.

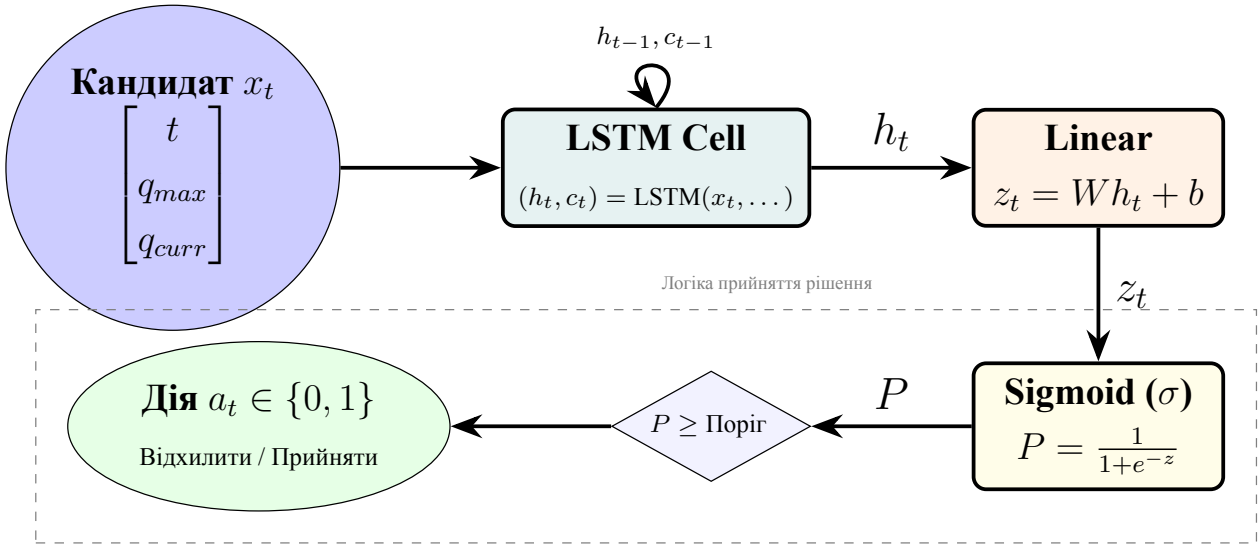

Рис. 2.1. Загальна архітектура спроєктованої рекурентної нейронної мережі

На кожному кроці часу t мережа отримує тривимірний вектор стану, що описує поточну ситуацію та накопичену історію спостережень:

$$x_t = [t,\ q_{\text{max}},\ q_{\text{curr}}],$$

де t — номер кроку, q_{curr} — якість поточного елемента, а q_{max} — максимальне значення серед уже переглянутих кандидатів. Таким чином,

$$x_t \in \mathbb{R}^3.$$

Обробка послідовності здійснюється за допомогою рекурентного блоку типу Long Short-Term Memory (LSTM), реалізованого через модуль `LSTMCell`. На кожному кроці оновлюються прихований стан h_t та стан пам'яті c_t відповідно до співвідношення:

$$(h_t, c_t) = \text{LSTMCell}(x_t, h_{t-1}, c_{t-1}),$$

де

$$h_t, c_t \in \mathbb{R}^{64}.$$

Внутрішня структура LSTM-комірки, представлена на рис. 2.2, реалізує механізм керованого потоку інформації через систему обчислювальних вентилів — гейтів, кожен з яких виконує специфічну функцію.

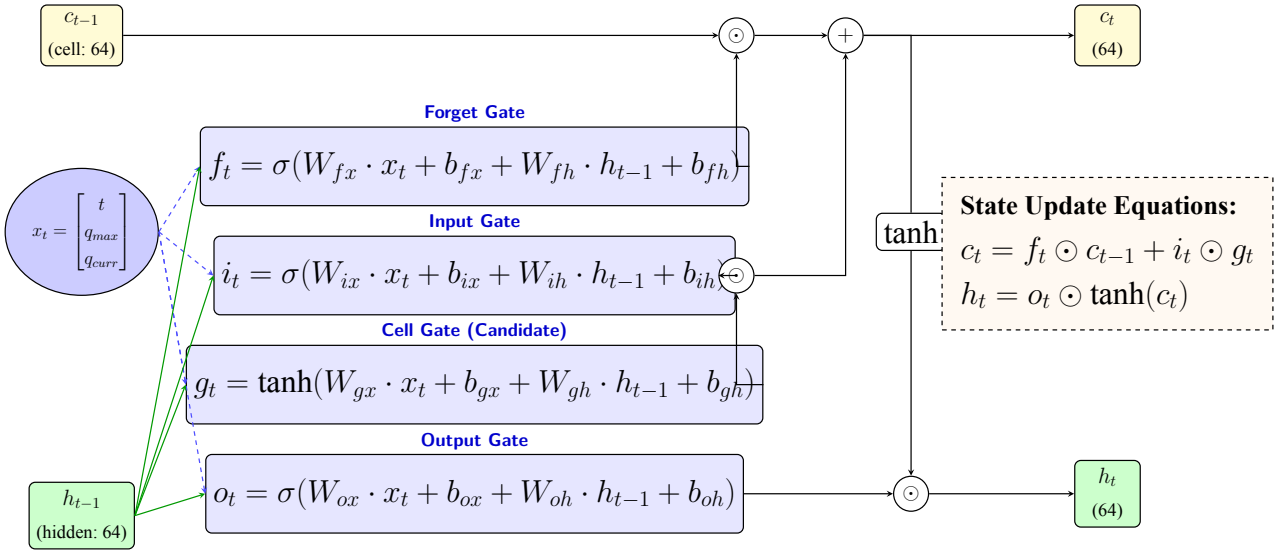

Рис. 2.2. Внутрішня структура обчислювального блоку LSTM з 64 прихованими параметрами

Формально функціонування комірки описується системою рівнянь, що відповідає чотирьом основним вентилям. Гейт забування f_t визначає, яку частину попереднього стану пам'яті c_{t-1} слід зберегти:

$$f_t = \sigma(W_{fx} \cdot x_t + b_{fx} + W_{fh} \cdot h_{t-1} + b_{fh}).$$

Вхідний гейт i_t та кандидатний гейт g_t разом визначають, яку нову інформацію слід додати до стану пам'яті:

$$i_t = \sigma(W_{ix} \cdot x_t + b_{ix} + W_{ih} \cdot h_{t-1} + b_{ih}),$$

$$g_t = \tanh(W_{gx} \cdot x_t + b_{gx} + W_{gh} \cdot h_{t-1} + b_{gh}).$$

Вихідний гейт o_t керує тим, яка частина оновленого стану пам'яті буде передана у прихований стан:

$$o_t = \sigma(W_{ox} \cdot x_t + b_{ox} + W_{oh} \cdot h_{t-1} + b_{oh}).$$

Оновлення стану пам'яті та прихованого стану виконується відповідно до співвідношень:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t,$$

$$h_t = o_t \odot \tanh(c_t),$$

де $\sigma(\cdot)$ — сигмоїдна функція активації, $\tanh(\cdot)$ — гіперболічний тангенс, а $\odot$ позначає покомпонентне множення (Адамарів добуток).

У результаті прихований вектор h_t містить стиснуте представлення всієї історії епізоду до моменту t включно, що дозволяє наступним шарам мережі формувати рішення з урахуванням повного контексту.

Отриманий прихований стан h_t подається на вихідний шар, який формує логіт рішення через лінійне перетворення:

$$stop\_logit_t = W_p h_t + b_p,$$

де $W_p \in \mathbb{R}^{1\times 64}$ — вагова матриця, а b_p — параметр зміщення.

