

Міністерство освіти і науки України

Український державний університет науки і технологій

Факультет Комп'ютерні технології та системи
Кафедра Комп'ютерні інформаційні технології

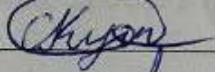
Пояснювальна записка

до кваліфікаційної роботи
магістра

на тему: «Прогнозування результатів командних змагань на основі
конструктивного підходу та МАІ»

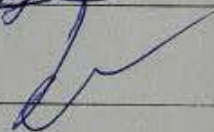
за освітньою програмою **121 Інженерія програмного забезпечення**
зі спеціальності: **121 Інженерія програмного забезпечення**

Виконав: студент групи «ПЗ2221»



/Сергій КУМΠΑН/

Керівник:



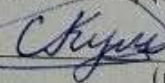
/Віктор ШИНКАРЕНКО/

Нормоконтролер:



/Світлана ВОЛКОВА/

Засвідчую, що у цій роботі немає запозичень з
праць інших авторів без відповідних посилань
Студент



(підпис)

Дніпро – 2024 рік
Факультет Комп'ютерних технологій і систем
Кафедра Комп'ютерні інформаційні технології
Рівень вищої освіти Магістр
Спеціальність Інженерія програмного забезпечення

«ЗАТВЕРДЖУЮ»
Завідувач кафедри
проф. Гарячкін В.М.
(підпис)
«__» _____ 2023 р.

ЗАВДАННЯ

до дипломної роботи на здобуття ОС магістр
(освітній ступінь)
студента групи (ПЗ2221) 951-М Кумпана Сергія Володимировича
(номер групи) (ПІБ)

1 Тема дипломної роботи: Прогнозування результатів командних змагань на основі конструктивного підходу та МАІ затверджена наказом по університету від «11» лютого 2022 р. № 173ст.

2 Термін подання студентом закінченої роботи «17» січня 2024 р.

3 Вихідні дані до дипломної роботи _____

4 Зміст пояснювальної записки (перелік питань до розробки) Призначення, постановка задачі та огляд аналогів, методика проведення дослідження, проектування і розробка інструментального програмного забезпечення, дослідження ефективності роботи алгоритму, висновки.

5 Перелік демонстраційного матеріалу Постановка задачі, опис аналогів, методи вирішення, механізми реалізації, аналіз результатів, висновки.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва розділів дипломного проекту	Термін виконання розділів (роботи) проекту	Примітка
1	Вступ	1.09.2023 – 10.09.2023	
2	Аналіз сучасного стану дослідження проблеми за науковими літературними джерелами	10.09.2023 – 15.09.2023	
3	Постановка задачі, технічне завдання	15.09.2023 – 30.09.2023	
4	Створення тестової програми	01.10.2023 – 15.10.2023	
5	Перші тестування	15.10.2023 – 22.10.2023	30%
6	Аналіз результатів	22.10.2023 – 11.11.2023	60%
7	Оформлення пояснювальної записки	11.11.2020 – 01.01.2024	100%
8	Демонстраційні матеріали	02.01.2024 – 17.01.2024	

Дата видачі завдання «11» лютого 2020 р.

Керівник дипломного проекту

(підпис)

Шинкаренко В.І.

(ПІБ)

Завдання прийняв до виконання

(підпис)

Кумпан С.В.

(ПІБ)

РЕФЕРАТ

Об'єктом дослідження є ефективність прогнозування результатів спортивних змагань за допомогою методу аналізу ієрархій.

Метою є визначення ступеню ефективності застосування МАІ для вирішення підібних задач

Пояснювальна записка складається зі вступу, 4 розділів, висновків, бібліографічного списку та 2 додатків:

– у вступі описується підстава для розробки, її актуальність. Складається із 3 сторінок;

– у першому розділі висвітлено аналіз сучасного стану дослідження застосування МАІ у різних сферах людської діяльності. Складається з 14 сторінок;

– у другому розділі надано обґрунтування експериментального методу дослідження. Складається з 14 сторінок;

– у третьому розділі представлена розробка й тестування інструментального забезпечення для дослідження. Складається з 11 сторінок;

– у четвертому розділі описано виконані дослідження. Складається з 6 сторінки;

– додатки містять технічне завдання, робочий проект, опубліковані тези та статтю.

Таблиць – 4, рисунків – 17, бібліографія – джерел.

Ключові слова: метод аналізу ієрархій, прогнозування результатів змагань, матриця попарних порівнянь.

Зміст

ВСТУП	6
1 АНАЛІЗ СУЧАСНОГО СТАНУ ДОСЛІДЖЕННЯ ПРОБЛЕМИ ЗА НАУКОВИМИ ЛІТЕРАТУРНИМИ ДЖЕРЕЛАМИ	9
1.1 Призначення та сфера застосування.....	9
1.2 Постановка задачі.....	10
1.3 Огляд останніх досліджень і публікацій.....	11
1.4 Огляд програмних аналогів.....	15
1.4.1 FiveThirtyEight	16
1.4.2 KickForm	20
1.4.3 Windrawwin.....	21
Висновки до першого розділу.....	24
2 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ ДЛЯ ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ СПОРТИВНИХ ЗМАГАНЬ НА ОСНОВІ КОНСТРУКТИВНО-ПРОДУКЦІЙНОГО ПІДХОДУ І МЕТОДУ АНАЛІЗУ ІЄРАРХІЙ.....	25
2.1 Внутрішнє проектування.....	25
2.1.1 Опис функціональних характеристик	25
2.1.2 Вхідні дані.....	26
2.1.3 Вихідні дані.....	27
2.1.4 Модулі програми.....	27
2.2 Зовнішнє проектування	29
2.2.1 Середовище розробки і мова програмування.....	30
2.2.2 Обґрунтування експериментального методу дослідження ефективності застосування МАІ для прогнозування результатів командних змагань	32
2.2.2.1 Аналіз МАІ.....	32
2.2.2.2 Методика застосування МАІ.....	35
2.2.2.3 Застосування МАІ для прогнозування результатів командних змагань	38
Висновки до другого розділу	40
3 РОЗРОБКА І ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ КОМАНДНИХ ЗМАГАНЬ ЗА ДОПОМОГОЮ МАІ.....	41
3.1 Застосування принципів програмування	41
3.2 Проектування архітектури системи.....	44
3.3 Аналіз методів, використаних для тестування програми.....	49
Висновки до третього розділу.....	51
4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТОЧНОСТІ ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ КОМАНДНИХ ЗМАГАНЬ ЗА ДОПОМОГОЮ МЕТОДУ АНАЛІЗУ ІЄРАРХІЙ.....	52
4.1 Підготовка до експерименту	52
4.2 Проведення експерименту.....	54

ВСТУП

Дослідження у сфері Методу Аналізу Ієрархій ведуться з моменту його розробки в сімдесятих роках двадцятого століття. З того часу і по цей день спостерігається постійне збільшення наукових публікацій, присвячених цьому методу. Перш за все, це пов'язана з великою кількістю сфер, застосування МАІ до яких є оптимальним. Зі впровадженням штучного інтелекту у все більшу кількість сфер людської діяльності, автоматизація процесу прийняття рішень в умовах великої кількості альтернатив зі складною ієрархією критеріїв стає більш актуальною з кожним днем.

Актуальність роботи. Прогнозування результатів спортивних змагань є дуже розповсюдженою сферою на сьогоднішній день. Це пов'язано зі зростаючою увагою до спорту у багатьох розвинутих країнах, та країнах, що розвиваються. В деякому сенсі це обумовлено розвитком інформаційних технологій та технологій телекомунікації, адже тепер люди з усього світу можуть спостерігати за змаганнями в прямому ефірі, вболіваючи за свого улюбленого спортсмена чи команду, і вони більше не вимушені їхати в інші міста чи країни, чи чекати поки результат надрукують в газеті.

Таким чином прогнозування результатів змагань є дуже популярним. Ним займаються велика кількість телевізійних каналів, друкованих та інтернет видань, що спеціалізуються на тих чи інших видах спорту. Також є безліч людей, які пропонують всім бажаючим купити такі прогнози через мережу інтернет. Але в більшості випадків такі прогнози є результатом суб'єктивної думки автора прогнозу, яка заснована на його власному досвіді, а отже на неї можуть впливати багато факторів, як пов'язаних, так і не пов'язаних безпосередньо зі змаганням. Такі прогнози ніколи не будуть системними, а отже може піддаватися сумнівам.

На результати спортивних змагань впливає величезна кількість факторів, таких як фізичний та ментальний стан гравців, випадкових травм чи вердиктів рефері, які неможливо передбачити до початку змагання. Таким чином постає

питання, чи може бути релевантним прогноз на базі матриці попарних порівнянь, яка виходить в результаті роботи алгоритму МАІ.

Об'єкт дослідження. Об'єктом дослідження є процес прогнозування результатів спортивних змагань.

Предмет дослідження. Предметом дослідження є оцінка точності прогнозування результатів спортивних змагань за допомогою Методу Аналізу Ієрархій, в залежності від кількості відсутніх альтернатив.

Мета. Дослідити можливість прогнозування результатів спортивних змагань за допомогою Методу Аналізу Ієрархій.

Завдання. В процесі дослідження необхідно було виконати такі завдання:

- Реалізувати інструментальне програмне середовище, яке буде прогнозувати результати спортивних змагань.
- Реалізувати методи оцінки точності прогнозування результатів спортивних змагань.
- Дослідити вплив кількості відсутніх альтернатив на точність прогнозування

Методи дослідження. Для встановлення залежності точності прогнозування від кількості відсутніх альтернатив проводилася серія емпіричних експериментів. Під час цих експериментів програма обробляла декілька чемпіонатів. За допомогою метода аналізу ієрархій робиться прогноз сили команд, дані за якими відсутні. В результаті проведення експериментів формувалася статистична інформація що до залежності точності прогнозування від кількості відсутніх альтернатив.

Наукова новизна. Досліджено залежність точності прогнозування результатів спортивних змагань в залежності від кількості відсутніх альтернатив. На відміну від існуючих алгоритмів прогнозування за методом часових рядів, МАІ використовує тільки актуальні дані, в той час як часові ряди за самою своєю

суттю мають використовувати дані за довгий період часу, які можуть бути нерелевантними.

Практична значимість. Результати виконаних досліджень дозволяють оцінити точність роботи Методу Аналізу Ієрархій при його застосуванні до прогнозування результатів чемпіонатів з футболу. Інструмент дослідження може використовуватися для оцінки точності роботи алгоритму у випадку його застосування до прогнозування результатів змагань з інших спортивних дисциплін.

Апробація результатів дослідження: Результати дослідницької роботи доповідались на семінарах кафедри КІТ 09.11.2023.

Публікації за темою роботи. Підготовлено проект статті
Прогнозування результатів командних змагань за допомогою МАІ.

1 АНАЛІЗ СУЧАСНОГО СТАНУ ДОСЛІДЖЕННЯ ПРОБЛЕМИ ЗА НАУКОВИМИ ЛІТЕРАТУРНИМИ ДЖЕРЕЛАМИ

1.1 Призначення та сфера застосування

Прогнозування результатів спортивних змагань є надзвичайно популярною сферою на сьогоднішній день. Однією з причин цього є популярність азартних ігор серед усіх верств населення, адже сьогодні букмекерський бізнес можна вести через мережу інтернет, і все більше і більше людей отримують можливість поставити на перемогу тієї чи іншої команди. В результаті це приваблює велику кількість шахраїв, які обіцяють стовідсоткову гарантію вірності своїх прогнозів і пропонують людям купувати їх за «невелику» в порівнянні з потенціальним виграшем плату, таким чином отримуючи нечесний дохід за рахунок людей, які ризикують програти все.

Втім не можна стверджувати, що спроби спрогнозувати результати футбольних змагань не мають жодного сенсу. Є велика кількість даних, яку можна аналізувати перед кожним матчем. Такими даними можуть бути об'єктивні фактори, такі як наприклад статистика виступів команд за деяку кількість матчів. Ця статистика може включати в себе середню кількість забитих та пропущених, час володіння м'ячем, відсоток реалізації голевих моментів, відсоток вдалих передач м'яча і так далі. Іншими факторами для аналізу можуть бути розміщення гравців на полі та склади команд. Втім оцінка таких факторів вже вимагає експертної оцінки, неминучим ризиком якої в свою чергу є людський фактор, адже кожен експерт володіє власним унікальним досвідом у тих чи інших аспектах. Таким чином експертна оцінка тієї ж самої команди різними експертам може суттєво відрізнятись.

Таким чином з'являється необхідність дослідження можливостей прогнозування результатів спортивних змагань за допомогою різних алгоритмів. У відкритому доступі є деяка кількість онлайн ресурсів і незліченна кількість

експертів, які пропонують прогнози результатів тих чи інших змагань. Втім в жодному з випадків не зрозуміло, як ці прогнози були зроблені. Інколи додаються якісь статистичні дані гри команд, але користувач може лише здогадуватися, які самі дані були використані при прогнозуванні, і якими алгоритмами вони були оброблені.

Таким чином метою розробки програмного продукту є дослідження такого питання: чи здатний обраний алгоритм прогнозувати результати спортивних змагань в цілому. І якщо здатен, тобто його точність від простого вгадування за допомогою підкидання монетки, наскільки точним може бути такий прогноз.

Сферою застосування такої програми можуть бути будь які змагання, як командні, так і одиночні. Втім необхідною умовою для роботи з обраним алгоритмом є формат змагань. Прогнозування може використовуватися лише для групових форматів змагань, де кожен учасник зіграє з кожним.

1.2 Постановка задачі

Розробка модулю прогнозування результатів спортивних змагань. Даний модуль використовує Метод аналізу ієрархій для прогнозування результатів. Даний модуль використовуватиме список вже зіграних ігор, заданих користувачем. При цьому в якості експерта для оцінки відносної сили команд в даній програмі виступає сама природа. Індикатором відносної сили команд є результат зіграного між ними матчу: одна з команд сильніша, якщо перемогла, або команди рівні за силою, якщо була нічия. Ступінь різниці в силі команд розраховується за відношенням забитих м'ячів. Якщо команди не грали одна з одною, результат їхньої зустрічі буде спрогнозований на основі відносної сили цих у порівнянні з усіма іншими командами. Результатом прогнозу буде відносна сила команд. У випадку якщо сила однакова, за прогнозом команди зіграють в нічию, у іншому випадку команда з більшою силою за прогнозом виграє.

Розробка модулю дослідження залежності точності прогнозування в залежності від кількості наявних результатів. Даний модуль розраховує точність прогнозування на основі прогнозів, зроблених алгоритмом прогнозування. Маючи результати всіх ігор чемпіонату, цей модуль виконує прогноз на основі частини ігор, наприклад якщо в чемпіонаті було двадцять турів, прогноз буде виконано на основі ігор, зіграних у перших десяти турах, а потім, маючи результати всіх ігор, підраховує відношення кількості правильно спрогнозованих ігор до загальної кількості ігор, які були спрогнозовані. Робота модуля продовжується, поступово збільшуючи кількість заздалегідь відомих результатів ігор. Результатом роботи програми є графік залежності точності прогнозування від початкової кількості даних.

Для користувача програма буде надавати наступні можливості:

- Введення початкових даних – результатів зустрічей команд між собою
- Прогнозування результатів змагань на основі деякої кількості відомих даних
- Аналіз точності прогнозування за допомогою Методу аналізу Ієрархій

1.3 Огляд останніх досліджень і публікацій

1.3.1 Загальні тенденції досліджень в області використання МАІ

Починаючи з моменту розробки Томасом Сааті в 1970-х роках, і до теперішнього часу методу аналізу ієрархій широко використовується для вирішення практичних задач і розвивається вченими всього світу. МАІ використовується для прийняття рішень в разі наявності деякого набору альтернатив у багатьох сферах: соціальній, економічній, індустріальній, політичній, екологічній, бізнесу та інших. МАІ може вирішувати такі задачі як: оцінка якості об'єктів, наприклад ефективності підрозділів компаній у різних

містах чи країнах, ефективності персоналу, аналіз балансу затрат, вигоди та ризиків для декількох проектів, маркетингові дослідження, планування найбільш ефективних заходів, тощо. Велика кількість досліджень показала гарну ефективність алгоритму при вирішенні задач прийняття правильного рішення за наявності великої кількості альтернатив, критеріїв та інших факторів, які впливають на фінальний вибір. Постійно пропонуються різні варіанти модифікації алгоритму як для розширення набору задач, які можуть бути вирішені цим алгоритмом, так і для покращення якості роботи алгоритму. Ключовим елементом роботи алгоритму є судження експертів що до відносної ваги тих чи інших критеріїв. Багато досліджень присвячені таким факторам, як вплив суб'єктивності оцінки на якість роботи алгоритму, стійкість ранжування при незначній зміні експертної оцінки, ступінь відмінності ранжування при оцінці різними експертами з однаковими знаннями, досвідом, тощо. Для вирішення цих питань пропонуються різні алгоритми, такі як наприклад опитування деякої кількості експертів з подальшим усередненням їх відповідей. До недоліків такого підходу відносять ризик появи випадків, коли відносні вагові коефіцієнти, отримані від деяких експертів кардинально відрізнялися від оцінок інших. Через це при підрахунку усередненого значення вагових коефіцієнтів шукана величина може суттєво змінитися. Для вирішення таких ситуацій в МАІ є вбудований показник якості експертної оцінки – індекс узгодженості, який відповідає за ступінь відповідності числової та транзитивної узгодженості заданим параметрам максимальної кількості помилок експертних оцінок. Втім відсіювання експертів з високим ступенем відносної злагоженості не завжди дає потрібний ефект, адже кардинальна різниця в експертних судженнях не обов'язково говорить про неузгодженість тієї чи іншої оцінки. Таким чином постає питання про розробку нових алгоритмів, які будуть відповідати за відбір експертів, або перехід до таких процедур, які дозволять відмовитися від експертної оцінки повністю і перейти до об'єктивних оцінок факторів.

1.3.2 Приклади практичних досліджень використання МАІ

1.3.2.1 Інструмент визначення актуальності прискореного будівництва мостів у порівнянні із класичними способами будівництва.

Це дослідження було проведено департаментом цивільної та екологічної інженерії університету міста Сіракузи, Сполучені Штати Америки у 2016 році. У цьому дослідженні було запропоновано двоетапний алгоритм прийняття рішення що до доцільності використання прискореного методу будівництва інфраструктури. В якості першого етапу як раз і було запропоновано МАІ. Для проведення розрахунків було запропоновано велику кількість різних факторів різного характеру. Прикладами таких факторів є

- Важливість об'єкту інфраструктури для розвитку населених пунктів, що знаходяться поблизу, а саме надання доступу до найближчих об'єктів цивільного використання, таких як школи, лікарні, підприємства і таке інше.
- Вплив на інші об'єкти транспортної інфраструктури, такі як автомобільні дороги, залізничні шляхи, водні транспортні вузли, і так далі.
- Вплив об'єкт на зовнішнє середовище протягом його будівництва і подальшої експлуатації

За результатами проведених досліджень очікується, що запропонована модель стане базою для подальшої розробки і впровадження більш ефективних методів прийняття рішень у проектах будівництва мостів, а негативний вплив пов'язаний із ремонтом і відновленням критичної інфраструктури буде суттєво зменшено.

1.3.2.2 Вибір університетського факультету

Це дослідження було проведено університетом Блумсбург, штат Пенсильванія у 2005 році. Головною ціллю дослідження було поліпшення процесу прийняття рішення стосовно подальшої наукової діяльності і функціонування університету. До таких рішень входили: вибір факультету студентом, вибір

кандидата на працевлаштування до університету, вибір обладнання для закупівель і так далі. В якості проблематики прийняття таких рішень зазначалася відсутність чітко визначеного підходу до вирішення таких питань. Це призводило до того, що кожного разу, як поставала необхідність найму на роботу нового кандидата, створювалася нова комісія, у якої був новий підхід до вирішення подібних питань. Це призводило до кардинально різного рівня кандидатів, яких наймали на роботу.

Для вирішення цього питання було запропоновано використання МАІ. До переваг використання цього метода було віднесено ефективність прийняття рішень в умовах великої кількості критеріїв різної значущості, можливість видалення непотрібних критеріїв або додавання нових без впливу на роботу алгоритму, можливість зміни вагових коефіцієнтів критеріїв в залежності від особливостей факультету, який використовує алгоритм для прийняття рішення.

В результаті впровадження алгоритму у процес прийняття рішень були виявлено, що результати роботи комісій, які використовують МАІ є оптимальними і враховують всі точки зору, побажання та зауваження людей, відповідальних за прийняття рішень.

1.3.2.3 Оцінка загальної якості програмного забезпечення

Метою цього дослідження була розробка методології оцінки якості програмного забезпечення. На відміну від інших алгоритмів, коли програми оцінюються на етапі тестування з приводу коректності їх роботи, або оцінюються вже готові програми, але порівнюються вони лише за одним критерієм, запропонований у цьому дослідженні підхід надає можливість оцінювати готові програми за багатьма різними факторами.

