

**УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ**

ФАКУЛЬТЕТ УПРАВЛІННЯ ПРОЦЕСАМИ ПЕРЕВЕЗЕНЬ

Кафедра управління експлуатаційною роботою

МЕТОДИЧНІ ВКАЗІВКИ

**до практичних занять
та виконання індивідуальних завдань**

**з дисципліни
«СУЧАСНІ ІНФОРМАЦІЙНІ СИСТЕМИ
В УПРАВЛІННІ ЗАЛІЗНИЧНИМИ ПІДРОЗДІЛАМИ»**

Харків – 2022

Методичні вказівки розглянуто та рекомендовано до друку на засіданні кафедри управління експлуатаційною роботою 16 травня 2022 року, протокол № 12.

Методичні вказівки рекомендовано для здобувачів вищої освіти факультету «Управління процесами перевезень» спеціальності 275.02 «Транспортні технології (залізничний транспорт)» другого (магістерського) рівня освітньо-професійних програм «Організація перевезень і управління на транспорті», «Організація міжнародних перевезень», «Митний контроль на транспорті» та освітньо-наукової програми «Організація перевезень і управління на транспорті» усіх форм навчання.

Укладачі:

професор Т. В. Бутько,
доценти В. М. Прохоров,
Г. О. Прохорченко,
Д. В. Шумик

Рецензент

професор О. М. Огар

ЗМІСТ

Вступ.....	4
Теоретична частина.....	6
Практична робота 1. Розв’язання задач із застосуванням генетичного алгоритму.....	7
Практична робота 2. Розв’язання задач із застосуванням нейронної мережі.....	17
Практична робота 3. Розв’язання задач із застосуванням нечіткої логіки.....	28
Практична робота 4. Опис технологічних процесів із застосуванням UML-діаграм.....	40
Список літератури.....	49

ВСТУП

Ефективне управління будь-яким виробничим процесом, особливо таким складним, як залізничні перевезення, може бути реалізовано лише на основі всеосяжної, достовірної та підготовленої для аналітичної обробки інформації. Впровадження інформаційних технологій є одним з найбільш важливих та ефективних заходів щодо інноваційного розвитку системи управління перевізним процесом. Одними з новітніх трендів інноваційного розвитку систем управління підприємств різних галузей виробництва є цифровізація та цифрова трансформація виробничих процесів. Як відомо, однією з найважливіших галузей виробництва є транспортна галузь. Отже, найбільш динамічні та масштабні зміни останніх років, пов'язані з впровадженням сучасних технологій, відбуваються саме на транспорті. І хоча залізниці, з огляду на консервативність їх виробничого процесу, вважаються одними з найбільш традиційних елементів транспортної галузі, потенціал сучасних цифрових технологій є таким, що навіть тут за їх допомогою може бути створена істотна частина доданої вартості. З огляду на це та на ту виключну роль залізничного транспорту, яку він відіграє в транспортній системі України, а також завдяки зусиллям фахівців, він займає лідируючу позицію у сфері інформатизації та телекомунікацій серед усіх видів транспорту.

Дисципліна «Сучасні інформаційні системи в управлінні залізничними підрозділами» допомагає здобувачам вищої освіти у вивченні основ процесів обробки інформаційних потоків в контексті управління технологічним процесом залізничних перевезень, і сприяє підготовці саме таких фахівців, які будуть здатні утримувати цю лідируючу позицію залізничного транспорту у сфері інформаційних технологій і надалі.

Передбачено, що в ході освоєння матеріалів при виконанні практичних робіт у здобувачів формуються такі компетенції:

1 ціннісно-сміслова компетентність (формування та розширення світогляду студента в галузі використання існуючих інформаційних технологій при управлінні перевізним процесом на залізничному транспорті та формування сучасних

інтелектуальних технологій на автоматизованих робочих місцях (АРМ) оперативних працівників);

2 загальнокультурна компетентність (розуміння культурних, історичних та регіональних особливостей, що склалися в Україні та за її межами в області управління експлуатаційною роботою залізничного транспорту);

3 навчально-пізнавальна компетентність (формування у здобувача зацікавленості про стан та перспективи розвитку інтелектуальних транспортних технологій при прийнятті управлінських рішень; методу їх формування з метою розвитку креативної складової компетентності; оволодіння навичками комп'ютерного моделювання; здатність здобувача формувати цілі дослідження; проводити класифікацію складних ситуацій в процесі прийняття рішень, зокрема в умовах вирішення слабо-структурованих проблем в процесі функціонування залізничного транспорту України);

4 інформаційна компетентність (розвиток вмінь студента до самостійного пошуку, аналізу, структурування та відбору потрібної інформації в області автоматизації процесів прийняття управлінських рішень в перевезенні вантажів та пасажирів за допомогою сучасних інформаційних технологій);

5 комунікативна компетентність (розвиток у здобувача навичок роботи в команді шляхом реалізації групових проектів в області формування локальних та просторово-розподілених СППР, вміння презентувати власний проект та кваліфіковано вести дискусію у досліджуваній сфері);

6 компетентність особистісного самовдосконалення (елементи духовного й інтелектуального саморозвитку; підтримка постійної жаги до самовдосконалення та самопізнання, шляхом постійного пошуку нетрадиційних підходів до проблеми формування інтелектуальних транспортних технологій).

ТЕОРЕТИЧНА ЧАСТИНА

При виконанні теоретичної частини здобувачі готують відповідь на теоретичне питання за темами дисципліни відповідно до останньої цифри у заліковій книжці (таблиця 1).

Таблиця 1 – Теми для виконання теоретичної частини роботи

Остання цифра залікової книжки	Теоретичне питання
0	Методи прийняття рішень в умовах невизначеності (нечітка логіка)
1	Байесові мережі. Приклади задач, які розв'язуються при їх застосуванні
2	Модель штучного нейрону. Аналогія з біологічним нейроном. Одношаровий персептрон
3	Штучні нейронні мережі. Галузь застосування
4	Генетичні алгоритми (ГА). Загальні відомості. Приклади задач, які розв'язуються за допомогою ГА
5	Основні відмінності ГА від інших способів оптимізації. Визначення основних понять – популяція, ген, хромосома
6	Класифікація штучних нейронних мереж
7	Принципи побудови і функціонування прикладних систем штучного інтелекту
8	Мультиагентні системи. Поняття, особливості, приклади розв'язуваних задач
9	Нечітка логіка. Поняття, особливості, приклади розв'язуваних задач

Основою доповіді здобувача є визначення термінів, пов'язаних з розвитком інформаційного суспільства, його характерних властивостей. Всі використовувані терміни мають бути зрозумілими доповідачу. Він зобов'язаний пояснити їх у разі появи питань. Інформація, що наводиться в доповіді, не має співпадати з лекційним матеріалом, що викладений у відповідному розділі дисципліни на сайті <https://do.kart.edu.ua>.

При підготовці до доповіді дозволяється користуватися літературою, що знаходиться у відкритому доступі.

Вимоги до доповіді: мова доповіді – українська, обсяг у межах 3–5 сторінок тексту шрифту Times New Roman 14 з інтервалом 1,5 на аркуші А4 з полями 2 см з усіх боків.

Структура доповіді:

- назва;
- вступ (важливість пропонованої теми);
- викладення матеріалу;
- висновок;
- список використаних джерел.

Практична частина складається з чотирьох задач. Перед виконанням кожної практичної роботи необхідно ознайомитися з теоретичним матеріалом.

ПРАКТИЧНА РОБОТА 1

Розв'язання задач із застосуванням генетичного алгоритму

Теоретичний матеріал для виконання завдання

У процесі проєктування, планування та виробництва, зокрема, і на залізничному транспорті, може виникати багато завдань, які мають велику область пошуку рішень. Наприклад, задача побудови графіка руху поїздів на одноколіїній дільниці.

Задача побудови ГРП, головним чином, полягає в тому, щоб відшукати для кожного поїзда послідовність прослідування станцій на дільниці з урахуванням вирішення конфліктних ситуацій з іншими поїздами та за умови дотримання експлуатаційних обмежень залізничної інфраструктури. Одночасне прокладання великої кількості поїздів призводить до величезного простору пошуку рішень.

Найбільш складною є задача побудови графіка руху поїздів на одноколіїній дільниці. На перший погляд ця задача може здатися не дуже складною і такою, що може бути вирішена перебором усіх можливих варіантів прямування поїздів дільницею. Дійсно, якщо розглянути задачу, в якій на одному плановому періоді часу два поїзди з різних напрямків мають

проїхати через одну станцію на одноколіійній дільниці, виникне лише одна конфліктна ситуація, рисунок 1.

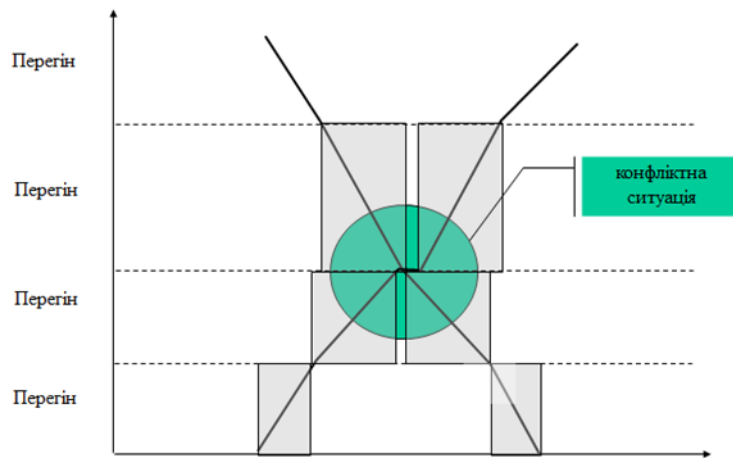


Рисунок 1 – Побудова ГРП на одноколіійній дільниці

Рішення цієї конфліктної ситуації відповідає вирішенню завдання побудови ГРП. Область вирішення такої задачі налічує лише два можливих варіанти схрещення (перший поїзд зупиняється на станції, другий – прослідує без зупинки, і навпаки). Але якщо збільшувати розмірність задачі (кількість пар поїздів), то можна побачити, що розмір області рішень збільшується дуже швидко [1].

Якщо представити конфліктну ситуацію з двома варіантами рішення в якості параметра, що характеризує умову прокладання поїздів, який приймає дискретні значення: 1 – поїзд має зупинку при схрещенні або 0 – поїзд не має зупинки, то при кількості конфліктних ситуацій в ГРП, рівній (мається на увазі сумарну кількість схрещень і обгонів), кількість можливих варіантів графіка руху поїздів визначається за виразом:

$$W = 2^R$$

У випадку 50 конфліктних ситуацій в ГРП кількість можливих варіантів графіка складає $W=2^{50}=11,3 \cdot 10^{14}$. Якщо для оцінювання одного варіанту потрібно виконати всього 100 операцій, то загальна обчислювальна робота при швидкості розрахунків комп'ютерних пристроїв 10 млн. операцій за секунду

займе близько 400 років. Це доводить, що при великих значеннях R знаходження оптимального графіка простим перебором неможливе навіть при використанні найбільш швидкодіючого пристрою.

Побудова ГРП на двоколінійній дільниці також є складною задачею. Вже при $\max N_{вант} = 25$; $N_{вант} = 20$; $I_p = 10xв$ можлива кількість варіантів двоколійного графіка складе 50 000.

Доведено, що задача розрахунку ГРП навіть теоретично не може мати ефективних алгоритмів, які можуть знайти оптимальне розв'язання. Методом повного перебору варіантів цю задачу розв'язати також неможливо, тому що для цього потрібен колосальний об'єм обчислювальних ресурсів. Отже, немає сенсу витратити зусилля на пошуки таких алгоритмів, які б могли розв'язати цю задачу точно, а потрібно розглянути можливості створення такого алгоритму, за допомогою якого можливо було б знаходити ГРП, близький до оптимального, протягом прийняттого часового інтервалу і який можливо було реалізувати у вигляді комп'ютерної програми. Оскільки точне розв'язання задачі для реального залізничного полігону неможливе, потрібно розглянути можливість застосування сучасних алгоритмів [1].

Для вирішення завдання, наведеного вище, доцільно розглянути можливість застосування алгоритмів комбінаторної оптимізації. Алгоритмам комбінаторної оптимізації приділяється велика увага останніми роками.

Досягнення в комбінаторній оптимізації поділяється на два підкласи. Перший містить ті проблеми, які можна ефективно вирішити. Для прикладу: лінійне програмування, підбирання, мережеві проблеми. Другий містить проблеми, які не мають алгоритму, який розв'язує кожен випадок за поліноміальний час. Отже, є випадки які потребують супер-поліноміального або й експоненціального часу для оптимального розв'язання. Багато відомих проблем і задач відносять до цього класу, напевне, найвідомішим прикладом є задача комівояжера [2]. Очевидно, що складні задачі мають оброблятися на практиці. Грубо кажучи, це можна зробити двома типами алгоритмів. У першому випадку використовуються алгоритми оптимізації, які знаходять можливий оптимальний розв'язок, з використанням великої кількості часу, та обчислень. У другому випадку застосовують

евристичні алгоритми, які дають змогу знайти наближені розв'язки, виконуючи невелику кількість обчислень. Алгоритм імітації відпалу один з найвідоміших локальних пошукових алгоритмів [3].