Імовірність вибору дії зупинки визначається через сигмоїдну функцію активації:

$$P(a_t = 1 \mid s_t) = \sigma(stop\_logit_t),$$

де

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$

Фінальна дія $a_t \in \{0, 1\}$ визначається на основі порогового значення τ:

$$a_t = \begin{cases} 1, & \text{якщо } P(a_t = 1 \mid s_t) \geq \tau, \\ 0, & \text{інакше.} \end{cases}$$

Таким чином, модель приймає одне з двох можливих рішень — зупинити процес вибору та прийняти поточного кандидата або продовжити перегляд

наступних кандидатів.

Поєднання рекурентного механізму LSTM та збереження прихованого стану між часовими кроками дозволяє ефективно враховувати часові залежності між спостереженнями та формувати рішення на основі всієї історії переглянутих кандидатів.

2.6. Процес навчання спроєктованої нейронної мережі

Навчання спроєктованої LSTM-моделі є динамічним та інтерактивним процесом, у якому агент самостійно формує навчальні дані через безпосередню взаємодію із середовищем, поступово вдосконалюючи власну стратегію на основі накопиченого досвіду.

У двосторонній задачі секретаря ця взаємодія набуває особливого характеру: кожна зі сторін незалежно і одночасно приймає рішення щодо поточного кандидата, не маючи інформації про рішення іншої. Взаємна згода досягається лише тоді, коли обидва агенти одночасно виявляють інтерес до поточного кандидата. Саме ця умова взаємності визначає складність задачі — стратегія агента має враховувати не лише власну оцінку кандидатів, але й невідому апріорі поведінку протилежної сторони, що унеможливлює пряме аналітичне розв’язання.

Навчання організоване в епізодичному режимі. Кожен епізод являє собою одне повне проходження двосторонньої задачі секретаря — від моменту появи першого кандидата до завершення процесу вибору. Цикл навчання складається з чотирьох послідовних етапів: ініціалізація середовища та внутрішнього стану мережі, моделювання взаємодії агента із середовищем з накопиченням траєкторій поведінки, обчислення функції втрат на основі отриманих даних та оновлення параметрів мережі методом градієнтного спуску. Повторення цього циклу тисячі разів забезпечує поступове підвищення якості стратегії агента та її адаптацію до умов двосторонньої взаємодії.

2.6.1 Ініціалізація епізоду

На початку кожного епізоду необхідно привести систему до чітко визначеного початкового стану, що гарантує повну незалежність окремих епізодів один від одного та забезпечує коректне порівняння різних стратегій в однако-

вих умовах. Середовище скидається до початкового стану:

$$s_0 = \text{env.reset()}.$$

Початкове спостереження формується як тривимірний вектор:

$$x_0 = [0,\ \xi^T,\ 0], \quad \xi \sim U[0,1],$$

де перший компонент відповідає номеру поточного кроку, другий — якості першого кандидата, третій — максимальній якості серед переглянутих кандидатів, що генерується випадково відповідно до рівномірного розподілу. Випадкова генерація якості кандидатів відображає стохастичну природу двосторонньої задачі, в якій жодна зі сторін не володіє апріорною інформацією про майбутніх кандидатів.

Паралельно з ініціалізацією середовища обнуляються внутрішні стани LSTM-мережі:

$$h_0 = [0,\ 0,\ \ldots,\ 0]^T \in \mathbb{R}^{64}, \quad c_0 = [0,\ 0,\ \ldots,\ 0]^T \in \mathbb{R}^{64}.$$

Ця процедура є принципово важливою: вона гарантує, що кожен новий епізод розпочинається з «чистої пам'яті», без будь-якого впливу інформації з попередніх взаємодій, що забезпечує коректне навчання незалежних стратегій для кожної послідовності кандидатів.

2.6.2 Взаємодія агента із середовищем

Після ініціалізації розпочинається основний цикл взаємодії, в якому агент послідовно переглядає кандидатів та приймає рішення. На кожному кроці t агент отримує спостереження

$$x_t = [t,\ q_{\text{max}},\ q_{\text{curr}}],$$

що містить інформацію про поточний момент часу, найкращого раніше зустрінутого кандидата та якість поточного претендента. Це спостереження передається у LSTM-комірку, архітектуру якої детально описано у підрозділі 2.5:

$$(h_t, c_t) = \text{LSTMCell}(x_t,\ h_{t-1},\ c_{t-1}).$$

Отриманий прихований стан h_t кодує всю накопичену до поточного моменту інформацію про послідовність кандидатів і одночасно використовується двома вихідними головами мережі для обчислення ймовірності зупинки та оцінки функції цінності:

$$P(a_t = 1 \mid s_t) = \sigma(W_p h_t + b_p), \quad V(s_t) = W_v h_t + b_v.$$

На основі обчисленої ймовірності агент стохастично вибирає дію та передає її у середовище:

$$a_t \sim \text{Bernoulli}(P(a_t = 1 \mid s_t)), \quad (s_{t+1},\ \text{done},\ \text{info}) = \text{env.step}([a_t]).$$

Середовище одночасно з обробкою рішення агента генерує незалежне рішення протилежної сторони. Якщо агент обрав $a_t = 1$, проте протилежна сторона відхилила поточного кандидата, епізод продовжується без нарахування винагороди. Успішна зустріч фіксується виключно при одночасному збігу обох позитивних рішень — саме цей механізм взаємної згоди є ключовою відмінністю двосторонньої задачі від класичної односторонньої постановки. Разом з виконаною дією зберігаються логарифм ймовірності обраного рішення та ентропія поточного розподілу:

$$H_t = -P_t \log P_t - (1 - P_t) \log(1 - P_t).$$

2.6.3 Обчислення функції втрат та оновлення параметрів

Після завершення епізоду накопичені дані використовуються для навчання мережі. Першим кроком є обчислення дисконтованої термінальної винагороди:

$$R = r \cdot \gamma^T, \quad \gamma = 0.99,$$

де r — фінальна винагорода, а T — кількість кроків у епізоді. Коефіцієнт дисконтування природним чином стимулює агента приймати рішення раніше: у двосторонній задачі надмірна затримка збільшує ризик того, що обраний кандидат стане недоступним, оскільки протилежна сторона може прийняти іншу пропозицію.

Для оцінювання якості кожного прийнятого рішення обчислюється фун-

кція переваги:

$$A_t = R - V(s_t),$$

яка відображає різницю між фактично отриманою дисконтованою винагородою та прогнозом критика для поточного стану. Додатне значення переваги свідчить про те, що рішення виявилось кращим за очікування, і відповідна дія має заохочуватись. Від'ємне значення, навпаки, сигналізує про необхідність коригування стратегії.

Загальна функція втрат, відповідно до методу Actor-Critic (підрозділ 2.4), об'єднує три складові:

$$L = L_{\text{policy}} + c_v L_{\text{value}} + c_e L_{\text{entropy}},$$

де

$$L_{\text{policy}} = -\sum_{t=0}^{T} \log \pi(a_t|s_t) \cdot A_t, \quad L_{\text{value}} = \frac{1}{2} \sum_{t=0}^{T} \big(V(s_t) - R\big)^2, \quad L_{\text{entropy}} = -\sum_{t=0}^{T} H_t.$$

Коефіцієнти $c_v = 0.5$ та $c_e = 0.01$ обрано емпірично. Ентропійна регуляризація відіграє особливо важливу роль у двосторонній задачі: вона запобігає передчасній збіжності агента до субоптимальної стратегії та стимулює його досліджувати різноманітні варіанти взаємодії, що є критично необхідним для знаходження ефективних умов взаємної згоди.