Для дослідження було запропоновано оцінювати програму за великою кількістю факторів, до яких входили: доступність програми, доступність документації як для користувачів, так і для системних адміністраторів, програмістів і так далі, функціональність, захист, стабільність, користувацький

інтерфейс, потенціал модернізації програми і так далі.

У результаті проведеного дослідження було відзначено перевагу використання шкали відношень Сааті, яка оперує поняттями краще, гірше, набагато краще, набагато гірше перед використанням кількісної шкали при порівнянні якості роботи програмного забезпечення за деякими критеріями. Втім також було зазначено, що не варто цілком спиратися на один, який би то не було, метод оцінки. Вказано, що МАІ буде гарним доповненням до деяких інших методів, таким чином висновок буде найбільш оптимальним.

1.3.2 Загальні тенденції досліджень в області використання конструкційно-продукційного підходу.

Конструкційно-продукційні структури є відносно новим напрямком в науці, і дослідження у даній сфері ведуться лише декількома групами вчених. Втім в останніх публікаціях відзначена перспективність методу для формалізації опису тих чи інших об'єктів досліджень. В свою чергу, в сучасних умовах стрімкого розвитку штучного інтелекту це відкриває нові підходи до розробки програмного забезпечення. Завдяки формальному опису об'єктів та алгоритмів стає можливим автоматизація процесу розробки та оптимізації програмного забезпечення для використання одного алгоритму для вирішення великого спектру задач, однакових за своєю концепцією. Такий підхід дозволить застосовувати алгоритм для різних сфер, не вносячи змін в програмне забезпечення зовсім, або модифікуючи лише окремі модулі, без переробки алгоритму від початку і до кінця.

1.4 Огляд програмних аналогів

Існують інтернет сервіси, які намагаються прогнозувати результати футбольних змагань на основі аналізу статистики, інших даних та різних алгоритмів. Для аналізу було взято декілька веб сервісів, які пропонують користувачам прогнози результатів футбольних матчів у вигляді ймовірності

перемоги тієї чи іншої команди або нічії, кількості голів, які будуть забиті кожною командою, вірогідність того, що обидві команди заб'ють хоч один гол і так далі.

1.4.1 FiveThirtyEight

Це відомий сайт, який надає статистичні дані та аналітику для різних видів спорту, включаючи футбол. На цьому сайті можна побачити прогнози результатів окремих матчів, а саме вірогідність перемоги тієї чи іншої команди, або нічії, а також статистику що до реалізації голевих моментів, стандартних положень і так далі.

		WIN	DRAW
5/27	RB Leipzig 4 Schalke 04 2	60%	22%
	RBL	SCH	
	Adjusted goals	3.7	2.1
	Shot-based xG	2.7	1.5
	Non-shot xG	2.8	1.9
5/27	Union Berlin 1 Werder Bremen 0	54%	25%
	FCU	BRE	
	Adjusted goals	1.1	0.0
	Shot-based xG	0.8	0.4
	Non-shot xG	2.0	0.9
5/27	VfL Bochum 3 Leverkusen 0	32%	26%
	BOC	LEV	
	Adjusted goals	2.2	0.0
	Shot-based xG	1.6	1.0
	Non-shot xG	2.0	1.2
5/27	Eintracht 2 SC Freiburg 1	41%	27%
	FRA	FRE	
	Adjusted goals	2.1	1.1
	Shot-based xG	2.4	1.7
	Non-shot xG	0.9	1.6
5/27	Dortmund 2 Mainz 2	71%	17%
	DOR	MNZ	
	Adjusted goals	2.1	2.1
	Shot-based xG	4.5	1.5
	Non-shot xG	3.5	1.2
5/27	1. FC Köln 1 Bayern Munich 2	18%	20%
	COL	BMU	
	Adjusted goals	1.1	2.1
	Shot-based xG	1.3	1.9
	Non-shot xG	1.2	0.8
5/27	Gladbach 2 FC Augsburg 0	54%	23%
	MGB	AUG	
	Adjusted goals	2.1	0.0
	Shot-based xG	2.7	0.5
	Non-shot xG	2.1	1.1
5/27	Wolfsburg 1 Hertha BSC 2	50%	25%
	WLF	HER	
	Adjusted goals	1.1	2.1
	Shot-based xG	2.4	1.5
	Non-shot xG	2.3	1.1

Рисунок 1.1 – Результат роботи сервісу FiveThirtyEight

Прогнози на даному ресурсі засновані на суттєво переглянній версії рейтингу ESPN Soccer Power Index (SPI), розробленого Нейтом Сільвером у 2009 році. SPI рейтинг являє собою очікувану силу команди і складається з двох частин: атакуючий рейтинг команди, який показує скільки голів вона має забити у грі проти середньої команди на нейтральному полі, та рейтинг гри команди в обороні, який показує скільки м'ячів команда має пропустити у тих самих умовах. У підсумку ці два рейтинги загалом утворюють рейтинг SPI, який показує відсоток отриманих очок, три за перемогу, одне за нічию та нуль за поразку. Маючи рейтинги команд можна спрогнозувати як результат зустрічі

двох команд для різних форматів гри, таких як виїзний товариський матч, турнірний матч або фінал чемпіонату.

Перед початком турнірного сезону, SPI рейтинг базується на двох факторах: рейтинг команди наприкінці минулого сезону, та її ринкова цінність, яка розраховується на основі грошової цінності кожного гравця команди, якщо його буде виставлено на трансферний ринок. З плином сезону рейтинг команди змінюється після кожного матчу на основі рівня гри команди та сили її опонента. Важливим моментом є те, що рейтинг команд може знизитися навіть у разі перемоги, якщо результат гри був нижчим за очікуваний.

Втім результат на табло не завжди відповідає тому, що очікували глядачі враховуючи свої власні враження від рівня гри двох команд. Тому для зменшення впливу випадковості на рейтинг використовуються три параметра: скориговані голи, очікувана кількість реалізації гольових моментів та очікувана кількість голів в залежності від параметрів гри команди біля штрафного майданчика опонента.

Скориговані голи – цей параметр враховує умови, за яких було забито кожен гол. При значенні за замовчанням, що дорівнює одиниці, цінність голів, забитих коли у команди було більше гравців на полі, або коли команда вже лідирувала при підрахуванні цього параметра буде нижчою, і ця цінність буде зменшуватися ще більше, якщо гол було забито наприкінці матчу. В той самий час цінність голів, які мають критичне значення для результату матчу, буде більшою. У результаті цей параметр буде близьким до фактичного рахунку у матчі, хоча не обов'язково, що буде йому дорівнювати.

Очікувана кількість реалізації гольових моментів характеризує скільки м'ячів команда мала забити за матч. При підрахунку цього параметра враховуються вірогідність реалізації кожного окремого гольового моменту. Вона розраховується на основі відстані до воріт у момент удару, кута, частини тіла, і влучності гравця, який цей удар зробив. Сума цих імовірностей буде відповідати

другому параметру. Він може суттєво відрізнятись від фактичної кількості забитих голів як у більшу, так і у меншу сторону.

Очікувана кількість голів в залежності від параметрів гри команди біля штрафного майданчика опонента. Вона розраховується на основі кількості дій, які могли призвести до удару поворотах, не враховуючи сам удар. До таких дій відносяться кутові, пас у штрафний майданчик, перехоплення м'яча на чужій половині поля і таке інше. При підрахунку цього параметра також враховується вірогідність того, що цей тип дії спричинить гол, а також те, який самий гравець або пар гравців розіграли кожен момент. Як і для другого параметра, сума усіх імовірностей і буде шуканим значенням.

Значення цих параметрів будуть враховані для зміни рейтингу атаки команди, а значення параметрів опонента буде враховано для зміни рейтингу захисту першої команди.

Після розрахунку рейтингів команд, відбувається сам прогноз результату матчу. Він розраховується таким чином, щоб рахунок у матчі відповідав кількості

	0	1	2	3	4	5+
0	5%	12%	16%	14%	9%	9%
1	2%	6	7	6	4	4
2	<1%	1	2	1	1	1
3	<1%	<1	<1	<1	<1	<1
4	<1%	<1	<1	<1	<1	<1
5						

Рисунок 1.2 Приклад прогнозу результату матчу

м'ячів, яку кожна команда має забити для того, щоб відповідати своєму рейтингу атаки. Фінальний прогноз робиться з урахуванням важливості матчу для кожної команди, перевага домашнього поля і так далі.

Після розрахунку імовірностей того чи іншого результату на їх основі відбувається прогнозу результату цілого сезону. Важливою деталлю є те, що після прогнозу кожного матчу, рейтинг команди змінюється на основі його основі.

Останнім фактором, який впливає на прогноз чемпіонату є сила рідної ліги команди. Переважну кількість матчів команди грають у домашніх чемпіонатах, тому їх рейтинг має бути перерахований при прогнозі результатів таких турнірів як Ліга Чемпіонів чи Ліга Європи.

Таким чином, алгоритм, який використовується сайтом, робить прогноз результатів футбольних матчів і цілих чемпіонатів засновуючись на широкому наборі статистичних даних, які враховують як рівень гри тієї чи іншої команди протягом років, так і успішність виступу команди у поточному чемпіонаті та рівень гри кожного окремого гравця у команді.

1.4.2 KickForm

Це безкоштовний онлайн сервіс, який надає прогнози для різних футбольних ліг і змагань. Окрім вірогідності перемоги тієї чи іншої команди або нічиєї, на сайті також можна побачити прогноз кількості голів, забитих кожною командою, прогноз кількості м'ячів, забитих за матч у цілому, а також імовірність того, що обидві команди заб'ють хоча б один гол.

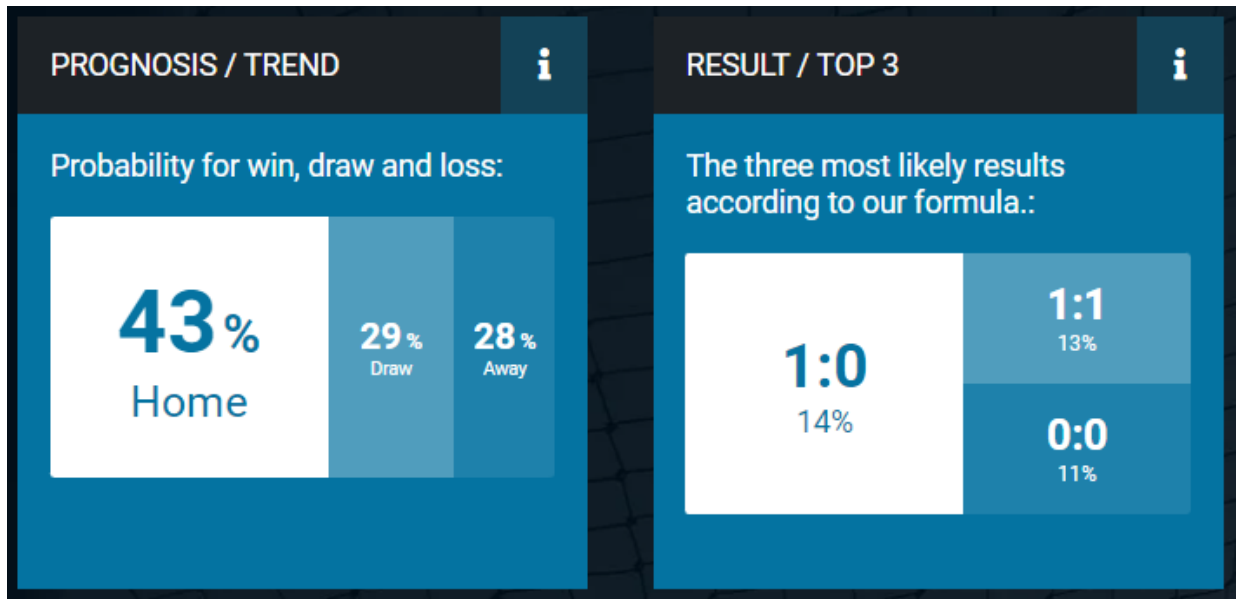


Рисунок 1.3 Прогноз вірогідності перемоги та рахунку

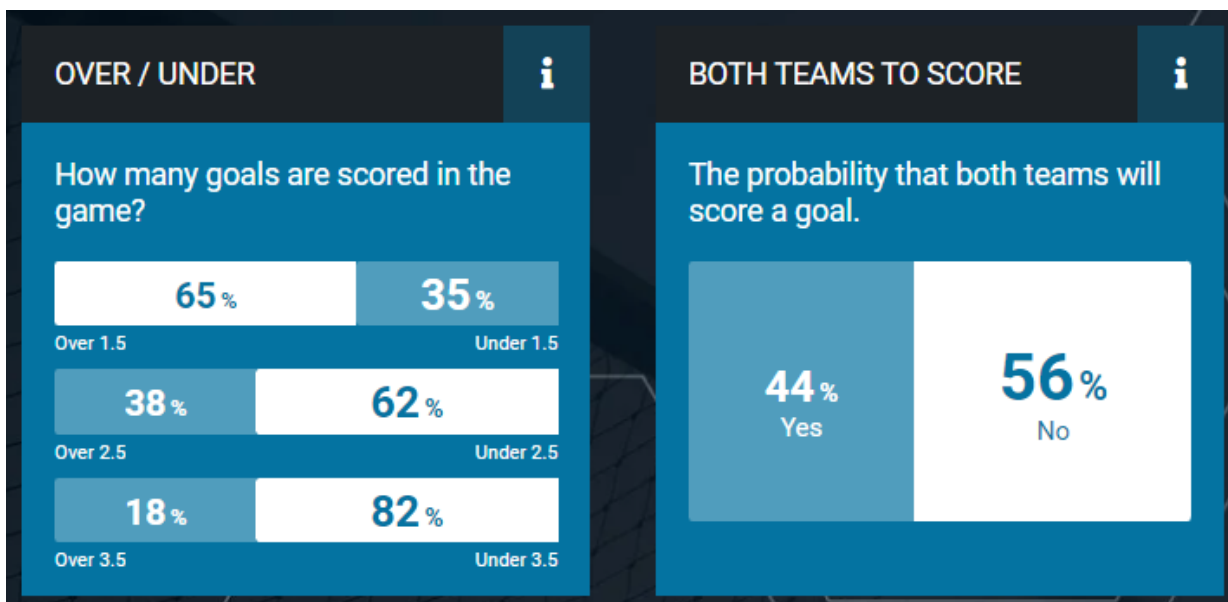


Рисунок 1.4 Прогноз кількості забитих голів

Веб сервіс KickForm надає прогнози результатів футбольних матчів, але не надає жодної інформації що до того, які саме дані були використані для прогнозу, чи які алгоритми були використані для їх обрахунку. Таким чином користувач не ніяк не може проаналізувати актуальність цих прогнозів і якось скоригувати їх, спираючись на власний досвід. Отже він вимушений або цілковито покладатися на «магічні» цифри, або зовсім їх не використовувати, адже значущість таких прогнозів не піддається жодному аналізу.

1.4.3 Windrawwin

Ще дин безкоштовний онлайн-сервіс, який надає прогнози для футбольних матчів та інших спортивних змагань. Окрім того, цікавим елементом

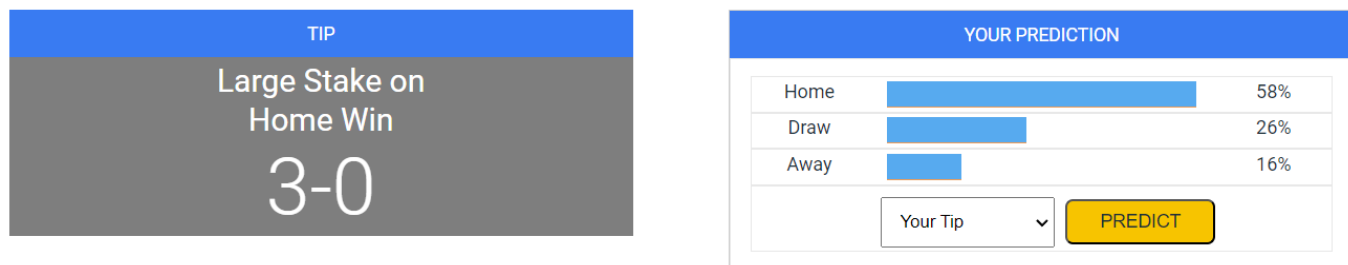


Рисунок 1.5 Прогнози користувачів сайту

сайту є можливість бачити прогнози інших користувачів та робити прогнози самому.

Хоча веб сервіс і не надає жодної інформації що до того, які саме дані та алгоритми використовувалися для прогнозування результату зустрічі, для кожного матчу присутня інформація що до успішності гри команд у турнірі. До цієї інформації входять такі дані, як загальна кількість зіграних ігор, перемоги та поразки в них, перемоги та поразки у домашніх матчах та на виїзді, кількість забитих та пропущених м'ячів. Таким чином можна зробити припущення, що для прогнозування результатів гри використовувалися саме ці дані.

Таблиця 1.1-опис переваг та недоліків аналогів

Назва аналога	Переваги	Недоліки
FiveThirtyEight	Веб ресурс надає детальний опис алгоритму, який використовується для прогнозів результатів спортивних змагань, а також перелік даних, на яких ці прогнози засновані. Результатом прогнозу є не тільки перемога чи поразка тієї чи іншої команди, а й зміна індексів рейтингу цієї команди, які можна використовувати для ефективного прогнозу не	Прогнози надаються для обмеженої кількості спортивних, отже неможливо отримати прогноз результатів змагань, які не мають світової популярності.

	тільки результату окремих ігор, а і плину чемпіонатів від початку і до кінця.	
KickForm	Веб ресурс надає не тільки вірогідність перемоги та поразки тієї чи іншої команди, а і деякі додаткові прогнози, такі як вірогідність певного рахунку, кількість м'ячів, забитих за весь матч обома командами, вірогідність того, що обидві команди заб'ють хоча б один гол.	Веб ресурси не надають жодної інформації що до того, який саме алгоритм був використаний або на основі яких даних був зроблений той чи інший прогноз. Таким чином, користувач ніяк не може оцінити значущість значень, які пропонує сайт. Прогнози надаються для
Windrawwin	Веб ресурс надає імовірність перемоги тієї чи іншої команди або нічий, а також деяку інформацію що до рівня гри команди протягом турніру. До цієї інформації відносяться дані що до кількості перемог вдома і на виїзді, кількості забитих і	обмеженої кількості футбольних ліг, отже неможливо отримати прогноз результатів змагань, які не мають світової популярності.

	пропущених м'ячів і так далі.	
--	-------------------------------	--

Висновки до першого розділу

Аналіз предметної сфери показав недостатню розповсюдженість програмних аналогів на ринку, при великому попиті на даний вид діяльності. Лише один з розглянутих аналогів надає опис алгоритму та даних, які він використовує для прогнозування результатів спортивних змагань. Усі інші програмні засоби, що знаходяться у вільному доступі, не надають жодної інформації що до використовуваних ними даних та алгоритмів. Таким чином якість наданих прогнозів залишається сумнівною.

Втім навіть ті програмні продукти, які зазначають алгоритм своєї роботи, не використовують для прогнозування МАІ. Втім розглянуті дослідження у сфері використання МАІ чітко говорять про перспективність даного напрямку. Результати дослідження щодо використання МАІ для прогнозування результатів спортивних змагань можуть дати відповідь на питання, що до доцільності використання МАІ для вирішення даної задачі. Тому варто провести серію експериментів, яка продемонструє ефективність даного алгоритму для продукування прогнозів.

2 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ ДЛЯ ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ СПОРТИВНИХ ЗМАГАНЬ НА ОСНОВІ КОНСТРУКТИВНО-ПРОДУКЦІЙНОГО ПІДХОДУ І МЕТОДУ АНАЛІЗУ ІЄРАРХІЙ

2.1 Внутрішнє проектування

2.1.1 Опис функціональних характеристик

Програмний комплекс виконує прогноз результатів футбольних матчів за допомогою методу аналізу ієрархій, надаючи можливість зробити прогноз починаючи з туру, номер якого вказує користувач. Виконавши прогноз після кожного туру, починаючи з того, який був вказаний користувачем, програма зберігає кількість правильних та хибних прогнозів. На основі цих даних програма підраховує точність прогнозування в залежності від кількості наявних даних. На останньому етапі програма будує графік залежності точності прогнозування. Код програми написано на мові програмування Visual Basic for Applications(VBA). Для зберігання результатів даних та дослідів використовуються файли у форматі .xlsm. Цей формат був обраний, так як саме він дозволяє працювати з інструментами розробника для збереження і використання макросів.

Програмний додаток має забезпечувати наступний функціонал

- обробити вхідні дані, трансформували їх із списку ігор у турнірну таблицю.
- створити окрему таблицю, яка буде зберігати лише результати тих ігор, які були зіграні за задану кількість турів.
- побудувати таблицю МАІ, використовуючи кількість забитих і пропущених м'ячів для розрахунку вагових коефіцієнтів відносної сили команд.

