

Міністерство освіти і науки України
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«НАЦІОНАЛЬНИЙ ГІРНИЧИЙ УНІВЕРСИТЕТ»



М.А. Демиденко

СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

Навчальний посібник

Дніпропетровськ

НГУ

2016

УДК 330.47+(075.8)

БКК 65.050.2

ДЗ0

Рекомендовано вченою радою як навчальний посібник по дисципліні "Системи підтримки прийняття рішень" для студентів напряму підготовки 6.050102 "Економічна кібернетика", (Протокол № 5 від 22.12.2015).

Рецензенти:

Мещеряков Л.І.- д-р техн. наук, проф.(ДНВЗ "Національний гірничий університет"; кафедра «Програмного забезпечення комп'ютерних систем»)

Ковальчук К.Ф. д-р техн. наук, проф.(Національна металургійна академія України, декан факультету економіки та менеджменту)

Демиденко М.А.

ДЗ0

Системи підтримки прийняття рішень : навч. посіб. / М.А. Демиденко; Нац. гірн. ун-т. — Електрон. текст. дані. – Д. : 2016. – 104 с. – Режим доступу: <http://ntu.org.ua> (дата звернення: 17.10.2015). – Назва з екрана.

ISBN 978-912-350-293-9

Викладено основні поняття і методологічні основи систем підтримки прийняття рішень . Матеріал подано на рівні, доступному студентам, які знайомі з курсом інформатики та математики для економістів. Методи, що розглядаються в посібнику, ілюструються великою кількістю прикладів. Посібник має на меті навчити студентів застосовувати сучасні системи підтримки прийняття рішень для отримання оптимальних управлінських рішень. Посібник базується на досвіді викладання дисципліни «Системи підтримки прийняття рішень» в ДВНЗ «Національний гірничий університет», призначений для студентів вищих учбових закладів і може бути корисним для економістів, плановиків, менеджерів та маркетингологів.

УДК 330.47+(075.8)

БКК 65.050.2

ISBN 978-912-350-293-9

© М.А. Демиденко, 2016

© НГУ , 2016

Будьте ввічливі з зубрилами і ботаніками. Не виключено, що незабаром ви будете працювати на одного з них.

Білл Гейтс



Системи підтримки прийняття рішень

ЗМІСТ

Передмова	1
ТЕМА 1. Місце систем підтримки прийняття рішень в інформаційних системах та визначення методик отримання і обробки знань	2
Алгоритми прийняття управлінських рішень.....	3
Взаємодія учасників при прийнятті рішень.....	6
Менеджмент знань.....	8
Класифікація знань за їх атрибутами.....	9
Поняття системи обробки знань	10
Висновки.....	11
Контрольні запитання.....	11
ТЕМА 2. Основи систем підтримки прийняття рішень	12
Типова архітектура для систем підтримки прийняття рішень.....	12
Текстові СППР	13
Гіпертекстові, Web орієнтовані СППР	13
СППР орієнтовані на використання баз даних та сховищ даних.....	14
Табличні СППР	15
Орієнтовані на моделі СППР	15
СППР, які використовують штучний інтелект.....	15
Гібридні СППР.....	16
Групові СППР	17
Висновки.....	18
Контрольні запитання.....	18
ТЕМА 3. Бази даних і сховища даних для підтримки прийняття рішень	19
Архітектура сховищ даних.....	21
Моделі представлення даних у сховищах даних (БСД).	22
Вітрини даних.....	30
Поточний стан і майбутній розвиток сховищ даних.....	34
Висновки.....	37
Контрольні запитання.....	37
ТЕМА 4. Моделі в системах СППР.....	38

Вибір критеріїв в оптимізаційні моделях систем СППР	39
Економіко-математичні моделі лінійного програмування	41
Цілочислові задачі лінійного програмування	42
Задачі дробово-лінійного програмування	42
Нелінійне моделювання. Загальна постановка задач нелінійного програмування	43
Алгоритм розв'язання задачі методом множників Лагранжа	44
Моделі, які застосовують математичний апарат теорії ігор.....	45
Постановка задачі	45
Розв'язок гри із сідловою точкою	45
Розв'язок гри, що не має сідлової точки	46
Розв'язок задач теорії ігор за допомогою методів лінійного програмування	47
Моделі які використовують нейронні мережі	49
Штучні нейронні мережі	52
Навчання штучної нейронної мережі	53
Методи і моделі прогнозу які застосовуються в системах підтримки прийняття рішень	54
Підготовка вихідних даних для прогнозування	55
Регресійні моделі прогнозування	62
Метод ковзної середньої, метод експоненціального згладжування	63
Метод статистичної екстраполяції	65
Програмне забезпечення для моделювання в системах СППР.....	66
Вільне (безкоштовне) оптимізаційне програмне забезпечення	66
Ліцензійне програмне забезпечення для розв'язання задач оптимізації ...	68
Безкоштовне програмне забезпечення для університетів	69
Висновки.....	70
Контрольні запитання	70
ТЕМА 5. Методи обробки даних в системах підтримки прийняття рішень	71
Online Analytical Processing (OLAP) он-лайн аналітична обробка в системах підтримки прийняття рішень.....	71
Дейтамайнінг в СППР	72
Методи добування даних в СППР.....	73
Висновки.....	91

ТЕМА 6. Ефективність запровадження систем підтримки прийняття рішень ..	92
Оцінка розрахункової економічної ефективності системи на етапі проектування	92
Оцінка економічної ефективності системи на етапі впровадження в експлуатацію	93
Висновки.....	96
Контрольні запитання	97
Список рекомендованої та використаної літератури	98
Предметний покажчик	99

Передмова

Стрімкий розвиток інформаційних систем, нарощування можливостей комп'ютерів, поява портативних пристроїв — планшетів і смартфонів, хмарні технології обробки даних, надали нові можливості менеджерам і керівникам підприємств в управлінні і підвищенні конкурентоздатності бізнесу. Одним з таких засобів є системи підтримки прийняття рішень. В навчальному посібнику подано матеріал, який викладається студентам спеціальності “Економічна кібернетика” в дисципліні “Системи підтримки прийняття рішень” обсягом 144 годин (16 лекцій).

В посібнику подано шість тем які розкривають базовий матеріал по дисципліні.

В першій темі подано основні термінологічні формулювання, показано місце СППР в загальному різноманітті інформаційних систем. Розглянуто методики і алгоритми отримання економічно ефективних управлінських рішень. Роз'яснені поняття менеджменту знань і способів отримання знань за допомогою СППР.

В другій темі обговорюються основи функціонування систем підтримки прийняття рішень. Розглянуто класифікацію СППР. їх структуру, особливості, відмінності і принципи взаємодії з користувачем.

В третій темі розглянуто принципи побудови баз та сховищ даних в системах підтримки прийняття рішень. Показано їх архітектуру, моделі представлення даних, вітрини даних, поточний стан і майбутній розвиток сховищ даних.

В четвертій темі, розглянуто економіко математичні моделі які використовуються в СППР для отримання оптимальних управлінських рішень. Обговорено методи вибору критеріїв в задачах оптимізації. Розглянуто моделі лінійного, цілочислового, дробово-лінійного програмування, а також методи нелінійного програмування. Приділена увага моделям теорії ігор і нейронних мереж. Розглянуто методи прогнозування для задач СППР.

В п'ятій темі вивчаються методи обробки даних в СППР. Розглянуто он-лайн аналітична обробка в системах підтримки прийняття рішень, дейтамайнінг в СППР, методи добування даних в СППР, збереження даних, дистиляція даних, візуалізація даних, крос-табуляція, програмні агенти, довірчі мережі.

В шостій темі подано матеріали для оцінки всіх факторів економічної ефективності СППР, розрахунку економічного ефекту на стадії розробки СППР, тобто, до впровадження в експлуатацію і розрахунок економічного ефекту для системи яка знаходиться у промисловій експлуатації

В посібнику додано перелік літературних джерел які необхідно використовувати для поглибленого вивчення дисципліни і самостійної роботи.

ТЕМА 1. Місце систем підтримки прийняття рішень в інформаційних системах та визначення методик отримання і обробки знань

Інформаційні системи розвиваються більше ніж півсторіччя з 1960х — 1970х років. Спочатку інформаційні системи будувалися як автоматизовані системи управління підприємством. Ці системи мали відповідати запитам менеджерів. Такі системи продукували велику кількість необхідних і корисних для менеджерів і управління документів. Розвиток автоматизованих систем управління в подальшому привів до того, що з'явилися системи підтримки прийняття рішень.

Ці системи відрізняються можливостями які дозволяють користувачам отримувати інформацію або знання з інформаційної системи яка була запрограмована і налаштована користувачем так, щоб урахувати досвід користувача і його підприємства, а також можливості математичних моделей та сховищ даних які поліпшують якість прийняття рішень. Подальший розвиток інформаційних систем призводить до побудови обчислювальних середовищ, які включають в себе електронну комерцію, нові можливості аналітики, можливості колективної співпраці, проектування, розробку нової продукції.

Взаємозв'язок систем підтримки прийняття рішень з іншими дисциплінами які вивчають і використовують сучасні методи управління показано на рис. 1 .