Перед оновленням параметрів здійснюється відсікання градієнтів за нормою, що є стандартною практикою при навчанні рекурентних мереж для запобігання явищу вибуху градієнтів під час зворотного поширення помилки через велику кількість часових кроків:

$$\text{якщо } \|\nabla_\theta L\| > 1, \quad \nabla_\theta L \leftarrow \frac{\nabla_\theta L}{\|\nabla_\theta L\|}.$$

Параметри мережі оновлюються адаптивним оптимізатором Adam, який індивідуально підбирає ефективну швидкість навчання для кожного параметра на основі історії градієнтів:

$$\theta \leftarrow \theta - \alpha \frac{\hat{m}}{\sqrt{\hat{v}} + \varepsilon}, \quad \alpha = 3 \times 10^{-4}, \quad \varepsilon = 10^{-7},$$

де $\hat{m}$ та $\hat{v}$ — оцінки першого та другого моментів градієнтів відповідно. Така адаптивність дозволяє ефективно долати хаотичні коливання в просторі параметрів та забезпечує значно стабільнішу збіжність порівняно з базовим стохастичним градієнтним спуском.

2.6.4 Цикл навчання та аналіз прогресу

Описаний цикл повторюється протягом $10\,000$ епізодів, протягом яких агент поступово накопичує досвід взаємодії та вдосконалює свою стратегію. Для відстеження динаміки навчання використовується експоненціальне ковзне середнє (EMA) винагороди, яке згладжує короткострокові стохастичні коливання та дозволяє спостерігати загальний тренд:

$$\text{EMA}_t = \beta \cdot \text{EMA}_{t-1} + (1 - \beta) \cdot r_t, \quad \beta = 0.98.$$

Стале зростання EMA є свідченням того, що агент успішно навчається узгоджувати власну стратегію з поведінкою протилежної сторони та дедалі ефективніше знаходить моменти взаємної згоди.

На рисунку 2.3 показано динаміку якості кандидатів, яких обирав агент протягом навчання.

Рис. 2.3. Прогрес якості обраного кандидата

На рисунку 2.4 відображено зміну абсолютного рангу обраних кандидатів.

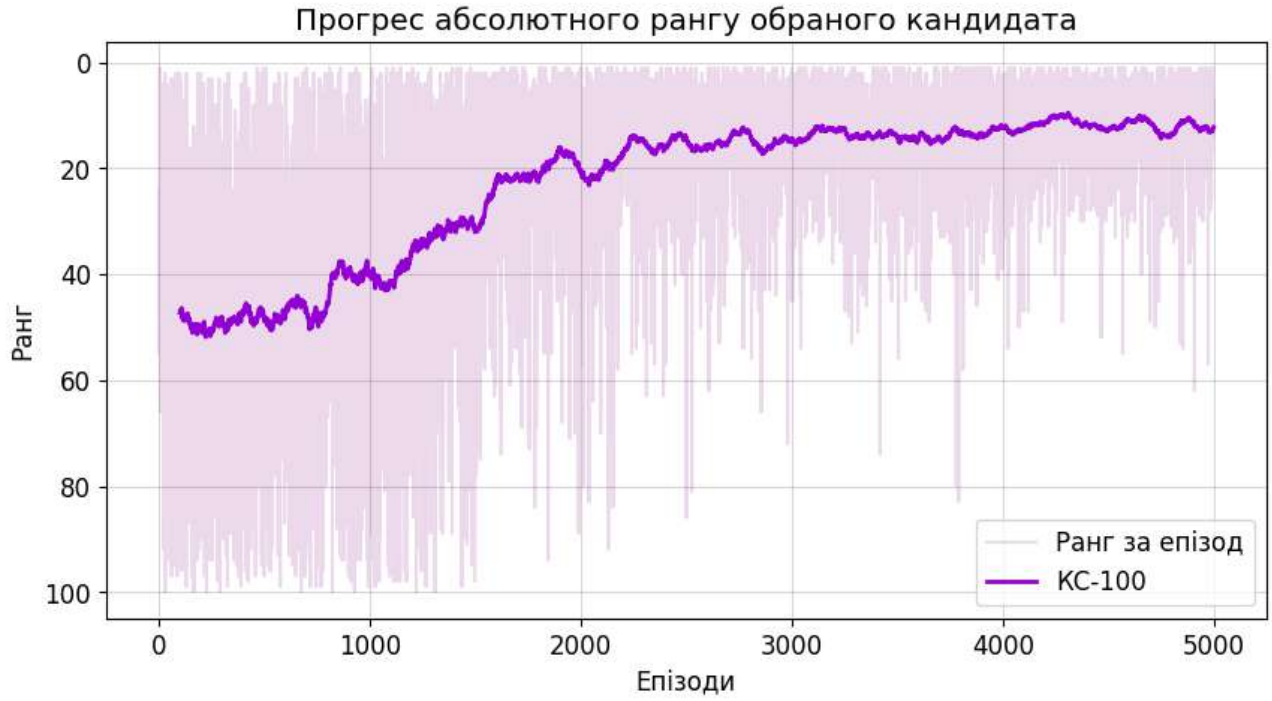

Рис. 2.4. Прогрес абсолютного рангу обраного кандидата

Стале зростання середньої якості та зменшення середнього рангу обраних кандидатів є переконливим підтвердженням того, що модель успішно формує внутрішню стратегію оптимальної зупинки, адаптовану до умов двосторонньої взаємодії.

На рисунку 2.5 показано динаміку вибору кроку зупинки протягом навчання. Світло-зелена область відображає значення окремих епізодів, тоді як темно-зелена крива представляє ковзне середнє.

Рис. 2.5. Динаміка вибору кроку зупинки під час навчання

На початкових етапах навчання поведінка агента є нестабільною та хаотичною, однак із накопиченням досвіду стратегія поступово стабілізується та набуває виразного характеру. На рисунку 2.6 показано коефіцієнт успішності — частку епізодів, у яких агент досяг взаємної згоди саме з найкращим доступним кандидатом.