- за допомогою транзитивного замикання визначити відносну силу команд, ігри між якими ще не були зіграні на момент туру, заданого користувачем.
- отримавши прогноз сили команд, порівняти його з фактичними результатами ігор
- розрахувати точність прогнозування
- побудувати графік залежності точності прогнозування від кількості зіграних ігор

2.1.2 Вхідні дані

Дані, необхідні для роботи програми такі:

- Список зіграних матчів. Список складається з назв команд, які брали участь у матчі та рахунку. Формат клітин, які зберігають назви команд – текстовий. У одній клітині знаходяться обидві назви команд, розділені двокрапкою. Їм відповідає клітина з рахунком. Формат клітин, які зберігають рахунок – часовий. Години відповідає кількості м'ячів, забитих першою командою, хвилини – кількості м'ячів, забитих другою. Секунди при роботі програми не враховуються.
- Додаткові дані. До них входять – кількість ігор у одному турі, кількість вже зіграних турів або номер туру, з якого почнеться прогнозування. Ця цифра має бути такою, щоб добуток номеру обраного туру та кількості ігор у турі був меншим або дорівнював кількості ігор, дані за якими були надані. Також до додаткових даних входять кількість турів, які були заплановані. Всі зазначені дані мають бути цілими числами у відповідних клітинах.
- Список команд, які приймали участь у турнірі, у порядку, обраному користувачем

2.1.3 Вихідні дані

Вихідні дані програми такі:

- Таблиця MAI, яка містить вагові коефіцієнти відносної сили команд, розраховані на основі відношення м'ячів, забитих і пропущених у матчах цих команд.
- Таблиця, яка містить результати прогнозу відносної сили команд, зробленого за допомогою транзитивного замикання, а також фактичні результати прогнзованих матчів для більшої зручності їх порівняння користувачем.
- Дані що до точності прогнозування для кожного туру починаючи з того, який був заданий користувачем.
- Графік залежності точності прогнозування від кількості вже зіграних турів.

2.1.4 Модулі програми

Програма складається з декількох модулів, які являють собою макроси, написані на мові програмування VBA.

Перший модуль – CompleteTheLeagueTable

Він виконує побудову турнірної таблиці, використовуючи список ігор. В шапці таблиці як по горизонталі, так і по вертикалі, знаходяться назви команд, які приймали участь у турнірі, в однаковому порядку. Цей порядок визначається користувачем при введенні цих команд. Таблиця заповнюється таким чином, що на перетині рядка і стовпчика двох різних команд буде зазначено рахунок за результатом їх зустрічі, або клітинка буде пустою, якщо команди іще не зустрічалися одна з одною. Клітини на перетині рядка і стовпця однієї і тієї ж команди не використовується, тому залишається пустим. Клітини таблиці знаходяться у часовому форматі – години та хвилини. Години відповідають кількості м'ячів, забитих першою командою, хвилини – кількості м'ячів, забитих другою.

Другий модуль – SecondTable

В цілому він відповідає першому модулю. Він заповнює другу турнірну таблицю, але відрізняється тим, що таблиця буде заповнена від початку і до кінця всіма іграми, які були зіграні протягом турніру. Вона необхідна для подальшого порівняння прогнозованих результатів із фактичними.

Третій модуль – LoseWinColors

Цей модуль позначає пусті клітини у першій таблиці різними кольорами, засновуючись на даних із другої таблиці. Зеленим кольором, якщо перша команда перемогла, жовтим – у разі нічиєї, і червоним, якщо перша команда програла.

Четвертий модуль – ZapMAI

Цей модуль створює нову таблицю на основі даних із першої таблиці. Він використовує рахунок зустрічей тих команд, результати яких відомі на момент туру, що обробляється для розрахунку вагового коефіцієнту відносної сили команд. Вагові коефіцієнти являють собою цілі числа від одного до дев'яти, якщо перша команда сильніша, і від 1 до 1/9, якщо слабша. Якщо дані щодо результату зустрічі команд відсутні, клітина залишається пустою.

П'ятий модуль – Interpol

Цей модуль виконує інтерполяцію для розрахунку відносної сили команд, які ще не зустрічалися одна з одною. Для цього він використовує відомі дані відносної сили тих команд, відносна сила яких прогнозується, та всіх інших команд. В результаті роботи цього модулю отримується повністю заповнена таблиця з ваговими коефіцієнтами відносної сили команд.

Шостий модуль – LoseWinByMAI

Цей модуль на основі даних із третьої таблиці доповнює дані у першій таблиці, заповнюючи пусті клітини спрогнозованими результатами. Важливо зазначити, що результатом прогнозу є не кількість голів, які мають забити перша і друга команди, а те, яка з команд сильніша, тобто має забити більше голів при зустрічі цих двох команд. Якщо за прогнозом перша команда сильніша, у клітину

записується знак «+», якщо вона слабкіша – знак «-». Якщо за прогнозом команди рівні, в таблицю записується знак «=».

Сьомий модуль – accuracy

Для його роботи необхідно, щоб перша таблиця вже була заповнена прогнозами, і кольоровими позначками фактичних результатів. Цей модуль розраховує точність прогнозу для даного туру, а саме відношення кількості правильних прогнозів до загальної кількості прогнозованих даних.

Восьмий модуль – GraphData

GraphData – останній модуль програми, який автоматизує процес дослідження впливу кількості вже відомих результатів ігор на точність прогнозу. Цей модуль покроково викликає усі вище вказані модулі для отримання дослідницьких даних що до точності прогнозів результатів спортивних змагань за допомогою MAI. Ця процедура повторюється починаючи з прогнозу результатів чемпіонату на основі даних туру, заданого користувачем. В кожному циклі модуль повторює процедуру для нового туру. Останнім туром, для якого робиться прогноз є тур, після якого все ще відсутні дані що до результатів зустрічі хоча б однієї пари команд. Результатом роботи модуля є набір даних, які будуть використані для побудови графіку залежності точності прогнозування результатів спортивних змагань від кількості вже зіграних турів.

2.2 Зовнішнє проектування

Як при розробці програмного забезпечення, так і при проведенні досліджень важливу роль відіграє вибір мови програмування, на якій здійснюється написання програми. Обрана мова програмування має надавати можливість реалізації усіх функціональних вимог програми якомога більш оптимальним шляхом. Не менш важливим етапом є вибір середовища програмування. Різні програми надають різні можливості що до налагодження роботи програмного забезпечення, що розробляється, різні можливості

моніторингу ресурсів, які використовує програма, а також різний ступінь допомоги програмісту при розробці у вигляді підказок, автоматичних підстановок, вказівок на помилки і так далі.

Для задачі розробки алгоритму з використанням методу аналізу ієрархій важливо провести детальний аналіз із різними аналогами, які можуть за тими чи іншими аспектами бути кращими у роботі з табличними даними. Іншим важливим питанням при проведенні досліджень є робота з графіками. Обрана середа розробки має надавати необхідні можливості для наочної демонстрації результатів досліджень. Робота з графіками має бути зручною та інтуїтивно зрозумілою. Має бути можливість побудови різних видів графіків для більш гнучкої демонстрації отриманих експериментальних даних.

2.2.1 Середовище розробки і мова програмування

В якості середовища розробки був обраний табличний процесор Microsoft Office Excel. Програма для роботи з таблицями Microsoft Excel була розроблена компанією Microsoft а її перша версія для операційної системи Windows була випущена у 1987 році. Починаючи з першої версії і до актуальної версії Microsoft Office Excel 2019 програма постійно удосконалювалася, додаючи велику кількість нових функцій користувацького інтерфейсу, в той самий час залишаючись зручним інструментом для роботи з таблицями масового користувача.

Наступні фактори стали причиною вибору програми Microsoft Office Excel в якості середовища розробки:

- Метод аналізу ієрархій за своєю суттю спирається на велику кількість роботи з таблицями. Microsoft Excel надає широкі можливості для створення, редагування, та зміни зовнішнього виду таблиць, а також надає велику кількість вбудованих функцій для обробки даних в цих таблицях.

- Microsoft Excel надає широкий набір інструментів для відображення дослідницьких даних, включаючи побудову великої кількості різних типів графіків. Важливим фактором є можливість користувача редагувати елементи графіків таким чином, щоб якомога краще продемонструвати результати проведених досліджень.
- Починаючи з 1993 року в Microsoft Excel інтегровано мову програмування Visual Basic for Applications. VBA є гарним додатком до програми і надає широкий спектр можливостей для автоматизації процесу обробки даних і проведення досліджень. А поява у останніх версіях цієї мови елементів об'єктно-орієнтованого програмування іще більше розширює можливості VBA.
- Microsoft Excel є стандартним вибором для роботи з таблицями у багатьох сферах. Таким чином велика кількість користувачів з різним рівнем володіння програмою матиме можливість скористатися розробленим програмним додатком або результатами проведених досліджень.
- Формат файлів, у яких Microsoft Excel зберігає дані та програмні елементи є стандартним для багатьох інших редакторів таблиць. Таким чином імпорт електронних таблиць у інші програм на різних операційних системах та пристроях є дуже зручним і простим процесом.

В якості мови програмування була обрана мова Visual Basic for Applications. Це мова програмування, яка вбудована у деякі продукти Microsoft Office, такі як Microsoft Word, Microsoft Excel, Microsoft Access, а також до деяких інших програм, таких як AutoCAD, CorelDRAW і так далі. Вона була створена у 1993 році компанією Microsoft на базі мови програмування Visual Basic для розширення функціоналу їх програмних продуктів елементів пакету Microsoft Office.

Наступні фактори стали причиною вибору мови Visual Basic for Applications в якості мови програмування:

- Visual Basic for Applications інтегрована у програму Microsoft Office Excel, що дозволяє суттєво розширити можливості програміста по створенню програм з комплексними розрахунками, автоматизацію проведення досліджень і так далі.
- Відносна простота мови, що дозволяє навіть недосвідченому програмісту, не знайомому із нею, швидко розібратися у нюансах її застосування.
- Інтеграція мови у табличний процесор Microsoft Office Excel дозволяє користувачам, не знайомих із програмуванням, використовувати програми, написані іншими програмістами мовою Visual Basic for Applications у вигляді макросів.
- Автоматизація взаємодії між різними продуктами пакету Microsoft Office.

2.2.2 Обґрунтування експериментального методу дослідження ефективності застосування МАІ для прогнозування результатів командних змагань

Завдання розробки нових ефективних алгоритмів прогнозування результатів спортивних змагань набуває все більшої актуальності з кожним роком. З урахуванням швидкого розвитку сучасних технологій, що відповідають за зберігання і розповсюдження великих об'ємів інформації, постає питання, які саме алгоритми могли б бути найбільш ефективними для виконання прогнозу. Для проведення дослідження було обрано метод аналізу ієрархій, так як він надає можливість порівнювати команди за їх силою в контексті цілого чемпіонату в цілому, .

2.2.2.1 Аналіз МАІ

Метод аналізу ієрархій був розроблений Томасом Л. Сааті в 1970-х роках, а у співпраці з Ернестом Форманом 1983 році був розроблений алгоритм вибору експертів. На сьогоднішній день МАІ використовується у всьому світі у великій

кількості різних ситуацій для прийняття рішень у таких сферах як бізнес, промисловість, державне управління, освіта, охорона здоров'я і так далі.

Сутність МАІ полягає у парному порівнянні між собою факторів, які впливають на прийняття рішення. Оцінка кожного з факторів визначається експертом, базуючись на його власному досвіді і знаннях. Ця оцінка виставляється кожному фактору у числовому еквіваленті. Судження експертів виставляються відповідно до шкали відношень Сааті і поділяються на:

Таблиця 2.1 Шкала відношень Сааті

Оцінка	Значення оцінки
1	Альтернативи однаково цінні
2	Проміжна оцінка між абсолютною рівністю і невеликою перевагою
3	Незначна перевага альтернативи
4	Проміжна оцінка між незначною і середньою перевагою
5	Середня перевага альтернативи
6	Проміжна оцінка між середньою і сильною перевагою
7	Сильна перевага альтернативи
8	Проміжна оцінка між сильною і абсолютною перевагою
9	Абсолютна перевага альтернативи

З отриманих парних порівнянь складається квадратна матриця, де елемент a_{ij} є відношенням оцінок двох елементів з відповідними індексам. Головна діагональ є одиничною, так як на ній кожен фактор порівнюється сам із собою, і таке відношення дорівнює одиниці. Коефіцієнти i та j змінюються від одного до n , де n це загальна кількість факторів. Так як кожен фактор порівнюється один з

одним, отримана матриця буде зворотною симетричною, тобто клітина i у стовбці j містить оцінку переваги елемента i над елементом j , а клітина j у стовбці i містить зворотнє число

$$a_{ij} = \frac{1}{a_{ji}}. \quad (1)$$

Індекс узгодженості(CI) є критерієм якості роботи експерта, тобто чи не суперечить він сам собі. Перевірка на транзитивну узгодженість дозволяє перевірити значущість експертних оцінок. Наприклад якщо експерт оцінив фактор «А» як кращий за фактор «В», а фактор «В» як кращий за фактор «С», то логічним припущенням буде те, що фактор «А» також має бути кращим за фактор «С». Погана узгодженість матриці може призвести до значних проблем у роботі алгоритму і навіть унеможливити процес пошуку найкращого рішення.

Індекс узгодженості розраховується за формулою:

$$CI = \frac{\lambda - n}{n - 1} \quad (2)$$

Де λ – власне значення максимального власного вектора.

n – кількість альтернатив.

Індекс узгодженості порівнюється із середнім значенням узгодженості(RI), отриманим при заповненні матриці випадковими величинами.

Приклад значень середньої узгодженості для таблиць різного розміру(n – кількість альтернатив).

Таблиця №2.2 Значення середньої узгодженості для таблиць різного розміру

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0,58	0,90	1,12	1,24	1,32	1,41	1,45	1,49

Розділивши значення індексу узгодження на значення середньої узгодженості, ми отримуємо відношення узгодженості(CR), яке і буде означати те, наскільки експерт коректний у власних порівняннях.

$$CR = \frac{CI}{RI} \quad (3)$$

В залежності від того, яке дослідження проводиться, може бути обраний різний задовільний рівень узгодженості. У більшості випадків вона вважається задовільною, якщо CR менше 0,1, інколи межі допустимої узгодженості можуть бути розширені до величини 0.2. У випадку, якщо відношення узгодженості більше за ці величину, оцінки такого експерта вважаються некоректним, і виключається із досліджень.

2.2.2.2 Методика застосування МАІ

Застосування МАІ поділяється на 5 кроків.

- Крок перший: побудова моделі задачі у вигляді ієрархії. До цієї моделі входять: ціль, критерії та альтернативи. Кожен елемент ієрархії, що знаходиться вище, вважається батьківським для всіх елементів, які знаходяться на один рівень нижче нього і пов'язані з ним, а елементи що знаходяться нижче, в свою чергу є дочірніми

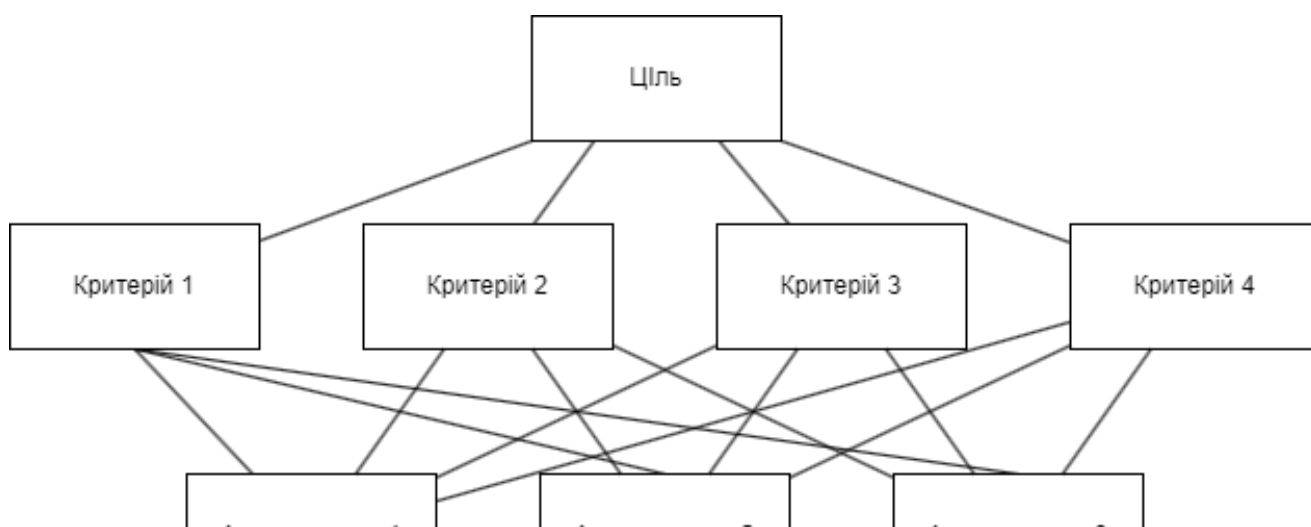


Рисунок 2.2 Ієрархічна структура задачі

елементами для пов'язаних з ними елементів, які знаходяться на один рівень вище.

- Крок другий: необхідно визначити пріоритети, тобто визначити вагові коефіцієнти відносної важливості або переваги того чи іншого елемента ієрархічної структури. Використовуючи шкалу відношень Сааті будуються матриці попарних порівнянь.

Таблиця 2.3 матриця попарних порівнянь

	A_1	A_2	...	A_n
A_1	1	K_{12}	...	K_{1n}
A_2	$\frac{1}{K_{12}}$	1	...	K_{2n}
...	1	...
A_n	$\frac{1}{K_{1n}}$	$\frac{1}{K_{1n}}$...	1

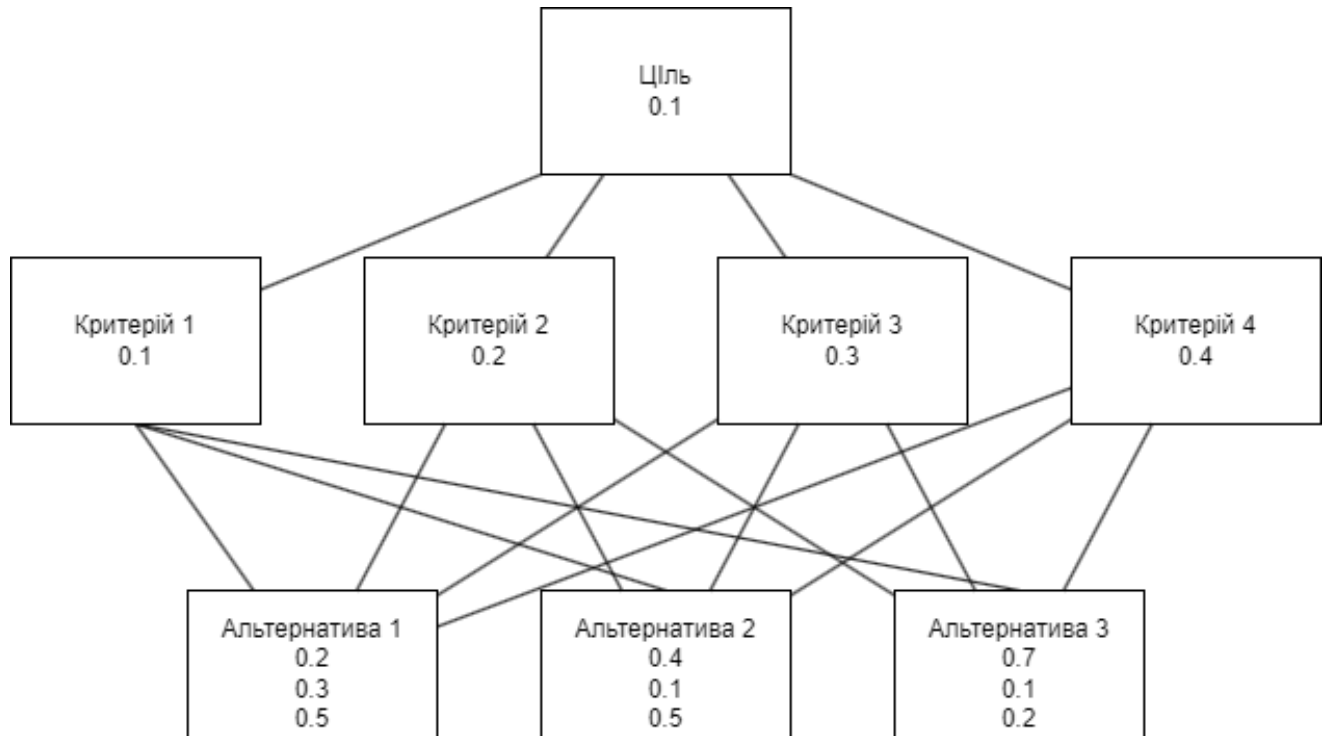


Рисунок 3.2 Ієрархічна структура задачі з пріоритетами

- Третій крок: знаходження вектору пріоритетів, яке поділяється на три кроки:
 - o Знайти середнє геометричне для кожного рядка матриці парних зрівнянь.
 - o Знайти суму всіх знайдених елементів.
 - o Отримати вектор локальних пріоритетів, розділивши кожне середнє геометричне на їх суму.
- Четвертий крок: перевірка на узгодженість, розраховавши відношення узгодження. У випадку, якщо відношення узгодження буде незадовільним, має бути проведена нова експертна оцінка.
- П'ятий крок: на базі парного порівняння альтернатив вагових коефіцієнтів відносної важливості критеріїв кожній з альтернатив присвоюється оцінка. На основі цих оцінок приймається фінальне рішення.