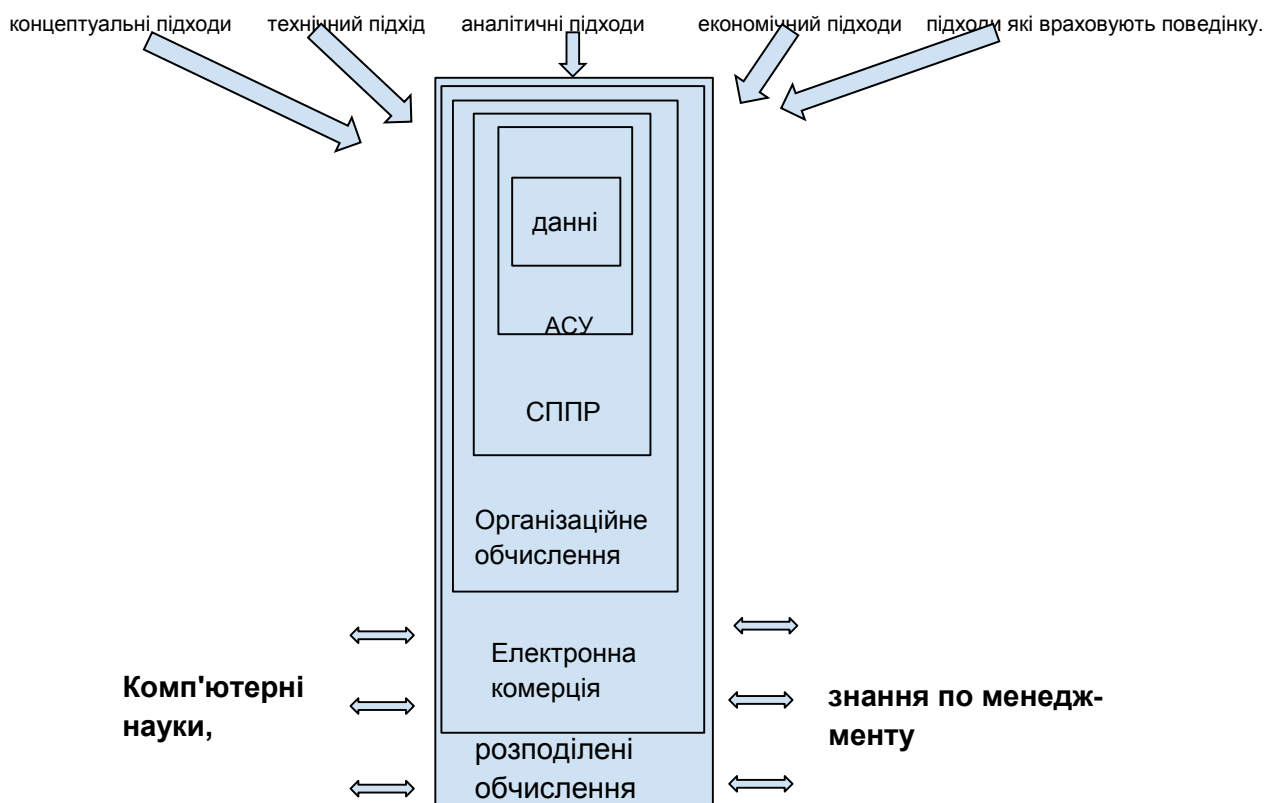


Рисунок 1. Місце систем прийняття рішень в інформаційних системах управління

Алгоритми прийняття управлінських рішень

Прийняття управлінських рішень можна розглядати як вибір з багатьох можливостей - альтернатив. Наприклад вибір стратегій поведінки, вибір якихось об'єктів. Таким чином, система підтримки прийняття рішень повинна допомагати менеджеру робити такий вибір. Традиційний метод прийняття керівництвом рішень в системах управління показано на рис. 2 .

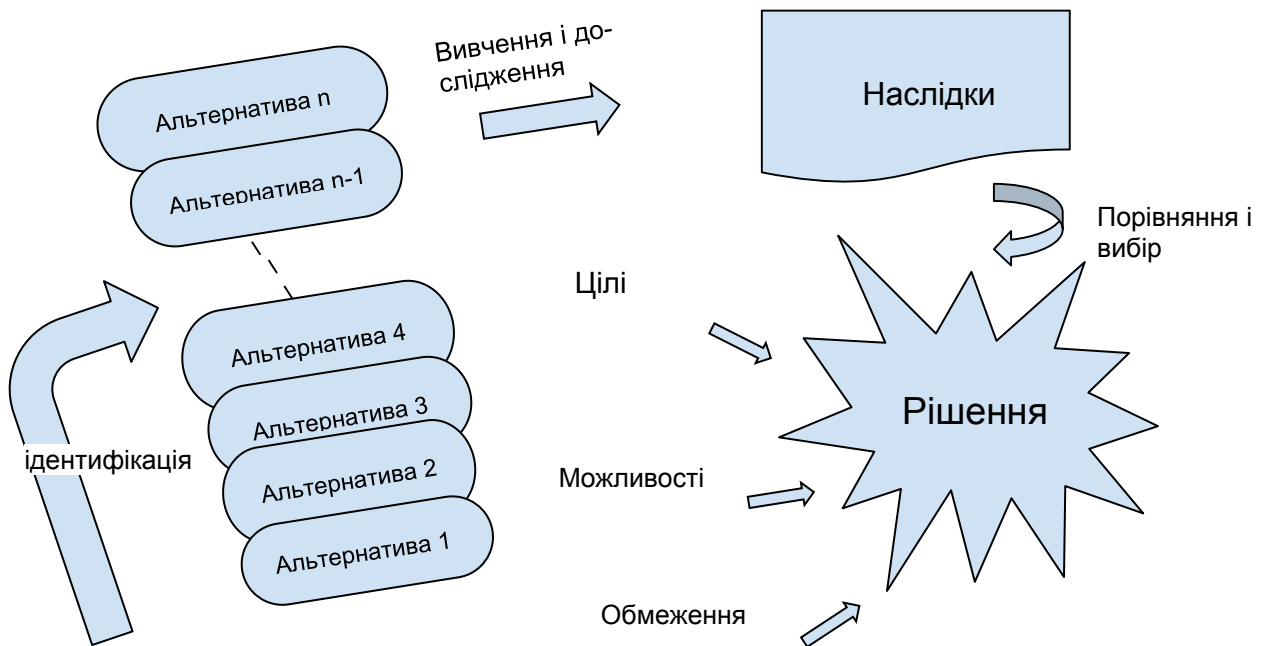


Рисунок 2. Традиційний метод прийняття рішень

На практиці кількість альтернатив може бути великою, тому менеджер вручну не може ефективно і якісно проаналізувати наслідки прийняття всіх альтернатив. В цій ситуації ефективною є система підтримки прийняття рішень, яка надає можливість за допомогою комп'ютера проаналізувати наслідки всіх альтернатив і змоделювати поведінку системи.

Розвиток науки управління привів до того що традиційний метод прийняття рішень не є найкращим. Розроблено низку нових методів серед яких виділяється метод прийняття рішень базований на знаннях.

Прийняття рішень в системі базованій на знаннях показано на рис. 3.

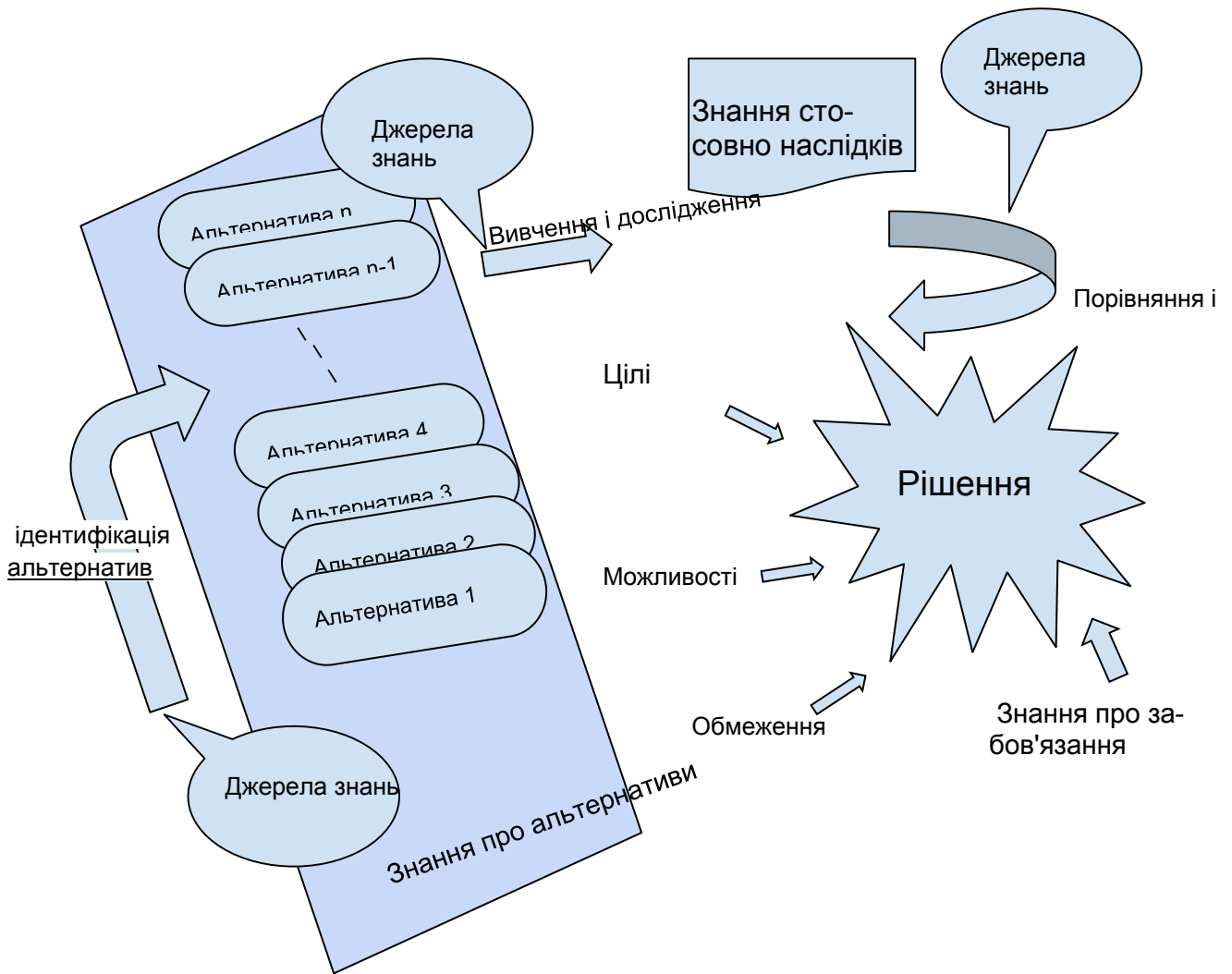


Рисунок 3 . Система прийняття рішень базована на знаннях.