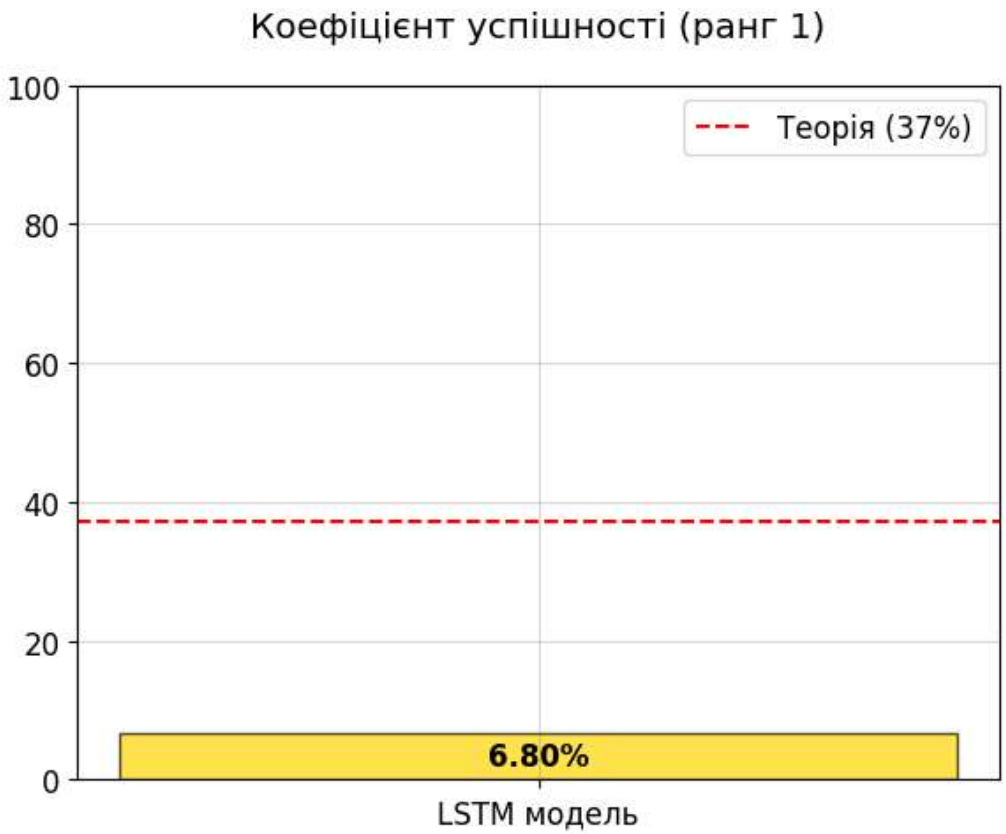

Рис. 2.6. Коефіцієнт успішності моделі

Червона пунктирна лінія відповідає теоретичній межі успішності класичної односторонньої задачі секретаря ($\approx 37\%$). У двосторонній постановці досягнення цієї межі є принципово неможливим, оскільки успішний результат вимагає одночасної та незалежної згоди обох сторін, що суттєво знижує імовірність збігу. Тим не менш, отриманий результат переконливо демонструє здатність нейронної мережі виробляти ефективну стратегію поведінки в умовах двосторонньої невизначеності.

2.6.5 Інференс та детермінований режим

Після завершення навчання агент переходить у режим інференсу, де стохастичний вибір дій замінюється детермінованим правилом прийняття рішення на основі порогового значення:

$$a_t = \begin{cases} 1, & P(a_t = 1 \mid s_t) \geq 0.5, \\ 0, & \text{інакше.} \end{cases}$$

Перехід до детермінованого режиму є важливим для коректного оцінювання якості навченої моделі: стохастичний вибір під час тестування вносив би додаткову дисперсію в результати, що ускладнювало б об'єктивне порівняння стратегій. У двосторонній задачі детермінований режим дозволяє стабільно відтворювати навчену поведінку агента та коректно оцінювати частоту взаємних збігів із середовищем, що моделює протилежну сторону.

Якість навченої стратегії оцінюється на серії з $N = 1000$ незалежних симуляцій:

$$\bar{R} = \frac{1}{N} \sum_{i=1}^{N} R_i.$$

Середня винагорода $\bar{R}$ є основною метрикою ефективності стратегії. Додатково аналізується стандартне відхилення винагороди, яке характеризує стабільність поведінки агента, та частка епізодів, у яких відбулася взаємна згода саме на найкращому доступному кандидаті.

2.7. Алгоритм моделювання двосторонньої задачі секретаря

Для дослідження ефективності розроблених нейромережевих стратегій у двосторонній задачі секретаря було спроєктовано та реалізовано кероване імітаційне середовище. На відміну від класичних моделей оптимальної зупинки, де рішення приймається однією стороною, розроблена симуляція базується на принципах мультиагентної взаємодії з асиметричним інформаційним полем. Керуючий цикл симуляції здійснює ітераційне моделювання взаємодії агентів методом Монте-Карло:

```
def run_two_side_simulation(environment, agent1, agent2, episodes=1000):
    episodes_info = []
    for _ in range(episodes):
        obs = environment.reset()
        agent1.reset()
        agent2.reset()
        terminated = False
        while not terminated:
            actions = [agent1.make_decision([obs[0]]),
                       agent2.make_decision([obs[1]])]
            obs, terminated, info = environment.step(actions)
        episodes_info.append(info)
```

```
return episodes_info
```

Ініціалізація епізоду

На початку кожного епізоду $m \in \{1, \ldots, M\}$, де $M = 1000$, виконується процедура скидання станів системи:

```
obs = environment.reset()
agent1.reset()
agent2.reset()
```

Цей крок реалізує дві операції. По-перше, середовище стохастично формує нову послідовність кандидатів і повертає стартовий вектор спостережень:

$$S_0 = \big(s_0^{(1)},\, s_0^{(2)}\big) \in \mathcal{S}.$$

По-друге, оскільки агенти реалізовані на основі архітектури LSTM, метод `agent.reset()` анулює приховані стани та стани комірок пам'яті:

$$h_0 = 0_d, \quad c_0 = 0_d.$$

Це гарантує повну незалежність поточного епізоду від досвіду, накопиченого в попередніх взаємодіях.

Децентралізоване формування дій

На кожному кроці t відбувається паралельне опитування стратегій обох агентів:

```
actions = [agent1.make_decision([obs[0]]), agent2.make_decision([obs[1]])]
```

Відповідно до концепції частково спостережуваного марковського процесу прийняття рішень (POMDP), агенти не мають доступу до повного стану середовища, а оперують виключно власними векторами спостережень $s_t^{(1)}$ та $s_t^{(2)}$. Спільний вектор дій формується як:

$$A_t = \big(a_t^{(1)},\, a_t^{(2)}\big), \quad \text{де} \quad a_t^{(i)} \sim \pi_i\big(\cdot \mid s_t^{(i)},\, h_t^{(i)}\big),$$

де π_1 та π_2 — політики першого та другого агентів відповідно, а $a_t^{(i)} \in \{0, 1\}$ —

бінарне рішення (0 — відхилити, 1 — обрати).

Умова взаємної згоди та переходи середовища

Вектор дій передається у середовище для розрахунку наступного кроку:

```
obs, terminated, info = environment.step(actions)
```

Умова завершення епізоду підпорядковується правилу двосторонньої згоди, яке виражається через індикаторну функцію:

$$I_t = a_t^{(1)} \wedge a_t^{(2)} = a_t^{(1)} \cdot a_t^{(2)}.$$

Прапорець завершення епізоду визначається диз'юнкцією двох умов — досягнення взаємної згоди або вичерпання горизонту вибору:

$$\text{terminated} = \begin{cases} \text{True}, & \text{якщо } I_t = 1 \text{ або } t = T_{\max} - 1, \\ \text{False}, & \text{інакше.} \end{cases}$$

Якщо $I_t = 0$, система переходить до стану S_{t+1}, а рекурентні мережі агентів оновлюють внутрішні стани:

$$h_{t+1}^{(i)},\, c_{t+1}^{(i)} = \text{LSTMCell}\big(s_t^{(i)},\, h_t^{(i)},\, c_t^{(i)}\big).$$

Оцінювання за методом Монте-Карло

Після завершення кожного епізоду фінальні метрики зберігаються для подальшого аналізу:

```
episodes_info.append(info)
```

Оскільки аналітичне знаходження оптимальних стратегій для двосторонніх задач є обчислювально неможливим через прокляття розмірності, алгоритм використовує статистичне усереднення за методом Монте-Карло. Очікувана спільна цінність стратегій оцінюється емпірично як середнє арифметичне виграшу обох агентів по всіх M епізодах:

$$V_{\text{joint}} = \mathbb{E}\left[\frac{R^{(1)} + R^{(2)}}{2}\right] \approx \frac{1}{2M} \sum_{m=1}^{M} \Big(R_m^{(1)} + R_m^{(2)}\Big),$$

де $R_m^{(1)}$ та $R_m^{(2)}$ — індивідуальні винагороди першого та другого агентів у m-му епізоді симуляції.