Локальний пошук – пошук, що здійснюється алгоритмами локального пошуку, групою алгоритмів, у яких пошук ведеться тільки на підставі поточного стану, а раніше пройдені стани не враховуються й не запам'ятовуються. Основною метою пошуку є не знаходження оптимального шляху до цільової точки, а оптимізація деякої цільової функції, тому задачі, розв'язувані подібними алгоритмами, називають задачами оптимізації. Для опису простору станів у таких задачах використовують ландшафт простору станів, у цьому поданні задача зводиться до пошуку стану глобального максимуму (або мінімуму) на даному ландшафті. На початку 1980-х Kirkpatrick (1983) і незалежно Cerny (1985) ввели концепції відпалу у комбінаторній оптимізації. Спочатку ці концепції значною мірою були побудовані на аналогії між фізичним процесом відпалу твердих тіл і проблемою вирішення великих комбінаторних оптимізацій [2, 3].

Серед стохастичних методів оптимізації особливо добре зарекомендували себе на практиці методи, які використовують закономірності і принципи, запозичені у самої природи, такі як методи еволюційної оптимізації, методи ройового інтелекту, алгоритми імітації відпалу. Перші дві групи відносяться до так званих популяційних методів, оскільки використовують системи, що складаються з агентів (популяцій агентів).

Як правило, під **агентом** розуміється деяка точка в просторі пошуку рішень завдання, а процес оптимізації полягає в переміщенні агентів в цьому просторі. При цьому методи еволюційної оптимізації передбачають створення на кожному кроці нових популяцій агентів з урахуванням досвіду, отриманого попередніми популяціями. Методи ройового інтелекту використовують деякі правила, задають непрямий обмін інформацією між агентами. Алгоритми ройового інтелекту принципово відрізняються від еволюційних, оскільки не вимагають створення на кожному кроці нових популяцій шляхом відбору і схрещування агентів попередньої популяції, а

використовують колективні децентралізовані переміщення агентів однієї популяції, без процедур відбору, знищення старих і породження нових агентів. Спосіб визначення, чи належить той чи інший популяційний алгоритм до колективного інтелекту: в формулах, які задають поведінку (переміщення, міграцію) агентів рою, має бути присутнім об'єкт, що забезпечує непрямий обмін інформацією між агентами.

До еволюційних алгоритмів відносяться:

- алгоритм зростаючих дерев;
- алгоритм еволюції розуму;
- бактеріальна оптимізація;
- гармонійний пошук;
- генетичний алгоритм;
- культурний алгоритм;
- зозулин пошук;
- меметичний алгоритм;
- сорняковий алгоритм.

Ройовий інтелект включає в себе такі алгоритми:

- алгоритм гравітаційного пошуку;
- алгоритм динаміки формування річок;
- алгоритм колонії мурах;
- алгоритм кажанів;
- алгоритм роїння бактерій;
- алгоритм рою бджіл;
- алгоритм рою частинок;
- алгоритм світлячків;
- алгоритм зграї вовків;
- гармонійний пошук;
- інтелектуальні краплі води;
- мавпячий алгоритм;
- пошук косяком риб;
- стохастичний дифузний пошук;
- покращений алгоритм пошуку зозулі;
- електромагнітний пошук.

ЗАДАЧА 1

Розв'язання задач із застосуванням генетичного алгоритму

Постанова задачі. В умовному залізничному вузлі є шість місць, звідки локомотиву необхідно забрати вагони за умови проходження найкоротшої відстані для зменшення непродуктивних пробігів. Умовний полігон наведено на рисунку 2.

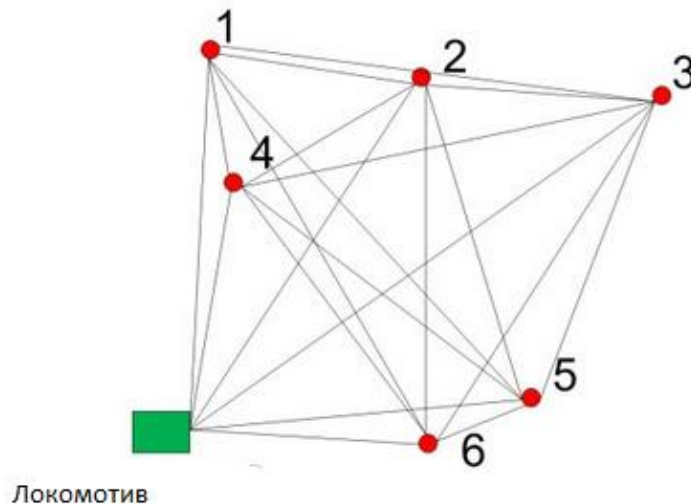


Рисунок 2 – Умовний полігон залізничного вузла

Відстані від кожної точки задано у вигляді матриці відстаней, рисунок 3.

	1	2	3	4	5	6
1		6	24	4	40	38
2	6		7	5	30	32
3	24	7		34	27	31
4	4	5	34		23	25
5	40	30	27	23		4
6	38	32	31	25	4	

Рисунок 3 – Матриця відстаней

Для виконання задачі кожний студент змінює дані відповідно до передостанньої цифри у заліковій книжці, таблиця 2.

Таблиця 2 – Вихідні дані щодо зміни у матриці відстаней

Передостання цифра залікової книжки	Зміни у матриці відстаней
0	Збільшити всі дані на 1
1	Збільшити всі дані на 9
2	Збільшити всі дані на 2
3	Зменшити всі дані на 2
4	Збільшити всі дані на 2
5	Збільшити всі дані на 10
6	Збільшити всі дані на 6
7	Збільшити всі дані на 7
8	Збільшити всі дані на 5
9	Зменшити всі дані на 1

Рішення завдання – це послідовність проходження контрольних точок.

Для розв’язання подібних комбінаторних задач можна використовувати класичні методи, але при збільшенні розмірності задачі для їх використання потрібні значні витрати часу. Генетичні алгоритми надають можливість швидкої генерації прийнятних розв’язків задач, які неможливо розв’язати іншими аналітичними методами, уникаючи повного перебору і значно скорочуючи часові витрати.

При застосуванні традиційних методів оптимізації і пошуку, у разі навіть незначної зміни параметрів середовища, всі обчислення доводиться проводити заново. Еволюційний підхід дозволяє аналіз і адаптацію вже створеної популяції до нових умов середовища, чим теж скорочує час роботи алгоритму і реалізує принцип машинної адаптації і навчання.

У класичному генетичному алгоритмі початкова популяція створюється випадковим чином. Розмір популяції (всю кількість хромосом позначимо як N), який не змінюється впродовж всього алгоритму, є фіксований. Кожна особина генерується як L – бітний рядок, де L – довжина коду особини. Кожна особина – це вирішення поточної проблеми. Більш придатна особина – це придатний розв’язок. Ці особливості роблять генетичний алгоритм порівняно з іншими оптимізаційними алгоритмами таким, що дає кращий результат.

Канонічний ГА складається з таких кроків:

- 1 генеруємо початкову популяцію з n хромосом;
- 2 обчислюємо для кожної хромосоми її життєздатність;
- 3 за допомогою обраного методу відбору обираємо пару хромосом-батьків;
- 4 виконуємо схрещування батьків із імовірністю p_c . Отримуємо двох нащадків;
- 5 виконуємо мутацію нащадків із імовірністю p_m ;
- 6 повторюємо 3–5 доки не буде сформовано нове покоління з n хромосом;
- 7 повторюємо 2–6 доки не досягнемо умови зупинки алгоритму.

У нашому випадку алгоритм починається з того, що створюється квадратна матриця відстаней, змінена за варіантом.

Далі формується перша популяція, що складається з випадково згенерованих особин (тобто послідовності забирання вагонів зі станцій).

У роботі кожного здобувача має бути створена популяція з чотирьох особин, відмінна від наведеної у прикладі.



Рисунок 4 – Приклад початкової популяції

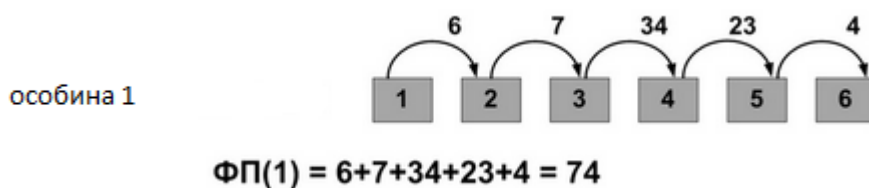
Далі обчислюється функція приналежності (ФП) – довжина шляху кожної особини. У нашому прикладі це буде сума відстаней від точки до точки в обраному маршруті.

$$\text{ФП} = P(1)+P(2)+P(3)+P(4)+P(5)+P(6),$$

де $P(1) \dots P(6)$ – відстань між точками у відповідному переході з матриці відстаней.

Нам необхідно знайти мінімальну відстань, тому, чим менше значення ФП для особини, тим краще.

Порахуємо функції придатності для першої особини:



За прикладом обрахувати інші ФП та провести сортування популяції від особини з найменшою довжиною до особини з найбільшою. Отже, найперша особина в масиві популяції матиме найкоротший шлях. Однак це буде не самий оптимальний маршрут, а всього лише кращий з перших, випадково згенерованих.

Для пошуку найкращого маршруту запускається цикл, який буде працювати до тих пір, поки користувач його не зупинить. У цьому циклі буде генеруватися нова популяція на основі попередньої, потім знову буде проводитися сортування, і т. д. Для генерації нової популяції в циклі проводиться схрещування двох особин, що стоять поруч, взятих з популяції, сформованої на попередній ітерації, потім наступних двох, і так до тих пір, поки кількість нових особин залишається менше заданого у вхідних параметрах відсотка нових особин. При схрещуванні двох особин виходять також дві особини. Всі нові особини записуються в кінець нового масиву популяції, який є копією поточного. Отже, особи з найбільшою довжиною шляху замінюються новими особинами.

1	3	2	4	5	6	$\Phi П(1) = 63$
1	4	5	3	2	6	$\Phi П(2) = 93$
6	3	2	5	4	1	$\Phi П(3) = 95$
2	1	6	5	3	4	$\Phi П(4) = 109$

Маємо найкращі результати зверху, а для двох останніх особин застосуємо алгоритм схрещення, який називається

«частково відображуване схрещування». Суть цього алгоритму полягає в такому:

Нехай є дві батьківські хромосоми $ch\ 1,1 = \{ 6,3,2,5,4,1 \}$ та $ch\ 2,1 = \{ 2,1,6,5,3,4 \}$.

Випадковим чином знаходимо точки розриву та вважатимемо, що розрив стався так:

$ch\ 1,1 = \{ 6,3|2,5|4,1 \}$

$ch\ 2,1 = \{ 2,1|6,5|3,4 \}$.

При формуванні нащадків $ch\ 2,1$ та $ch\ 2,2$ спочатку проводимо обмін частин, що знаходяться між точками розриву:

$ch\ 2,1 = \{ x,x|6,5|x,x \}$

$ch\ 2,2 = \{ x,x|2,5|x,x \}$ – тобто для нащадка $ch\ 2,2$ беремо частину, що знаходиться між точками розриву, з $ch\ 1,1$.

Далі розставляємо позиції, що залишилися, від відповідних хромосом зліва направо до виникнення конфлікту (якщо номери повторюються у вже сформованій частині хромосоми). Якщо номер повторюється, то ставимо не його, а номер із сусідньої хромосоми в тій позиції, в якій був знайдений конфлікт в поточній хромосомі. Так триває до повного розв'язання конфліктів. Отримуємо нащадків:

$ch\ 2,1 = \{ 2,3|6,5|4,1 \}$

$ch\ 2,2 = \{ 6,1|2,5|3,4 \}$

Тоді нова популяція виглядатиме так:

1 3 2 4 5 6

1 4 5 3 2 6

2 3 6 5 4 1

6 1 2 5 3 4

Для кожного нащадка необхідно обрахувати ФП.

Однак якщо проводити схрещування двох ідентичних особин, то виявиться, що їх нащадки виявляться ідентичними своїм батькам. У підсумку в популяції буде вже чотири ідентичні особини, а оскільки наступна популяція формується на підставі попередньої, то рано чи пізно вся нова популяція складатиметься з ідентичних особин. Щоб цього уникнути, після створення двох нових особин проводиться перевірка на їх ідентичність один одному, і якщо виявляється, що вони ідентичні, то виконується мутація кожної особини. Мутація являє собою перестановку двох випадково обраних позицій у кожній особини.

Для наочності студенту необхідно навести блок-схему генетичного алгоритму.

Критерій зупинки генетичного алгоритму. У роботі необхідно знайти найкоротший шлях проходження локомотива за вагонами з відвідуванням всіх контрольних точок. Абсолютно правильну відповідь буде отримано, якщо перебрати всі варіанти, а їх дуже багато навіть для шести точок. Тому генетичний алгоритм шукає не правильне рішення, а оптимальне, виходячи з умов, які задає користувач.

Критерій зупинки – умова, за яким генетичний алгоритм зупиняє свою роботу.

У роботі можна вказати, наприклад, один з таких критеріїв зупинки:

- сумарний шлях менше 100;
- час роботи алгоритму не більше 1 год;
- число циклів алгоритму 10;
- впродовж 3 поколінь не з'являються особини краще тих, які були та ін.

ПРАКТИЧНА РОБОТА 2

Розв'язання задач із застосуванням нейронної мережі

Теоретичний матеріал для виконання завдання

Штучні нейронні мережі (ШНМ, англ. artificial neural networks, ANN), або конективістські системи (англ. connectionist systems) – це обчислювальні системи, натхнені біологічними нейронними мережами, що складають мозок тварин. Такі системи навчаються задач (поступово покращують свою продуктивність на них), розглядаючи приклади, загалом без спеціального програмування під задачу [4].