2.2.2.3 Застосування МАІ для прогнозування результатів командних змагань

Ключовим елементом для використання методу аналізу ієрархій є побудова матриці попарних порівнянь. В той самий час, одним з основних способів показати перебіг командного змагання є турнірна таблиця. Така таблиця може зберігатися у формі матриці з розміром n на n , де n це кількість команд, які приймають участь у змаганнях. Ця матриця буде утримувати у собі результати ігор між кожною парою команд, а саме інформацію про кількість голів, забитих першою та другою командами. При цьому головна діагональ матриці, на якій перетинаються рядки і стовпці однієї і тієї ж самої команди, буде складатися з одиниць, або може бути пустою. Отже побудована таким чином матриця за своїми властивостями подібна до матриці попарних порівнянь, яка використовується при роботі алгоритму МАІ.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
Дома / На виїзді	АРС	АСТ	БЛВ	БСЛ	ВВА	ВУЛ	КПР	ЛИВ	МС	МЮ	НОР	НЬО	САН	СТО	СУО	ТОТ	УНГ	ФУЛ	ЧЕЛ	ЗВЕ	
3	Арсенал		03:00	07:01	03:00	03:00	01:01	01:00	00:02	01:00	01:02	03:03	02:01	02:01	03:01	01:00	05:02	01:02	01:01	00:00	01:00
4	Астон Вилла	01:02		03:01	01:02	01:02	00:00	02:02	00:02	00:01	00:01	03:02	01:01	00:00	01:01	00:02	01:01	02:00	01:00	02:04	01:01
5	Брайтон																				
6	Брайтон	04:03	01:01		01:02	01:02	01:02	03:02	02:03	00:04	00:02	02:00	00:02	02:00	01:02	04:02	01:02	00:01	03:01	00:01	00:01
7	Вест Бромвіч	00:00	01:02	02:01		02:02	01:01	02:01	03:01	02:03	00:05	01:02	00:02	00:02	05:00	01:01	01:04	01:02	00:03	01:05	00:02
8	Вест Бромвіч	02:03	00:00	03:00	02:01		02:00	01:00	00:02	00:00	01:02	01:02	01:03	04:00	00:01	01:02	01:03	01:02	00:00	01:00	00:01
9	Вулвергетон	00:03	02:03	00:02	02:03	01:05		00:03	00:03	00:02	00:05	02:02	01:02	02:01	01:02	02:02	00:02	03:01	02:00	01:02	00:00
10	Вулвергетон	02:01	01:01	01:01	00:04	01:01	01:02		03:02	02:03	00:02	01:02	00:00	02:03	01:00	03:00	01:00	03:01	00:01	01:00	01:01
11	Ливерпуль	01:02	01:01	01:01	03:01	00:01	02:01	01:00		01:01	01:01	01:01	03:01	01:01	00:00	00:00	00:00	01:02	00:01	04:01	03:00
12	Манчестер Сіті	01:00	04:01	03:00	02:00	04:00	03:01	03:02	03:00		01:00	05:01	03:01	03:03	03:00	04:00	03:02	03:00	03:00	02:01	02:00
13	Манчестер Сіті	08:02	04:00	02:03	03:00	02:00	04:01	02:00	02:01	01:06		02:00	01:01	01:00	02:00	02:00	03:00	05:00	01:00	03:01	04:04
14	Норвіч Сіті	01:02	02:00	03:03	02:00	00:01	02:01	02:01	00:03	01:06	01:02		04:02	02:01	01:01	03:01	00:02	01:01	01:01	00:00	02:02
15	Ньюкасл Юнайтед	00:00	02:01	03:01	02:00	02:03	02:02	01:00	02:00	00:02	03:00	01:00		01:01	03:00	00:00	02:02	01:00	02:01	00:03	02:01
16	Сандерленд	01:02	02:02	02:01	02:02	02:02	00:00	03:01	01:00	01:00	00:01	03:00	00:01		04:00	02:00	00:00	01:02	00:00	01:02	01:01
17	Сток Сіті	01:01	00:00	03:01	02:02	01:02	02:01	02:03	01:00	01:01	01:01	01:00	01:03	00:01		02:00	02:01	02:02	02:00	00:00	01:01
18	Суссекс Сіті	03:02	00:00	03:00	03:01	03:00	04:04	01:01	01:00	01:00	00:01	02:03	00:02	00:00	02:00		01:01	00:00	02:00	01:01	00:02
19	Тоттенхем	02:01	02:00	02:00	03:00	01:00	01:01	03:01	04:00	01:05	01:03	01:02	05:00	01:00	01:01	03:01		03:01	02:00	01:01	02:00
20	Уїган Атлетик	00:04	00:00	03:03	01:03	01:01	03:02	02:00	00:00	00:01	01:00	01:01	04:00	01:04	02:00	00:02	01:02		00:02	01:01	01:01
21	Фулхем	02:01	00:00	01:01	02:00	01:01	05:00	06:00	01:00	02:02	00:05	02:01	05:02	02:01	02:01	00:03	01:03	02:01		01:01	01:03
22	Челсі	03:05	01:03	02:01	03:00	02:01	03:00	06:01	01:02	02:01	03:03	03:01	00:02	01:00	01:00	04:01	00:00	02:01	01:01		03:01
23	Зевітон	00:01	02:02	01:01	01:02	02:00	02:01	00:01	00:02	01:00	00:01	01:01	03:01	04:00	00:01	01:00	01:00	03:01	04:00	02:00	

Рисунок 2.3 Турнірна таблиця

Таким чином постає питання, чи можна використовувати МАІ для спроб спрогнозувати результати спортивних змагань. Для цього, користуючись деякими математичними перетвореннями, можна спробувати перетворити рахунок, який був у матчі між двома командами, у ваговий коефіцієнт відносної сили цих команд. Такі коефіцієнти можуть бути використані у подальшій роботі алгоритму для розрахунку вагових коефіцієнтів відносної сили команд, які ще не грали між собою у даному турнірі.

Із цього постає друге питання, яке стосується використання експертної оцінки для визначення вагових коефіцієнтів відносної сили двох команд. Така оцінка завжди є суб'єктивною, заснованою на власних знаннях та досвіді кожного окремого експерта. Отже така оцінка сильно залежить від людського фактору, який і так, разом із великою кількістю інших випадкових імовірностей, значною мірою впливає на результат тих чи інших командних змагань. Іншим недоліком використання експертної оцінки людей може бути ризик того, що оцінки різних експертів будуть кардинально відрізнятися одна від одної. Такі оцінки все ще можуть відповідати вимогам алгоритму МАІ що до якості експертної точки зору, відносно узгодження може бути нижчим за максимально значення, яке визначено як максимально допустиме для даної задачі.

Отже друге питання полягає в тому, чи можна використовувати результати зустрічей двох учасників турніру як основу для розрахунку відносної сили двох команд. В такому разі, експертом для нашої матриці попарних порівнянь буде виступати сама природа. Такий підхід повністю виключає вплив людського фактору при побудові матриці попарних порівнянь. Про те невідомо, яким саме чином це вплине на загальну точність роботи алгоритму прогнозування. Саме тому це питання потребує детального дослідження.

Висновки до другого розділу

В цьому розділі було проведено аналіз особливостей та необхідних характеристик середі та мови програмування, які будуть використовуватися для подальшого проведення досліджень. Спроектовано структуру програми, описані ключові вимоги до окремих її модулів. Також був проведений аналіз особливостей роботи методу аналізу ієрархій і обґрунтовано необхідність дослідження можливості його використання для прогнозування результатів командних змагань.

3 РОЗРОБКА І ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ КОМАНДНИХ ЗМАГАНЬ ЗА ДОПОМОГОЮ МАІ

3.1 Застосування принципів програмування

При розробці будь якого програмного забезпечення необхідно враховувати особливості різних етапів життєвого циклу програмного продукту. Це має величезне значення, адже навіть після завершення розробки програми і введення її у експлуатацію або навіть виведення її на ринок програма може й надалі супроводжуватися програмістами. Для цього можуть бути різні підстави. В деяких випадках з'являється необхідність у модифікації програми для того, щоб вона відповідала функціональним вимогам, які змінилися. У іншому випадку може з'явитися необхідність для повного перероблення програмного продукту, зміни чи видалення окремих частин чи інтеграція програми із іншим програмним забезпеченням. Не виключена і вірогідність наявності помилок в початковій програмі, які мають бути виправлені на етапі її експлуатації.

Таким чином на стадії розробки програмного забезпечення необхідно приділяти увагу використанню патернів та парадигм програмування. Це дозволить не тільки спростити розробку програмного продукту, який буде відповідати поставленим функціональним задачам, а й суттєво спростити його подальше використання на етапі експлуатації. Такий підхід також може продовжити час життєвого циклу програми, що між іншим може збільшити потенційні прибутки від її реалізації на ринку.

Отже першим кроком при програмуванні додатку має бути обрання парадигми програмування. Перш за все, обрана методологія має допомогти нам вирішити задачі, поставлені технічним завданням. Під час створення програмного забезпечення, найбільше значення мають два фактори: складність програмування і ресурсна ефективність роботи програми. Нажаль часто ці фактори слабо чи зовсім не сумісні. Таким чином з'являється необхідність

детального аналізу функціональних вимог програми.

У даній роботі стоїть задача реалізувати алгоритм МАІ для подальшого прогнозування результатів командних змагань. За своєю сутністю, метод аналізу ієрархій включає в себе набір математичних операцій, а також обробку матриць. Таким чином до оброблюваних даних входять лише числа та матриці. Це означає, що використання парадигм програмування, на кшталт ООП не є доцільним. Це лише призведе до ускладнення програми непотрібними сутностями та збільшення часу роботи програми через вищий рівень абстракції.\

В той же час є необхідність спростити процес розробки, оптимізувавши його для можливості виконання командою. Цього можна досягти, розбивши програму на маленькі процедури, кожна з яких буде виконувати свою, конкретно поставлену задачу. Це дасть програмі зрозумілу структуру та значно облегшить процес тестування.

Проаналізувавши поставлені вимоги, в якості парадигми програмування було обрано процедурне програмування. Концептуально це програмування, при якому послідовно виконуваний код можна об'єднати у більші за розміром цілісні елементи – підпрограми або процедури. Кожна з таких процедур може виконувати свою невелику задачу.

Переваги та особливості процедурного програмування:

- Простота розробки і зрозумілість коду: У структурі коду використовуються підпрограми(процедури), кожна з яких має на меті виконання однієї конкретної задачі. Такий підхід дозволяє спростити процес розробки і покращити структуру програми, що в свою чергу робить її код більш зрозумілим навіть для програмістів, не знайомих з даною програмою.
- Можливість написання великих програм: розбиття великої програми на маленькі підзадачі дозволяє полегшити розробку і подальшу підтримку на етапі експлуатації.

- Легкість у тестуванні та відлагодженні: розділення на маленькі процедури дозволяє легко відстежити джерело помилок, тим самим суттєво спрощуючи процес налагодження та тестування програми.
- Доступність та простота вивчення: концептуальна простота процедурного програмування суттєво спрощує процес вивчення тих мов, які користуються цією парадигмою.
- Часова ефективність: відсутність додаткового навантаження на обчислювальну спроможність через відсутність високого рівня абстракції, властивого деяким іншим парадигмам програмування, таким як наприклад ООП.

Наступним важливим елементом при розробці програмного забезпечення є використання принципів проектування, які спрямовані на створення надійних, ефективних та зручних у використанні систем.

До таких принципів відносяться:

- Single Responsibility Principle – SRP: кожен модуль системи має мати одну єдину функцію. Дотримання цього принципу буде означати, щ при внесенні змін у дин елемент системи, інші частини не зазнаватимуть жодного впливу. Це дозволить кожному елементу мати власну причину на існування, а отже зробить систему більш підтримуваною та гнучкою для доповнення програми новим функціоналом без необхідності редагувати вже існуючі елементи.
- Don't Repeat Yourself – DRY: не повторюй себе. Цей принцип у деякому сенсі перекликається із принципом SRP і заохочує уникати повторення коду у програмі. Замість дублювання одного і того ж самого коду у різних місцях програміст має витратити зусилля на реалізацію функцій, які будуть використовуватися декілька разів протягом виконання програми. Такий принцип позитивно впливає

на якість програмного продукту в цілому. Його застосування дозволяє значно спростити процес пошуку і виправлення помилок. Іншою перевагою застосування цього принципу є значне спрощення подальшої підтримки програми на етапі експлуатації.

- You Ain't Gonna Need It – YAGNI: тобі це не знадобиться. Це принцип, який говорить про те, що не варто включати у програму функціональність, яка на даний момент не є необхідною. Уникання зайвої складності, такої як додавання непотрібного функціоналу, чи функціоналу, необхідність якого на даному етапі розробки не є очевидною не тільки збільшує загальну складність програмного продукту, а й викликає за собою надмірні затрати часу на процес розробки і значно збільшує ризик появи помилок. Це в свою чергу значно збільшує час, необхідний на тестування програми, тобто з'являється такий собі мультиплікативний ефект.

3.2 Проектування архітектури системи.

В даному розділі описано процедури, які були створені для реалізації алгоритму та архітектура взаємодії між ними.

Глобальні дані:

- Список всіх ігор, які були зіграні протягом турніру. Список складається із двох клітин: назви команд учасників гри у першій клітині, кількість м'ячів, забитих кожною із команд – у другій .
- Перша турнірна таблиця. Зберігає в собі дані про рахунок зустрічей між парами команд. Розмір матриці - n на n , де n це кількість команд, які приймали участь у турнірі. В залежності від туру, обраного користувачем, матриця може бути не повною. Протягом роботи алгоритму, ця таблиця буде доповнена індикаторами у вигляді кольору, які позначають фактичні результати ігор, а також

умовними позначками, які будуть позначати прогнозовані результати цих ігор.

- Друга турнірна таблиця. Ця таблиця буде повністю заповнена. Вона необхідна для подальшого порівняння прогнозованих результатів із фактичними.
- Таблиця МАІ. Таблиця попарних порівнянь, створених на основі рахунків тих зустрічей команд, результати яких вже відомі.

Реалізовані процедури:

`CompleteTheLeagueTable` – створює турнірну таблицю, використовуючи список ігор.

Вхідні дані:

- Список ігор.

Вихідні дані:

- Матриця для заповнення першої таблиці.

`SecondTable` – в цілому аналогічна першій процедурі, але відрізняється тим, що створює матрицю для заповнення другої турнірної таблиці.

`LoseWinColors` - позначає пусті клітини у першій таблиці різними кольорами таким чином, щоб перемозі першої команди відповідав зелений колір, нічий – жовтий, а поразці – червоний.

Вхідні дані:

- Матриця, яка зберігає в собі дані про рахунок зустрічей між усіма парами команд.

Вихідні дані:

- Модифікатор кольору відсутніх елементів першої матриці.

`ZarMAI` - створює таблицю, яка містить у собі попарні порівняння.

Вхідні дані:

- Матриця, яка зберігає в собі дані про рахунок зустрічей між парами команд.

Вихідні дані:

- Матриця попарних порівнянь, яка містить у собі лише ті елементи, які відповідають відомим результатам зустрічей команд.

Interpol - виконує інтерполяцію для розрахунку відносної сили команд, які ще не зустрічалися одна з одною.

Вхідні дані:

- Матриця попарних порівнянь.

Вихідні дані:

- Дані що до відносної сили команд, які були відсутні у початковому стані матриці попарних порівнянь.

LoseWinByMAI - доповнює дані у першій таблиці, заповнюючи пусті клітини спрогнозованими результатами.

Вхідні дані:

- Матриця попарних порівнянь.

Вихідні дані:

- Дані що до прогнозу результату зустрічі для тих пар команд, які ще не грали одна з одною.

ассигасу - модуль розраховує точність прогнозу для туру, визначеного користувачем.

Вхідні дані:

- Перша таблиця, яка вже заповнена як результатами відомих ігор, так і спрогнозованими результатами ігор, які ще не були зіграні, а також індикаторами їх фактичного результату.

Вихідні дані:

- відношення кількості правильних прогнозів до загальної кількості прогнозованих даних.

GraphData - автоматизує процес дослідження впливу кількості вже

відомих результатів ігор на точність прогнозу.

Як такі власні вхідні та вихідні дані для цього модуля відсутні. Він

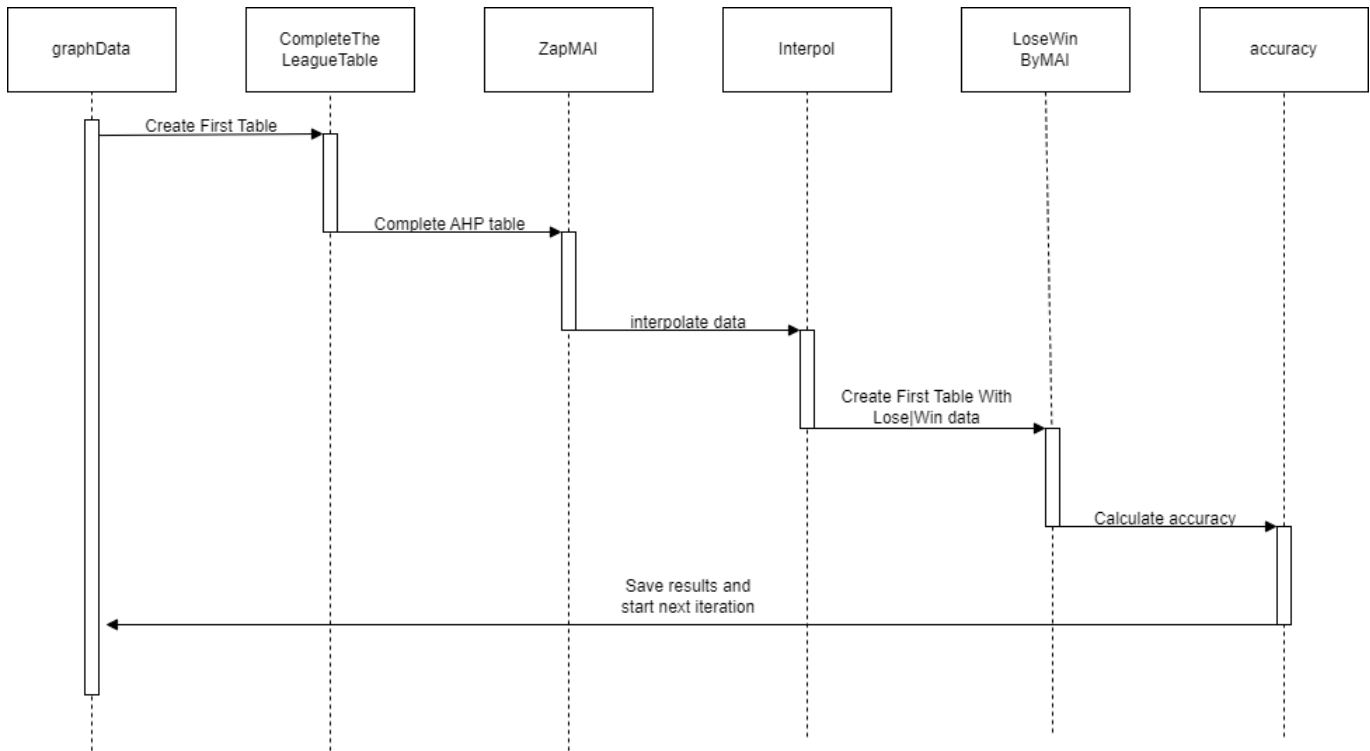


Рисунок 3.1 Діаграма послідовності

відповідає за автоматизовану роботу всіх інших модулів, і в результаті його роботи клітини, необхідні для побудови графіка, будуть заповнені даними що до точності прогнозу для кожного туру.

Окрім автоматичного виконання дослідження, користувачу надається можливість виклику окремих процедур через механізм макросів, вбудований у табличний процесор Microsoft Excel. Таким чином користувач може прослідкувати за результатами роботи програми на кожному етапі проведення експерименту.