Такий підхід дозволяє звести мультиагентну оцінку до єдиного критерію системної ефективності. Стабільне зростання цієї величини в ході серії експериментів свідчить про те, що LSTM-агенти адаптуються до стратегій одне одного та знаходять ефективні умови взаємної згоди в умовах асиметрії інформації.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ДВОСТОРОННЬОЇ ЗАДАЧІ СЕКРЕТАРЯ НЕЙРОННИМИ МЕРЕЖАМИ В ПОРІВНЯННІ З ПОРОГОВОЮ СТРАТЕГІЄЮ

Для проведення обґрунтованого порівняльного аналізу ефективності двох принципово різних підходів до розв'язання двосторонньої задачі оптимальної зупинки — класичної порогової стратегії, що базується на аналітичному принципі $1/e$, та спроєктованої LSTM-моделі, навченої методом градієнта політики — було розроблено та реалізовано комплексну схему статистичного моделювання методом Монте-Карло. Вибір саме цього методу зумовлений стохастичною природою досліджуваної задачі: оскільки якості кандидатів та порядок їх надходження є випадковими величинами, аналітичне обчислення характеристик стратегій у двосторонній постановці є надзвичайно ускладненим через взаємозалежність рішень агентів. Емпірична оцінка через велику кількість незалежних реалізацій випадкового процесу є найбільш надійним підходом.

Для порогової стратегії було обрано поріг 10% — значно нижчий за класичний 37%. Це зумовлено специфікою двосторонньої постановки: класичний поріг $1/e$ оптимізований для односторонньої задачі і у двосторонній постановці призводить до надто частих відмов через невзаємність рішень.

Параметри експериментального середовища

Експериментальне середовище було сконфігуровано з урахуванням балансу між обчислювальною складністю та статистичною значущістю результатів. Кількість кандидатів у кожному епізоді зафіксована на рівні $N = 100$ — це значення достатньо велике для прояву асимптотичних властивостей класичної порогової стратегії та водночас помірне для прийнятної швидкості симуляції. Загальна кількість незалежних симуляційних епізодів становила 3000, що згідно з центральною граничною теоремою забезпечує статистичну похибку оцінок на рівні 1–2% — цілком достатньо для виявлення суттєвих відмінностей між підходами.

Якості кандидатів генерувалися як незалежні однаково розподілені випадкові величини з рівномірного розподілу $U(0, 1)$. Вибір цього розподілу зумовлений його канонічністю для теоретичних досліджень задачі секретаря, ней-

тральністю щодо апріорних припущень про форму розподілу якостей, а також можливістю одночасної інтерпретації значення якості як абсолютної оцінки та як квантиля рангового розподілу.

Метрики оцінювання

Для всебічної кількісної характеристики поведінки стратегій було обрано чотири взаємодоповнюючі показники. **Середня якість обраної пари** обчислюється як середнє арифметичне значень якості кандидатів, обраних обома сторонами в межах одного епізоду, усереднене за всіма симуляціями. **Середній абсолютний ранг пари** визначається аналогічно для абсолютних рангів обраних кандидатів. **Середній крок зупинки** фіксує середній номер кроку, на якому обидва агенти досягли взаємної згоди. **Коефіцієнт успішності** являє собою емпіричну ймовірність ідеального подвійного матчингу — частку епізодів, у яких обидва агенти одночасно обрали кандидата з абсолютним рангом 1.

Результати порівняльного аналізу

Розпочнемо аналіз із коефіцієнта успішності — метрики, що характеризує здатність стратегії досягти ідеального результату.

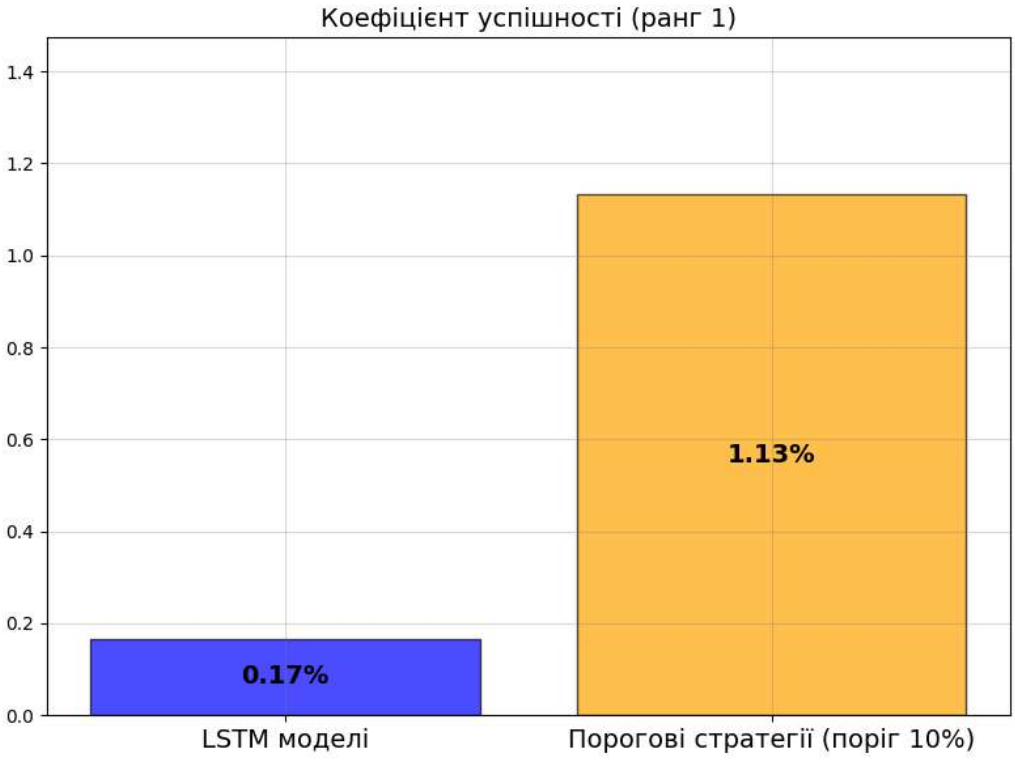

Рис. 3.1. Коефіцієнт успішності (ранг 1)

На рисунку 3.1 порогова стратегія демонструє коефіцієнт успішності на рівні 1.13%, тоді як LSTM-модель досягає лише 0.17%. Цей результат пояснюється фундаментальною відмінністю у цільових функціях двох підходів: порогова стратегія конструктивно орієнтована на пошук абсолютного максимуму вибірки, тоді як LSTM-модель оптимізує математичне сподівання якості через термінальну винагороду у функції втрат. Нейромережа свідомо жертвує ймовірністю досягнення рангу 1 заради значно вищої середньої якості рішення та суттєвого зниження ризику отримання незадовільного результату. Саме це підтверджується наступною метрикою.