В такій системі альтернативи аналізуються не тільки фахівцем, але і за допомогою комп'ютера та програмного забезпечення. В системі досліджуються наслідки застосування альтернатив, далі аналізуються інші джерела знань. Далі отримані знання обробляються з урахуванням цілей, критеріїв, обмежень і виробляються нові знання які приводять до прийняття науково обґрунтованих рішень. Таким чином система підтримки прийняття рішень являє собою виробничу систему яка виробляє нові знання.

Системи підтримки прийняття рішень можуть бути корисними для розробки слабо структурованих рішень . Особливості таких рішень показано в таблиці 1. СППР дають можливість виробити правильне рішення в умовах невизначеності чи часткової невизначеності та браку інформації та знань.

Таблиця 1

Структуровані рішення	Неструктуровані рішення
Регулярні, повторювані рішення	Ненадійні, рідко повторювані рішення
Засновані на стабільних умовах	Засновані на випадкових і умовах і даних
Альтернативи чіткі	Альтернативи неясні
Наслідки альтернатив визначені	Наслідки альтернатив невизначені
Критерії вибору чітко визначені	Критерії вибору неоднозначно визначені
Спеціальні знання відомі	Спеціальні знання необхідний для прийняття рішень невідомі
Необхідне знання легко доступні	Необхідне знання недоступно
Заздалегідь відомо повний набір кроків, для досягнення рішення	В процесі прийняття рішень застосовуються нестандартне мислення, мозковий штурм, синтез, аналогія.
Опора на традиції	Опора на розвідку, творчість, прозоріння, винахідливість

Процес прийняття рішень в СППР полягає у трьох фазах.

Перша фаза дослідницька . На цій фазі досліджуються знання та інформація з внутрішніх та зовнішніх джерел. В процесі дослідження розуміння проблеми стає більш ясними і обґрунтованим, альтернативи і їх наслідки більш зрозумілими.

Друга фаза - проектування. На цьому етапі менеджер формулює знання про результати альтернативних дій оцінює їх наслідки для функціонування системи. На цьому етапі менеджер може виробляти додаткові знання .

Третя фаза - вибір рішення . На цьому етапі менеджер виконує вибір між альтернативами з урахуванням їх наслідків які були проаналізовані і досліджені на першій і другій фазах.

Прийняте рішення може бути застосовано не одразу але через деякий час, який підтвердить правильність прийнятого рішення . Всі три фази не завжди йдуть послідовно вони можуть перекривати одна одну і процес прийняття рішення може бути ітеративним.

Метою застосування систем підтримки прийняття рішень є підвищення можливості і здібності менеджера для прийняття правильних і науково обґрунтованих рішень .

Взаємодія учасників при прийнятті рішень

Взаємодія учасників прийняття рішень є складною. Часто прийняття рішень має ієрархічну структуру, в якій вищий менеджер приймає рішення із застосуванням пропозицій нижчестоящих менеджерів. Структурованість може бути виражена в тому що менеджер приймає рішення за допомогою інших спеціалістів які його консультують. Такий спосіб прийняття рішень проілюстровано на рис. 4. В такій системі прийняття рішень вищий менеджер вирішує які поради враховувати і який консультант є компетентним в розробці того чи іншого рішення.

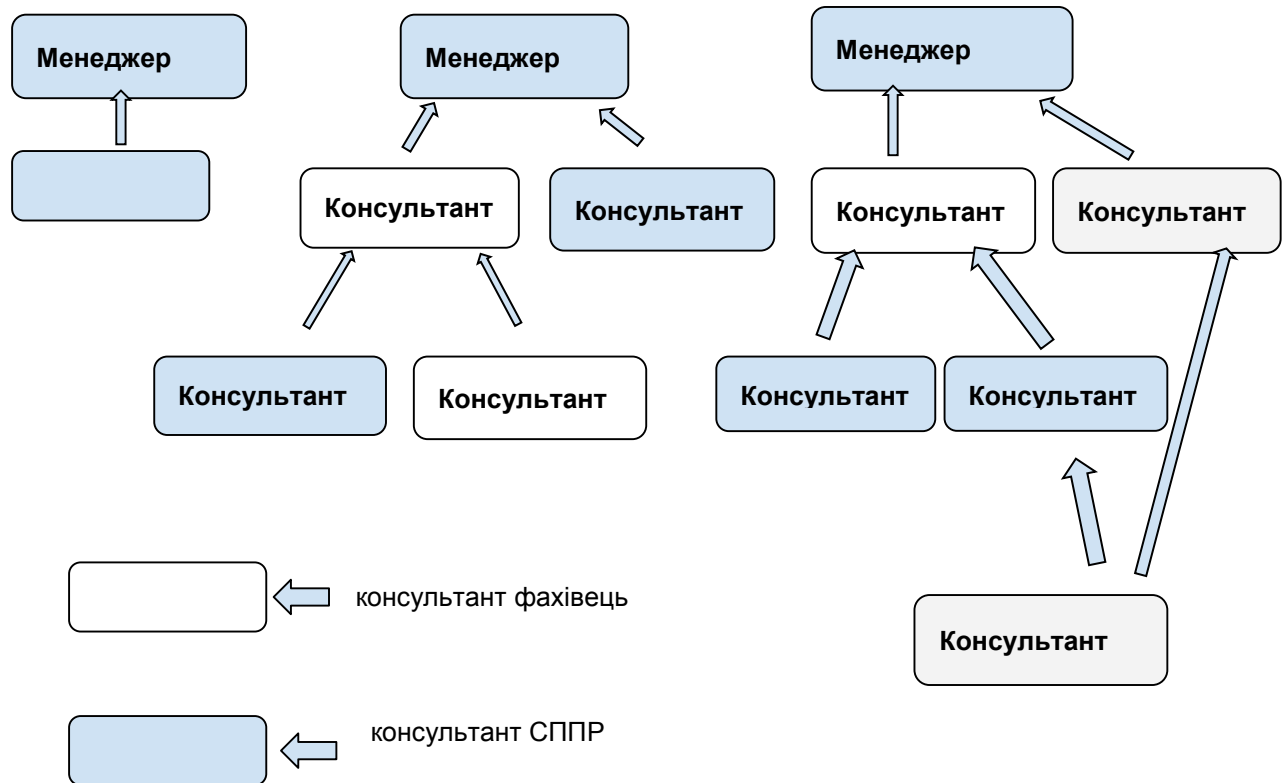


Рисунок 4 . Прийняття рішень в ієрархічній структурі

Існує більш цілісний підхід для урахування зусиль всіх учасників прийняття рішення. Це метод багатостороннього прийняття рішень Рис.5. В цьому методі існує центральний центр управління, який об'єднує всіх учасників процесу і дозволяє отримувати реальному часі всі надані пропозиції, оцінювати їх і приймати узгоджений усіма результат. Таким центром може бути комп'ютер системи підтримки прийняття рішень.

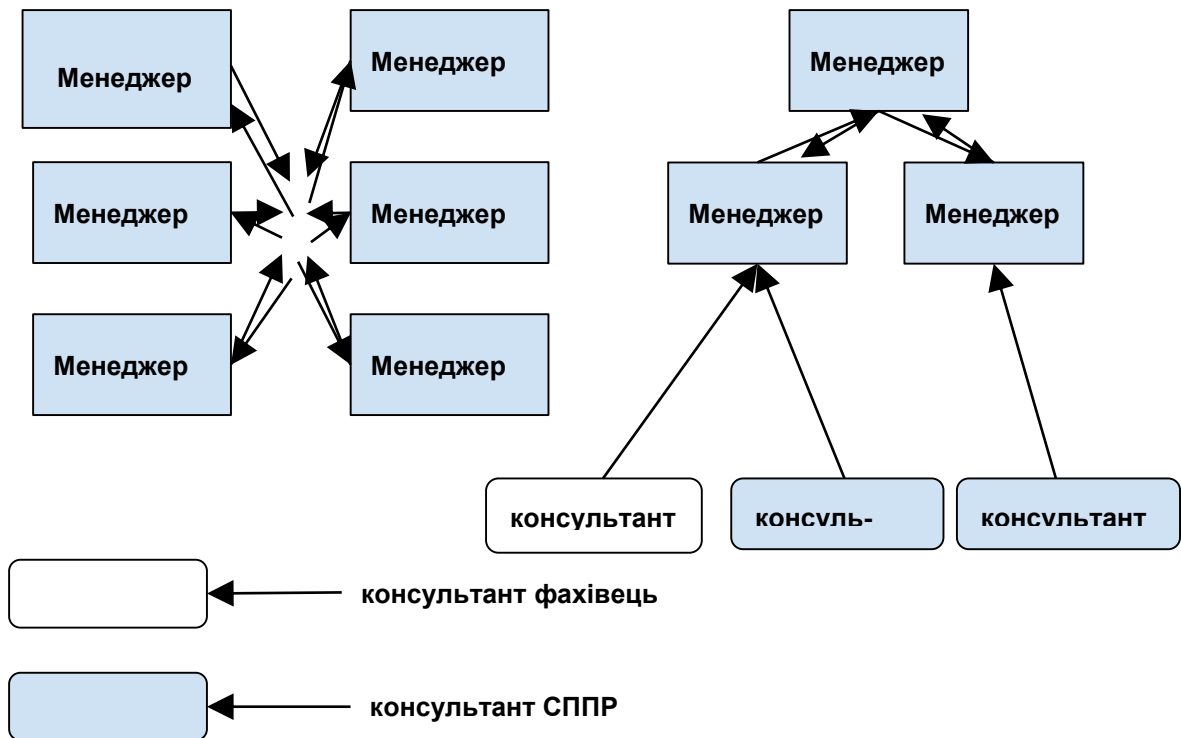


Рисунок 5 Методика багатостороннього прийняття рішень

Така система має декілька переваг. В системі бере участь більше учасників, тобто більше знань може бути застосовано при прийнятті рішень, більша кількість учасників одночасно обробляє проблему тому прийняте рішення може бути більш якісним.