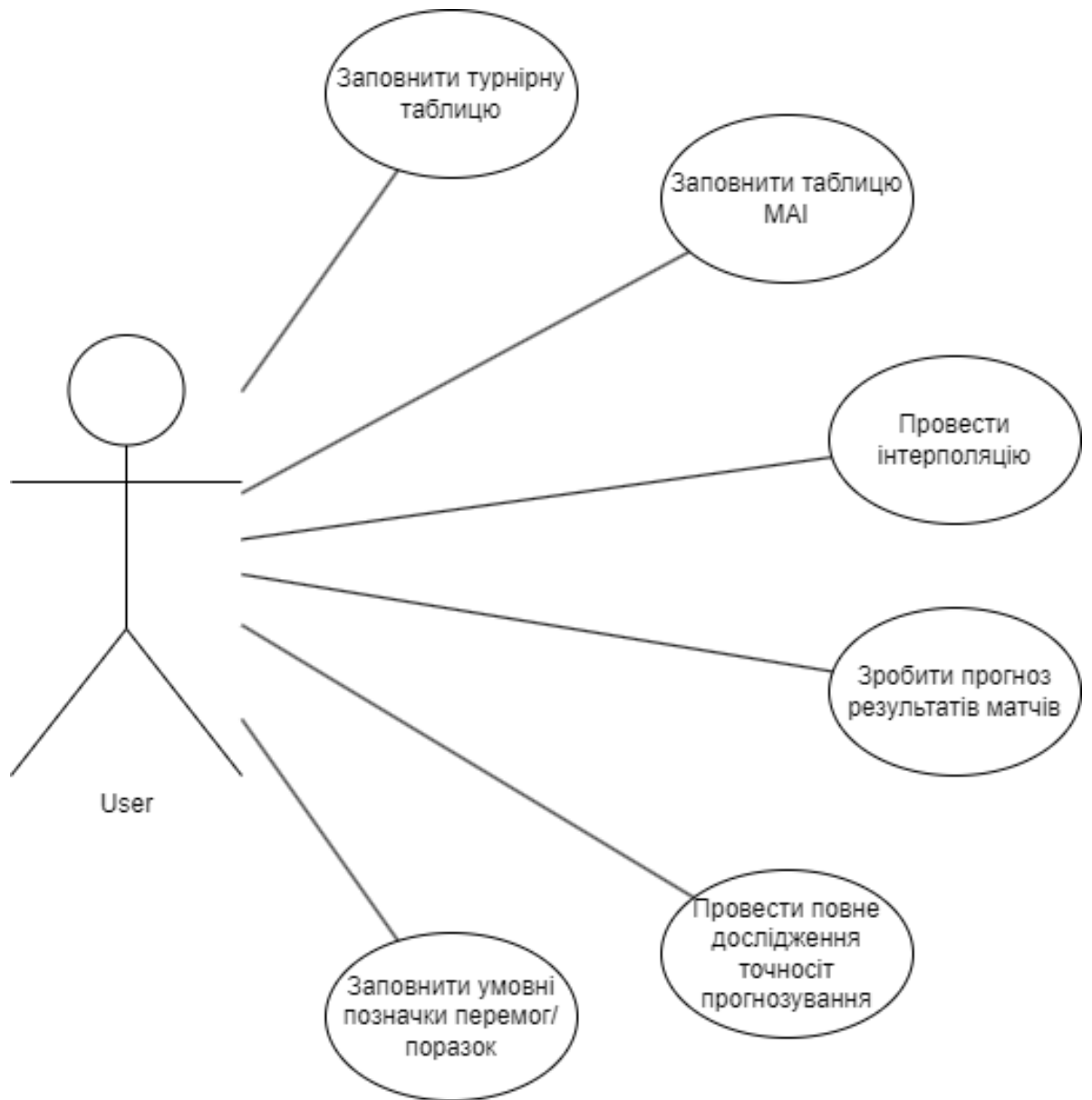


Рисунок 3.2 Діаграма прецедентів

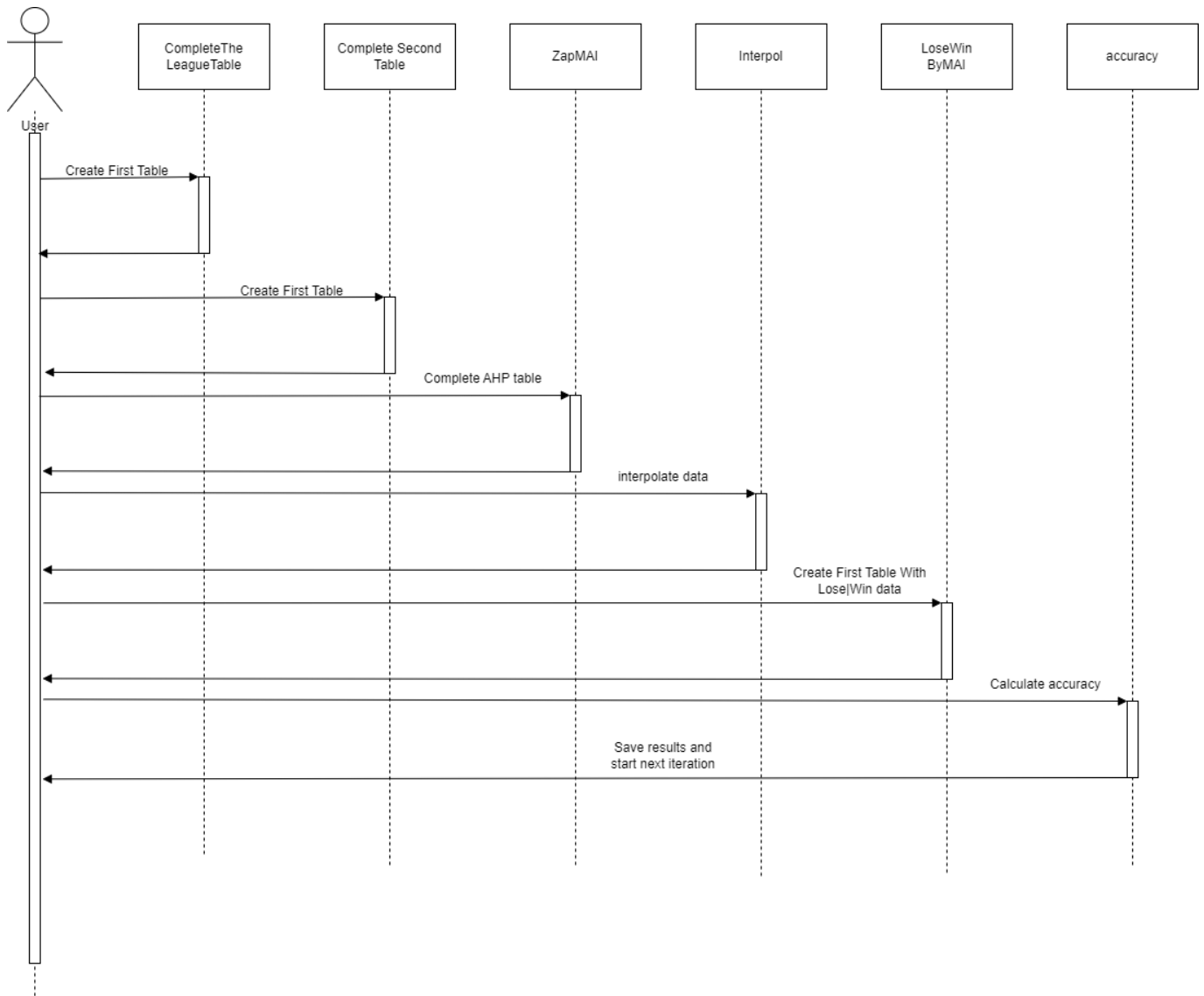


Рисунок 4.3 Діаграма послідовності користувача

3.3 Аналіз методів, використаних для тестування програми

Тестування програмного забезпечення є невід’ємною частиною розробки. Використання різних підходів до тестування, поєднаних в одну ретельно продуману та структуровану систему не тільки допомагає уникнути значної частини помилок, а й забезпечить суттєву економію часу розробника. Ціллю тестування є забезпечення коректності роботи програми, а також перевірка таких факторів як:

- Виконання програмою очікуваних від неї функцій і відповідність їм.

- Стабільність роботи програми при стресових умовах, під високим навантаженням системи, тощо.
- Швидкість виконання, витрата ресурсів комп'ютера та інші параметри продуктивності.
- Вразливості програми від можливих атак.

Перевірити коректність роботи програми можна двома методами: чорної та білої скриньки. Тестування методом чорної скриньки полягає у перевірці коректності роботи окремих модулів або програми в цілому, базуючись на перевірці відповідності вихідних даних вхідним. При цьому тестові дані розробляються на основі специфікації програми чи її функціонального елемента, а також технічного завдання. Характерною особливістю тестування методом чорного ящика є те, що під час його виконання повністю ігноруються внутрішня структура роботи функції чи програми, що тестується. Таким чином тестування методом чорного ящика можна використовувати для перевірки коректності роботи окремих функцій на етапі написання програми, а також для перевірки відповідності програми специфікаціям та технічному завданню вже після завершення процесу розробки. Перевагою такого методу є те, що тестування може проводити не лише кваліфікований програміст, а й простий користувач, якому для перевірки коректності роботи програми достатньо просто знати, що програма, яка перевіряється, має робити.

Тестування білою скринькою, або структурне тестування на основі коду, використовує інформацію про специфіку внутрішньої реалізації програми для проектування тестових сценаріїв. Тестування проводиться спираючись на особливості внутрішньої структури програми та її окремих частин. Суттєвим недоліком такого підходу є необхідність абсолютного розуміння програмістом структури, логіки та функціональних особливостей різних частин програми, що тестується.

Тестування білою скринькою включає в себе такі критерії покриття:

- Рядків коду – для порівняння того, скільки рядків коду виконується при виконанні тесту, та скільки всього рядків налічується у кодї програми в цілому.
- Операторів – перевіряє, щоб кожен оператор було виконано хоча б один раз.
- Гілок коду – визначає, яка кількість гілок в умовних операторах була виконана у порівнянні з загальною кількістю гілок у кодї програми в цілому.
- Умов коду – визначає, яка кількість умов була використана під час проведення тесту, а також чи була кожна така умова використана та невикористана хоча б один раз.

Такий набір критеріїв дозволяє оцінити ефективність тестування та розробити нові набори тестів для більш ефективного покриття всієї програми.

Висновки до третього розділу

Розроблена архітектура інструментального забезпечення для дослідження ефективності використання MAI для прогнозування результатів командних змагань. Завдяки використанню модулів, кожен з яких відповідає за окрему задачу, було передбачено можливість модернізації та розширення функціоналу програмного продукту.

Завдяки використанню загальновідомих шаблонів проектування розробникам з різним рівнем компетенції буде легко зрозуміти функціональні особливості та принципи роботи програми.

Система готова до впровадження і використання.

4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТОЧНОСТІ ПРОГНОЗУВАННЯ РЕЗУЛЬТАТІВ КОМАНДНИХ ЗМАГАНЬ ЗА ДОПОМОГОЮ МЕТОДУ АНАЛІЗУ ІЄРАРХІЙ

4.1 Підготовка до експерименту

В якості експериментальних даних були взяті результати чемпіонатів Англії з футболу з 2008-го по 2018-й рік. В якості прикладу ми розглянемо прогнозування результатів чемпіонату Англії з футболу сезону 2011–го – 2012-го року.

На першому етапі підготовки даних до проведення експерименту необхідно заповнити інформацію про результати ігор даного чемпіонату.

13 августа	БЛЭКБЕ РН : ВУЛ ВЕРХЭМ ПТОН	подробнее» 1:2 (1:1)	
13 августа	УИГАН : НОРВИЧ	1:1 (1:1) подробнее»	
13 августа	КПР : БО ЛТОН	0:4 (0:1) подробнее»	
13 августа	ЛИВЕРП УЛЬ : СА НДЕРЛЕ НД	подробнее» 1:1 (1:0)	
13 августа	ФУЛХЭМ : АСТОН ВИЛЛА	подробнее» 0:0 (0:0)	
13 августа	НЬЮКАС Л : АРСЕ НАЛ	0:0 (0:0) подробнее»	
14 августа	СТОК СИТИ : ЧЕЛСИ	подробнее» 0:0 (0:0)	

Рисунок 4.1 Приклад заповнених даних про результати ігор

Наступним етапом підготовки до експерименту є заповнення турнірної таблиці даними, наявними на момент обраного туру, до прикладу так буде виглядати турнірна таблиця після 20-ти турів чемпіонату.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
Дома / На виїзді	АРС	АСТ	БЛЕ	БОЛ	ВБА	ВУЛ	КПР	ЛИВ	МС	МЮ	НОР	НЬО	САН	СТО	СУО	ТОТ	УІГ	ФУЛ	ЧЕЛ	ЭВЕ		
Арсенал				03:00	03:00	01:01	01:00	00:02						02:01	03:01	01:00			01:01		01:00	
Астон Вилла	01:02		03:01		01:02	00:00		00:02		00:01	03:02	01:01			00:02		02:00					
Блекберн Роверс	04:03			01:02	01:02	01:02			00:04						01:02	04:02	01:02				00:01	00:01
Болтон Уондерерс		01:02				01:01			02:03	00:05	01:02	00:02	00:02	05:00						01:05	00:02	
Вест Бромвіч Альбіон				02:01		02:00		00:02	00:00	01:02					00:01		01:03	01:02	00:00		00:01	
Вулвергемптон Уондерерс							00:03				02:02	01:02	02:01	01:02	02:02	00:02	03:01	02:00	01:02			
Вундербіркенхед		01:01	01:01	00:04	01:01				02:03	00:02	01:02	00:00	02:03							01:00		
Ліверпуль			01:01	03:01		02:01	01:00		01:01	01:01	01:01	03:01	01:01		00:00							
Манчестер Сіті	01:00	04:01			03:01			03:00			05:01	03:01		03:00	04:00		03:00				02:00	
Манчестер Юнайтед	08:02		02:03			04:01			01:06		02:00	01:01	01:00			03:00	05:00			03:01		
Норвіч Сіті	01:02		03:03		00:01		02:01						04:02	02:01	01:01	03:01	00:02		01:01			
Ньюкасл Юнайтед	00:00		03:01		02:03					03:00						00:00	02:02	01:00	02:01	00:03	02:01	
Саутгемптон		02:02	02:01		02:02				01:00				00:01		04:00			01:02	00:00	01:02	01:01	
Сток Сіті		00:00	03:01				02:03	01:00		01:01			01:03				02:01	02:02	02:00	00:00		
Свонсі Сіті		00:00		03:01	03:00		01:01			00:01				00:00	02:00		01:01	00:00	02:00			
Тоттенхем Хотспур	02:01	02:00		03:00	01:00		03:01	04:00	01:05					01:00						01:01	02:00	
Уїган Атлетик	00:04		03:03	01:03			02:00	00:00			01:01			01:04			01:02		00:02	01:01		
Фулгем	02:01	00:00	01:01	02:00			06:00	01:00	02:02	00:05							01:03				01:03	
Челсі	03:05	01:03			02:01	03:00		01:02	02:01			03:01				04:01			01:01		03:01	
Евертон		02:02		01:02		02:01	00:01	00:02		00:01	01:01				00:01	01:00		03:01				

Рисунок 4.2 Приклад вигляду турнірної таблиці перед проведенням експерименту

У результаті ми отримуємо таблиці, яка містить у собі дані про результат тих зустрічей, які вже відбулись. А фактичний результат матчів, які ще не були

зіграні на момент обраного туру, умовно позначений відповідним кольором. На цьому підготовка до експерименту закінчується.

4.2 Проведення експерименту

Першим кроком проведення експерименту є заповнення таблиці попарних порівнянь засновуючись на наявних даних що до результатів футбольних матчів, які вже відбулись.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	Дома / На виїзде	АРС	АСТ	БЛБ	БОЛ	ВБА	ВУЛ	КПР	ЛИВ	МС	МЮ	НОР	НЬЮ	САН	СТО	СУО	ТОТ	УИГ	ФУЛ	ЧЕЛ	ЭВЕ
55	Арсенал				4,00	4,00	1,00	3,00	0,25					3,00	4,00	3,00			0,33		3,00
56	Астон Вилла			4,00		0,33	1,00		0,25		0,33	3,00	1,00			0,25		4,00			
57	Блэкберн Роверс		0,25		0,33	0,33	0,33			0,20					0,25	4,00	0,33			0,33	0,33
58	Болтон Уондерерс	0,25		3,00			1,00			0,33	0,17	0,33	0,25	0,25	6,00					0,20	0,33
59	Вест Бромвич Альбион	0,25	3,00	3,00			4,00		0,25	1,00	0,33				0,33		0,25	0,33	1,00		0,33
60	Вулвергемптон Уондерерс	1,00	1,00	3,00	1,00	0,25		0,25			1,00	0,33	3,00	0,33	1,00	0,25	4,00	4,00	0,20		
61	Квінз Парк Рейнджерс	0,33					4,00			0,33	0,25	0,25	1,00	0,33						3,00	
62	Ліверпуль	4,00	4,00			4,00				0,25	1,00	1,00	4,00	1,00		1,00					
63	Манчестер Сіті		5,00	3,00	1,00		3,00	4,00			5,00	4,00		4,00	5,00		4,00			4,00	4,00
64	Манчестер Юнайтед		3,00		6,00	3,00		4,00	1,00			4,00	0,25	3,00			4,00	6,00		4,00	
65	Норвіч Сіті		0,33		3,00		1,00	4,00	1,00	0,20	0,25		4,00	3,00	1,00	4,00	0,25		1,00		
66	Ньюкасл Юнайтед		1,00		4,00		3,00	1,00	0,25	0,25	4,00	0,25				1,00	1,00	3,00	3,00	0,25	3,00
67	Сандерленд	0,33			4,00		0,33	3,00	1,00		0,33	0,33			5,00			4,00	1,00	0,33	1,00
68	Сток Сіті	0,25		4,00	0,17	3,00	3,00			0,25		1,00		0,20			3,00	1,00	4,00	1,00	
69	Свонсі Сіті	0,33	4,00	0,25			1,00		1,00	0,20		0,25	1,00				1,00	1,00	4,00		
70	Тоттенхем Хотспур				3,00		4,00	4,00			0,25	4,00	1,00		0,33	1,00				1,00	4,00
71	Уїган Атлетик		0,25			3,00	0,25			0,25	0,17		0,33	0,25	1,00	1,00			0,25	1,00	
72	Фулхем	3,00				1,00	0,25					1,00	0,33	1,00	0,25	0,25			4,00	4,00	0,25
73	Челсі				3,00	5,00		5,00	0,33				0,25	4,00	3,00	1,00		1,00	1,00		4,00
74																					

Рисунок 4.3 Приклад частково заповненої таблиці попарних порівнянь

Другим кроком є інтерполяція відсутніх даних. У результаті буде отримано повну таблицю попарних порівнянь.