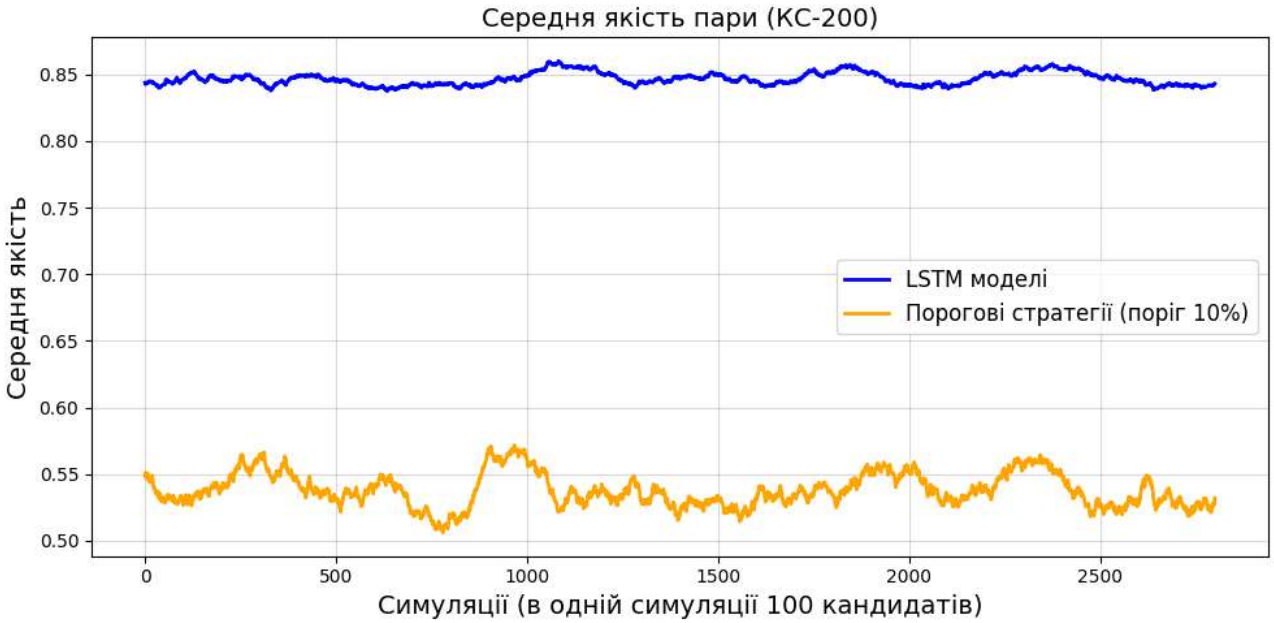

Рис. 3.2. Порівняння середніх якостей пар

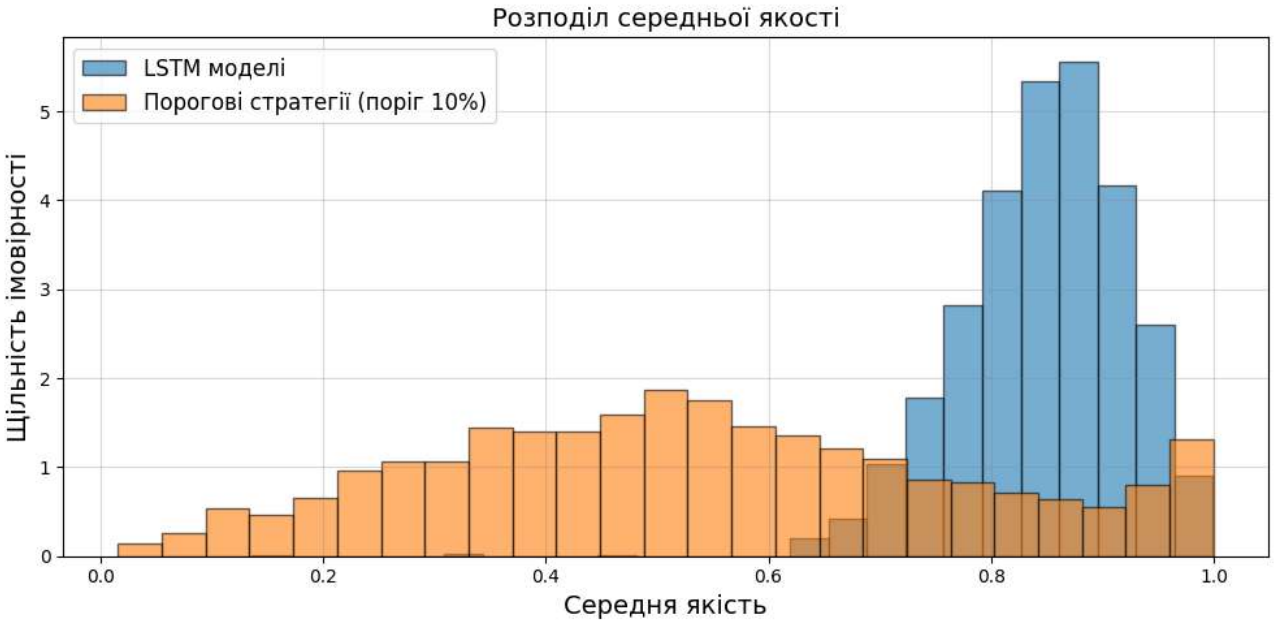

Рис. 3.3. Розподіл середніх якостей пар

На рисунку 3.2 суттєва перевага LSTM-моделі є очевидною: середня якість обраної пари стабільно тримається в межах 0.84–0.86, тоді як порогова стратегія забезпечує лише 0.53–0.57. Розрив між підходами становить близько 0.30 одиниць якості. Крім того, крива LSTM-моделі має значно меншу амплітуду коливань, що свідчить про вищу стабільність нейромережевої стратегії.

Розподіл якостей на рисунку 3.3 наочно підтверджує цю закономірність. Для LSTM-моделі він є явно скошеним у бік високих значень з вираженою мо-

дою близько 0.82–0.88 та практично відсутністю результатів нижче 0.65. Порогова стратегія, навпаки, генерує широкий розподіл з модою близько 0.45–0.55 та значною кількістю випадків із низькою якістю, включно зі значеннями у діапазоні 0.0–0.3. Це пояснюється тим, що при невзаємному виборі порогові агенти часто змушені приймати останнього кандидата незалежно від його якості, що призводить до практично випадкових результатів.

Не менш важливою характеристикою стратегії є момент прийняття рішення.