На рисунку 6 показано інша організація багатостороннього прийняття рішень. В цій системі група консультантів знаходиться у центрі і консультує групу менеджерів які приймають рішення.

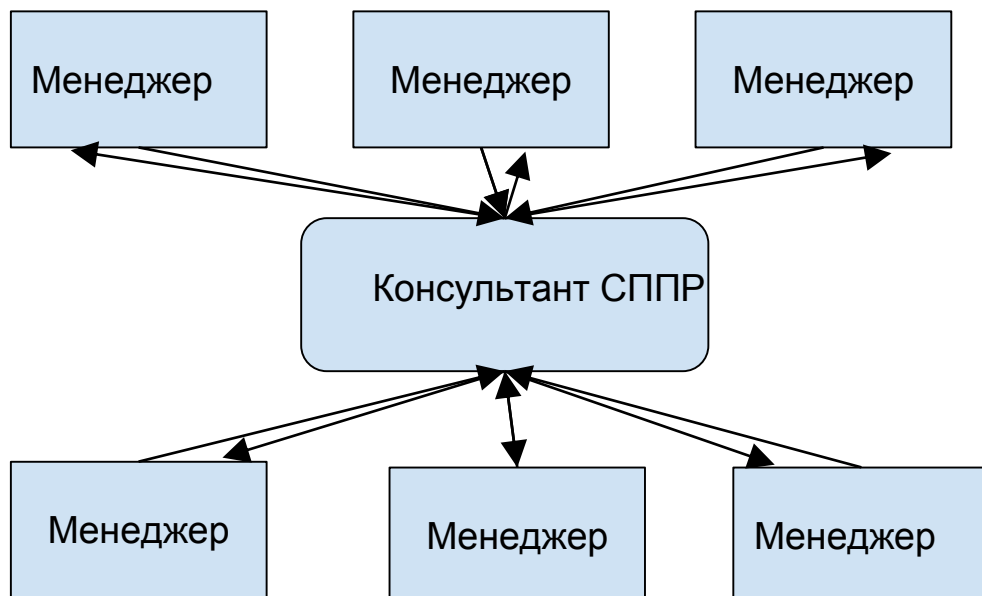


Рисунок 6. Приклади багатосторонньої системи прийняття рішень.

В системах підтримки прийняття рішень застосовуються всі позначені вищі методи. Не існує загального критерію згідно якого можна прийняти одні методи, а інші відкинути. В кожному конкретному випадку вибирають метод який найкраще підходить для конкретної системи управління.

Менеджмент знань

Прийняття рішень сучасних системах є наукомістким, тому є необхідними обробка та управління знаннями. Так само, як комп'ютерні науки формують технологічну основу для реалізації систем підтримки прийняття рішень, управління знаннями утворює інтелектуальну основу для розробки, вивчення і застосування систем підтримки прийняття рішень.

Алгоритм отримання знань показано на рисунку 7.

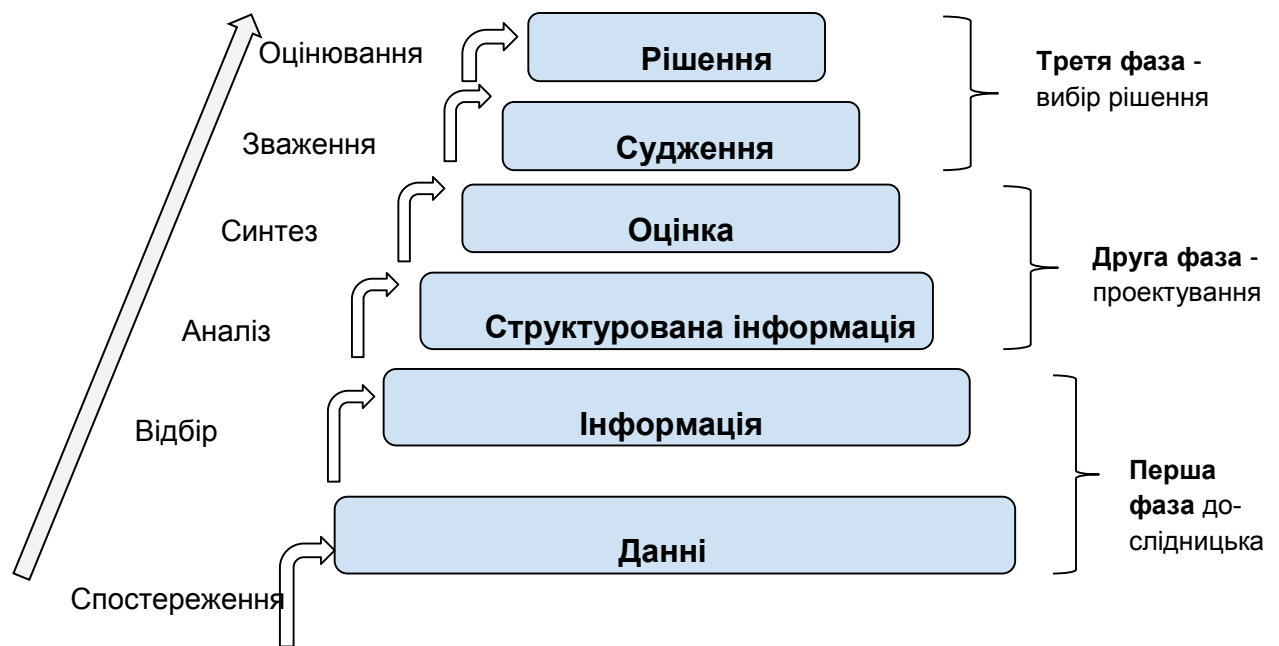


Рисунок 7 Алгоритм отримання знань

На фазі дослідження менеджер або система управління отримує дані які обробляються і вилучається інформація. Інформація це щось нове, що ми отримали з цих даних. Далі отримана інформація на фазі проектування структурується, приймаються судження які аналізуються і наприкінці приймається управлінське рішення.

Проілюструємо фази отримання знань за допомогою таб. 2

Фази отримання знань

Таблиця 2

Фази отримання знань	Результат
Данні	240
Інформація	240 є рівнем холестерину
Структурована інформація	240 є поточним рівнем холестерину для Петренка В.
Оцінка	Рівень холестеролу для Петренка В. є дуже високим
Судження	Здоров'я Петренка В є в даний час в небезпеці
Рішення	Направити Петренка В. в госпіталь, терапевтичне відділення на дослідження і лікування протягом 7 днів .

Класифікація знань за їх атрибутами.

Розглянемо атрибути знань. Атрибути знань це виміри за якими класифікують знання в СППР. Можливість атрибутів може змінюватися у часі і в залежності від вимог. Тому атрибути які були важливі деякий час назад, можуть стати неважливим в поточному часі . Важливість атрибутів залежить від способів обробки інформації наприклад цифрова чи аналогова . Класифікація атрибутів знань дозволяє створити ефективну систему підтримки прийняття рішень .

Класифікація атрибутів знань за їх природою .

Таблиця 3

Атрибут	Природа атрибута
Режим	Отримання знань у неявному вигляді чи в явному вигляді
Тип	Описовий метод чи метод обмірковування
Предметна область	Предметна область де використовується знання (наприклад, маркетинг, політика, техніка, виробництво, сільське господарство)
Застосованість	Діапазон від локального до глобального.
Рівень управління	Оперативне управління, середньотермінове і стратегічне управління
Використання	Використання в практичних , наукових, мілітарних та інших цілях
Доступність	З приватним або публічним доступом
Корисність	Визначення корисності і важливості для досягнення цілей і застосування цих знань
Термін дії	Термін часу протягом якого ці знання можуть бути застосовані
Уміння	Ступінь умінь, які застосовані у цих знаннях

Атрибут	Природа атрибута
Джерело	Походження знань
Вік	Діапазон часу, в якому визначаються знання як нові і як застарілі
Термін життя	Термін зберігання знань
Мінливість	Встановлюються критерії за якими можуть змінюватися знання
Місцезнаходження	Вказується місце де знаходиться це знання. (географічне, організаційне, або технічне)
Абстрактність	Визначається діапазон, в якому знання характеризуються як конкретні і абстрактні
Здатність до програмування	Визначається, чи можливо ефективно запрограмувати використання цих знань

Поняття системи обробки знань

Як було вказано вище на рис 5 і 6, система обробки знань містить щонайменше два основних рівні. Консультаційний центр або люди які надають консультації менеджеру, який приймає рішення утворюють перший рівень обробки знань. Особи які приймають рішення створюють другий рівень обробки знань.