Дома / На выезде	АРС	АСТ	БЛЕ	БОЛ	ВБА	ВУЛ	КПР	ЛИВ	МС	МЮ	НОР	НЬЮ	САН	СТО	СВО	ТОТ	УИГ	ФУЛ	ЧЕЛ	ЭВЕ	
55																					
56	Арсенал	1,00	2,00	3,00	4,00	4,00	1,00	3,00	0,25	0,33	0,50	1,00	1,00	3,00	4,00	3,00	1,00	3,00	0,33	1,00	3,00
57	Астон Вилла	0,50	1,00	4,00	1,00	0,33	1,00	1,00	0,25	0,25	0,33	3,00	1,00	1,00	1,00	0,25	0,50	4,00	2,00	1,00	1,00
58	Блэкберн Роверс	0,33	0,25	1,00	0,33	0,33	0,33	0,33	0,17	0,20	0,14	0,33	0,33	0,33	0,25	4,00	0,33	1,00	2,00	0,33	0,33
59	Болтон Уондерерс	0,25	1,00	3,00	1,00	1,00	1,00	1,00	0,25	0,33	0,17	0,33	0,25	0,25	6,00	2,00	1,00	2,00	2,00	0,20	0,33
60	Вест Бромвич Альбион	0,25	3,00	3,00	1,00	1,00	4,00	1,00	0,25	1,00	0,33	1,00	1,00	1,00	0,33	1,00	0,25	0,33	1,00	0,50	0,33
61	Вулверзхемптон Уондерерс	1,00	1,00	3,00	1,00	0,25	0,50	0,25	0,25	0,20	1,00	0,33	3,00	0,33	1,00	0,25	4,00	4,00	0,20	1,00	
62	Куінз Парк Рейнджерс	0,33	1,00	3,00	1,00	1,00	4,00	1,00	0,50	0,33	0,25	0,25	1,00	0,33	1,00	2,00	0,50	3,00	3,00	3,00	2,00
63	Ливерпуль	4,00	4,00	6,00	4,00	4,00	4,00	2,00	1,00	0,25	1,00	1,00	4,00	1,00	2,00	1,00	1,00	4,00	3,00	1,00	3,00
64	Манчестер Сити	3,00	4,00	5,00	3,00	1,00	4,00	3,00	4,00	0,50	1,00	5,00	4,00	3,00	4,00	5,00	2,00	4,00	6,00	2,00	4,00
65	Манчестер Юнайтед	2,00	3,00	7,00	6,00	3,00	5,00	4,00	1,00	0,50	4,00	0,25	3,00	4,00	3,00	4,00	6,00	5,00	4,00	4,00	
66	Норвич Сити	1,00	0,33	3,00	3,00	1,00	1,00	4,00	1,00	0,20	0,25	0,25	4,00	3,00	1,00	4,00	0,25	3,00	1,00	1,00	2,00
67	Ньюкасл Юнайтед	1,00	1,00	3,00	4,00	1,00	3,00	1,00	0,25	0,25	4,00	0,25	0,20	2,00	2,00	1,00	1,00	3,00	3,00	0,25	3,00
68	Сандерленд	0,33	1,00	3,00	4,00	1,00	0,33	3,00	1,00	0,33	0,33	0,33	0,50	0,20	5,00	2,00	1,00	4,00	1,00	0,33	1,00
69	Сток Сити	0,25	1,00	4,00	0,17	3,00	3,00	1,00	0,50	0,25	1,00	0,50	0,20	0,17	3,00	3,00	1,00	4,00	1,00	1,00	
70	Сунси Сити	0,33	4,00	0,25	0,50	1,00	1,00	0,50	1,00	0,20	0,33	0,25	1,00	0,50	0,33	0,14	1,00	1,00	4,00	0,33	1,00
71	Тоттенхэм Хотспур	1,00	2,00	3,00	1,00	4,00	4,00	2,00	1,00	0,50	0,25	4,00	1,00	1,00	0,33	1,00	0,13	3,00	5,00	1,00	4,00
72	Уиган Атлетик	0,33	0,25	1,00	0,50	3,00	0,25	0,33	0,25	0,17	0,33	0,33	0,25	1,00	1,00	0,33	0,11	0,25	1,00	1,00	
73	Фулхэм	3,00	0,50	0,50	0,50	1,00	0,25	0,33	0,33	0,17	0,20	1,00	0,33	1,00	0,25	0,25	0,20	4,00	0,11	0,20	0,25
74	Челси	1,00	1,00	3,00	5,00	2,00	5,00	0,33	1,00	0,50	0,25	1,00	4,00	3,00	1,00	3,00	1,00	1,00	5,00	0,11	4,00

Рисунок 4.4 Повністю заповнена таблиця попарних порівнянь

На третьому кроці початкова таблиця заповнюється результатами прогнозу на основі отриманої сили команд.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	Домашня команда	АРС	АСТ	БЛЕ	БОЛ	ВВА	ВУЛ	КПР	ЛИВ	МС	МЮ	НОР	НЬЮ	САН	СТО	СУВ	ТОТ	УИГ	ФУЛ	ЧЕЛ	ЭВЕ
3	Арсенал		+	+	03:00	03:00	01:01	01:00	00:02	-	-	=	=	02:01	03:01	01:00	=	+	01:01	=	01:00
4	Астон Вилла	01:02		03:01	=	01:02	00:00	=	00:02	-	00:01	03:02	01:01	=	=	00:02	-	02:00	+	=	=
5	Блэкберн Роверс	04:03	-		01:02	01:02	01:02	-	-	00:04	-	-	-	-	01:02	04:02	01:02	=	+	00:01	00:01
6	Волтон Уондерерс	-	01:02	+		=	01:01	=		02:03	00:05	01:02	00:02	00:02	05:00	+	=	+	+	01:05	00:02
7	Вест Бромвич Альбион	-	+	+	02:01		02:00	=	00:02	00:00	01:02	=	=	=	00:01	=	01:03	01:02	00:00	-	00:01
8	Вулвергемптон Уондерерс	=	=	+	=	-		00:03	-	-	-	02:02	01:02	02:01	01:02	02:02	00:02	03:01	02:00	01:02	=
9	Жуина Парк Рейнджерс	-	01:01	01:01	00:04	01:01	+		-	02:03	00:02	01:02	00:00	02:03	=	+	-	+	+	01:00	+
10	Ливерпуль	+	+	01:01	03:01	+	02:01	01:00		01:01	01:01	01:01	03:01	01:01	+	00:00	=	+	+	=	+
11	Манчестер Сити	01:00	04:01	+	+	=	03:01	+	03:00		=	05:01	03:01	+	03:00	04:00	+	03:00	+	+	02:00
12	Манчестер Юнайтед	08:02	+	02:03	+	+	04:01	+	=	01:06		02:00	01:01	01:00	+	+	03:00	05:00	+	03:01	+
13	Норвич Сити	01:02	-	03:03	+	00:01	=	02:01	=	-	-		04:02	02:01	01:01	03:01	00:02	+	01:01	=	+
14	Ньюкасл Юнайтед	00:00	=	03:01	+	02:03	+	=	-	03:00	-		+	+	00:00	02:02	01:00	02:01	00:03	02:01	
15	Саусгемптон	-	02:02	02:01	+	02:02	-	+	=	01:00	-	00:01		04:00	+	=	01:02	00:00	01:02	01:01	
16	Сток Сити	-	00:00	03:01	-	+	+	02:03	01:00	-	01:01	=	01:03	-	+	02:01	02:02	02:00	00:00	=	
17	Сундсвалль	-	00:00	-	03:01	03:00	=	01:01	=	-	00:01	-	=	00:00	02:00		01:01	00:00	02:00	-	=
18	Тоттенхем Хотспур	02:01	02:00	+	03:00	01:00	+	03:01	04:00	01:05	-	+	=	01:00	-	=		+	+	01:01	02:00
19	Уингтон Атлетик	00:04	-	03:03	01:03	+	-	02:00	00:00	-	-	01:01	-	01:04	=	=	01:02		00:02	01:01	=
20	Фулхэм	02:01	00:00	01:01	02:00	=	-	06:00	01:00	02:02	00:05	=	-	=	-	-	01:03	+		-	01:03
21	Челси	03:05	01:03	+	+	02:01	03:00	-	01:02	02:01	-	03:01	+	+	=	04:01	=	=	01:01		03:01
22	Эвертон	-	02:02	+	01:02	+	02:01	00:01	00:02	-	00:01	01:01	-	=	00:01	01:00	-	03:01	+	-	
23																					

Рисунок 4.5 Результат прогнозу переможців матчів

Результатом проведення експерименту є графік залежності точності прогнозування від кількості відсутніх даних.

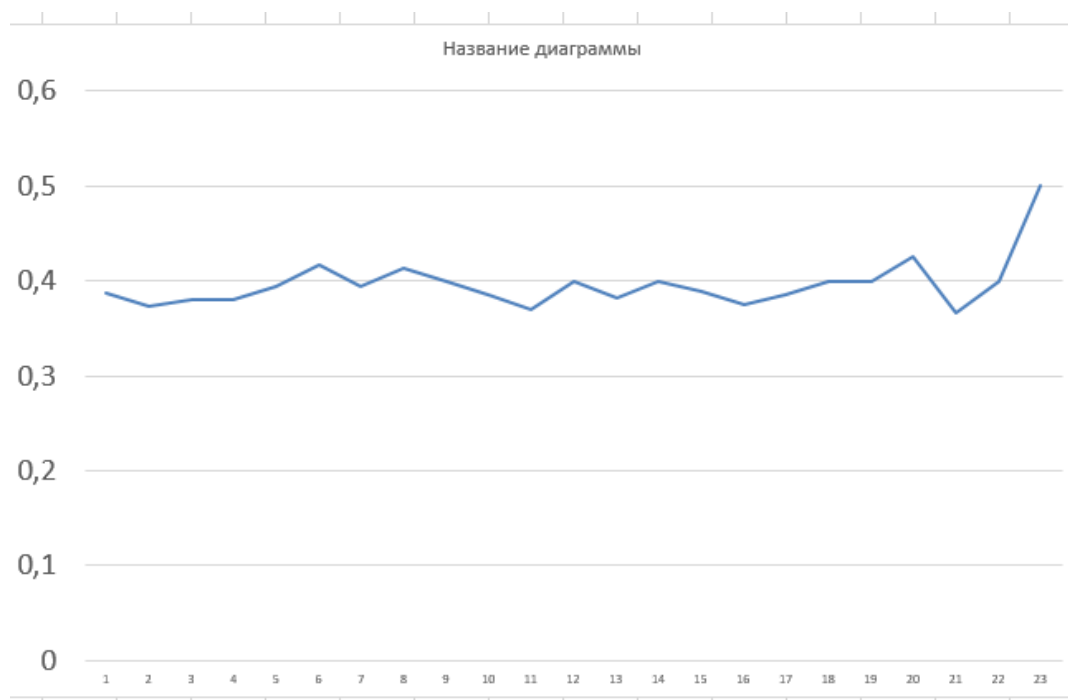


Рисунок 4.6 Графік залежності точності прогнозування від кількості відсутніх альтернатив

Висновки до четвертого розділу.

В результаті проведених досліджень отримано дані, які показують що в цілому застосування методу аналізу ієрархій до прогнозування результатів спортивних змагань має деякий сенс. Середня точність прогнозування складає близько 40%, в той час як при прогнозуванні випадковим методом, за допомогою так званої монетки, точність складає 33%. Також цікавим спостереженням є те, що точність прогнозування загалом не залежить від кількості вже зіграних матчів, а от на останньому етапі впала до 33%. Таким чином, метод аналізу ієрархій продемонстрував деяку ефективність у прогнозуванні результатів командних змагань. Отже подальші дослідження щодо вдосконалення роботи алгоритму або його комбінації з іншими методами прогнозування є доцільними.

ВИСНОВКИ

В роботі було досліджено ефективність застосування методу аналізу ієрархій для прогнозування результатів спортивних змагань. В результаті досліджень було виявлено, що використання МАІ має деякий рівень ефективності при вирішенні даного питання. В той же час можна зробити припущення, що самого по собі МАІ не достатньо для того, щоб робити надійні прогнози. Таким чином постає питання, чи можна використовувати МАІ у комбінації з іншими методами, і якщо так, то якими саме.

БІБЛІОГРАФІЧНИЙ СПИСОК

1. Thomas L. Saaty The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation.
2. Viktor Shynkarenko, Tetiana Vasetska , and Iryna Vyshnyakova Analytic Hierarchy Process Sustainability at the Significant Number of Alternatives Ranking.
3. Analytic hierarchy process [ел.ресурс]
https://en.wikipedia.org/wiki/Analytic_hierarchy_process.
4. Ossama Salem, Baris Salman, Sudipta Ghorai Accelerating construction of roadway alternative techniques and procurement methods.
5. John R. Grandzol Improving the Faculty Selection Process in Higher Education: A Case for the Analytic Hierarchy Process

ДОДАТКИ

ДОДАТОК А
Технічне завдання

ЗАТВЕРДЖУЮ
Проректор
Українського державного
університету науки і технології
Анатолій РАДКЕВИЧ

Прогнозування результатів командних змагань на основі конструктивного
підходу та МАІ

Технічне завдання
ЛИСТ ЗАТВЕРДЖЕННЯ
1116130.01361-01-ЛЗ

Завідувач кафедри КІТ
_____Вадим ГОРЯЧКІН
Керівник розробки
_____Віктор Шинкаренко_____
Виконавець
_____Сергій Кумпан
Нормоконтролер
_____Світлана ВОЛКОВА

ЗАТВЕРДЖЕНО
1116130.01361-01

Прогнозування результатів командних змагань на основі конструктивного
підходу та МАІ

Технічне завдання

Листів 18

2024

1. ВВЕДЕННЯ

Завданням для даної дипломної роботи було обрано розробку програми що за допомогою методу аналізу ієрархій прогнозує результат чемпіонатів з футболу.

Аналіз проблеми прийняття рішень в МАІ починається з побудови ієрархічної структури, яка включає ціль, критерії, альтернативи і інші фактори, що впливають на вибір. Кожний елемент ієрархії може являти собою різні аспекти вирішуваної задачі, причому до уваги можуть бути прийняті як матеріальні, так і нематеріальні фактори, вимірювані кількісні і якісні характеристики, об'єктивні дані і суб'єктивні експертні оцінки.

Наступним етапом аналізу є визначення пріоритетів, які мають відносну важливість або перевагу елементів побудованої ієрархічної структури, за допомогою парних порівнянь. Безрозмірні пріоритети дозволяють обґрунтуванню порівнювати різнорідні фактори, що є особливістю МАІ. На заключному етапі аналізу виконується синтез(лінійне згортання) пріоритетів на ієрархії, в результаті якої обраховуються пріоритети альтернативних рішень відносно головної цілі. Кращою вважається альтернатива з максимальним значенням пріоритету.

В даній роботі пропонується формалізація процесу прийняття рішень на основі класичного методу аналізу ієрархій засобами конструктивно-продукційних структур. Використання конструктивно-продукційних структур для формалізації даної моделі може бути перспективним підходом для подальшої автоматизації процесу прогнозування результатів змагань і переходу від окремого випадку на зразок футбольних змагань до загального алгоритму.

Особливість конструктивно-продукційних структур складається в формуванні множини конструкцій за допомогою операцій зв'язування, підстановки і допоміжних операцій, які задаються правилами аксіоматики. Конструкції формуються в результаті виконання операцій виводу.

Визначення 1. Узагальненою конструктивно-продукційною структурою

(ОКПС) назвемо:

$$C = \langle M, \Sigma, \Lambda \rangle, (1)$$

Де M – носій структури, Σ – сигнатура, яка складається з множин можливих імен багатомісних операцій і відношень: зв'язування, підстановки, виводу і допоміжних операцій; Λ - конструктивна аксіоматика, яка включає множину визначень, аксіом, правил, властивостей, інструкцій і іншого. Множини M та Σ визначаються аксіоматикою.

2. ПРИЗНАЧЕННЯ ТА СФЕРА ЗАСТОСУВАННЯ

Під час виконання роботи буде створено програму, яка буде прогнозувати результати чемпіонатів з футболу за допомогою МАІ. Також процес прогнозування буде формалізовано за допомогою конструкційно-продукційних структур.

3. ВИМОГИ ДО ПРОГРАМИ

4.1. Вимоги до функціональних характеристик

Вимоги до функціональних характеристик наступні:

- ввід даних;
- автоматичне підрахування результатів;
- вивід результатів;
- данні вводяться з клавіатури та миші.

Вхідні данні:

Інформація про матчі, яка включає: номер тура, команду хазяїв, команду гостей, результат хазяїв, результат гостей.

Вихідні данні:

Вихідні данні виводяться на екран у імені команди.

4.2 Вимоги до надійності

Вимоги до надійності наступні:

- наявність архівної копії тексту програми на зовнішньому носії;
- повинно бути безперебійне постачання електроенергії;
- своєчасне виконання процедур резервного копіювання даних.

4.3. Умови експлуатації

Для забезпечення надійного функціонування програмного продукту користувачу необхідно дотримуватися таких умов:

- температура у приміщенні +21-+25 С°, відносна вологість повітря 40-60%;
- стан технічних засобів має задовольняти відповідним нормам і вимогам.

Працювати з програмою може людина яка володіє навичками роботи с персональним комп'ютером та ознайомена з керівництвом користувача.

4.4. Вимоги до складу і параметрів технічних засобів

Вимоги до складу і параметрів технічних засобів наступні:

- присутність монітора комп'ютера - пристрої виведення інформації про роботу, проведену програмним забезпеченням;

- присутність комп'ютерної миші і клавіатури - пристроїв введення інформації під час роботи з програмою;
- для успішного скачування програми потрібно не менше 120 МБ вільного дискового простору;
- процесор: core2duo;
- оперативна пам'ять: 1024 МВ ОЗУ;
- відеокарта: AMD Radeon HD7470;
- звукова карта: Manli C-Media;

4.5. Вимоги до інформаційної і програмної сумісності

Вимоги до інформаційної і програмної сумісності наступні:

- операційна система windows XP чи вище;
- середовище програмування Unity 3.5.

4.6. Вимоги до маркування та упаковки

Пакування виробу(носіїв і програмної документації) повинно передбачати захист від механічних, кліматичних електромагнітних впливів. На упаковці має бути зазначено назва виробу, розробник, контактні данні, рік розробки.

Рис. 1. Маркування програмного продукту

4.7. Вимоги до транспортування і зберігання

Транспортування може виконуватися будь-яким способом що виключає механічний та електромагнітний вплив на носії інформації. Зберігання на жорсткому диску.

Місце зберігання повинне бути сухим з відсутністю доступу прямих сонячних струменів та відсутністю пилу.

Строк зберігання обумовлений збереженням інформації на носії. Рекомендується проводити профілактичні роботи перевірки якості носіїв кожні 6 місяців.

4. ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

До складу програмної документації мають входити:

- Текст програми;

Вся програмна документація повинна задовольняти вимогам державного стандарту до оформлення програмних документів (ГОСТ 19.101-77).

5. ТЕХНІКО-ЕКОНОМІЧНІ ПОКАЗНИКИ

При виконанні роботи було розроблено програмне рішення для дослідження точності прогнозування результатів командних змагань за допомогою МАІ.

Метою даного розділу є розробка техніко-економічного обґрунтування для розроблення нового програмного забезпечення заради оцінки запланованих витрат, а також дати висновок щодо доцільності його розробки та розгортання.

Спершу треба оцінити обсяг витрачених людино-годин спираючись на обсяг програмного коду. Для визначення запланованих трудовитрат використовуються кількісна та якісна оцінка.

Constructive Cost Model (COCOMO) [1] орієнтується на вимірюваний розмір проекту S в рядках коду LOC (KLOC), а обсяг трудових ресурсів у людино-місяцях, розрахунок наведено у формулі 5.1.

$$E = a \cdot S^b \cdot EAF \quad (5.1)$$

де E – необхідні людські ресурси на розробку проекту (в людино-місяцях);

S^b – обсяг коду (в KLOC);

EAF – фактор уточнення витрат (effort adjustment factor).

Для простих систем $a = 2,4$; $b = 1,05$.

Загальна довжина коду програмного забезпечення становить 446 рядків. Трудовитрати у людино-місяцях розраховано у формулі 5.2.

$$E = 2,4 * 446^{1,05} * 1 = 1\ 156,4 \quad (5.2)$$

Далі наведені розрахунки для обчислення вартості розробки програмного комплексу. За передбачувані напрямки витрат прийняті:

- грошовий оклад та премії працівників;
- відрахування на соціальні потреби;

- накладні витрати;
- витрати на персональний комп'ютер та ліцензійні програмні засоби, що застосовуються при розробці.

Оцінка основної заробітної плати (ОЗП) інженера-програміста, що займається розробкою програмного продукту, базується на кількості розробників, заробітної плати в перерахунок на години та тривалості розробки (у годинах). Розрахунок заробітної плати відбувається за даними середньої заробітної плати програміста в м. Дніпро [2] і представлений у табл. 5.1.

Таблиця 5.1 – Фонд місячної заробітної плати

№п /п	Посада виконавця	Оклад, грн/міс	Кількість		Сума зарплати грн.
			чол.	місяців	
1	інженер-програміст	20000	1	4,85	97000

Розробка представленого у цій роботі програмного продукту буде проводитися одним програмістом в період з 01.02.2023 до 25.06.2023, що становить 21 робочий тиждень або 105 днів. Робочий час на тиждень приймається за 40 годин. Погодинна оплата кваліфікованого інженера-програміста становить 115.47 грн/год. Таким чином, витрачений робочий час за форм. 5.3 розраховано у форм. 5.4:

$$t_{\text{розробки}} = N_{\text{чол}} \cdot N_{\text{тиж}} \cdot N_{\text{год}}, \quad (5.3)$$

де $N_{\text{чол}}$ – кількість виконавців, чол;

$N_{\text{тиж}}$ – тривалість розробки;

$N_{\text{год}}$ – витрати робочого часу, год;

$$t_{\text{розробки}} = 1 \cdot 21 \cdot 40 = 840 \text{ чол/год.} \quad (5.4)$$

ОЗП визначається за формулою 5.5:

$$\text{ОЗП} = t_{\text{розробки}} \cdot N \cdot K_{\text{КВ}}, \quad (5.5)$$

де $t_{\text{розробки}}$ – витрати праці у чол/год;

N – зарплата за годину;

K_{KB} – коефіцієнт кваліфікації програміста, прийнято 0,75.

ОЗП становитиме за формулою 5.6:

$$\text{ОЗП} = 840 \cdot 115,47 \cdot 0,75 = 72746 \text{ грн.} \quad (5.6)$$

Відрахування на оплату соціальних внесків вираховуються у відсотковому відношенні від суми заробітної плати [3]:

$$C_{\text{соц}} = \frac{\text{ОЗП} \cdot 22\%}{100\%}$$

$$C_{\text{соц}} = \frac{72746 \cdot 22\%}{100\%} = 16004 \text{ грн.} \quad (5.7)$$

Отримані результати за пунктами (5.6) та (5.7) треба скласти. В результаті вони утворюють суму 88750 грн., яку визначено основними прямими витратами.

Накладні витрати [4] включають загальногосподарські витрати на організацію робочого процесу: витрати на комунальні послуги в офісі, амортизація будівель, зарплату штатного обслуговуючого персоналу та інше. Вони визначаються в процентах (30 – 40%) від суми прямих витрат і розраховані за форм. 5.8 у 5.9:

$$C_{\text{накл}} = \frac{(\text{ОЗП} + C_{\text{соц}}) \cdot 35\%}{100\%}; \quad (5.8)$$

$$C_{\text{накл}} = \frac{(72746 + 16004) \cdot 35\%}{100\%} = 31062,75 \text{ грн.} \quad (5.9)$$

Впродовж усього терміну використання технічного устаткування підприємство щорічно витрачає певні кошти, пов'язані з її експлуатацією.