ШНМ ґрунтується на сукупності з'єднаних вузлів, що називають штучними нейронами (аналогічно до біологічних нейронів у головному мозку тварин). Кожне з'єднання (аналогічне синапсові) між штучними нейронами може передавати сигнал від одного до іншого. Штучний нейрон, що

отримує сигнал, може обробляти його й потім сигналізувати штучним нейронам, приєднаним до нього [5].

Нейрон – базовий елемент нейронних мереж, має входи, забезпечені вагами, зміщення, підсумовуючий елемент і вихідну активаційну функцію. Нейрон виконує функцію адаптивного суматора зі змінними входними ваговими коефіцієнтами, сумарний вихідний сигнал якого піддається лінійній або нелінійній обробці, утворюючи підсумковий вихідний сигнал.

Функція активації нейрону – функція лінійного або нелінійного перетворення стану нейрона. Ця функція перетворює суму зважених входних сигналів нейрона в його вихід – сигнал.

Архітектура – опис числа шарів в нейронній мережі, передавальних (активаційних) функцій кожного шару, кількості нейронів в кожному шарі і зв'язків між шарами.

Шар – множина нейронів (вузлів), що мають спільні входні або вихідні сигнали.

Вхідний шар – шар нейронів (вузлів), утворений сукупністю елементів, на входи яких подаються входні дані нейронної мережі.

Прихований шар – шар, що містить нейрони, на які не надходять входні дані і з яких не зчитуються вихідні дані нейронної мережі.

Вихідний шар – шар нейронів (вузлів), вихідні сигнали яких утворюють вихідний сигнал нейронної мережі.

Нейронна мережа – структура з'єднаних між собою нейронів, яка характеризується топологією, властивостями вузлів, а також правилами навчання або тренування для отримання бажаного вихідного сигналу.

ШНМ є системою з'єднаних і взаємодіючих між собою простих процесорів (штучних нейронів). Такі процесори зазвичай досить прості (особливо порівняно з процесорами, використовуваними в персональних комп'ютерах). Кожен процесор подібної мережі має справу тільки з сигналами, які він періодично отримує, і сигналами, які він періодично посиляє іншим процесорам. І, тим не менше, будучи з'єднаними в досить велику мережу з керованим взаємодією, такі окремо прості процесори разом здатні виконувати досить складні завдання.

Працює штучна нейронна мережа так: на входи нейронів надходять сигнали, які підсумовуються. При цьому враховується синаптична маса, тобто значущість кожного з входів. Далі, вхідні сигнали одних нейронів надходять на входи інших нейронів. Маса кожного такого зв'язку може бути позитивною (збуджуючі зв'язки) або негативною (гальмівні зв'язки). Вони визначають обчислення нейронної мережі, а отже, її пам'ять та поведінку [4, 5].

Але для того, щоб штучна нейронна мережа виконувала складні завдання, її необхідно навчити.

Штучні нейронні мережі не програмуються в звичайному розумінні цього слова, вони навчаються. Можливість навчання – одна з головних переваг нейронних мереж перед традиційними алгоритмами. Технічно, навчання полягає в знаходженні коефіцієнтів зв'язків між нейронами. У процесі навчання нейронна мережа здатна виявляти складні залежності між вхідними даними й вихідними, а також здійснювати узагальнення [6].

Для процесу навчання необхідно мати модель зовнішнього середовища, у якій функціонує нейронна мережа, – потрібну для вирішення задачі інформацію. По-друге, необхідно визначити, як модифікувати вагові параметри мережі.

Існують три загальні парадигми навчання: «з вчителем», «без вчителя» (самонавчання) та змішана. У першому випадку нейронна мережа має у своєму розпорядженні правильні відповіді (виходи мережі) на кожен вхідний приклад. Ваги налаштовуються так, щоб мережа виробляла відповіді найбільш близькі до відомих правильних відповідей. Навчання без вчителя не вимагає знання правильних відповідей на кожен приклад навчальної вибірки. У цьому випадку розкривається внутрішня структура даних та кореляція між зразками в навчальній множині, що дає змогу розподілити зразки по категоріях. При змішаному навчанні частина ваг визначається за допомогою навчання зі вчителем, у той час, як інша визначається за допомогою самонавчання.

Типи нейронних мереж:

- 1 одношарові з прямими зв'язками;
- 2 багатшарові з прямими зв'язками;
- 3 одношарові зі зворотними зв'язками (рекурентні);
- 4 багатшарові рекурентні.

Галузі застосування нейронних мереж

Розпізнавання образів та класифікація. Як образи можуть виступати різні за своєю природою об'єкти: символи тексту, зображення, зразки звуків і т. д. При навчанні мережі пропонуються різні зразки образів із зазначенням того, до якого класу вони відносяться. Коли мережі пред'являється якийсь образ, на одному з її виходів має з'явитися ознака того, що образ належить цьому класу. У той же час на інших виходах має бути ознака того, що образ до даного класу не належить.

Прийняття рішень та управління. Це завдання близьке до задачі класифікації. Класифікації підлягають ситуації, характеристики яких надходять на вхід нейронної мережі. На виході мережі має з'явитися ознака рішення, яке вона прийняла.

Кластеризація. Під кластеризацією розуміється розбиття множини вхідних сигналів на класи, при тому, що ні кількість, ні ознаки класів заздалегідь не відомі. Після навчання така мережа здатна визначати, до якого класу належить вхідний сигнал.

Прогнозування. Здібності нейронної мережі до прогнозування безпосередньо впливають з її здатності до узагальнення та виділення прихованих залежностей між вхідними та вихідними даними. Після навчання мережа здатна передбачити майбутнє значення якоїсь послідовності на основі декількох попередніх значень або якихось існуючих зараз чинників.

Стиснення даних і асоціативна пам'ять. Здатність нейромереж до виявлення взаємозв'язків між різними параметрами дає можливість висловити дані великої розмірності більш компактно, якщо дані тісно взаємопов'язані між собою. Зворотній процес – відновлення вихідного набору даних з частини інформації – називається асоціативною пам'яттю [6].

ЗАДАЧА 2

Розрахунок щодо симуляції виводу багат шарової повнозв'язної нейронної мережі

Постанова задачі. Необхідно представити архітектуру повнозв'язної тришарової нейронної мережі та обчислити її вихідний сигнал згідно з вихідними даними.

На рисунку 5 наведена повнозв'язна тришарова нейронна мережа. Мережа має три входи x_1 , x_2 та x_3 , сигнал кожного входу потрапляє до кожного нейрону першого шару. Кожен нейрон першого шару сполучається з кожним нейроном другого шару. Вихідний сигнал y мережі формується за допомогою третього шару, який містить лише один нейрон.

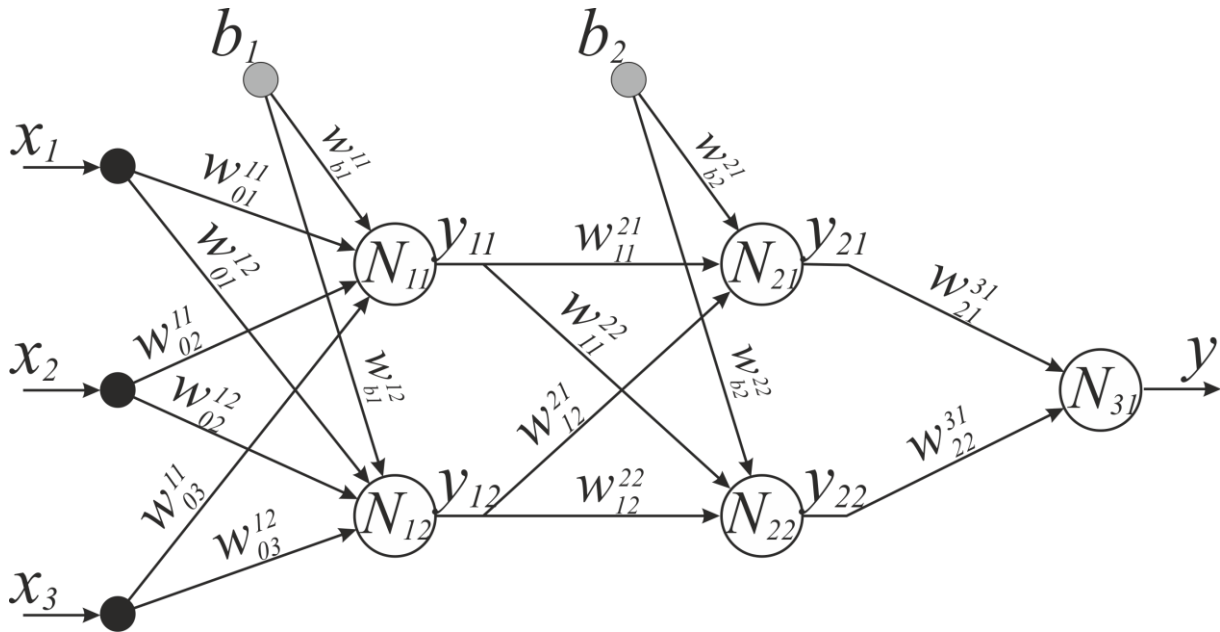


Рисунок 5 – Архітектура тришарової нейронної мережі прямого розповсюдження сигналу

Нейрони першого та другого шарів мережі мають логістичну функцію активації, яку можна обчислити за формулою:

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

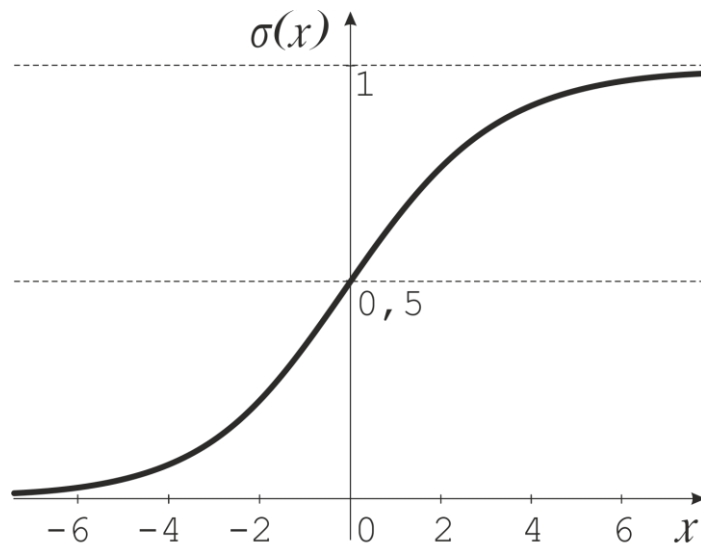


Рисунок 6 – Логістична функція активації

Нейрон третього (вихідного) шару має функцію активації типу softplus, яку можна обчислити за формулою:

$$\varphi(x) = \ln(1 + e^x).$$

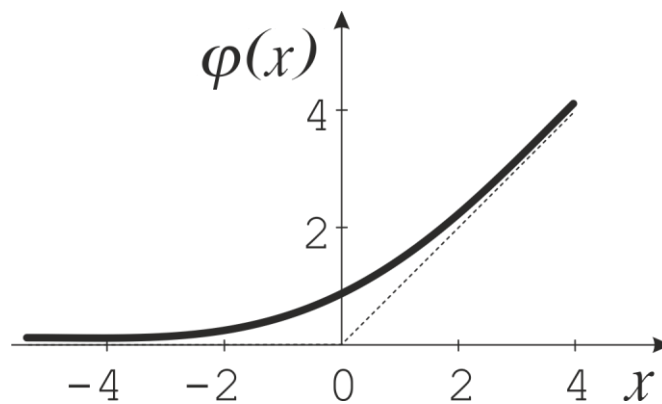


Рисунок 7 – Функція активації типу softplus

Розрахунки наведено для вихідних даних:

1 Значення вхідних сигналів $x_1 = 5$, $x_2 = 2$, $x_3 = 1$ (здобувач має вибрати свій варіант відповідно до даних, наведених у таблиці 3).

Таблиця 3 – Дані щодо значень вхідних сигналів для розрахунку за варіантом

Передостання цифра залікової книжки	0	1	2	3	4	5	6	7	8	9
x_1	2	3	1	5	9	7	4	3	4	1
x_2	5	4	6	2	-2	3	8	1	2	5
x_3	-1	8	3	1	3	4	-2	2	3	1

2 Ваги вхідних сигналів нейронів першого шару $w_{01}^{11} = -0,5$, $w_{02}^{11} = 0,6$, $w_{03}^{11} = -0,6$, $w_{01}^{12} = 0,8$, $w_{02}^{12} = 0,9$, $w_{03}^{12} = 0,5$; ваги вхідних сигналів нейронів другого шару $w_{11}^{21} = 2,5$, $w_{12}^{21} = -3$, $w_{11}^{22} = 0,7$, $w_{12}^{22} = 0,9$, ваги вхідних сигналів нейрону третього шару $w_{21}^{31} = 0,5$, $w_{22}^{31} = 1,7$, (здобувач має вибрати значення відповідно до даних, наведених у таблиці 4).