Згідно фундаментальних понять управління знаннями, на першому рівні є п'ять основних класів обробки знань : отримання, відбір, генерація, асиміляція, і розповсюдження.

На другому рівні обробки знань існує чотири етапи обробки: керівництво знаннями, координація знань, контроль знань, вимірювання знань, Таб.4 .

Форми обробки знань, які відбуваються при прийнятті рішень Таблица 4.

Рівень	Клас обробки	Опис
Перший рівень	Отримання знань	Отримання знань джерел зовнішніх порівняно до особи яка приймає рішення. Обробка знань для наступного прийняття рішень.
	Відбір знань	Відбір знань для прийняття рішення
	Генерація знань	Розробка (генерація, створення) нових знань на базі отриманих та відібраних знань
	Асиміляція знань	Переклад знань на внутрішню мову системи підтримки прийняття рішень
	Розповсюдження знань	Запровадження отриманих знань в управлінні системою, підприємством
Другий рівень	Керівництво знаннями	Створення умов, які полегшать плідне застосування обробки знань
	Координація знань	Керування залежностями в Управлінні знаннями, щоб гарантувати, що відповідні менеджери і ресурси задіяні адекватно в потрібний час прийняття рішення.

Рівень	Клас обробки	Опис
	Контроль знань	Забезпечення того, що менеджери і ресурси, необхідні у процесі прийняття рішення доступні в достатній якості і, відповідно до вимог управління об'єктом .
	Вимірювання знань	Оцінка вартості ресурсів, знань , і їх впровадження в рамках прийняття рішення.

Висновки

Знання і управління знаннями є невід'ємною частиною системи управління та систем підтримки прийняття рішень. Існує п'ять методів обробки інформації першого рівня і чотири методи обробки інформації другого рівня. Система підтримки прийняття рішень, яка реалізує або сприяє сполученню цих дев'яти класів обробки знань може реально покращити продуктивність управлінських рішень.

Розробники систем підтримки прийняття рішень мають враховувати наступні фактори:

- Наукомісткій характер процесу прийняття рішень;
- Процес прийняття рішень орієнтується на знання;
- Застосування атрибутів знань;
- Застосування багаторівневих і багатосторонніх схем прийняття рішень;
- Застосування дворівневої системи обробки знань яка приймає активну участь в розробці управлінських рішень;
- Застосування в системах підтримки прийняття рішень і методів управління знаннями в сучасній науці та практиці підвищується.

Контрольні запитання

1. Яке місце займає система СППР серед інформаційних систем?
2. Чим відрізняється система управління базована на знаннях від традиційної системи управління?
3. Які особливості розробки слабо структурованих рішень?
4. Які особливості взаємодії учасників при прийнятті рішень в ієрархічній структурі управління і системі, що використовує багатосторонній метод прийняття рішень?
5. Що таке менеджмент знань? Дайте визначення менеджменту знань.
6. Дайте класифікацію знань з їх атрибутами.

ТЕМА 2. Основи систем підтримки прийняття рішень

Типова архітектура для систем підтримки прийняття рішень

Структура систем підтримки прийняття рішень включає в себе наступні елементи

1. Лінгвістична система;
2. Система подання даних та інформації;
3. Система обробки задач;
4. Системи обробки задач конкретних предметних областей, системи знань.

На рис. 8 показана взаємодія елементів і структура системи підтримки прийняття рішень.

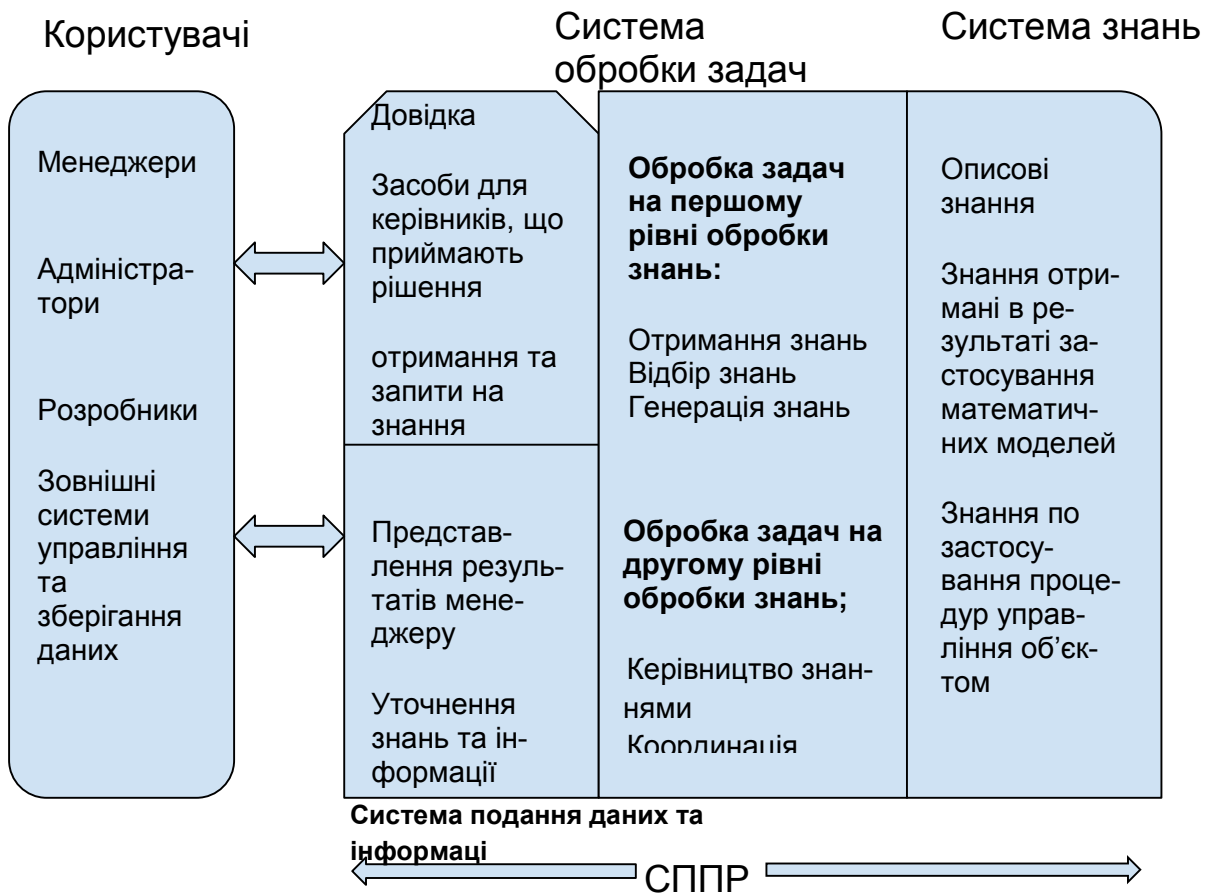


Рисунок 8. Архітектура системи підтримки прийняття рішень

У загальній архітектурі СППР, ми бачимо важливі і фундаментальні аспекти, загальні для всіх систем підтримки прийняття рішень. Конкретна система підтримки прийняття рішень буде містити чотири вказаних елемента, але треба буде додатково сформулювати особливості цієї системи, які враховують вимоги, що висуваються до цієї системи.

Основні ідеї архітектури систем підтримки прийняття рішень були розроблені у 70 - 80-х роках двадцятого сторіччя . У подальшому ці ідеї були конкретизовані і розроблені у декількох напрямках . Створенні інформаційної системи можна класифікувати наступним чином.

- Текстові СППР;
- Гіпертекстові СППР;
- Орієнтовані на використання баз даних та сховищ даних СППР;
- Табличні СППР;
- Орієнтовані на моделі СППР;
- СППР, які використовують штучний інтелект;
- Гібридні СППР;
- Групові СППР.

Текстові СППР

Протягом багатьох років розробники, інженери, менеджери використовували текстову інформацію, наприклад, книжки , журнали, рекламну продукцію, як базу для прийняття рішень. Тому існує клас систем підтримки прийняття рішень які збирають, накопичують, обробляють, представляють користувачу інформацію у текстовому вигляді. Сучасні інформаційні системи дозволяють виконувати інтелектуальний аналіз тексту і виявляти нові знання для розв'язання конкретної проблеми управління чи технології.

Наприклад розробляється новий продукт. Необхідно отримати інформацію про технічні характеристики вимоги і особливості цього продукту. Інформаційна система збирає текстову інформацію про подібні розробки та такі продукти і представляє розробнику її у текстовому вигляді. Розробник розглядає інформацію, аналізує і генерує нову інформацію і знання для прийняття науково обґрунтованого рішення по розробці цього продукту .

Текстові системи підтримки прийняття рішень займають значне місце в сучасних інформаційних системах.

Гіпертекстові, Web орієнтовані СППР

Сучасні системи обробки текстової інформації базовані на електронній обробці текстів. Така обробка вимагає стандартизації і представлення інформації в такому вигляді , щоби вона могла бути оброблена спеціальними програмами незалежних виробників. Для цього було розроблено спеціальну мову обміну інформацією яка називається гіпертекст. Ця мова дозволяє за допомогою посилань зв'язати багато документів які пов'язані один з одним за змістом. Розвиток Інтернету надав більших можливостей гіпертекстовим СППР, які зараз є Web орієнтованими.