Витрати на експлуатацію [4] персонального комп'ютер визначаються на етапі планування розробки програмного засобу в залежності від вартості конкретної робочої станції. До експлуатаційних витрат відносяться:

- витрати на ремонт;
- оренда приміщення;
- вартість витратних матеріалів;
- заробітна плата ремонтника;
- додаткові витрати – обслуговування приміщення (прибирання, охорона, оренда, комунальні послуги);
- амортизаційні витрати на персональний комп'ютер та базовий пакет програмного забезпечення;
- витрати на електроенергію ($C_{ел}$).

Визначаються ці витрати за формулою 5.10:

$$C_{ел} = P \cdot B \cdot T_{розр}, \quad (5.10)$$

де P – потужність комп'ютера та допоміжних споживачів електричної енергії, розрахована потужність для комп'ютера та монітору прийнята 0,5 кВт/год;

B – вартість 1 кВт/година, в Україні за поточним тарифом складає 1,68 грн;

$T_{розр}$ – час роботи з ЕВМ, прирівнюється робочому часу.

Витрати на електроенергію розраховані у форм. 5.11:

$$C_{ел} = 0,5 \cdot 1,68 \cdot 840 = 705,6 \text{ грн.} \quad (5.11)$$

Для визначення заробітної плати ремонтника ($C_{рем}$) необхідно перерахувати його середньомісячну заробітну плату в години, що витрачаються за нормою на один персональний комп'ютер. Кількість комп'ютерів, що обслуговуються одним ремонтником приймається за 50, а

його зарплата за місяць – 10000 грн. [7]. Тоді на один комп'ютер буде витрачено коштів за форм. 5.12:

$$C_{\text{рем}} = \frac{C'_{\text{рем}}}{N_{\text{КОМ}}} \cdot T_{\text{міс}}, \quad (5.12)$$

де $C'_{\text{рем}}$ – середньомісячна заробітна плата;

$N_{\text{КОМ}}$ – кількість комп'ютерів на одного ремонтника.

$T_{\text{міс}}$ – час розробки програмного продукту, міс.

Заробітна плата ремонтника ($C_{\text{рем}}$) розрахована у форм. 5.13:

$$C_{\text{рем}} = \frac{10000}{50} \cdot 4,85 = 970 \text{ грн.} \quad (5.13)$$

Крім цього необхідно врахувати витрати на комплектуючі вироби ($C_{\text{КОМ}}$) [5], на які будуть замінені несправні при ремонті. Вони приймаються за 10% від вартості комп'ютера впродовж терміну його експлуатації. Тепер перерахуємо ці витрати з урахуванням часу створення програмного засобу у форм. 5.14:

$$C_{\text{КОМ}} = B_{\text{КОМ}} \cdot \frac{N_{\text{Д}}}{N_{\text{експ}} \cdot 365} \cdot \frac{10\%}{100\%}, \quad (5.14)$$

де $B_{\text{КОМ}}$ – вартість персонального комп'ютеру;

$N_{\text{Д}}$ – тривалість розробки програмного продукту у днях;

$N_{\text{експ}}$ – термін експлуатації персонального комп'ютеру.

Прийнята вартість ПК з клавіатурою, мишею і монітором 20 000 грн [11]. Сума витрат на комплектуючі частини протягом періоду розробки програми визначається у форм. 5.15:

$$C_{\text{КОМ}} = 20000 \cdot \frac{105}{5 \cdot 365} \cdot \frac{10}{100} = 115,07 \text{ грн.} \quad (5.15)$$

Сума на витратні матеріали ($C_{\text{ВМ}}$) [5] протягом всього терміну експлуатації також становить приблизно 10% від вартості комп'ютеру. Робоча станція оцінюється у 20 000 грн., термін її експлуатації – 5 років. Витрати такі ж, що й на комплектуючі вироби:

$$C_{\text{ВМ}} = C_{\text{КОМ}} = 115,07 \text{ грн.} \quad (5.16)$$

Амортизаційні відрахування на персональний комп'ютер (АПК) [6] визначені з положення, що амортизаційний період в даний час дорівнює терміну зміни поколінь обчислювальних машин і приймається за 3 роки. Отже, кожні 3 роки амортизаційні витрати дорівнюють вартості такого комп'ютера. Вирахуємо частку витрат на амортизаційні відрахування в період розробки програми:

$$\text{АПК} = V_{\text{КОМ}} \cdot \frac{N_{\text{Д}}}{N_{\text{експ}} \cdot 365} = 20000 \cdot \frac{105}{3 \cdot 365} = 1917,8 \quad (5.17)$$

Для визначення суми амортизаційних відрахувань на програмне забезпечення (АПЗ) [6] необхідно враховувати час розробки проекту в межах часу дії пакету прикладних програм та операційної системи, що були застосовані. Розробка проводилася в операційній системі Windows, оновлення версій якої прийнято здійснювати кожні 5 років, а за середовище розробки для поставлених задач обрано Visual Studio 2019, що поширюється за моделлю щомісячної підписки, тому амортизаційні відрахування на програмне забезпечення дорівнюють його вартості в перерахунок на час розробки.

Операційна система для розробки взята Windows 10, середовище розробки програм – Microsoft Visual Studio 2019 Professional, що оплачується щомісячно за підпискою у 1200 грн/міс.

(5.18)

$$АПЗ_W = 6204 \cdot \frac{105}{5 \cdot 365} = 357 \text{ грн.}$$

$$АПЗ_V = 1200 \cdot 5 = 6000 \text{ грн.}$$

(5.19)

Розрахунок амортизаційних відрахувань на програмне забезпечення зведений у табл. 5.2.

Таблиця 5.2 – Застосоване програмне забезпечення

Найменування програмного забезпечення	Вартість програмного забезпечення, грн	Джерело придбання	Амортизаційні відрахування, грн
Windows 10 Professional	6204	https://soft.rozetka.com.ua/microsoft_fqc_09131/p3936301/	460
Microsoft Office Excel	6000	https://www.microsoft.com/uk-UA/microsoft-365/buy/compare-all-microsoft-365-products	6000
Всього:	12204	-	6460

Додаткові витрати ($C_{\text{дод}}$) [6]: охорона, комунальні послуги, прибирання приміщень, тощо, досить складно оцінити, тому приймаються рівними 20% від заробітної плати інженера-програміста, а саме 3500 гривень на місяць.

Оренду офісу ($C_{\text{ор}}$) з площею 13 м² візьмемо 3000 гривень на місяць [8].

Загалом експлуатаційні витрати на один персональний комп'ютер будуть складати за форм. 5.20:

$$C_{\text{експ}} = C_{\text{ел}} + C_{\text{рем}} + C_{\text{КОМ}} + C_{\text{ВМ}} + АПК + АПЗ + C_{\text{дод}} + C_{\text{ор}} =$$

$$= 705,6 + 970 + 2 * 115,07 + 1917,8 + 6460 + 17500 + 15000 = 42783,54 \text{ грн} \quad (5.20)$$

Результати розрахунків приведено у табл. 5.3.

Таблиця 5.3 – Експлуатаційні витрати на ПК і ПЗ.

Найменування витрат	Витрати, грн
Витрати на електроенергію	705,6
Вартість витратних матеріалів	115,07
Витрати на ремонт	800
Амортизація персонального комп'ютера	1917,8
Амортизація програмного забезпечення	6460
Оренда приміщення	6000
Додаткові витрати	8750
Всього	42613,54

Підсумкові витрати на створення програмного продукту [6] вираховуються за форм. 5.21 у 5.22:

$$C_{\text{розробки}} = \text{ОЗП} + C_{\text{соц}} + C_{\text{накл}} + C_{\text{експ}}; \quad (5.21)$$

$$C_{\text{розробки}} = 65627 + 14438 + 28022,75 + 42783,54 = 150871,29 \quad (5.22)$$

Розрахунок витрат зведено у табл. 5.4.

Таблиця 5.4 – Кошторис витрат на розробку програмного засобу

Найменування витрат	Витрати, грн
Основна заробітна плата	65 627
Відрахування на соціальні потреби	14 438
Накладні витрати	28 022,75
Експлуатаційні витрати	42 783,54
Всього	150871,29

За розрахованими значеннями техніко-економічних показників проекту складено кошторис витрат на розробку актуального програмного

забезпечення. За результатами розрахунків встановлена вартість розробки програмного засобу складає 150 871,29 грн.

6. СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

Таблиця 6.1 – Стадії та етапи розробки

Стадії розробки	Етапи розробки	Терміни виконання
1. Технічне завдання (ТЗ)	Функціональне та експлуатаційне призначення	23.09.23 – 24.09.23
	Вхідні та вихідні дані	24.09.23 – 27.09.23
	Вимоги до програмної документації	27.09.23 – 29.09.23
	Техніко-економічні показники	29.09.23 – 01.10.23
	Узгодження та затвердження ТЗ	01.10.23 – 02.10.23
2. Робочий проект	Розробка та програмування функціоналу програми	03.10.23 – 16.11.23
	Розробка і реалізація інтерфейсу користувача	17.11.23 – 22.11.23
	Налагодження програми	23.11.23 – 02.12.23
	Розробка, узгодження та затвердження програмної документації відповідно до вимог ГОСТ 19.101-77	03.12.23 – 10.12.23
	Проведення випробувань і корегування програми та документації за результатами випробувань	11.12.23 – 31.12.23
3. Впровадження	Підготовка і передача програми та програмної документації замовнику	01.01.24 – 15.01.24

7. ПОРЯДОК КОНТРОЛЮ І ПРИЙМАННЯ

Контроль за виконанням роботи здійснює керівник розробки – (той хто на листу затвердження).

Приєм програмного продукту здійснює комісія у складі:

- Горячкін В.М.;
- Керівник розробки Шинкаренко В.І.

8. БІБЛІОГРАФІЧНИЙ СПИСОК

1. Модель СОСОМО. [Електронний ресурс]. Режим доступу: <https://ru.wikipedia.org/wiki/COCOMO>. Дата звернення: 25.11.21.
2. Дані щодо середньої зарплати програміста в Україні. [Електронний ресурс]. Режим доступу: <https://www.work.ua/ru/salary-dnipro-web-программист/> Дата звернення: 25.11.21.
3. Закон України «Про збір та облік єдиного внеску на загальнообов'язкове державне соціальне страхування» із змінами [від 1 липня 2021 року N 1617-IX](#)
4. Липаев В.В. Экономика производства сложных программных продуктов. – М Синтег, 2008. 432 с
5. Благодатских В.А. и др. Экономика, разработка и использование программного обеспечения ЭВМ. – М.: Финансы и статистика, 1995. - 286 с.
6. Фатрелл Р.Т., Шафер Д.Ф., Шафер Л.И.. Управление программными проектами. Достижение оптимального качества при минимуме затрат. М.: Издательский дом “Вильямс”, 2004. – 1125 с.
7. Дані щодо середньої зарплати програміста в Україні. [Електронний ресурс]. Режим доступу: <https://www.work.ua/ru/salary-компьютерный+мастер/> Дата звернення: 25.11.21.
8. Дані щодо оренди офісу в м. Дніпро. [Електронний ресурс]. Режим доступу: <https://www.olx.ua/d/obyavlenie/sdam-ofis-13i20m-mozhno-obedinit-arenda-kabineta-pomeschenie-salon-tsent-IDNpGGQ.html#5d8a44c1ac;promoted> Дата звернення: 25.11.21.
9. Ціна операційної системи Windows 10. [Електронний ресурс]. Режим доступу: https://soft.rozetka.com.ua/microsoft_fqc_09131/p3936301/ Дата звернення: 25.11.21.

ЗАТВЕРДЖЕНО
1116130.01361-01 12 01-ЛЗ

Програмний додаток для прогнозування результатів командних змагань

Текст програми

1116130.01361-01 12 01

Аркушів 28

2024

ТЕКСТ ПРОГРАМИ

```

'Номер строки - начало турнирной таблицы
Const rtTurnir As Integer = 30

'Номер строки - начало турнирной таблицы по МАИ
Const rtMAI As Integer = 55
'Номер строки - начало копии турнирной таблицы по МАИ
Const rtMAICopy As Integer = 100

'Смещение для строки - сумма
Const rSumMAI As Integer = 59
'Смещение для строки с S=
Const rResS As Integer = 60
'Смещение строки таблицы на листе "Результаты"
Const rRes As Integer = 27

'Нулевая оценка в таблице МАИ
Const FloatZero As Double = 0.0001

'Проценты для интерполяции
Const rProcent As Integer = 8
Const cProcent As Integer = 4

'Номера начальных столбцов для сохранения результатов
Const cResPlaceMAIMAI As Integer = 7      'по расхождению мест между МАИ до и после
интерполяции
Const cResEqPlaceMAIMAI As Integer = 17   'по доле совпадения мест между МАИ до и
после интерполяции
Const cResPlaceTurnirMAI As Integer = 27  'по расхождению мест между trad.таблицей
и МАИ после интерполяции
Const cResEqPlaceTurnirMAI As Integer = 37 'по доле совпадения мест между
trad.таблицей и МАИ после интерполяции
Const cResOSMAI As Integer = 47          'отношение согласованности МАИ после интерполяции
Const cResWeight As Integer = 57        'степень расхождения весов

'Начало таблицы для рез-в анализа по мат.ожиданию
Const cResAvg As Integer = 2
Const rResAvg As Integer = 10

'Начало таблицы для рез-в анализа по ср.квадр.отклонению
Const cResSq As Integer = 10

```

```
Const rResSq As Integer = 10
```

```
'Начало таблицы для рез-в анализа по дисперсии
```

```
Const cResD As Integer = 18
```

```
Const rResD As Integer = 10
```

```
Const rt90 As Integer = 15
```

```
Const ct90 As Integer = 4
```

```
Const rt95 As Integer = 16
```

```
Const ct95 As Integer = 4
```

```
Const rInt90 As Integer = 23
```

```
Const cInt90 As Integer = 4
```

```
Const rInt95 As Integer = 36
```

```
Const cInt95 As Integer = 4
```

```
'=====Доверительный интервал=====
```

```
'Начало таблицы с выборками
```

```
Const rAvgChoice As Integer = 5
```

```
Const cAvgChoicePlaceMAIMAI As Integer = 4      'по расхождению мест между МАИ до и  
после интерполяции
```

```
Const cAvgChoiceEqPlaceMAIMAI As Integer = 7    'по доле совпадения мест между МАИ  
до и после интерполяции
```

```
Const cAvgChoicePlaceTurnirMAI As Integer = 10  'по расхождению мест между  
трад.таблицей и МАИ после интерполяции
```

```
Const cAvgChoiceEqPlaceTurnirMAI As Integer = 13 'по доле совпадения мест между  
трад.таблицей и МАИ после интерполяции
```

```
Const cAvgChoiceOSMAI As Integer = 16          'отношение согласованности МАИ после  
интерполяции
```

```
Const cAvgChoiceWeight As Integer = 19        'степень расхождения весов
```

```
'Заполнение традиционной турнирной таблицы на активном листе
```

```
'n - количество команд
```

```
Sub ZapTradFunc(n As Integer)
```

```
Dim i, j, k, k1, k2, k3, k4, k5 As Integer
```

```
n = Cells(1, 1).Value
```

```

For i = 1 To n
    Cells(i + rtTurnir, 1).Value = Cells(i + 3, 1).Value 'название команды (по
вертикали)
    Cells(i + rtTurnir, 2 + n).Value = 0 'забитые
    Cells(i + rtTurnir, 3 + n).Value = 0 'пропущенные
    Cells(i + rtTurnir, 4 + n).Value = 0 'очки
Next i

For i = 1 To n
    Cells(rtTurnir, i + 1).Value = Cells(3, i + 1).Value 'название команды (по
горизонтали)
Next i

'Заполнение таблицы результатами игр команд
For i = 1 To n
    For j = 1 To n
        If i <> j Then

            k = Hour(Cells(i + 3, j + 1).Value) 'мячи первой команды
            k1 = Minute(Cells(i + 3, j + 1).Value) 'мячи второй команды

            'счет игры команд
            Cells(i + rtTurnir, j + 1).Value = Str(k) + ":" + Str(k1)

            'сумма забитых и пропущенных мячей
            Cells(i + rtTurnir, 2 + n).Value = Cells(i + rtTurnir, 2 + n).Value + k
            Cells(i + rtTurnir, 3 + n).Value = Cells(i + rtTurnir, 3 + n).Value + k1

            'За победу команда получает 3 очка, за ничью - 1 очко
            If k > k1 Then Cells(i + rtTurnir, 4 + n).Value = Cells(i + rtTurnir, 4 +
n).Value + 3
            If k = k1 Then Cells(i + rtTurnir, 4 + n).Value = Cells(i + rtTurnir, 4 +
n).Value + 1

            Cells(j + rtTurnir, 2 + n).Value = Cells(j + rtTurnir, 2 + n).Value + k1
            Cells(j + rtTurnir, 3 + n).Value = Cells(j + rtTurnir, 3 + n).Value + k

            If k < k1 Then Cells(j + rtTurnir, 4 + n).Value = Cells(j + rtTurnir, 4 +
n).Value + 3
            If k = k1 Then Cells(j + rtTurnir, 4 + n).Value = Cells(j + rtTurnir, 4 +

```

```
n).Value + 1
```

```
End If
```

```
Next j
```

```
Next i
```

```
'определение мест
```

```
For i = 1 To n
```

```
Cells(i + rtTurnir, 6 + n).Value = Cells(i + rtTurnir, 4 + n).Value
```

```
Next i
```

```
For j = 1 To n
```

```
k = 0
```

```
k1 = 1
```

```
k2 = 0
```

```
For i = 1 To n
```

```
If Cells(i + rtTurnir, 6 + n).Value > k Then
```

```
k = Cells(i + rtTurnir, 6 + n).Value
```

```
k1 = i + rtTurnir
```

```
k2 = k2 + 1
```

```
End If
```

```
Next i
```

```
While k2 > 0
```

```
For i = 1 To n
```

```
k4 = Cells(i + rtTurnir, 2 + n).Value - Cells(i + rtTurnir, 3 + n).Value
```

```
k5 = Cells(k1, 2 + n).Value - Cells(k1, 3 + n).Value
```

```
If (Cells(i + rtTurnir, 6 + n).Value = k) And (k4 > k5) Then
```

```
k1 = i + rtTurnir
```

```
End If
```

```
Next i
```

```
k2 = k2 - 1
```

```
Wend
```

```
Cells(k1, 6 + n).Value = 0
```

```
Cells(k1, 5 + n).Value = j
```

```
Next j
```

```
End Sub
```

```
Sub CompleteTheLeagueTable()
```

```
'Cells(5, 5).Value = Cells(27, 35).Value 'название команды (по вертикали)
```

```

'k = Hour(Cells(i + 3, j + 1).Value)      'мячи первой команды
'
'      k1 = Minute(Cells(i + 3, j + 1).Value)  'мячи второй команды
'      k = Hour(Cells(i + 3, j + 1).Value) - Minute(Cells(i + 3, j + 1).Value)
'      'разница голов K2 и K1
'      k1 = Hour(Cells(j + 3, i + 1).Value) - Minute(Cells(j + 3, i +
1).Value)
'Cells(rtTurnir, i + 1).Value = Cells(3, i + 1).Value  'название команды (по
горизонтали)
'Cells(i + rtTurnir, j + 1).Value = Str(k) + ":" + Str(k1)

```

```

Dim n, i, j, k1, k2, f, l, x, t1, t2 As Integer
n = Cells(3, 26).Value * 10
Cells(1, 2).Value = n
x = 20
i = 27
j = 35
For f = 1 To x
    For l = 1 To x
        Cells(f + 3, l + 1).ClearContents
        Cells(f + 3, l + 1).Interior.ColorIndex = 2
    Next l
Next f
For f = 1 To n
    Dim valueOne As String
    valueOne = Cells(i, j).Value
    k1 = Cells(i, j + 2).Value
    k2 = Cells(i, j + 3).Value
    If (IsEmpty(Cells(3 + k1, 1 + k2))) Then
        Cells(3 + k1, 1 + k2).Value = Left(valueOne, 3)
    Else
        Cells(4, 26).Value = Cells(3 + k1, 1 + k2).Value
        Cells(5, 26).Value = Left(valueOne, 3)
        't1 = Hour(Cells(4, 26).Value) + Hour(Cells(5, 26).Value)
        't2 = Minute(Cells(4, 26).Value) + Minute(Cells(5, 26).Value)
        Cells(6, 26).Value = Cells(4, 26).Value + Cells(5, 26)
        Cells(3 + k1, 1 + k2).Value = Cells(6, 26).Value
        'Cells(3 + k1, 1 + k2).Value.Hour = t1
        'Cells(3 + k1, 1 + k2).Value.Minute = t2
    End If
    i = i + 2

```

```

Next f
For f = 1 To x
    For l = 1 To x
        If f <> l Then
            If IsEmpty(Cells(f + 3, l + 1).Value) = True Then
                Cells(f + 3, l + 1).Interior.ColorIndex = Cells(f + 30, l +
1).Interior.ColorIndex
            End If
        End If
    Next l
Next f
End Sub