Таблиця 4 – Вихідні дані по кількості нейронів 1-го та 2-го шарів та їх ваги

Остання цифра залікової книжки	0	1	2	3	4
1	2	3	4	5	6
кількість нейронів 1-го шару	4	3	3	4	5
кількість нейронів 2-го шару	3	4	3	4	2

Продовження таблиці 4

1	2	3	4	5	6
ваги нейронів 1-го шару	$w_{01}^{11} = 0,5$ $w_{02}^{11} = -0,9$ $w_{03}^{11} = 1,5$ $w_{01}^{12} = 0,7$ $w_{02}^{12} = 1,9$ $w_{03}^{12} = -0,4$ $w_{01}^{13} = -0,2$ $w_{02}^{13} = 1,1$ $w_{03}^{13} = 1,3$ $w_{01}^{14} = 2,1$ $w_{02}^{14} = 1,6$ $w_{03}^{14} = -0,8$	$w_{01}^{11} = 1$ $w_{02}^{11} = -0,9$ $w_{03}^{11} = 1,5$ $w_{01}^{12} = 1,1$ $w_{02}^{12} = 0,3$ $w_{03}^{12} = 0,2$ $w_{01}^{13} = 0,1$ $w_{02}^{13} = 2,5$ $w_{03}^{13} = 1,7$	$w_{01}^{11} = 2$ $w_{02}^{11} = 0,8$ $w_{03}^{11} = 0,7$ $w_{01}^{12} = 0,6$ $w_{02}^{12} = 0,3$ $w_{03}^{12} = 1,8$ $w_{01}^{13} = -0,5$ $w_{02}^{13} = -0,5$ $w_{03}^{13} = 2,8$	$w_{01}^{11} = 1,5$ $w_{02}^{11} = 1,1$ $w_{03}^{11} = -0,1$ $w_{01}^{12} = 0,8$ $w_{02}^{12} = 0,8$ $w_{03}^{12} = 0,7$ $w_{01}^{13} = -0,9$ $w_{02}^{13} = -0,4$ $w_{03}^{13} = 0,4$ $w_{01}^{14} = 1,5$ $w_{02}^{14} = 1,7$ $w_{03}^{14} = 1,8$	$w_{01}^{11} = -0,7$ $w_{02}^{11} = 1,2$ $w_{03}^{11} = 1,4$ $w_{01}^{12} = 3,5$ $w_{02}^{12} = 0,6$ $w_{03}^{12} = 0,7$ $w_{01}^{13} = 0,9$ $w_{02}^{13} = 0,5$ $w_{03}^{13} = 1,9$ $w_{01}^{14} = 1,1$ $w_{02}^{14} = 0,6$ $w_{03}^{14} = -0,3$ $w_{01}^{15} = 0,9$ $w_{02}^{15} = -0,9$ $w_{03}^{15} = 0,7$
ваги нейронів 2-го шару	$w_{11}^{21} = 1,1$ $w_{12}^{21} = 0,7$ $w_{13}^{21} = 2,5$ $w_{14}^{21} = 1,5$ $w_{11}^{22} = 0,8$ $w_{12}^{22} = 0,7$ $w_{13}^{22} = 0,8$ $w_{14}^{22} = 0,3$ $w_{11}^{23} = -0,8$ $w_{12}^{23} = -1,8$ $w_{13}^{23} = 0,7$ $w_{14}^{23} = -0,1$	$w_{11}^{21} = 0,8$ $w_{12}^{21} = 0,4$ $w_{13}^{21} = 0,9$ $w_{11}^{22} = 1,2$ $w_{12}^{22} = -0,6$ $w_{13}^{22} = 1,6$ $w_{11}^{23} = 1,1$ $w_{12}^{23} = 2,1$ $w_{13}^{23} = 1,5$ $w_{11}^{24} = 0,8$ $w_{12}^{24} = 1,3$ $w_{13}^{24} = 1,7$	$w_{11}^{21} = 0,7$ $w_{12}^{21} = 0,1$ $w_{13}^{21} = -0,7$ $w_{11}^{22} = 1,5$ $w_{12}^{22} = 1,2$ $w_{13}^{22} = 1,7$ $w_{11}^{23} = 2,4$ $w_{12}^{23} = 1,9$ $w_{13}^{23} = 0,9$	$w_{11}^{21} = -0,7$ $w_{12}^{21} = 1,1$ $w_{13}^{21} = 2,1$ $w_{14}^{21} = 2,3$ $w_{11}^{22} = 1,4$ $w_{12}^{22} = 2,8$ $w_{13}^{22} = 3,1$ $w_{14}^{22} = -0,1$ $w_{11}^{23} = 1,8$ $w_{12}^{23} = 1,7$ $w_{13}^{23} = 0,7$ $w_{14}^{23} = 0,4$ $w_{11}^{24} = 0,8$ $w_{12}^{24} = 1,5$ $w_{13}^{24} = -0,6$ $w_{14}^{24} = -0,9$	$w_{11}^{21} = -0,4$ $w_{12}^{21} = 1,9$ $w_{13}^{21} = 1,1$ $w_{14}^{21} = -0,9$ $w_{15}^{21} = 1,4$ $w_{11}^{22} = 1,8$ $w_{12}^{22} = 3,3$ $w_{13}^{22} = 1,3$ $w_{14}^{22} = -0,5$ $w_{15}^{22} = 0,8$
ваги нейрону 3-го шару	$w_{21}^{31} = 0,4$ $w_{22}^{31} = 1,8$ $w_{33}^{31} = 3,6$	$w_{21}^{31} = 0,5$ $w_{22}^{31} = 2,5$ $w_{23}^{31} = 1,3$ $w_{24}^{31} = -0,1$	$w_{21}^{31} = -0,8$ $w_{22}^{31} = 2,5$ $w_{23}^{31} = 1,6$	$w_{21}^{31} = 0,6$ $w_{22}^{31} = 0,8$ $w_{23}^{31} = 0,8$ $w_{24}^{31} = 1,3$	$w_{21}^{31} = 1,6$ $w_{22}^{31} = -0,7$

Продовження таблиці 4

Остання цифра залікової книжки	5	6	7	8	9
1	2	3	4	5	6
кількість нейронів 1-го шару	4	2	2	2	3
кількість нейронів 2-го шару	2	4	5	3	2
ваги нейронів 1-го шару	$w_{01}^{11} = -0,3$ $w_{02}^{11} = 1,5$ $w_{03}^{11} = -0,2$ $w_{01}^{12} = 3,6$ $w_{02}^{12} = 1,2$ $w_{03}^{12} = 1,9$ $w_{01}^{13} = 0,1$ $w_{02}^{13} = 0,6$ $w_{03}^{13} = 0,7$ $w_{01}^{14} = 0,3$ $w_{02}^{14} = 0,8$ $w_{03}^{14} = -0,6$	$w_{01}^{11} = 2,4$ $w_{02}^{11} = -0,9$ $w_{03}^{11} = 1,8$ $w_{01}^{12} = 0,5$ $w_{02}^{12} = 0,8$ $w_{03}^{12} = 1,7$	$w_{01}^{11} = 3,1$ $w_{02}^{11} = 0,7$ $w_{03}^{11} = 0,3$ $w_{01}^{12} = 0,7$ $w_{02}^{12} = 1,9$ $w_{03}^{12} = 1,4$	$w_{01}^{11} = 1,4$ $w_{02}^{11} = 1,1$ $w_{03}^{11} = 3,1$ $w_{01}^{12} = -0,7$ $w_{02}^{12} = 0,4$ $w_{03}^{12} = 0,6$	$w_{01}^{11} = 3,2$ $w_{02}^{11} = 1,5$ $w_{03}^{11} = 2,1$ $w_{01}^{12} = -0,9$ $w_{02}^{12} = 1,2$ $w_{03}^{12} = 0,6$ $w_{01}^{13} = 0,1$ $w_{02}^{13} = 4,1$ $w_{03}^{13} = 0,6$
ваги нейронів 2-го шару	$w_{11}^{21} = 2,1$ $w_{12}^{21} = 0,1$ $w_{13}^{21} = 0,8$ $w_{14}^{21} = 0,4$ $w_{11}^{22} = 0,7$ $w_{12}^{22} = 0,8$ $w_{13}^{22} = 1,6$ $w_{14}^{22} = 1,4$	$w_{11}^{21} = 1,3$ $w_{12}^{21} = 1,2$ $w_{11}^{22} = 0,8$ $w_{12}^{22} = 0,7$ $w_{11}^{23} = 0,3$ $w_{12}^{23} = 1,3$ $w_{11}^{24} = 1,1$ $w_{12}^{24} = 0,6$	$w_{11}^{21} = 1,6$ $w_{12}^{21} = 0,1$ $w_{11}^{22} = 0,5$ $w_{12}^{22} = 1,4$ $w_{11}^{23} = 1,2$ $w_{12}^{23} = 0,8$ $w_{11}^{24} = 1,7$ $w_{12}^{24} = 0,3$ $w_{11}^{25} = 0,9$ $w_{12}^{25} = 1,8$	$w_{11}^{21} = 1,1$ $w_{12}^{21} = 1,4$ $w_{11}^{22} = 0,6$ $w_{12}^{22} = 1,8$ $w_{11}^{23} = 0,3$ $w_{12}^{23} = 0,9$	$w_{11}^{21} = 1,2$ $w_{12}^{21} = 0,8$ $w_{13}^{21} = 1,1$ $w_{11}^{22} = 0,8$ $w_{12}^{22} = -0,6$ $w_{13}^{22} = 0,9$

Продовження таблиці 4

1	2	3	4	5	6
ваги нейрону 3-го шару	$w_{21}^{31} = 1,8$ $w_{22}^{31} = 0,7$	$w_{21}^{31} = 2,5$ $w_{22}^{31} = 3,1$ $w_{23}^{31} = 0,3$ $w_{24}^{31} = 0,4$	$w_{21}^{31} = 0,8$ $w_{22}^{31} = 1,2$ $w_{23}^{31} = 0,8$ $w_{24}^{31} = 1,9$ $w_{25}^{31} = -0,8$	$w_{21}^{31} = 2,1$ $w_{22}^{31} = -0,9$ $w_{23}^{31} = 1,9$	$w_{21}^{31} = 2,9$ $w_{22}^{31} = 1,5$

3 Величина сигналів зміщення для нейронів першого та другого шарів завжди дорівнює одиниці $b_1 = 1$, $b_2 = 1$ відповідно. Ваги сигналів зміщення для нейронів першого шару становлять $w_{b1}^{11} = 1,2$, $w_{b1}^{12} = -6$, для нейронів другого шару $w_{b2}^{21} = 0,9$, $w_{b2}^{22} = 1,6$ (здобувач має вибрати значення відповідно до даних, наведених у таблиці 5)

Таблиця 5 – Вихідні дані ваги сигналу зміщення нейронів

Передостан- ня цифра залікової книжки	0	1	2	3	4
1	2	3	4	5	6
ваги сигналу зміщення нейронів 1- го шару	$w_{b1}^{11} = -0,4$ $w_{b1}^{12} = 1,8$ $w_{b1}^{13} = 4,5$ $w_{b1}^{14} = 0,6$	$w_{b1}^{11} = 0,6$ $w_{b1}^{12} = 1,9$ $w_{b1}^{13} = 3,8$	$w_{b1}^{11} = 1,7$ $w_{b1}^{12} = 1,9$ $w_{b1}^{13} = 0,3$	$w_{b1}^{11} = 1,8$ $w_{b1}^{12} = 0,1$ $w_{b1}^{13} = 1,1$ $w_{b1}^{14} = 0,3$	$w_{b1}^{11} = 0,1$ $w_{b1}^{12} = 1,9$ $w_{b1}^{13} = 0,1$ $w_{b1}^{14} = 2,9$ $w_{b1}^{15} = 2,1$
ваги сигналу зміщення нейронів 2- го шару	$w_{b2}^{21} = 0,2$ $w_{b2}^{22} = 1,9$ $w_{b2}^{23} = 3,1$	$w_{b2}^{21} = 1,7$ $w_{b2}^{22} = 0,3$ $w_{b2}^{23} = 0,9$ $w_{b2}^{24} = -0,4$	$w_{b2}^{21} = 0,6$ $w_{b2}^{22} = 0,9$ $w_{b2}^{23} = 4,2$	$w_{b2}^{21} = 0,5$ $w_{b2}^{22} = 1,9$ $w_{b2}^{23} = 2,5$ $w_{b2}^{24} = -0,7$	$w_{b2}^{21} = 2,6$ $w_{b2}^{22} = 1,8$

Продовження таблиці 5

Передостан- ня цифра залікової книжки	5	6	7	8	9
ваги сигналу зміщення нейронів 1-го шару	$w_{b1}^{11} = 0,8$ $w_{b1}^{12} = 0,4$ $w_{b1}^{12} = 2,8$ $w_{b1}^{14} = 1,3$	$w_{b1}^{11} = 1,8$ $w_{b1}^{12} = 0,6$	$w_{b1}^{11} = 1,1$ $w_{b1}^{12} = 2,3$	$w_{b1}^{11} = -0,7$ $w_{b1}^{12} = 1,8$	$w_{b1}^{11} = 2,2$ $w_{b1}^{12} = 1,1$ $w_{b1}^{13} = 0,8$
ваги сигналу зміщення нейронів 2-го шару	$w_{b2}^{21} = 2,8$ $w_{b2}^{22} = 1,9$	$w_{b2}^{21} = 3,1$ $w_{b2}^{22} = -0,9$ $w_{b2}^{23} = 1,7$	$w_{b2}^{21} = -0,1$ $w_{b2}^{22} = 1,8$ $w_{b2}^{23} = 1,1$ $w_{b2}^{24} = 0,8$ $w_{b2}^{25} = 0,3$	$w_{b2}^{21} = 1,6$ $w_{b2}^{22} = 0,9$ $w_{b2}^{23} = 3,1$	$w_{b2}^{21} = 1,8$ $w_{b2}^{22} = 2,4$