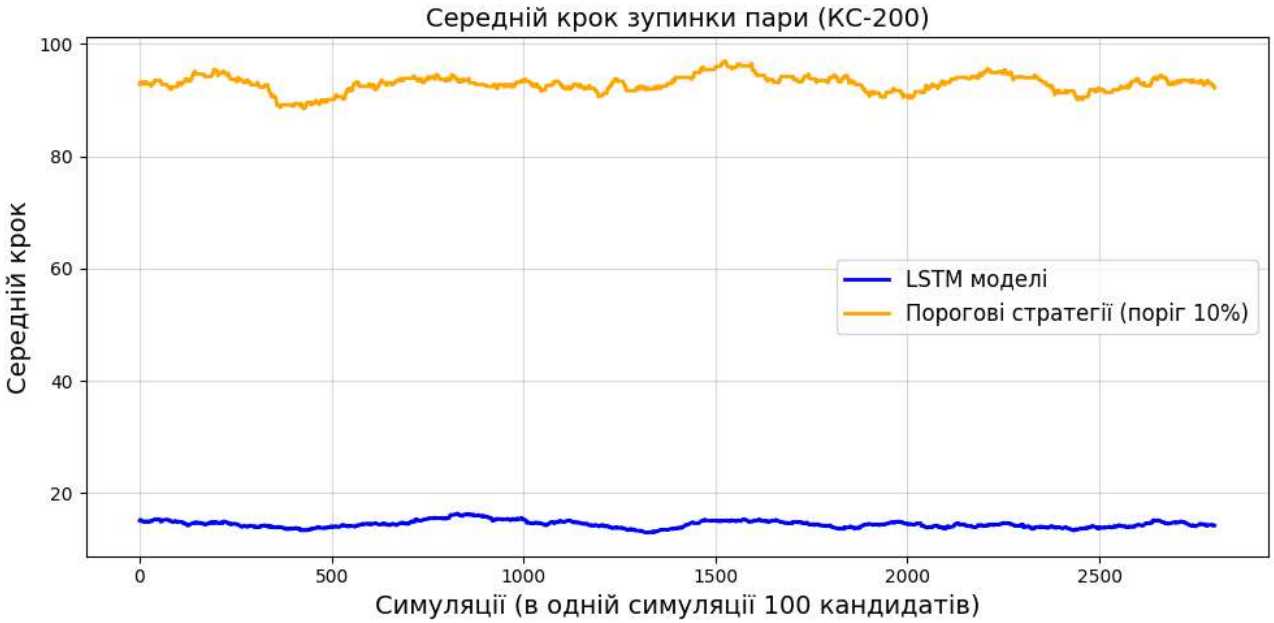

Рис. 3.4. Порівняння середніх кроків зупинки пар

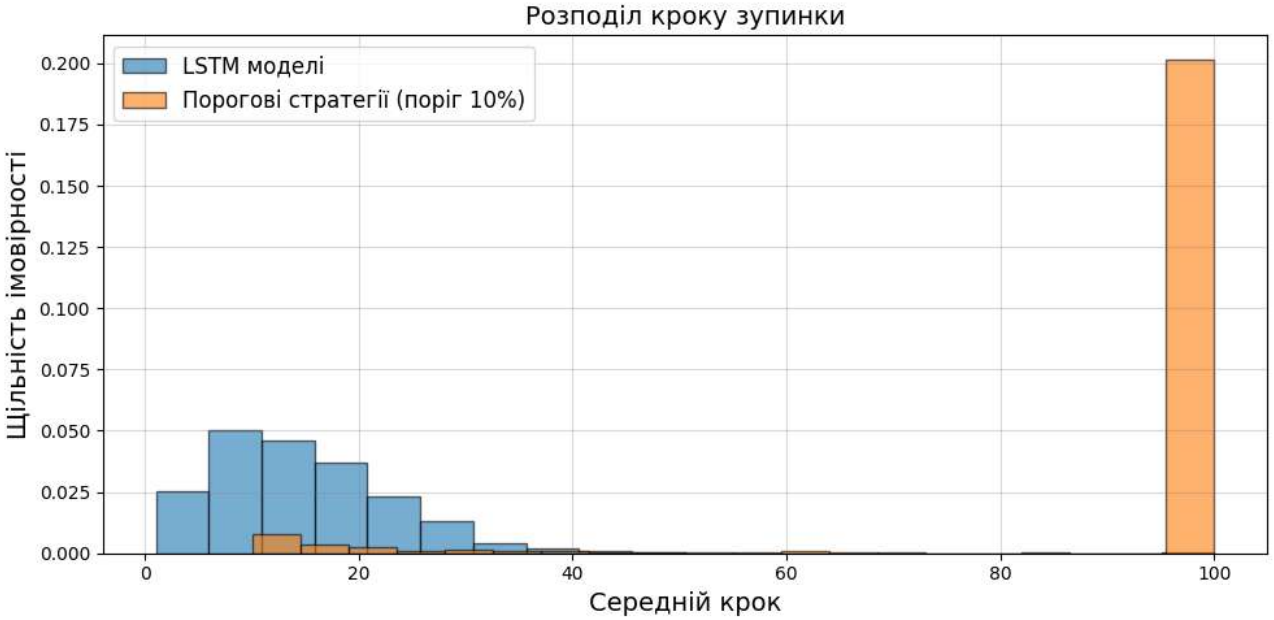

Рис. 3.5. Розподіл середніх кроків зупинки пар

На рисунку 3.4 показано, що LSTM-модель приймає рішення в середньому на 13–16 кроці зі 100 можливих, що свідчить про здатність швидко ідентифікувати якісного кандидата без необхідності перегляду значної частини вибірки. Порогові стратегії, навпаки, доходять майже до кінця вибірки — середній крок зупинки коливається в межах 90–95. Така поведінка зумовлена тим, що при незалежних рішеннях за фіксованим порогом ймовірність одночасного збігу обох позитивних рішень є вкрай низькою, що призводить до систематичного затягування процесу.

Розподіл кроків зупинки на рисунку 3.5 підтверджує цей висновок. Для LSTM-моделі він сконцентрований у діапазоні 5–30 кроків з модою близько 10–15, що відповідає швидкому та впевненому прийняттю рішень. Для порогових стратегій спостерігається яскраво виражений пік у точці 100 — близько 20% усіх епізодів закінчуються вимушеним прийняттям останнього кандидата. Це є фундаментальною проблемою порогових стратегій у двосторонній задачі: незалежність рішень двох агентів за однаковим фіксованим правилом унеможливлює своєчасне досягнення взаємної згоди.

Аналіз абсолютних рангів обраних кандидатів дозволяє підсумувати загальну картину.