Веб-системи підтримки прийняття рішень є системи підтримки прийняття рішень, які доступні в Інтернеті. Web орієнтовані СППР можна ідентифікувати за наступними характеристиками:

- Мають зручний доступ на Web;
- Підтримують окремих осіб, клієнтів, співробітників, менеджерів, групи в процесі прийняття рішень, незалежно від їх фізичного розташування або часу доступу;
- Мають специфічний інтерфейс користувача властивий для Інтернет додатків;
- Використовують процеси прийняття рішень, які частково структуровані або неструктуровані і на різних етапах процесу прийняття рішень можуть бути виконані в Інтернеті;
- Використовують досвід, бази знань, документи, моделі та методи пошуку знань, великої і різноманітної групи користувачів, які є доступними через Інтернет.

Web орієнтовані СППР широко використовуються в електронній комерції. Системи надають можливість клієнтам виконати оптимальний вибір продукції, сервісів післягарантійного обслуговування. Система допомагає клієнту сформулювати критерії відбору товару, далі система знаходить у базах даних необхідної клієнту товар. СППР також забезпечує підтримку клієнта після продажу товару, а саме, повернення товару, ремонт технічне обслуговування.

Web орієнтовані СППР використовуються у сфері дизайну поліграфічної продукції, дозволяють моделювати формат видання, загальну концепцію дизайну, колірне оформлення, шрифтове оформлення, оформлення текстових блоків, графічне оформлення, декоративні елементи та інше.

Спектр застосування Web орієнтованих СППР дуже широкий і включає комерційну, біржову, виробничу, наукову і культурну діяльність.

СППР орієнтовані на використання баз даних та сховищ даних

Цей клас інформаційних систем орієнтований на використання спеціалізованих баз даних. Найчастіше такі системи використовують реляційні бази даних і сховища даних. Такі бази даних побудовані на відношеннях - таблицях які пов'язані між собою. Правила побудови таблиць базуються на п'яти правилах нормалізації таблиць баз даних. Архітектуру бази даних формулює системний аналітик. Функціонування бази даних програмується із застосуванням мови SQL (*Structured query language*) — декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних і її модифікації, системи контролю за доступом до бази даних.

Починаючи з 1990, з'явився спеціальний клас систем баз даних, відомих як сховища даних. Сховище даних є велика колекція даних з декількох інтегрованих операційних систем, орієнтованих на конкретну предметну область, Зміст сховища постійно накопичується і ніколи не оновлюється чи переписується або видаляється. Сховище

даних може мати вигляд реляційної бази даних або багатовимірної бази даних. Технологія сховищ даних була спеціально задумана і розроблена, щоб підвищити продуктивність систем підтримки прийняття рішень.

Табличні СППР

СППР використовують бази даних. Бази даних формуються з таблиць. Табличні СППР дозволяють не тільки створювати, відображати коригувати таблиці але і виконувати інтелектуальну обробку даних які містяться у цих таблицях . Це надає більше можливостей користувачам, порівняно з текстовими СППР чи орієнтованими на використання баз даних СППР.

Таблиці баз даних можуть містити константи або формули для обчислення, обробку тексту, а також гіпертекстові посилання. Обчислення які містяться в клітинках можуть бути використані для генерації нових знань. Табличні СППР використовують для аналізу роботи підприємства, тенденцій розвитку ринку, прогнозування, статистичного аналізу даних, кластерного аналізу. Табличні СППР широко застосовуються на підприємствах.

Орієнтовані на моделі СППР

Такі СППР мають декілька інструментів пошуку рішень, які можуть бути застосовані для різних типів економіко математичних моделей . Розв'язання задачі формується як групування низки пошуків рішення які послідовно або паралельно розв'язують проблему сформульовану в моделі прийняття рішень. Користувач може коригувати, змінювати та видаляти моделі та інструменти пошуку в процесі розв'язання задачі . Отримані матеріали можуть бути збережені як шаблони для подальших досліджень чи пошуку рішення.

Системи підтримки прийняття рішень які орієнтовані на моделі використовуються для моделювання складних систем управління. У поточному часі такі моделі використовують системі як SAP і ORACAL.

СППР, які використовують штучний інтелект

Цей тип СППР розвивався у галузі штучного інтелекту який використовує методи і алгоритми які дозволяють комп'ютеру робити висновки і пропозиції на основі обчислень за спеціальними методами і програмами. За цією методикою комп'ютер може зробити висновок про правильність прийнятого рішення чи якогось висновку в конкретній ситуації . Комп'ютер може відстежувати і аналізувати причинно наслідкові взаємовідносини між факторами процесу прийняття рішень.

В систему закладаються декілька алгоритмів штучного інтелекту який дозволяє зробити систему гнучкою і здатною вирішувати складні проблеми управління .

Такі системи називаються експертними системами, тому що моделюють поведінку експерта при прийнятті рішення. Системи такого типу корисні в тому випадку коли немає експертів з конкретних проблем управління. Штучний інтелект доступний 24 години на добу, не вимагає високих гонорарів і не створює конфліктів. Тому такі системи дозволяють отримати якісні результати з меншими витратами .

Гібридні СППР

Гібридні СППР представляють собою комбінацію розглянутих вище систем. Більшість систем СППР , які використовуються в поточному часі є гібридними системами.

Для конкретних вимог і умов розробляється система СППР яка містить позитивні властивості різних систем і мінімізує негативні сторони систем, що комбінуються. Є два основні підходи до інтеграції СППР : гніздування та синергія. Методи гніздування передбачають передачу властивостей вкладної системи до системи яка прийняла цю підсистему. Наприклад, Word може прийняти чи використовувати деякі властивості таблиці Excel яка вкладена в документ Word.

У синергетичному підході до інтеграції різних систем, немає вкладеності, немає домінуючої технології, немає вкладених систем і методів. Всі методи інтегровані в єдиний інструмент, який дозволяє їх використовувати незалежно один від іншого, або разом декілька методів в межах однієї операції.

На рисунку 9 показана структура гібридної системи підтримки прийняття рішення.

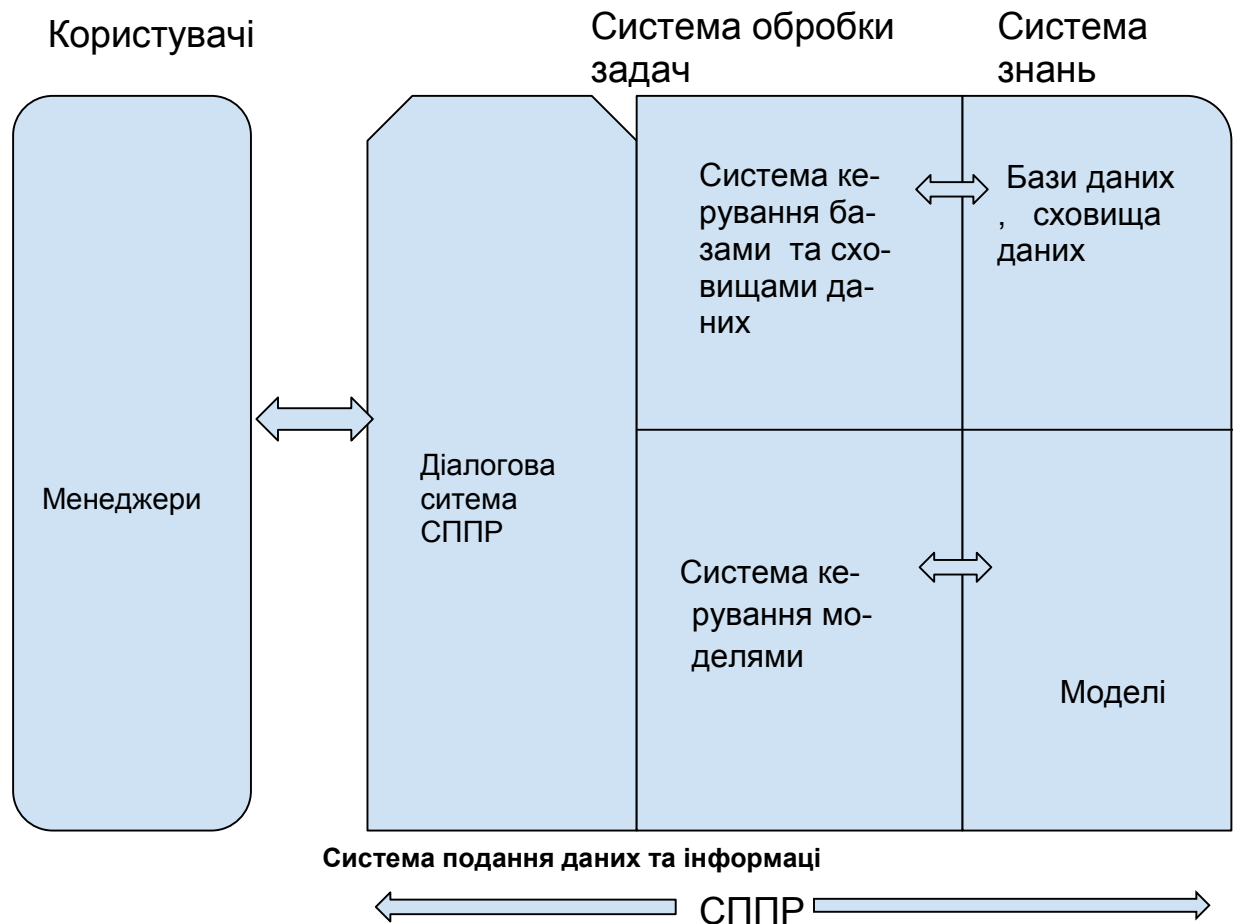


Рисунок 9 Структура гібридної системи СППР

Групові СППР

Групова система підтримки прийняття рішень — це інформаційна система, яка підтримує роботу групи людей, що мають спільне завдання та мету, і яка забезпечує інтерфейс до загальнодоступного (розподіленого) інформаційного середовища. Групові СППР використовують технології хмарного програмного забезпечення. Групові СППР системи керування моделями та БД, систему керування інтерфейсом користувача, систему керування поштою, а також низку моделей і функцій керування моделями, необхідних для задоволення потреб усіх учасників групи, і забезпечують доступ до інформації та агрегування її з різних джерел у множині форматів, що задовольняють різноманітні групові інформаційні потреби. Крім того, Групова СППР має бути зручною в користуванні для всіх учасників групи.