Розрахуємо вихідні сигнали нейронів першого шару:

$$y_{11} = \sigma \left(\sum_{i=1}^3 (x_i w_{0,i}^{11}) + b_1 w_{b1}^{11} \right) =$$

$$= \frac{1}{1 + \exp \left(- (5 \cdot (-0,5) + 2 \cdot 0,6 + 1 \cdot (-0,6) + 1 \cdot 1,2) \right)} = 0,3318;$$

$$y_{12} = \sigma \left(\sum_{i=1}^3 (x_i w_{0,i}^{12}) + b_2 w_{b1}^{12} \right) =$$

$$= \frac{1}{1 + \exp \left(- (5 \cdot 0,8 + 2 \cdot 0,9 + 1 \cdot 0,5 + 1 \cdot (-6)) \right)} = 0,5744 \cdot$$

Розрахуємо вихідні сигнали нейронів другого шару:

$$y_{21} = \sigma \left(\sum_{i=1}^2 (y_{1,i} w_{1,i}^{21}) + b_2 w_{b2}^{21} \right) =$$

$$= \frac{1}{1 + \exp \left(- (0,3318 \cdot 2,5 + 0,5744 \cdot (-3) + 1 \cdot 0,9) \right)} = 0,5016;$$

$$y_{22} = \sigma \left(\sum_{i=1}^2 (y_{1,i} w_{1,i}^{22}) + b_2 w_{b2}^{22} \right) =$$

$$= \frac{1}{1 + \exp(-(0,3318 \cdot 0,7 + 0,5744 \cdot 0,9 + 1 \cdot 1,6))} = 0,9129.$$

Розрахуємо вихідний сигнал нейрону третього шару, тобто вихідний сигнал мережі:

$$y = \varphi \left(\sum_{i=1}^2 (y_{2,i} w_{2,i}^{31}) \right) =$$

$$= \ln(1 + \exp(0,5016 \cdot 0,5 + 0,9129 \cdot 1,7)) = 1,9553.$$

ПРАКТИЧНА РОБОТА 3

Розв'язання задач із застосуванням нечіткої логіки

Теоретичний матеріал для виконання завдання

Нечітка логіка широко використовується для управління технічними пристроями і процесами. Розробка та застосування систем нечіткого висновку містить у собі ряд етапів, реалізація яких виконується за допомогою розглянутих раніше основних положень нечіткої логіки.

Інформацією, яка надходить на вхід системи нечіткого висновку, є отримані з датчиків параметри. Ці параметри відповідають реальним змінам процесу управління. Інформація, яка формується на виході системи нечіткого висновку, відповідає вихідним змінним процесу управління [7].

Системи нечіткого висновку призначені для перетворення значень вхідних змінних процесу управління у вихідні змінні на основі використання нечітких правил. Для цього системи нечіткого висновку мають містити базу правил нечітких продукцій і реалізовувати нечіткий висновок на основі посилянь або умов, представлених у формі нечітких лінгвістичних висловлювань.

Для отримання висновків у системах нечіткого логічного рішення запропоновані декілька алгоритмів, які базуються на

розподілі процесу висновку на ряд послідовних етапів. Отже, можливо не тільки досягнути певної систематизації розглянутих понять нечіткої логіки, а й отримати деяку загальну схему, на базі якої можна створювати індивідуальні алгоритми нечіткого висновку [8].

Основними етапами нечіткого висновку є такі дії (рисунк 8):

- 1 формування бази правил систем нечіткого висновку;
 - 2 фазифікація вхідних змінних;
 - 3 агрегування підумов у нечітких правилах;
 - 4 активізація або композиція висновків у нечітких правилах.
- Акумуляування висновків нечітких правил продукцій.



Рисунок 8 – Діаграма дій за формуванням висновку

Формування бази правил систем нечіткого висновку

База правил систем нечіткого висновку призначена для формального наведення емпіричних знань або знань експертів у тій чи іншій проблемній сфері. У системах нечіткого висновку використовуються правила нечітких продукцій, у яких умови і висновки сформульовані в термінах нечітких лінгвістичних висловлювань.

У контексті нечіткої логіки під фазифікацією розуміють не тільки окремий етап виконання нечіткого висновку, але й власне процес або процедуру знаходження значень функцій належності нечітких множин (термів) на основі звичайних (не нечітких) вхідних даних. Фазифікацію ще називають вступом до нечіткості. Метою етапу фазифікації є встановлення відповідності між конкретним (звичайно – числовим) значенням окремої базової вхідної змінної (параметра) системи нечіткого висновку і значенням функції належності відповідного їй терму вхідної лінгвістичної змінної. Після завершення цього етапу для всіх вхідних змінних мають бути визначені конкретні значення функцій належності за кожним з лінгвістичних термів, які використовуються в підумовах бази правил системи нечіткого висновку.

Агрегування є процедурою визначення ступеня істинності умов за кожним із правил системи нечіткого висновку.

Активация в системах нечіткого висновку є процедурою або процесом знаходження ступеня істинності кожного з висновків нечітких правил.

У випадку формування бази правил системи нечіткого висновку задаються вагові коефіцієнти для кожного правила (за замовчуванням передбачається, якщо ваговий коефіцієнт не заданий явно, його значення на рівні 1).

Акумуляція або акумулювання в системах нечіткого висновку є процедурою або процесом знаходження функції належності для кожної з вихідних лінгвістичних змінних множин.

Мета акумуляції полягає в тому, щоб об'єднати або акумулювати всі ступені істинності висновків (підвисновків) для одержання функції належності кожної з вихідних змінних. Причина необхідності виконання цього етапу полягає в тому, що підвисновки, які відносяться до однієї і тієї ж вихідної лінгвістичної змінної, належать різним правилам системи нечіткого висновку [8].

Дефазифікація нечіткого логічного висновку

Дефазифікація (defuzzification) в системах нечіткого висновку є процедурою або процесом знаходження реального фізичного (не нечіткого) значення для кожної з вихідних лінгвістичних змінних множини $W = \{\omega_1, \omega_2, \dots, \omega_s\}$

Мета дефазифікації полягає в тому, щоб, використовуючи результати акумуляції всіх вихідних лінгвістичних змінних, одержати звичайне кількісне значення кожної з вихідних змінних, тобто сигнал управління або рішення щодо управління, яке може бути використане спеціальними пристроями, зовнішніми відносно системи нечіткого висновку.

У сучасних системах управління пристрої та механізми здатні сприймати традиційні команди у формі кількісних значень відповідних сигналів управління. Саме з цієї причини необхідно перетворити нечіткі множини в деякі конкретні значення змінних. Тому дефазифікацію називають також приведенням до чіткості.

Формально процедура дефазифікації виконується в такий спосіб. До початку цього етапу передбачаються відомими функції належності всіх вихідних лінгвістичних змінних у формі нечітких множин: C'_1, C'_2, \dots, C'_s , де s — загальна кількість вихідних лінгвістичних змінних у базі правил системи нечіткого висновку. Далі послідовно розглядаються кожна з вихідних лінгвістичних змінних $\omega_j \in W$ і нечітка множина, що відноситься до неї C'_j . Результат дефазифікації для вихідної лінгвістичної змінної ω_j визначається у вигляді кількісного значення $y_j \in R$, одержаного за однією з математичних формул.

Етап дефазифікації вважається закінченим, коли для кожної з вихідних лінгвістичних змінних будуть визначені підсумкові кількісні значення у формі деякого дійсного числа, тобто у вигляді y_1, y_2, \dots, y_s , де s — загальна кількість вихідних лінгвістичних змінних у базі правил системи нечіткого висновку.

Для виконання числових розрахунків на етапі дефазифікації можуть бути використані такі основні математичні методи дефазифікації: *центру ваги, центру ваги для одноточкових множин, центру площі, лівого модального значення, правого модального значення* [8].

Метод центру ваги

Центр ваги (англ. Cog, COG, Centre of Gravity), або центроїд площі розраховується за формулою [8]:

$$y = \frac{\int_{Min}^{Max} x \cdot \mu(x) dx}{\int_{Min}^{Max} \mu(x) dx}.$$

У формулі використовуються такі позначення:

y – результат дефазифікації;

x – змінна, що відповідає вихідній лінгвістичній змінній ω ; $\mu(x)$ — функція належності нечіткої множини, що відповідає вихідній змінній ω після етапу акумуляції;

Min і Max — ліва й права точки інтервалу носія нечіткої множини вихідної змінної ω .

У випадку дефазифікації методом центру ваги звичайне (не нечітке) значення вихідної змінної дорівнює абсцисі центру ваги плоскої фігури, обмеженої графіком кривої функції належності відповідної вихідної змінної.

Графічна інтерпретація процедури дефазифікації методом центру ваги функції належності вихідної лінгвістичної змінної зображено на рисунку 8.

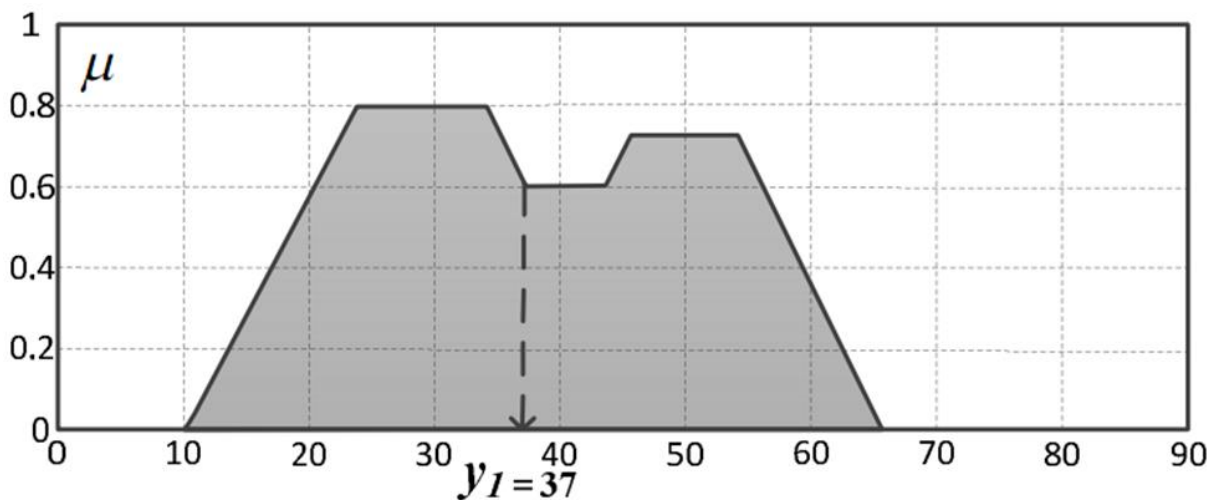


Рисунок 8 – Дефазифікація вихідної лінгвістичної змінної методом центру ваги

У цьому випадку $y_1 = 37^\circ$ (приблизне значення). Це значення є точкою рівноваги однорідної плоскої фігури, обмеженої ламаною кривою функції належності вихідної лінгвістичної змінної та віссю абсцис.

Метод центру ваги для одноточкових множин

Центр ваги (англ. COGS, Centre of Gravity for Singletons) для одноточкових множин розраховується за формулою [8]:

$$y = \frac{\sum_{i=1}^n x_i \cdot \mu(x_i)}{\sum_{i=1}^n \mu(x_i)},$$

де n – кількість одноточкових (одноелементних) нечітких множин, кожна з яких характеризує єдине значення розглянутої вихідної лінгвістичної змінної.

Графічна інтерпретація процедури дефазифікації методом центру ваги для одноточкових множин функції належності вихідної лінгвістичної змінної зображено на рисунку 9. У цьому випадку приблизно $y_1 = 41^\circ$.

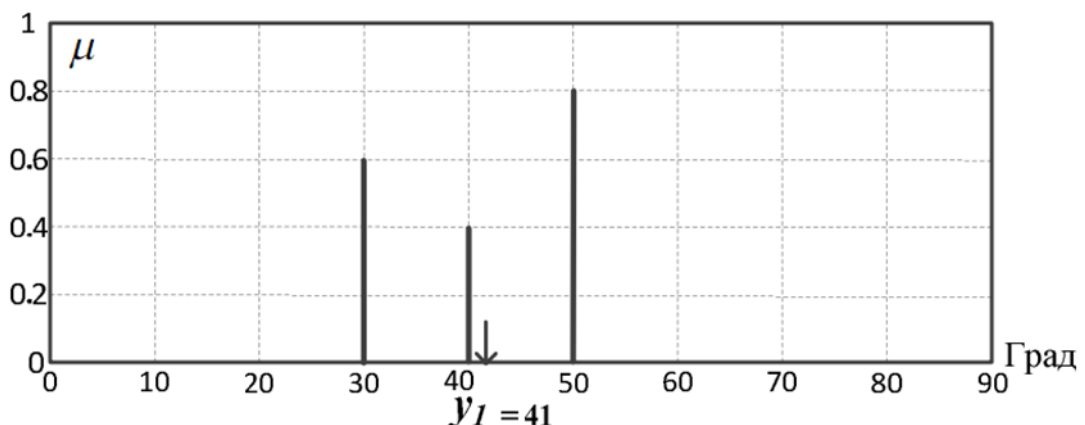


Рисунок 9 – Дефазифікація лінгвістичної змінної методом центру ваги для одноточкових множин

Метод центру площі

Центр площі (англ. CoA, COA, Centre of Area, Bisector of Area) розраховується, виходячи з рівняння, де значення u визначається з виразу [8]:

$$\int_{Min}^u \mu(x)dx = \int_u^{Max} \mu(x)dx .$$

Центр площі визначає точка на осі абсцис від перетину прямої паралельної осі ординат, яка ділить площу, обмежену графіком кривої функції належності відповідної вихідної змінної, на дві рівні частини, та осі абсцис. Іноді центр площі називають *бісектрисою площі*. Цей метод не може бути використаний у випадку одноточкових множин.