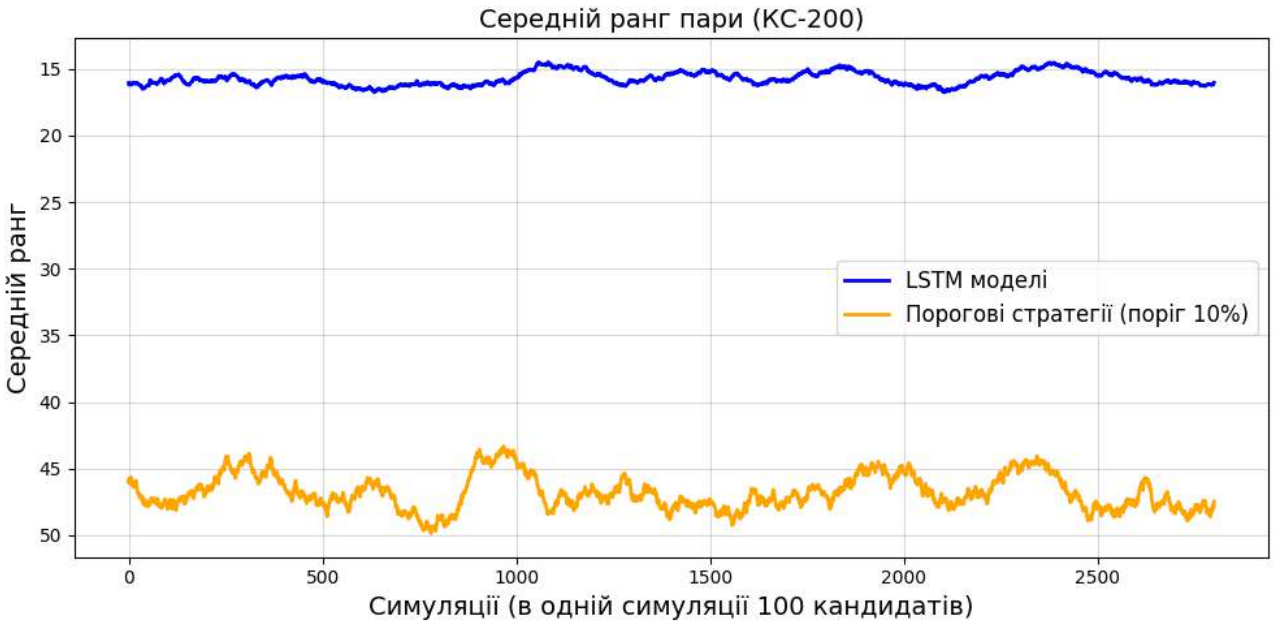

Рис. 3.6. Порівняння середніх рангів пар

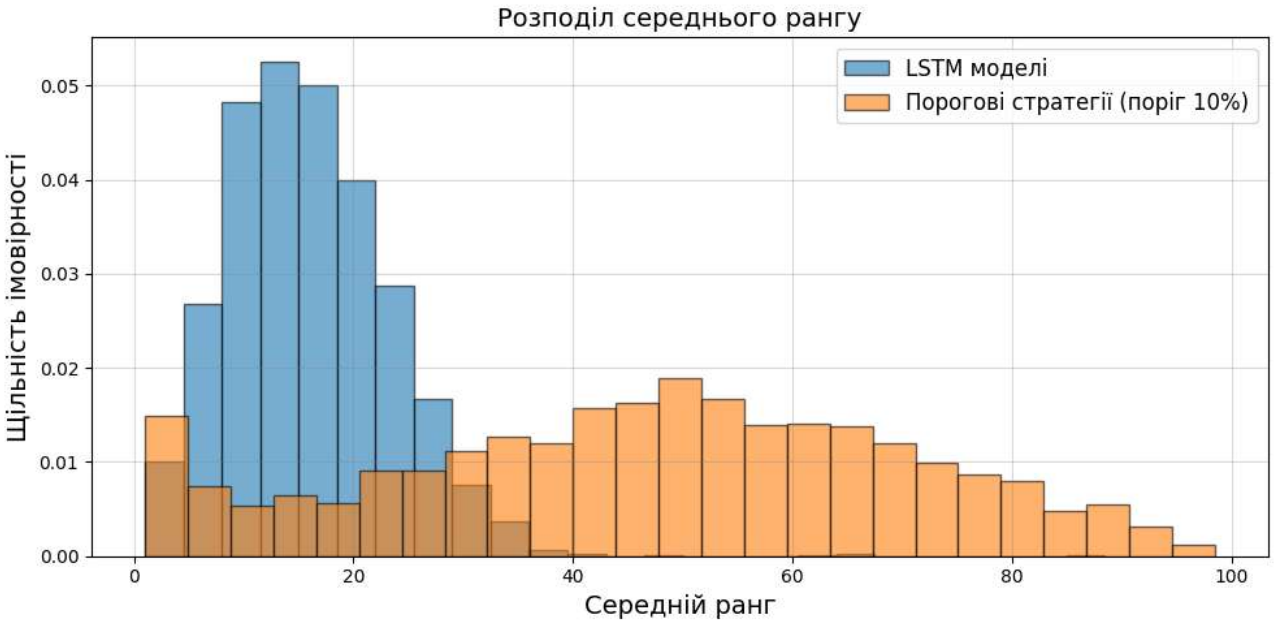

Рис. 3.7. Розподіл середніх рангів пар

На рисунку 3.6 LSTM-модель забезпечує середній ранг пари близько 15–17, тоді як порогові стратегії дають 42–47. Таким чином, нейромережа систематично знаходить кандидатів із верхніх 15% вибірки, тоді як порогові стратегії обирають кандидатів переважно з середини рейтингу.

Розподіл рангів на рисунку 3.7 наочно підтверджує цю асиметрію. Для LSTM-моделі він є вузьким і зосередженим у діапазоні 10–25 з вираженою модою близько 15–18, що свідчить про послідовну та передбачувану якість рішень. Розподіл порогових стратегій є значно ширшим — приблизно 30–90 — з модою близько 50 та помітним піком у нижніх рангах, що відображає регулярні випадки вимушеного вибору наприкінці вибірки.

Узагальнюючи результати всіх чотирьох метрик, можна зробити однозначний висновок: LSTM-модель значно перевершує класичну порогову стратегію в умовах двосторонньої задачі оптимальної зупинки за показниками середньої якості, стабільності результатів та ефективності використання вибірки. Ця перевага є наслідком здатності нейромережі адаптувати стратегію зупинки до умов взаємної залежності рішень, тоді як фіксований аналітичний поріг залишається принципово нечутливим до поведінки протилежної сторони.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було досліджено задачу оптимальної зупинки та особливості її класичних і сучасних підходів до розв'язання. Проведений аналіз показав, що класична задача секретаря має ефективний аналітичний розв'язок лише для одностороннього випадку, тоді як у двосторонніх моделях ефективність традиційних порогових стратегій суттєво знижується.

У роботі було розроблено та реалізовано модель на основі рекурентної нейронної мережі типу LSTM, яка навчається за допомогою методів навчання з підкріпленням та здатна приймати рішення в умовах послідовного надходження кандидатів. Було спроєктовано архітектуру мережі, реалізовано механізм взаємодії агента із середовищем та побудовано процес навчання і оптимізації параметрів моделі.

Проведені експерименти показали, що запропонована LSTM-модель демонструє суттєву перевагу над класичною пороговою стратегією за основними практичними метриками. Зокрема, модель забезпечує вищу середню якість сформованих пар, кращі ранги кандидатів та швидше прийняття рішень. Аналіз процесу навчання підтвердив поступове покращення стратегії агента та стабілізацію його поведінки.

Отримані результати свідчать про перспективність застосування рекурентних нейронних мереж та методів навчання з підкріпленням у задачах оптимальної зупинки та суміжних задачах прийняття рішень в умовах невизначеності. Перспективами подальших досліджень є ускладнення архітектури моделі, використання механізмів уваги та розширення підходу на багатоагентні сценарії.

СПИСОК ЛІТЕРАТУРИ

1. Eriksson K., Sjöstrand J., Strimling P. Optimal Expected Rank in a Two-Sided Secretary Problem. *Operations Research*. 2007. Vol. 55, No. 5. P. 921–931. URL: https://doi.org/10.1287/opre.1070.0403.

2. Brams S. J., Kilgour D. M. Revisiting the Secretary Problem. *The Mathematics Enthusiast*. 2026. Vol. 23, No. 3. P. 235–248. URL: https://doi.org/10.54870/1551-3440.1690.

3. Todd P. M., Miller G. F. From Pride and Prejudice to Persuasion: Satisficing in Mate Search. *Simple Heuristics That Make Us Smart* / ed. by G. Gigerenzer, P. M. Todd, ABC Research Group. New York : Oxford University Press, 1999. P. 287–308.

4. Freeman P. R. The Secretary Problem and Its Extensions: A Review. *International Statistical Review / Revue Internationale de Statistique*. 1983. Vol. 51, No. 2. P. 189–206. URL: https://doi.org/10.2307/1402748.

5. Chung J., Gulcehre C., Cho K., Bengio Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv*. 2014. URL: https://arxiv.org/abs/1412.3555.

6. Sutton R. S., Barto A. G. Reinforcement Learning: An Introduction. 2nd ed. Cambridge : MIT Press, 2018. URL: http://incompleteideas.net/book/the-book-2nd.html.

7. Konda V. R., Tsitsiklis J. N. Actor-Critic Algorithms. *Advances in Neural Information Processing Systems (NIPS)*. 2000. Vol. 12. P. 1008–1014.

8. Mnih V., Badia A. P., Mirza M., Graves A., Lillicrap T. P., Harley T., Silver D., Kavukcuoglu K. Asynchronous Methods for Deep Reinforcement Learning. *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*. 2016. P. 1928–1937. URL: https://doi.org/10.48550/arXiv.1602.01783.

9. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization. *3rd*

International Conference for Learning Representations (ICLR 2015). San Diego, 2015. URL: https://doi.org/10.48550/arXiv.1412.6980.

10. Williams R. J. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*. 1992. Vol. 8, no. 3–4. P. 229–256.

11. XIV Всеукраїнська наукова конференція молодих математиків : збірник праць. Київ, 2026. С. 141–142. URL: https://drive.google.com/file/d/1I5sdbEmrgRn5VO-oyP3HeXXXe9euYp76/view.

12. Melnyk D., Shchestyuk N., Zakharyichenko Y. The stochastic experiment for some generalizations of the secretary problem. *Mohyla Mathematical Journal*. 2025. Vol. 8. P. 40–45. URL: https://doi.org/10.18523/2617-70808202540-45.

ВИКОРИСТАННЯ ІНСТРУМЕНТІВ ШТУЧНОГО ІНТЕЛЕКТУ

У процесі підготовки кваліфікаційної роботи було використано мовну модель Claude (Anthropic) як допоміжний інструмент для покращення якості текстового оформлення. Зокрема, інструмент застосовувався для редагування та покращення лаконічності окремих фрагментів тексту, уточнення формулювань і стилістичного узгодження викладу в усіх розділах роботи.

Усі змістовні рішення, наукові результати, математичні викладки, програмна реалізація та висновки є результатом самостійної роботи автора. Згенеровані або відредаговані фрагменти тексту було критично переглянуто, перевірено на відповідність змісту дослідження та доопрацьовано автором перед включенням до роботи.