Висновки

В цій темі було розглянуто архітектуру СППР. Структура систем підтримки прийняття рішень включає в себе такі елементи: лінгвістична система; система подання даних та інформації; система обробки задач; системи обробки задач конкретних предметних областей.

Основними типами систем які використовуються є текстові СППР, гіпертекстові СППР, орієнтовані на використання баз даних та сховищ даних СППР, табличні СППР, орієнтовані на моделі СППР, СППР, які використовують штучний інтелект, гібридні СППР.

Контрольні запитання

1. Який вигляд має типова архітектура системи СППР?
2. Які інформаційні системи відносяться до текстових СППР?
3. Що таке Web орієнтовані СППР?
4. Роз'ясніть поняття табличних СППР.
5. Які СППР відносяться до систем орієнтованих на моделі?
6. Які СППР відносяться до інформаційних систем що використовують штучний інтелект?
7. Чому виникла необхідність використовувати гібридні СППР?
8. Дайте визначення групових СППР.

ТЕМА 3. Бази даних і сховища даних для підтримки прийняття рішень

Цінність і ефективність управлінських рішень залежить від якості отриманої інформації її точності і достовірності. Тому є дуже важливим правильна підготовка вихідних даних. Одним із ефективних методів підготовки даних є консолідація інформації.

Консолідація це комплекс методів і процедур призначених для отримання даних з різних джерел, забезпечення необхідного рівня їх інформативності та якості, перетворення до єдиного формату, в якому вони можуть бути завантажені в сховище даних або аналітичну систему. В основі консолідації лежить процес збору і організації зберігання даних у вигляді найбільш оптимальному з точки зору їх обробки на конкретній аналітичній платформі. Основними критеріями оптимальності з погляду консолідації даних є: забезпечення високої швидкості доступу до даних, компактність зберігання, автоматична підтримка цілісності структури даних, контроль несуперечності даних. В процесі консолідації даних вирішуються наступні завдання: вибір джерел даних, оцінку якості даних, зберігання, очищення, перенесення в сховище даних.

Консолідація інформації може трактуватися як бізнесова аналітика (англ. Business Intelligence) — процес перетворення даних в нові знання, які можуть бути використані для збільшення ефективності та конкурентоздатності підприємства.

Прикладна наукова галузь «консолідована інформація» — розвивається у трьох основних напрямках, а саме:

1. Консолідована інформація виробничої сфери, консолідована інформація не-виробничої сфери та консолідована інформація у системах спеціального призначення. Потенційними споживачами першого напрямку консолідованої інформації є бізнесові виробничі структури та керівники підприємств;
2. Консолідована інформація для державної влади різних рівнів;
3. Консолідована інформація спецслужб.

Схема консолідації даних показана на рис. 10.



Рисунок 10. Схема консолідації даних

Як видно з рис. 10 в основі консолідації лежить процес ETL (Extraction, Transformation, Loading) отримання, трансформація і завантаження. Цей процес вирішує завдання отримання даних з різнотипних джерел і їх перетворення до вигляду придатного до зберігання, а також завантаження у базу даних.

Накопичення консолідованої інформації виконується у спеціальних базах даних або сховищах даних. Розглянемо організацію баз даних і сховищ даних .

В системах підтримки прийняття рішень широко застосовуються сховища даних які відрізняються від існуючих баз даних.

Сховища даних є предметно орієнтованими, інтегрованими, такими, що змінюються у часі і накопичують дані але їх не видаляють. На відміну від реляційних баз даних у сховищах даних, у яких інформація тільки накопичується, але не видаляється і не коригується , нормалізація таблиць не є необхідною. В організаційному відношенні, сховища даних є частиною процесу пошуку, зберігання і представлення даних для цілей підтримки прийняття рішень в управлінні. На рис. 11 показано місце сховища даних всередині ланцюгу обробки інформації системі прийняття рішень.

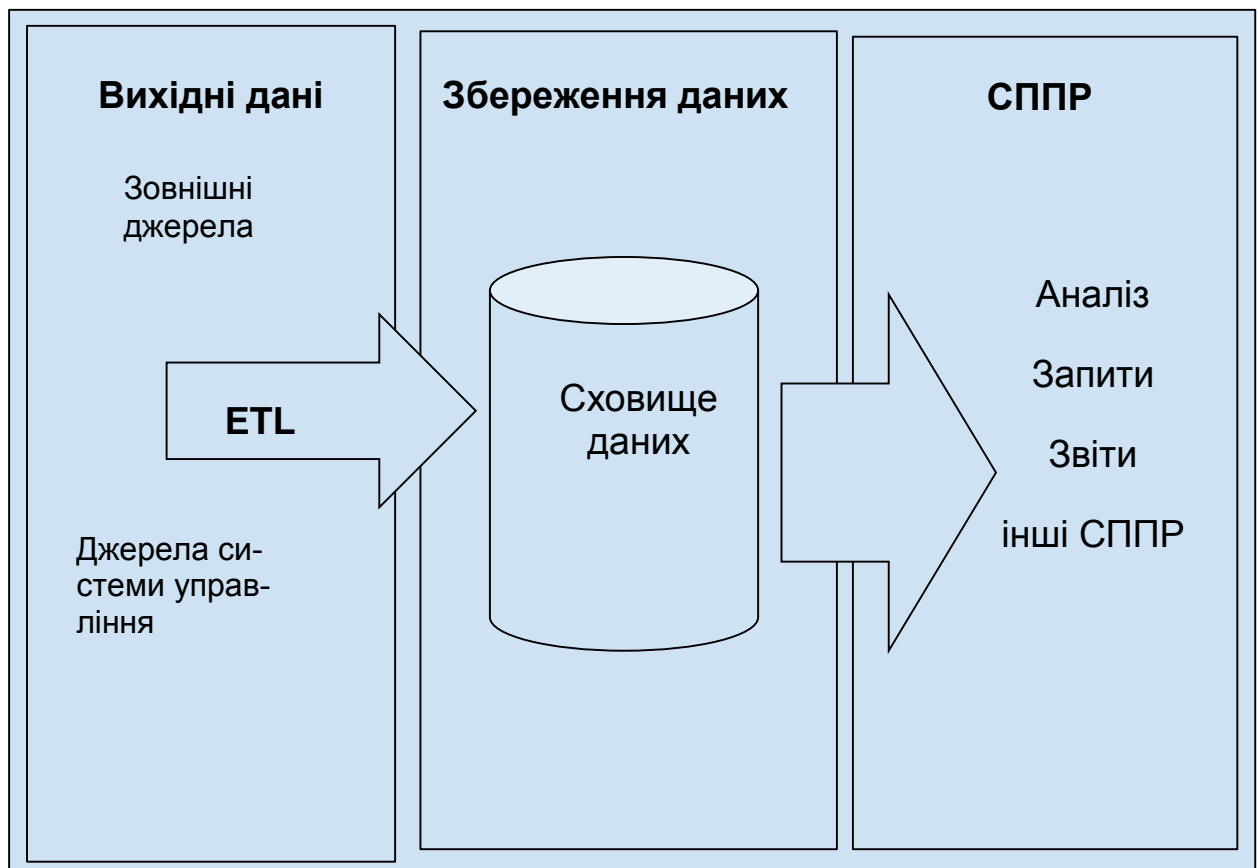


Рисунок 11. Загальна модель зберігання даних

Для того, щоб сховище даних виконувало належні йому функцію - підтримки процесу аналізу даних та прийняття рішення, воно повинне відповідати вимогам, які сформульовані одним з авторів концепції сховищ даних - Ральфом Кімболлом:

- висока швидкість отримання даних зі сховища;
- автоматична підтримка внутрішньої несуперечності даних;
- можливість отримання і порівняння зрізів даних;
- наявність зручних засобів для перегляду даних в сховищі;
- забезпечення цілісності і достовірності даних.

Для того, щоб задовольнити всі перелічені вимоги, для побудови і роботи сховища даних, як правило, використовується не одну програму, а систему, в яку входить декілька програмних продуктів. Одні з них є системою зберігання даних, інші - засоби їх перегляду, отримання, трансформації, завантаження, тощо.

Архітектура сховищ даних.

Розробка і побудова сховищ даних - це досить недешеве і трудомістке завдання. Успішність впровадження сховища даних багато в чому залежить від рівня інформатизації бізнес-процесів, інформаційних потоків на підприємстві, об'єму і структури даних, вимог до швидкості виконання запитів і періодичності оновлення сховища, характеру аналітичних завдань тощо. З метою наблизити сховище даних до умов і специфіки конкретної організації в даний час розроблено декілька архітектур сховищ даних: багатовимірні, реляційні, гібридні і віртуальні.

Основним завданням сховища даних є створення джерела якісних даних які доступні своєчасно у відповідному форматі для всіх користувачів для звітності та аналітичних цілей. Така проста конструкція сховища даних називається "Сховище даних підприємства" вона служить для інтегрованого накопичення організаційних даних із різних джерел інформації які записані у такому форматі якій підходить для моделей даних, а також для обробки персоналом. Інформація завантажується в сховище даних через процес ETL. Схема такого сховища даних та його архітектура показані на рис. 12 .