Графічна інтерпретація процедури дефазифікації методом центру площі функції належності вихідної лінгвістичної змінної зображена на рисунку 10. У цьому випадку $y_I=35^\circ$.

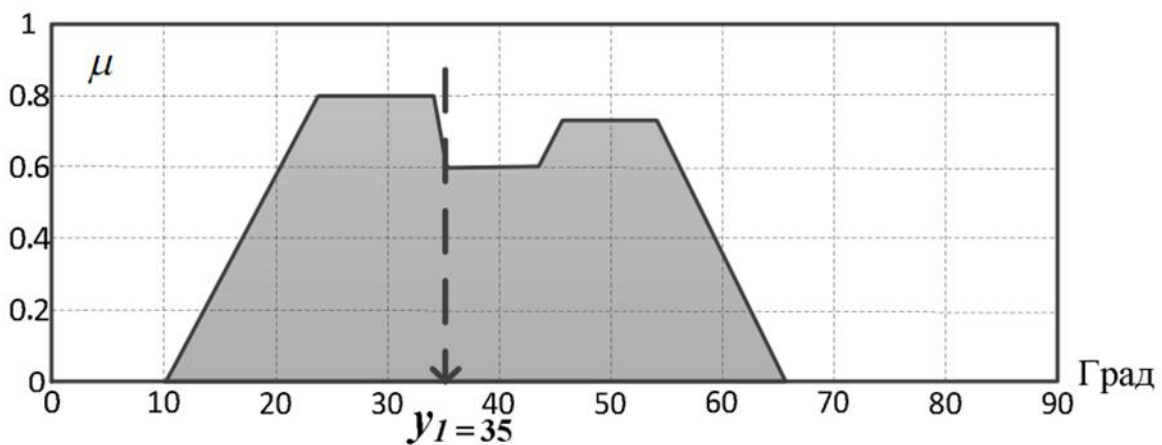


Рисунок 10 – Дефазифікація вихідної лінгвістичної змінної методом центру площі

Метод лівого модального значення

Ліве модальне значення (англ. LM, Left Most Maximum) розраховується за формулою [8]:

$$y = \min \{x_m\},$$

де x_m — модальне значення (мода) нечіткої множини, що відповідає вихідній змінній ω після акумуляції.

Тобто значення вихідної змінної визначається як мода нечіткої множини для відповідної вихідної змінної або найменша з мод (сама ліва), якщо нечітка множина має кілька модальних значень.

Графічна інтерпретація процедури дефазифікації методом лівого модального значення функції належності вихідної лінгвістичної змінної зображена на рисунку 11. У цьому випадку $y_1=24^\circ$.

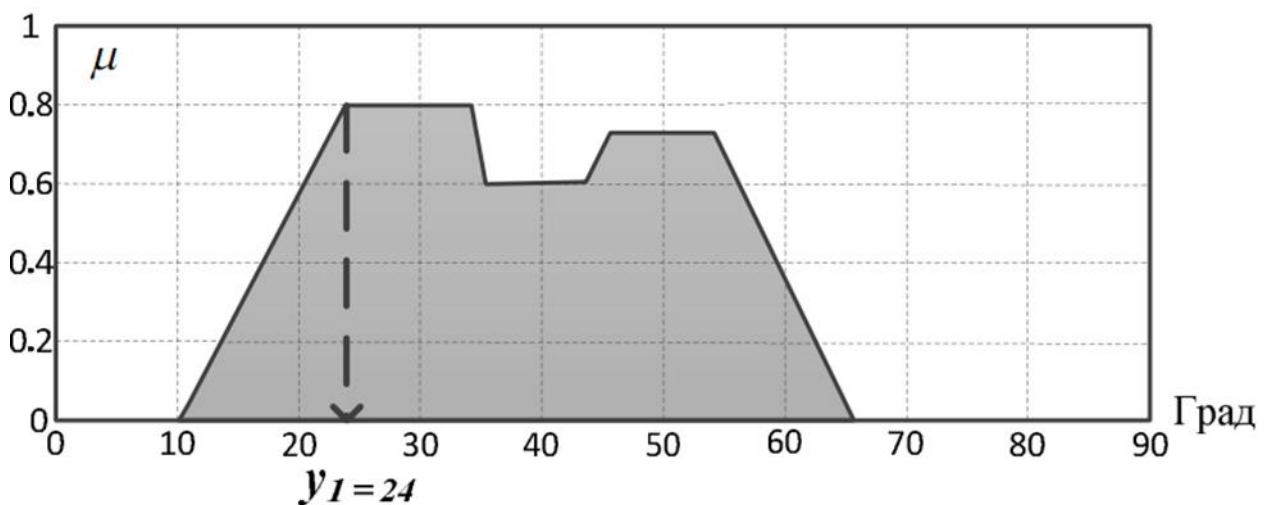


Рисунок 11 – Дефазифікація вихідної лінгвістичної змінної методом лівого модального значення

Метод правого модального значення

Праве модальне значення (англ. RM, Right Most Maximum) розраховується за формулою [8]:

$$y = \max \{x_m\},$$

де x_m — модальне значення (мода) нечіткої множини для вихідної змінної ω після акумуляції.

У цьому випадку значення вихідної змінної також визначається як мода нечіткої множини для відповідної вихідної змінної або найбільша з мод (сама крайня права), якщо нечітка множина має кілька модальних значень.

Графічна інтерпретація процедури дефазифікації функції належності вихідної лінгвістичної змінної зображена на рисунку 12. У цьому випадку $y_1=54^\circ$.

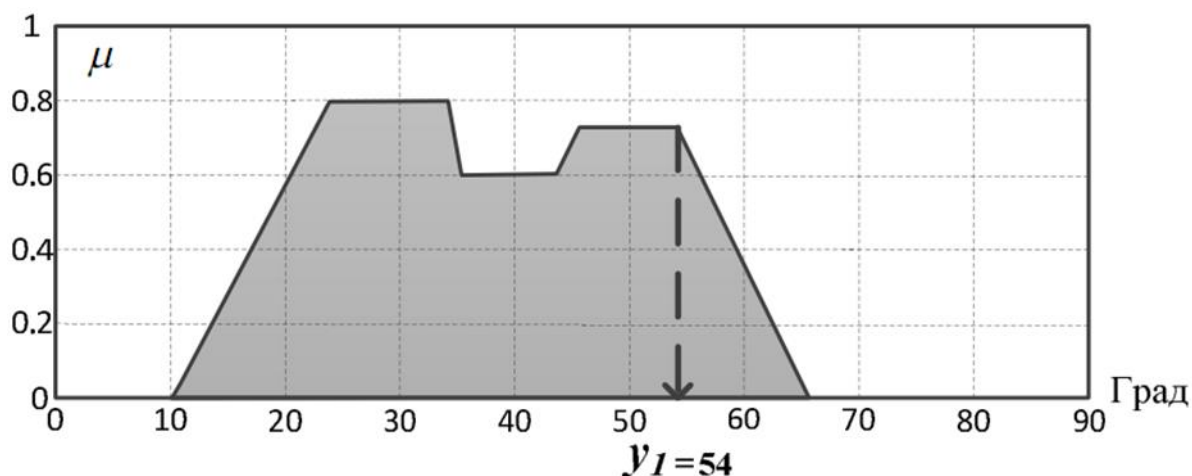


Рисунок 12 – Дефазифікація вихідної лінгвістичної змінної методом правого модального значення

ЗАДАЧА 3

Дефазифікація нечіткої змінної, представлені функцією приналежності

Постанова задачі. Необхідно подати функцію приналежності в графічному та аналітичному вигляді та здійснити операцію її дефазифікації за методом центроїду, результат якої подати також і в графічному вигляді згідно з варіантом завдання.

Нечітка змінна подана функцією належності до нечіткої множини трапецієвидного типу з параметрами $a=1$, $b=2$, $c=4$, $d=6$.

Функцію приналежності в графічному вигляді подано на рисунку 13.

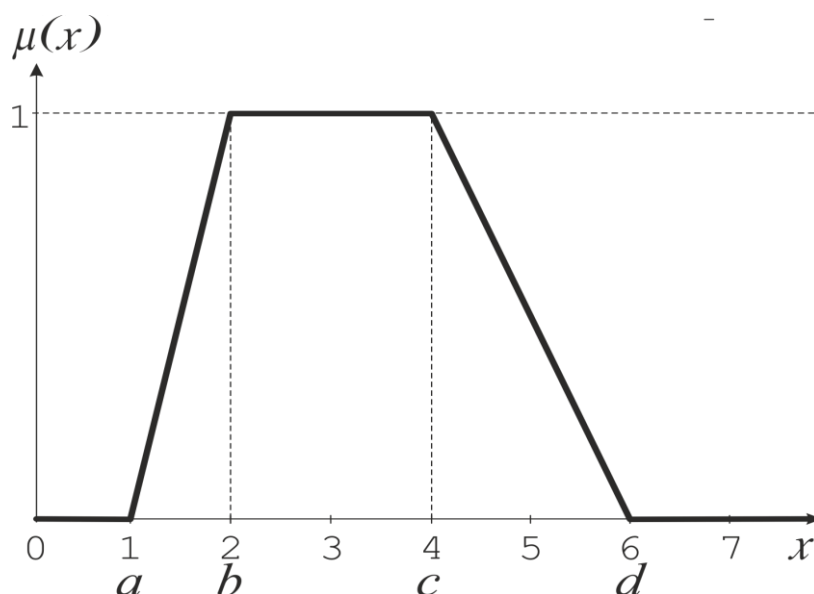


Рисунок 13 – Функція належності трапецієвидного типу

Аналitiчну залежність функції належності трапецієвидного типу в загальному вигляді можна записати:

$$\mu(x) = \begin{cases} 0 & , x \leq a \\ \frac{x-a}{b-a} & , a \leq x \leq b \\ 1 & , b \leq x \leq c \\ \frac{d-x}{d-c} & , c \leq x \leq d \\ 0 & , d \leq x \end{cases}$$

Отже, аналitiчну залежність даної функції належності можна подати у вигляді:

$$\mu(x) = \begin{cases} 0 & , x \leq 1 \\ \frac{x-1}{2-1} & , 1 \leq x \leq 2 \\ 1 & , 2 \leq x \leq 4 \\ \frac{6-x}{6-4} & , 4 \leq x \leq 6 \\ 0 & , 6 \leq x \end{cases} = \begin{cases} 0 & , x \leq 1 \\ x-1 & , 1 \leq x \leq 2 \\ 1 & , 2 \leq x \leq 4 \\ -0,5x+3 & , 4 \leq x \leq 6 \\ 0 & , 6 \leq x \end{cases}$$

Дефазифікація за методом центроїду являє знаходження абсциси центру тяжіння плоскої фігури, що обмежена віссю абсцис та функцією приналежності, тому загальна формула, яка відповідає даній операції, є такою:

$$x^* = \frac{\int \mu(x) x dx}{\int \mu(x) dx}.$$

Тоді для отриманої функції належності вона прийме такий вигляд:

$$x^* = \frac{\int_1^2 (x-1) x dx + \int_2^4 1 x dx + \int_4^6 (-0,5x+3) x dx}{\int_1^2 (x-1) dx + \int_2^4 1 dx + \int_4^6 (-0,5x+3) dx}.$$

Обчислимо визначені інтеграли у чисельнику та знаменнику та отримаємо кінцевий результат дефазифікації:

$$\begin{aligned} x^* &= \frac{\int_1^2 (x^2 - x) dx + \int_2^4 x dx + \int_4^6 (-0,5x^2 + 3x) dx}{\int_1^2 (x-1) dx + \int_2^4 1 dx + \int_4^6 (-0,5x+3) dx} = \\ &= \frac{\left(\frac{x^3}{3} - \frac{x^2}{2}\right)\Big|_1^2 + \frac{x^2}{2}\Big|_2^4 + \left(-\frac{x^3}{6} + \frac{3x^2}{2}\right)\Big|_4^6}{\left(\frac{x^2}{2} - x\right)\Big|_1^2 + x\Big|_2^4 + \left(-\frac{x^2}{4} + 3x\right)\Big|_4^6} = \\ &= \frac{\left(\frac{2^3}{3} - \frac{2^2}{2}\right) - \left(\frac{1^3}{3} - \frac{1^2}{2}\right) + \left(\frac{4^2}{2} - \frac{2^2}{2}\right) + \left(-\frac{6^3}{6} + \frac{3 \cdot 6^2}{2}\right) - \left(-\frac{4^3}{6} + \frac{3 \cdot 4^2}{2}\right)}{\left(\frac{2^2}{2} - 2\right) - \left(\frac{1^2}{2} - 1\right) + (4 - 2) + \left(-\frac{6^2}{4} + 3 \cdot 6\right) - \left(-\frac{4^2}{4} + 3 \cdot 4\right)} = \\ &= \frac{0,6667 + 0,1667 + 6 + 18 - 13,3333}{0 + 0,5 + 2 + 9 - 8} = \frac{11,5}{3,5} = 3,286 \end{aligned}$$

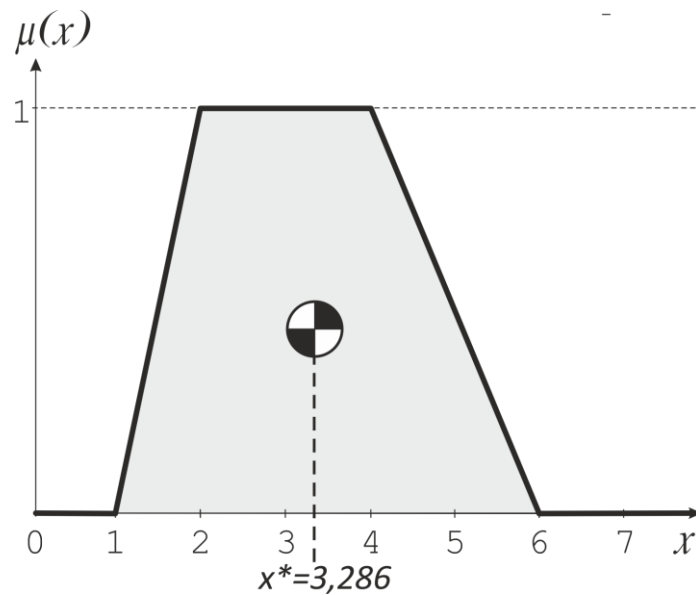


Рисунок 14 – Дефазифікація нечіткої множини

Дані для виконання індивідуального завдання наведено в таблиці 6.