```

```

Sub LoseWinColors()
Dim n, teamOneGoals, teamTwoGoals As Integer
n = Cells(3, 30).Value
For i = 1 To n
    For j = 1 To n
        Cells(i + 3, j + 1).Interior.ColorIndex = 2
        Cells(i + 30, j + 1).Interior.ColorIndex = 2
    Next j
Next i
For i = 1 To n
    For j = 1 To n
        If i <> j Then
            teamOneGoals = Hour(Cells(i + 3, j + 1).Value)
            teamTwoGoals = Minute(Cells(i + 3, j + 1).Value)
            If (teamOneGoals > teamTwoGoals) Then
                'Cells(i + 3, j + 1).Interior.ColorIndex = 43
                Cells(i + 30, j + 1).Interior.ColorIndex = 43
            ElseIf (teamOneGoals = teamTwoGoals) Then
                'Cells(i + 3, j + 1).Interior.ColorIndex = 36
                Cells(i + 30, j + 1).Interior.ColorIndex = 36
            Else
                'Cells(i + 3, j + 1).Interior.ColorIndex = 46
                Cells(i + 30, j + 1).Interior.ColorIndex = 46
            End If
        End If
    Next j
Next i
End Sub

```

```

        Next j
    Next i
End Sub

Sub SecondTable()
Dim n, teamOneGoals, teamTwoGoals As Integer
n = Cells(3, 30).Value
For i = 1 To n
    For j = 1 To n
        Cells(i + 30, j + 1).ClearContents
    Next j
Next i
For i = 1 To n
    For j = 1 To n
        If i <> j Then
            If IsEmpty(Cells(i + 3, j + 1).Value) = False Then
                teamOneGoals = Hour(Cells(i + 3, j + 1).Value)
                teamTwoGoals = Minute(Cells(i + 3, j + 1).Value)
                Cells(i + rtTurnir, j + 1).Value = Str(teamOneGoals) + ":" +
Str(teamTwoGoals)
            End If
        End If
    Next j
Next i

End Sub

'Заполнение матрицы МАИ на активном листе
'n - размер таблицы (количество команд чемпионата)
'ах, bx - подобранные коэффициенты для перехода от обычной таблицы чемпионата к
матрице МАИ.
Sub ZapMAI()
Dim n, i, j, k, k1 As Integer
Dim a, b, c, ax, bx As Double
n = Cells(3, 30).Value
ax = Cells(7, 26).Value
bx = Cells(8, 26).Value
    'Подготовка ячеек
    'Call PrepareMAI(n)
For i = 1 To n
    For j = 1 To n

```

```

Cells(i + 55, j + 1).ClearContents
Next j
Next i
'Заполнение матрицы МАИ от 1 до 9, от 1/9 до 1 (обратно-симметричной)
For i = 1 To n
  For j = i + 1 To n
    If i <> j Then
      If IsEmpty(Cells(i + 3, j + 1).Value) = False Then
        'разница голов K1 и K2
        k = Hour(Cells(i + 3, j + 1).Value) - Minute(Cells(i + 3, j + 1).Value)
        'разница голов K2 и K1
        k1 = Hour(Cells(j + 3, i + 1).Value) - Minute(Cells(j + 3, i + 1).Value)
        k = k - k1
        'K1 выиграла
        If k > 0 Then
          c = ax * k + bx
          k = Round(c)
          If k > 9 Then k = 9
          If k < 1 Then k = 1
          Cells(i + rtMAI, j + 1).Value = k
          Cells(j + rtMAI, i + 1).Value = 1 / k
        End If
        'Ничья
        If k = 0 Then
          Cells(i + rtMAI, j + 1).Value = 1
          Cells(j + rtMAI, i + 1).Value = 1
        End If
        'K1 проиграла
        If k < 0 Then
          c = -ax * k + bx
          k = Round(c)
          If k > 9 Then k = 9
          If k < 1 Then k = 1
          Cells(i + rtMAI, j + 1).Value = 1 / k
          Cells(j + rtMAI, i + 1).Value = k
        End If
      End If
    End If
  Next j
Next i
'формат "числовой" для ячеек матрицы

```

```

For i = 1 To n
    For j = 1 To n
        Cells(i + rtMAI, j + 1).NumberFormat = "0.00"
    Next j
Next i

'создание копии таблицы МАИ
'Call CopyTable(rtMAI + 1, 2, rtMAICopy, 2, n)

End Sub

'Проведение интерполяции для элементов матрицы МАИ
'n - размер таблицы (количество команд чемпионата)
Sub Interpol()
Dim i, i1, j, j1, k, n As Integer
n = Cells(3, 30).Value
Dim b As Boolean
Dim v As Double

b = False 'нулевое значение не найдено
While Not b
'Поиск нулевой оценки
    For i = 1 To n
        For j = i To n
            If (Cells(i + rtMAI, j + 1).Value < FloatZero) Then
                i1 = i
                j1 = j
                b = True 'нулевую оценку нашли
            End If
        Next j
    Next i

'Замена нулевой оценки на интерполированную
    If b Then
        k = 0
        For i = 1 To n
            If (Cells(i1 + rtMAI, i + 1).Value > FloatZero) And (Cells(i + rtMAI, j1 +
1).Value > FloatZero) Then
                Cells(i1 + rtMAI, j1 + 1).Value = Cells(i1 + rtMAI, j1 + 1).Value +
Cells(i1 + rtMAI, i + 1).Value * Cells(i + rtMAI, j1 + 1).Value
                k = k + 1
            End If
        Next i
    End If
End While
End Sub

```

```

        End If 'изменили aij=aik*akj !!!!!!
    Next i

    Cells(i1 + rtMAI, j1 + 1).Value = Cells(i1 + rtMAI, j1 + 1).Value / k

    k = 0
    For j = 1 To n
        If (Cells(j1 + rtMAI, j + 1).Value > FloatZero) And (Cells(j + rtMAI, i1 +
1).Value > FloatZero) Then
            Cells(j1 + rtMAI, i1 + 1).Value = Cells(j1 + rtMAI, i1 + 1).Value +
Cells(j1 + rtMAI, j + 1).Value * Cells(j + rtMAI, i1 + 1).Value
            k = k + 1
        End If
    Next j

    Cells(j1 + rtMAI, i1 + 1).Value = Cells(j1 + rtMAI, i1 + 1).Value / k

    'средняя проинтерполированная оценка
    v = (Cells(i1 + rtMAI, j1 + 1).Value + 1 / Cells(j1 + rtMAI, i1 + 1).Value) /
2

    'приведение оценки к диапазону от 1/9 до 9
    If Cells(i1 + rtMAI, j1 + 1).Value > 1 Then
        Cells(i1 + rtMAI, j1 + 1).Value = Round(v)
        If Cells(i1 + rtMAI, j1 + 1).Value > 9 Then Cells(i1 + rtMAI, j1 + 1).Value
= 9

        Cells(j1 + rtMAI, i1 + 1).Value = 1 / Cells(i1 + rtMAI, j1 + 1).Value
    Else
        v = (1 / Cells(i1 + rtMAI, j1 + 1).Value + Cells(j1 + rtMAI, i1 + 1).Value)
/ 2

        Cells(j1 + rtMAI, i1 + 1).Value = Round(v)
        If Cells(j1 + rtMAI, i1 + 1).Value > 9 Then Cells(j1 + rtMAI, i1 + 1).Value
= 9

        Cells(i1 + rtMAI, j1 + 1).Value = 1 / Cells(j1 + rtMAI, i1 + 1).Value
    End If
End If
    b = Not b
Wend
End Sub
Sub LoseWinByMAI()
Dim i, i1, j, j1, k, n As Integer

```

```

n = Cells(3, 30).Value
For i = 1 To n
    For j = 1 To n
        If i <> j Then
            If IsEmpty(Cells(i + 3, j + 1).Value) = True Then
                If Cells(i + 55, j + 1).Value > 1 Then
                    Cells(i + 3, j + 1).Value = "+"
                ElseIf Cells(i + 55, j + 1).Value = 1 Then
                    Cells(i + 3, j + 1).Value = "="
                Else
                    Cells(i + 3, j + 1).Value = "-"
                End If
            End If
        End If
    Next j
Next i
End Sub

Sub accuracy()
Dim i, i1, j, j1, k, n, b, predictionCount, correctResultCount As Integer
n = Cells(3, 30).Value
predictionCount = 0
correctResultCount = 0
    ' If (Cells(i + rtTurnir, 6 + n).Value = k) And (k4 > k5) Then
    '     k1 = i + rtTurnir
    ' End If
For i = 1 To n
    For j = 1 To n
        If i <> j Then
            If Cells(i + 3, j + 1).Interior.ColorIndex <> 2 Then
                predictionCount = predictionCount + 1
                If (Cells(i + 3, j + 1).Interior.ColorIndex = 43) And Cells(i + 3, j
+ 1).Value = "+" Then
                    correctResultCount = correctResultCount + 1
                ElseIf (Cells(i + 3, j + 1).Interior.ColorIndex = 36) And Cells(i
+ 3, j + 1).Value = "=" Then
                    correctResultCount = correctResultCount + 1
                ElseIf (Cells(i + 3, j + 1).Interior.ColorIndex = 46) And
Cells(i + 3, j + 1).Value = "-" Then
                    correctResultCount = correctResultCount + 1
                End If
            End If
        End If
    Next j
Next i
End Sub

```

```

        End If
    End If
Next j
Next i
Cells(Cells(4, 24).Value, 26).Value = predictionCount
Cells(Cells(4, 24).Value, 27).Value = correctResultCount
Cells(Cells(4, 24).Value, 28).Value = correctResultCount / predictionCount
Cells(4, 24).Value = Cells(4, 24).Value + 1
End Sub

Sub graphData()
Dim i, i1, j, j1, k, n, f, predictionCount, correctResultCount As Integer
n = Cells(3, 30).Value 'total number of teams
k = Cells(3, 34).Value 'total number of tours
f = Cells(3, 26).Value 'start tour number
Cells(3, 26).Value = 38
Call CompleteTheLeagueTable
Call SecondTable
Call LoseWinColors
Cells(3, 26).Value = f
For i = f To k
Call CompleteTheLeagueTable
Call ZapMAI
Call Interpol
Call LoseWinByMAI
Call accuracy
Cells(3, 26).Value = Cells(3, 26).Value + 1
Next i
End Sub

```

Прогнозування результатів командних змагань на основі конструктивного підходу та методу аналізу ієрархій

Кумпан С.В., Шинкаренко В.І. Український державний університет науки і технологій

Метод аналізу ієрархій (МАІ) використовується для прийняття рішень в разі наявності деякого набору альтернатив у багатьох сферах: соціальній, економічній, індустріальній, політичній, екологічній, бізнесу та інших. МАІ може вирішувати такі задачі як: оцінка якості об'єктів, наприклад ефективності підрозділів компаній у різних містах чи країнах, ефективності персоналу, аналіз балансу затрат, вигоди та ризиків для декількох проектів, маркетингові дослідження, планування найбільш ефективних заходів, тощо.

У даній роботі досліджується ефективність застосування МАІ у організації та прогнозуванні результатів спортивних змагань. Концептуальна ідея МАІ полягає у виборі із декількох альтернативних варіантів найкращого. У випадку спортивних змагань такими альтернативами є учасники змагань – окремі спортсмени або команди.

Найбільш ефективними у будь якому напрямі діяльності людини є задачі аналізу, оптимізації, прогнозу та прийняття рішень.

Розроблені конструктивно-продукційні моделі які призначені для вирішення задачі прогнозування спортивних результатів і моделюють процеси перетворення традиційних таблиць результатів футбольних матчів у відповідну таблицю для застосування МАІ та зворотнього перетворення.

Запропоновано алгоритм, який досліджує точність прогнозування результатів спортивних змагань в залежності від кількості відсутніх альтернатив, які будуть інтерпольовані на основі наявних коефіцієнтів МАІ.

Для експериментального дослідження використовується інформація щодо результатів матчів команд у футбольних чемпіонатах Англії.

Розроблено алгоритм, який на першому кроці розраховує відносні сили команд одна до одної за допомогою МАІ на основі неповної традиційної таблиці футбольних чемпіонатів. Роль експерта виконує природа, а саме результати гри команд безпосередньо на футбольному полі у конкретному матчі. Таким чином результат парних порівнянь залежить від багатьох випадкових факторів (дискваліфікацій гравців, травм, фізичного та психологічного стану гравців і т. і.).

На наступному кроці виконується розрахунок відсутніх коефіцієнтів відносної сили команд за допомогою інтерполяції на основі усередненої операції транзитивного замикання використовуючи коефіцієнти, отримані за допомогою МАІ. На основі отриманих коефіцієнтів виконується прогнозування результатів «незіграних» матчів.

На останньому етапі виконується розрахунок точності прогнозування в залежності від кількості зіграних турів чемпіонату.

За результатами експерименту формується графік точності прогнозування в залежності від туру, з якого почалось прогнозування.

Прогнозування результатів футбольних матчів за допомогою МАІ має досить невелику точність, хоча вона і більша, ніж прогнозування випадковим способом. На результат кожного матчу впливає велика кількість випадкових факторів, які змінюються з плином часу від одного матчу до іншого. Таким чином можна припустити, що прогнозування на основі класичного МАІ не є достатнім, щоб робити на його основі деякі висновки, або результати футбольних матчів не є в достатній мірі прогнозованими.

Для поліпшення точності прогнозування можна модифікувати алгоритм так, щоб при розрахунку відносної сили команд він враховував, чи був між ними зіграний лише один матч, чи два. Також можна додати алгоритм, який буде враховувати більше критеріїв, які команди показують протягом чемпіонату, наприклад середній час володіння м'ячем, середній час на тій чи іншій половині поля, середню кількість реалізованих моментів.

Проект статті

Прогнозування результатів командних змагань за допомогою МАІ

Сергій Кумпан, Віктор Шинкаренко

Факультет Комп'ютерні Інформаційні Технології

Український Державний Університет Науки і Технологій

Дніпро, Україна

sergejkumpan82@gmail.com

Ключові слова: метод аналізу ієрархій, прогнозування результатів змагань, матриця попарних порівнянь.

Вступ

На сьогоднішній день прогнозування результатів спортивних змагань є вкрай популярною сферою людської діяльності. Самі по собі спроби людей спрогнозувати переможця в яких небудь видах спорту, де учасники змагаються один з одним існували з давніх давен. Втім спроби наукового підходу до прогнозування з'явилися лише у 20-му столітті. Це пов'язано із таким факторами як: відкриття нових законів та алгоритмів теорії вірогідності та математичної статистики та розвиток математики як науки в цілому, винахід та швидкий розвиток електронних обчислювальних машин, поява способів зберігання великих обсягів даних та швидкого доступу до них.

Прогнозування результатів спортивних поєдинків є складним завданням, адже є велика кількість об'єктивних факторів та випадкових подій, які можуть вплинути на результат змагання. Тому існує велика кількість підходів, які намагаються врахувати всі фактори, або виявити і використати найважливіші з них для того, щоб зробити якомога більшу кількість вірних прогнозів.

Один з таких підходів полягає у спробі присвоїти кожному учаснику змагання якогось роду рейтинг, за яким можна буде порівнювати силу учасників змагання. Для кращого відображення сили тієї чи іншої команди рейтинг може складатися з декількох величин, наприклад рейтинг гри команди в атаці і у захисті.

Іншим підходом є використання методу аналізу ієрархій(МАІ) і принципу транзитивного замикання. В деякому

сенсі використання цього методу є природним для вирішення даної задачі для будь якого виду спорту, в якому є рахунок. Адже відношення наприклад забитих і пропущених м'ячів у футболі може виступати оцінкою відносної сили команд.

Окрім цих методів існують і інші підходи, втім важливо зазначити, що жоден з цих методів не дає абсолютної гарантії правильного вибору. В переважній більшості видів спорту величезну роль відіграють випадковості та людський фактор, особливо коли мова йде про прогноз результату однієї окремої гри.

Конструкційно-продукційні структури є відносно новим напрямком в науці. Методу є перспективним для формалізації опису тих чи інших об'єктів досліджень. В сучасних умовах стрімкого розвитку штучного інтелекту це відкриває нові підходи до розробки програмного забезпечення. Завдяки формальному опису об'єктів та алгоритмів стає можливим автоматизація процесу розробки та оптимізації програмного забезпечення для використання одного алгоритму для вирішення великого спектру задач, однакових за своєю концепцією. Такий підхід дозволить застосовувати алгоритм для різних сфер, не вносячи змін в програмне забезпечення зовсім, або модифікуючи лише окремі модулі, без переробки алгоритму від початку і до кінця.

Мета

Програмний продукт можна використати використовувати для прогнозування результатів будь яких командних змагань, які проходять у форматі чемпіонату. Завдяки використанню конструкційно-продукційного підходу буде можлива

подальша модернізація алгоритму, якщо виникне така необхідність. Основною метою роботи є визначення можливості та ефективності прогнозування результатів командних змагань на різних етапах чемпіонату.

Методика

Для проведення дослідів було обрано дані десяти чемпіонатів Англії з футболу. Для того, щоб можна було відслідкувати залежність точності прогнозування від кількості наявних альтернатив зберігається розбиття даних по турам. В кожному турі десять ігор.

В табличному процесорі Microsoft Office Excel за допомогою мови програмування Visual Basic for Applications (VBA) було створену програму, яка проводить серію експериментів.

Кожен експеримент починається з підготовки даних. Програма заповнює турнірну таблицю результатами матчів визначеної кількості турів. Таким чином у таблиці будуть як заповнені поля, що означає, що команди вже зустрічалися між собою протягом турніру, так і пусті клітини. Алгоритм буде прогнозувати результат зустрічей команд, які ще жодного разу не грали між собою.

На першому етапі програма заповнює таблицю МАІ таким чином, що коефіцієнт відносної сили команд підраховується на основі кількості голів, забитих цими командами під час зустрічі одна з одною.

На другому етапі за методом транзитивного замикання виконується переможця у зустрічі тих команд, які ще не грали між собою. Таким чином отримуються експериментальні дані спрогнозованих результатів футбольних матчів. Варто зазначити, що результатом прогнозу є перемога чи поразка команди, або нічия, а не рахунок, який міг би бути у матчі.

Результати

Усі спрогнозовані результати було порівняно з тими, що були насправді, і підрахована точність прогнозу на кожному етапі. В ході експерименту було побудовано таблицю попарних порівнянь МАІ.

МАІ	АРС	АСТ	БЛБ	БОЛ	ВБА	ВУЛ	КПР	ЛИВ
Домашня команда								
Арсенал	1,00	2,00	3,00	4,00	4,00	1,00	3,00	0,25
Астон Вилла	0,50	1,00	4,00	1,00	0,33	1,00	1,00	0,25
Блэкберн Рovers	0,33	0,25	1,00	0,33	0,33	0,33	0,33	0,17
Болтон Уондерерс	0,25	1,00	3,00	1,00	1,00	1,00	1,00	0,25
Вест Бромвич Альбйон	0,25	3,00	3,00	1,00	1,00	4,00	1,00	0,25
Вулверхэмптон Уондерерс	1,00	1,00	3,00	1,00	0,25	0,50	0,25	0,25

Рисунок 5 Таблиця попарних порівнянь МАІ

За результатами експериментів було виявлено, що при заповненні таблиці на 40 відсотків, точність прогнозування складає 37 відсотків. При подальшому доповненні таблиці новими даними точність поступово зростає до 50 відсотків при заповненні таблиці на 95 відсотків.



Рисунок 6 Залежність точності прогнозування від кількості відсутніх альтернатив

Наукова новизна та практична значимість

У цій статті вперше поставлено завдання розробити алгоритм прогнозування результатів командних змагань на основі конструкційно-продукційного підходу та МАІ. Виконано дослідження ефективності використання зазначених методів для прогнозування в залежності від кількості наявних альтернатив.

Висновки

Встановлено, що починаючи з 35 відсотків заповнення турнірної таблиці можна використовувати МАІ для прогнозування результатів командних змагань. Точність таких прогнозів буде невеликою, а зростання точності зі зростанням кількості початкових даних

становить близько 15 відсотків.

Метод аналізу ієрархій і демонструє деяку точність при прогнозуванні результатів командних змагань. Втім самого по собі цього методу недостатньо для надійних прогнозів, тому необхідна його комбінація з іншими методами.

БІБЛІОГРАФІЧНИЙ СПИСОК

1. Thomas L. Saaty The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation.
2. Viktor Shynkarenko, Tetiana Vasetska , and Iryna Vyshnyakova Analytic Hierarchy Process Sustainability at the Significant Number of Alternatives Ranking.
3. Analytic hierarchy process [ел.ресурс]
https://en.wikipedia.org/wiki/Analytic_hierarchy_process.
4. Ossama Salem, Baris Salman, Sudipta Ghora Accelerating construction of roadway alternative techniques and procurement methods.
5. John R. Grandzol Improving the Faculty Selection Process in Higher Education: A Case for the Analytic Hierarchy Process