Таблиця 6 – Вихідні дані для виконання розрахунку за варіантом

Остання цифра залікової книжки	0	1	2	3	4	5	6	7	8	9
a	1	2	4	2	1	2	3	3	1	2
b	5	4	7	3	6	6	6	5	7	5
Передостання цифра залікової книжки	0	1	2	3	4	5	6	7	8	9
c	7	8	9	10	8	7	11	10	12	8
d	11	15	12	11	13	12	15	14	16	16

ПРАКТИЧНА РОБОТА 4

Опис технологічних процесів із застосуванням UML-діаграм

Теоретичний матеріал для виконання завдання

Unified Modeling Language (англ. UML) – уніфікована мова моделювання. Якщо деталізувати назву, то modeling передбачає створення моделі, що описує об'єкт. Unified (універсальний, єдиний) – підходить для широкого класу проєктованих програмних систем, різних областей додатків, типів організацій, рівнів компетентності, розмірів проєктів [9].

UML описує об'єкт в єдиному заданому синтаксисі, тому де б не було намальовано діаграму, її правила будуть зрозумілими для всіх, хто знайомий з цією графічною мовою.

Сфера застосування UML:

1) проєктування. UML-діаграми використовують при моделюванні архітектури великих проєктів, це дає змогу об'єднати як великі, так і дрібні деталі і накреслити схему програми;

2) реверс-інжиніринг – створення UML-моделі з існуючого коду додатку, зворотна побудова. Може застосовуватися, наприклад, на проєктах підтримки, де є написаний код, але документація неповна або відсутня;

3) опис різних технологічних процесів, у тому числі таких, що відбуваються на залізничному транспорті. З моделей можна витягувати текстову інформацію і генерувати тексти – документувати. Текст і графіка будуть доповнювати один одного. Це особливо корисно при створенні та корегуванні технологічних процесів роботи різних підрозділів залізничного транспорту.

Види UML-діаграм

У мові UML є 12 типів діаграм, зокрема:

- структурні діаграми (діаграма класів, діаграма компонентів, діаграма композитної структури);
- діаграми кооперації (діаграма розгортання, діаграма об'єктів, діаграма пакетів, діаграма профілей);
- поведінкові діаграми (діаграма діяльності, діаграма станів, діаграма прецедентів);

- діаграми взаємодії (діаграма комунікації, діаграма огляду взаємодій, діаграма послідовності, діаграма синхронізації).

Не всі діаграми є широко застосовуваними, деякі з видів діаграм специфічні для певних систем і додатків.

Найбільш доступні UML-діаграми:

- діаграма прецедентів (англ. Use-case diagram);
- діаграма класів (англ. Class diagram);
- діаграма активностей (англ. Activity diagram);
- діаграма послідовності (англ. Sequence diagram);
- діаграма розгортання (англ. Deployment diagram);
- діаграма співробітництва (англ. Collaboration diagram);
- діаграма об'єктів (англ. Object diagram);
- діаграма станів (англ. Statechart diagram).

Розглянемо на прикладі операції придбання проїзного документа на залізничному транспорті застосування різних видів UML-діаграм.

Діаграма прецедентів (англ. Use-case diagram)

Діаграма прецедентів використовує 2 основних елементи [9]:

1) учасник (англ. Actor) – множина логічно пов'язаних ролей, виконуваних при взаємодії з прецедентами або сутностями (система, підсистема або клас). Учасником може бути людина, роль людини в системі чи інша система, підсистема або клас, які являють щось поза сутністю;

2) прецедент (англ. Use case) – опис окремого аспекту поведінки системи з точки зору користувача. Прецедент не показує, «як» досягається певний результат, а тільки «що» саме виконується.

Розглянемо приклад, в якому є 3 учасники: пасажир, касир та залізниця.

Прецеденти для пасажирів: робить запити щодо наявності проїзного документа (квитка), замовляє (бронює) квиток, сплачує за квиток.

Роль квиткового касира: друк та видача квитків, консультації і заповнення квитків, відміна квитків.

Роль залізниці: робота системи з бронювання квитків, повернення коштів за відмінений квиток, заповнення різних форм звітності та інші операції.

Діаграму, що зображує процес взаємодії, наведено на рисунку 15.

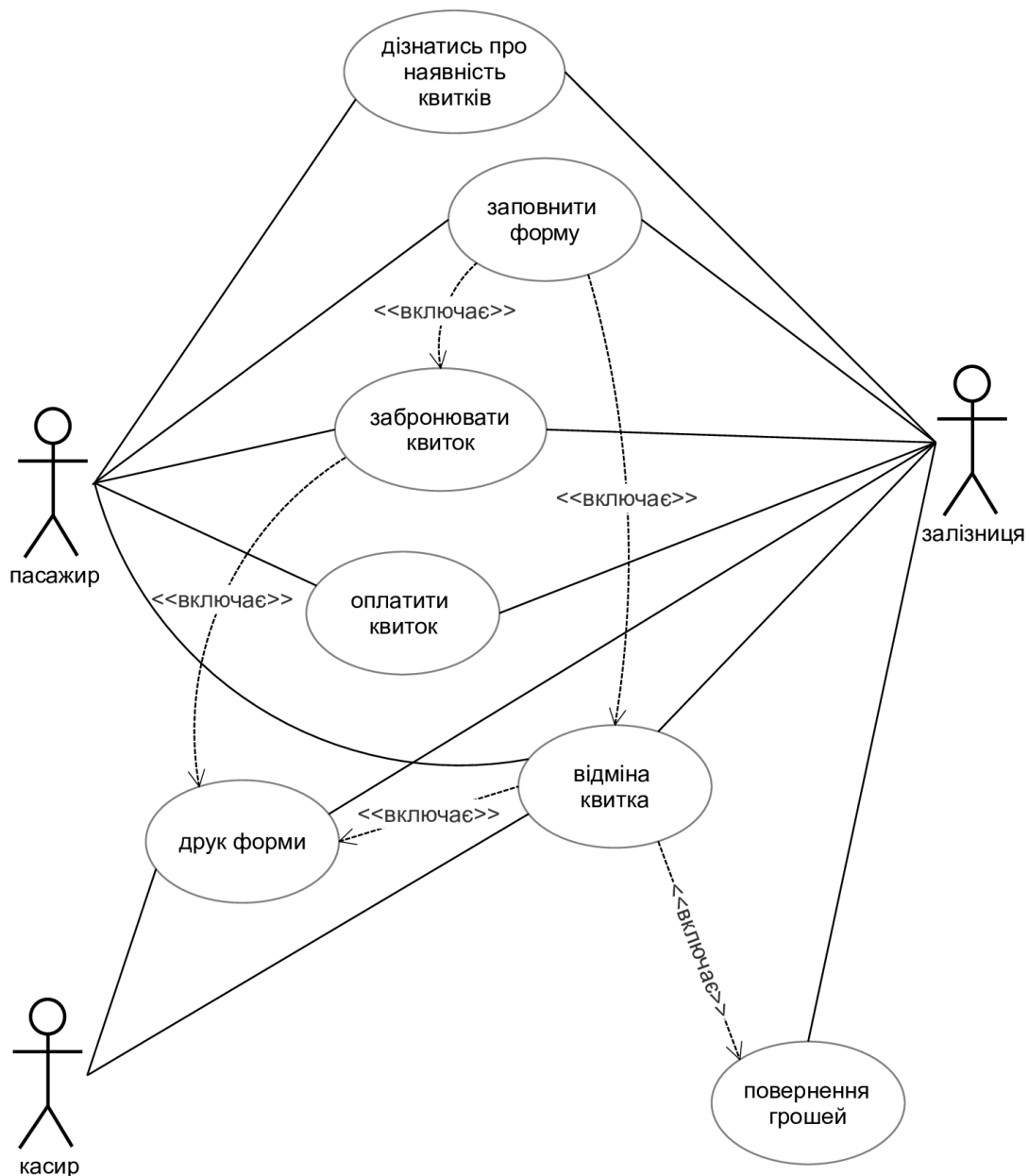


Рисунок 15 – Діаграма прецедентів на прикладі взаємодії пасажир та квиткового касира

Діаграма класів (англ. Class diagram)

Клас (англ. class,) – категорія речей, що мають загальні атрибути та операції [9]. Сама діаграма класів являє собою набір статичних, декларативних елементів моделі. Вона дає нам найбільш повне і розгорнуте уявлення про зв'язки в програмному коді, функціональність та інформацію про окремі класи. Додатки

генеруються часто саме з діаграми класів. Розглянемо на прикладі формалізації процесу бронювання проїзного документа на залізничному транспорті, рисунок 16.

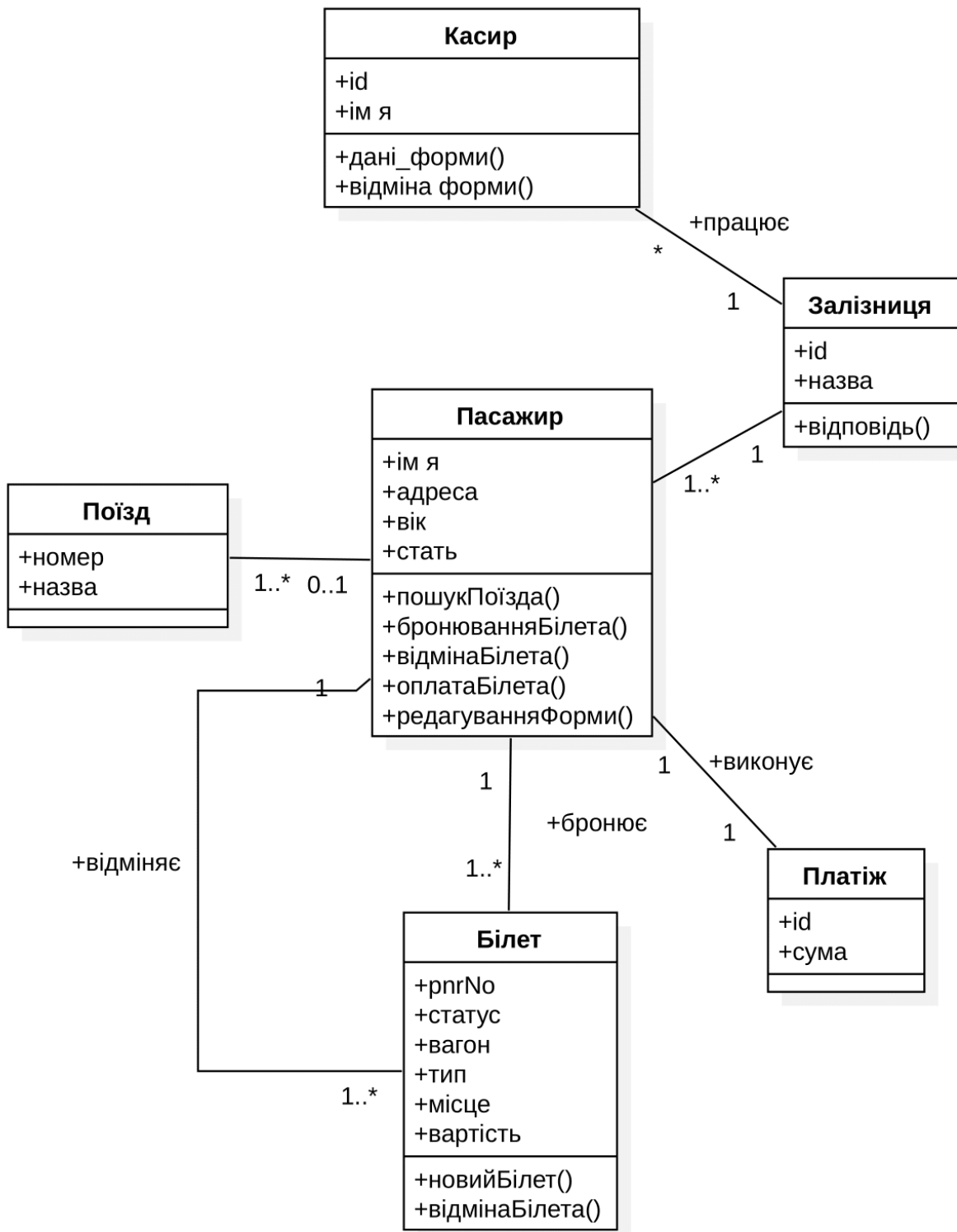


Рисунок 16 – Діаграма класів щодо формалізації процесу бронювання проїзного документа на залізничному транспорті

Для класу «пасажир» є таблиця, що містить атрибути: ім'я, адреса, вік, стать. І також показані зв'язки даної сутності з іншими, наприклад: пошук, бронювання, відміна або оплата з класом «білет»; виконує операцію оплати або відмінняє з класом «платіж» і т. п.

Діаграма активностей (англ. Activity diagram)

Вид діаграми, яка часто використовується на практиці. Діаграма активностей описує динамічні аспекти поведінки системи у вигляді блок-схеми, яка відображає бізнес-процеси, логіку процедур і потоки робіт-переходів від однієї діяльності до іншої. Дійсно, ми малюємо алгоритм дій (логіку поведінки) системи або взаємодії кількох систем [9].

На рисунку 17 приклад подібної діаграми формалізації процесу бронювання проїзного документа на пасажирський поїзд.

Діаграма активностей максимально доступно пояснює, які є інтеграції в системі. Актор (в нашому випадку покупець проїзного документа), що зайшов на сайт, робить запит щодо наявності перевізних документів. Далі у нас відбувається розгалуження: перевіряємо, чи є в наявності проїзні документи (так/ні). Якщо проїзних документів немає в наявності, відбувається вихід. Якщо проїзні документи є в наявності – відбувається замовлення проїзного документа, заповнення всіх інформаційних комірок (номер поїзда, дата відправлення, кількість пасажирів, ПБ пасажирів, наявність пільг, додаткові послуги, багаж та ін.), перевірка та підтвердження інформації для оформлення проїзного документа, оплата та друк проїзного документа.

Таку базову діаграму можна доповнити, розширити, вона може виступити частиною документації і дає загальне уявлення про роботу системи.

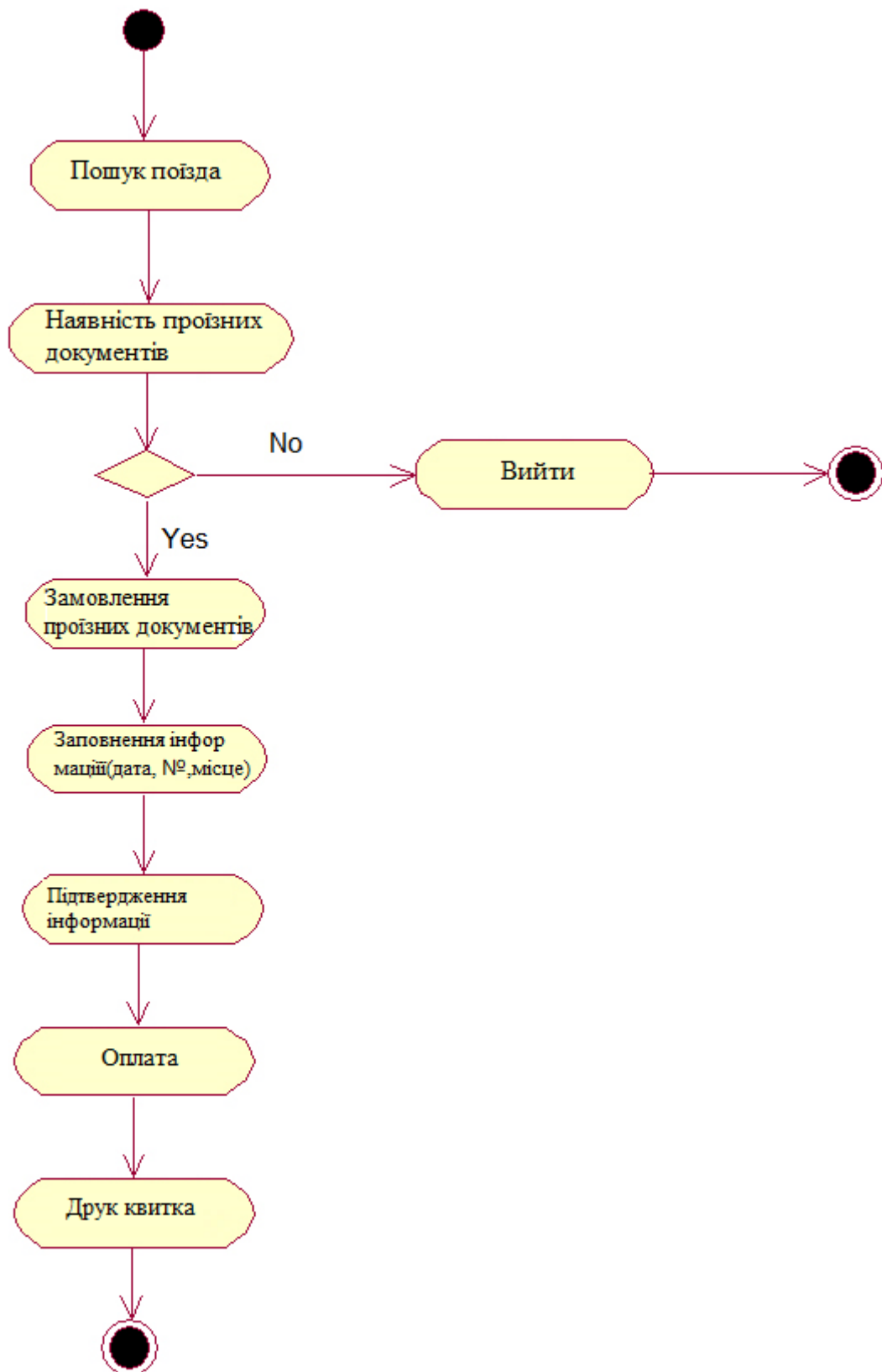


Рисунок 17 – Діаграма активностей формалізації процесу бронювання проїзного документа на пасажирський поїзд

Діаграма послідовності (англ. Sequence Diagram)

Використовується для уточнення діаграм прецедентів – описує поведінкові аспекти системи. Діаграма послідовності відображає взаємодію об'єктів в динаміці, в часі. При цьому інформація набуває вигляду повідомлень, а взаємодія об'єктів передбачає обмін цими повідомленнями в рамках сценарію [9].

За допомогою таких діаграм також можна подати формалізацію процесу бронювання проїзного документа на пасажирський поїзд, рисунок 18.



Рисунок 18 – Діаграма активностей для формалізації процесу бронювання проїзного документа на пасажирський поїзд

Діаграма розгортання (англ. **Deployment Diagram**)

Діаграма розгортання відображає графічне подання інфраструктури, на яку буде розгорнуто додаток: топологію системи і розподіл компонентів по її вузлах, а також з'єднання – маршрути передачі даних між вузлами. Діаграма допомагає раціональніше організувати компоненти, від чого залежить в числі іншого і продуктивність системи, а також вирішити допоміжні завдання, наприклад, пов'язані з безпекою [9].

На рисунку 19 наведено формалізацію процесу бронювання проїзного документа на залізничному транспорті декількома клієнтами.

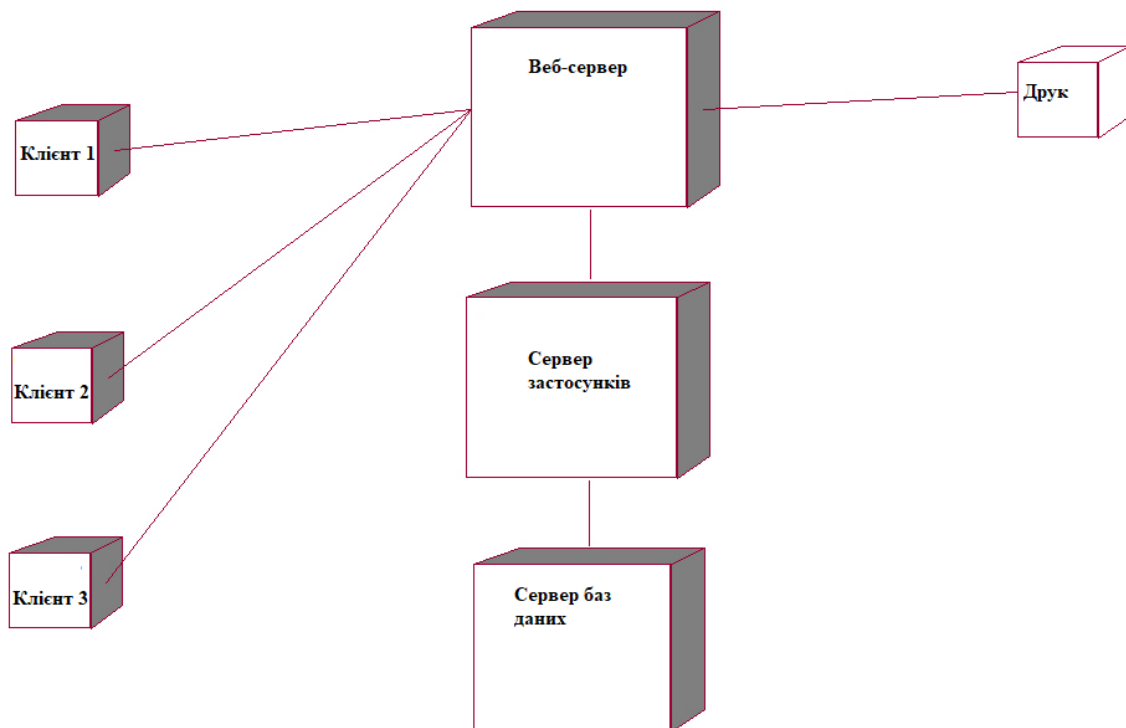


Рисунок 19 – Діаграма розгортання для формалізації процесу бронювання проїзного документа трьома клієнтами

ЗАДАЧА 5

Застосування UML-діаграм для технологічних процесів на залізничному транспорті

Постанова задачі. Необхідно представити процес, наведений у таблиці 6, у вигляді UML-діаграми на вибір здобувача.

Таблиця 6 – Вихідні дані для побудови UML-діаграм

Остання цифра залікової книжки	Процес
0	Оформлення документів для навантаження місцевого вагона на станції
1	Операції ДСП при прийманні поїзда на станцію
2	Приймання поїзда на неспеціалізовану колію
3	Відправлення пожежного поїзда
4	Відправлення відбудовного поїзда
5	Операції ДСП при відправленні поїзда зі станції
6	Оформлення документів для навантаження місцевого вагона на станції з небезпечним вантажем
7	Відправлення господарчого поїзда для проведення робіт на перегоні
8	Відправлення поїзда з підштовхуючим локомотивом, що прямує на весь перегін
9	Відправлення поїзда з підштовхуючим локомотивом, що прямує з поверненням на станцію відправлення

СПИСОК ЛІТЕРАТУРИ

1 Бутько Т. В., Прохорченко Г. О. Формування процедури автоматизації розробки графіку руху поїздів на основі алгоритму штучних бджолиних колоній. *Транспортні системи та технології перевезень*: зб. наук. праць ДНУЗТ ім. акад. В. Лазаряна. Дніпропетровськ: ДНУЗТ, 2015. Вип. 9. С. 10 – 15.

2 Teukolsky S. A., Vetterling W. T., and Flannery B. P. Simulated Annealing Methods. *Art of Scientific Computing*. W.H. Press. 2011. pp. 549–555.

3 Edmund K. Burke, Graham Kendall. Search Methodologies Introductory Tutorials in Optimization and Decision Support Techniques. 2014. P. 609.

4 Зорін Ю. М. Комп'ютерні системи штучного інтелекту: конспект лекцій. Веб-сайт. URL: <http://m.scs.kpi.ua/course/view.php?id=107> (дата звернення 10.01.2022).

5 Goodfellow I., Bengio Y. Deep Learning. 2016. Courville. P. 127.

6 Штучні нейронні мережі: що це таке? *Futurum: майбутнє сьогодні*: веб-сайт. URL: <https://futurum.today/shtuchni-neironni-merezhi-shcho-tse-take/> (дата звернення: 12.01.2022).

7 Кирик В. В. Математичний апарат штучного інтелекту в електроенергетичних системах: підручник. Київ: КПІ ім. Ігоря Сікорського, вид-во «Політехніка», 2019. 224 с.

8 Нечіткі висновки: конспект лекції з навчальної дисципліни «Сучасний штучний інтелект». *Житомирська політехніка*: веб-сайт. URL: <https://learn.ztu.edu.ua> (дата звернення: 01.02.2022).

9 UML для бізнес-моделювання: для чого потрібні діаграми процесів. *Evergreen*: веб-сайт. URL: <https://evergreens.com.ua/ua/articles/uml-diagrams.html> (дата звернення: 10.02.2022).

МЕТОДИЧНІ ВКАЗІВКИ

до практичних занять
та виконання індивідуальних завдань

з дисципліни
*«СУЧАСНІ ІНФОРМАЦІЙНІ СИСТЕМИ
В УПРАВЛІННІ ЗАЛІЗНИЧНИМИ ПІДРОЗДІЛАМИ»*

Відповідальний за випуск Прохорченко Г. О.

Підписано до друку 2022 р.
Умовн. друк. арк. 3,5. Тираж . Замовлення №

Видавець та виготовлювач Український державний університет
залізничного транспорту,
61050, Харків-50, майдан Фейєрбаха, 7.
Свідоцтво суб'єкта видавничої справи ДК № 6100 від 21.03.2018 р.