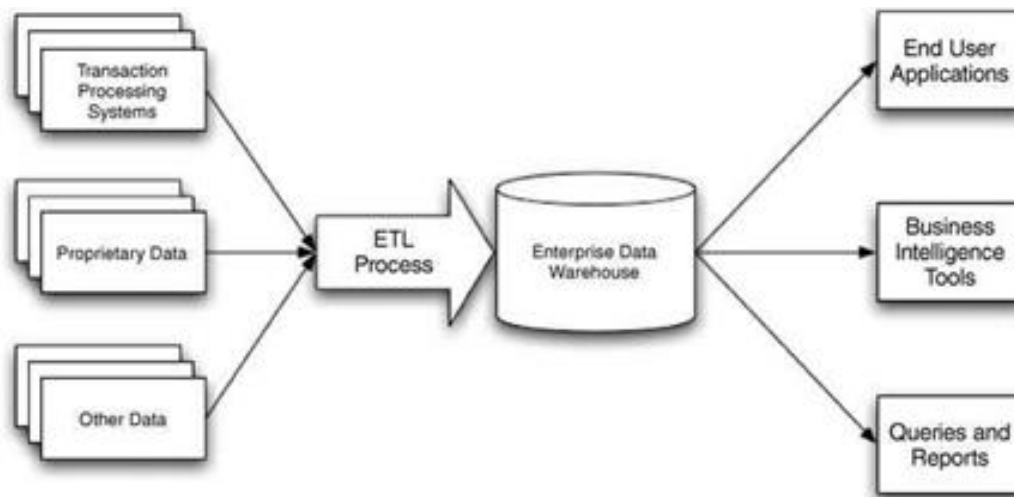


Рисунок 12. Схема сховища даних підприємства

Така організація сховища даних на підприємстві має недоліки оскільки вона складна в експлуатації тому, що дані отримуються із різних системних платформ і конфліктуючих форматів, які важко узгодити між собою. Тому така система може бути дорогою в експлуатації.

Альтернативою таким дорогим системам є створення вітрин даних на підприємствах. Архітектура і особливості вітрин даних буде розглянута нижче.

Моделі представлення даних у сховищах даних (БСД).

Багатовимірне сховище даних

Багатовимірна модель даних, яка лежить в основі побудови багатовимірних сховищ даних, оснований на концепції багатовимірних кубів або гіперкубів. Вони являють собою впорядковані багатовимірні масиви, які також часто називають OLAP-кубами (OLAP означає On-line Analytical Processing - оперативна аналітична обробка). Сховища даних побудованих на основі багатовимірності називають MOLAP (Multidimensional OLAP). Технологія OLAP - це методика оперативного отримання потрібної інформації з великих масивів даних і формування відповідних звітів.

Суть багатовимірного представлення даних полягає в наступному. Більшість реальних бізнес-процесів описуються великим набором показників, властивостей, атрибутів. Наприклад, для опису процесу продажу можуть знадобитися такі відомості, як найменування товарів або їх груп, інформація про постачальника і покупця, місто, де проводилися продажі, про ціни, кількості і суми проданих товарів. Окрім цього, для відстеження процесу в часі потрібно ввести такий атрибут, як дата та час. Якщо зібрати всю цю інформацію в таблицю, то вона виявиться складною для візуалізації та аналізу. Тому зберігання і обробка такої багатовимірної інформації в одній двовимірній таблиці є не ефективною.

Для пояснення суті проблеми проведемо геометричну аналогію. Уявіть собі 3-х вимірну фігуру - паралелепіпед і спроектуйте його на площину, а потім по отриманій плоскій проекції спробуйте оцінити форму і розміри початкової фігури. Це буде важко зробити, оскільки, по-перше, втрачена інформація про один вимір, а по-друге, фігура тепер представлена в абсолютно невластивому їй «плоскому» вигляді.

Приблизно те ж: саме можна сказати про інформацію, представлену декількома рядами даних. Кожен такий ряд (поле таблиці) можна розглядати як свого роду інформаційне вимірювання і тоді «плоска» таблиця може бути інтерпретована як результат «розплющення» багатовимірної інформаційної структури в абсолютно не властиву їй плоску форму. Щоб компенсувати втрату інформації від виключення одного або декількох вимірів, ускладнити структуру таблиці, що в більшості випадків приводить до того, що розібратися в ній стає дуже складно. Можна піти іншим шляхом, виконавши декомпозицію інформації в декілька простіших таблиць і зв'язати їх деяким набором зв'язків, перейшовши до реляційної моделі. Проте доведено, що реляційна модель не цілком оптимальна з точки зору аналізу даних, оскільки припускає високий ступінь нормалізації. Це знижує швидкість виконання запитів. Тому природним кроком була розробка багатовимірної моделі представлення даних, яка реалізується за допомогою багатовимірних кубів.

В основі багатовимірного представлення даних лежить їх розділення на дві групи - виміри і факти.

Виміри - це атрибути, найменування і властивості об'єктів, що беруть участь в бізнес-процесі. Вимірами можуть бути найменування товарів, назви фірм-постачальників і покупців, ПІБ людей, назви міст тощо. Виміри можуть бути і числовими, якщо відповідній категорії (наприклад, найменуванню товару) поставлений у відповідність числовий код, але у будь-якому випадку це дискретні дані, тобто такі, що приймають значення з обмеженого набору. Виміри якісно описують досліджуваний бізнес-процес. Факти - це дані, що кількісно характеризують бізнес-процес і можуть приймати нескінченну кількість значень. Приклади фактів - ціна товару або виробу, їх кількість, сума продажу або покупок, зарплата співробітників, сума кредиту, страхова винагорода тощо.

Багатовимірний куб можна розглядати як систему координат, осями якої є виміри - дата, товар, покупець. По осях відкладатимуться значення вимірів - дати, найменування товарів, назви фірм-покупців, ПІБ фізичних осіб. В такій системі кожному набору значень вимірів (наприклад, дата-товар-покупець) відповідатиме комірка, в якій можна розмістити числові показники (тобто факти), пов'язані з даним набором. Таким чином, між об'єктами бізнес-процесу та їх числовими характеристиками буде встановлений однозначний зв'язок.

Таким чином, інформація в багатовимірному сховищі даних є логічно цілісною. Це вже не просто набори строкових і числових значень, які у випадку з реляційною моделлю потрібно отримувати з різних таблиць, а цілісні структури типу «кому, що і скільки було продано на певний момент часу». Принцип організації багатовимірного кубу зображено на рис. 13. Переваги багатовимірного підходу є наступними:

- представлення даних у вигляді багатовимірних кубів є більш наочним, ніж сукупність нормалізованих таблиць реляційної моделі, структуру якої представляє тільки адміністратор БД;
- ширші можливості побудови аналітичних запитів до системи, що використовує БСД;
- використання багатовимірної моделі в деяких випадках дозволяє різко зменшити час пошуку в БСД, забезпечуючи виконання аналітичних запитів практично в режимі реального часу. Це пов'язано з тим, що агреговані дані обчислюються заздалегідь і зберігаються в багатовимірних кубах разом з деталізованими. Тому витратити час на обчислення агрегатів при виконанні запиту вже не потрібно;
- формування навіть найскладніших аналітичних запитів до БСД здійснюється без використання SQL з боку користувача.

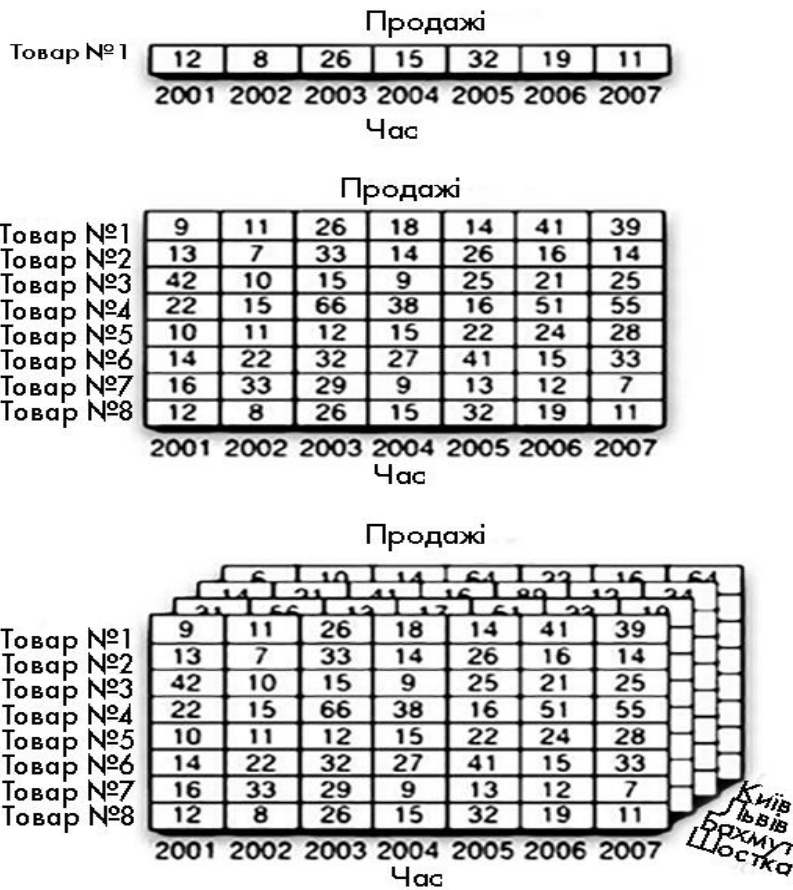


Рисунок 13. Принцип організації даних у багатовимірному кубі.

Використання багатовимірної моделі даних пов'язане і з певними труднощами. Так, для її реалізації потрібний великий об'єм пам'яті. Це пов'язано з тим, що при реалізації фізичної багатовимірності використовується значна кількість технічної інфор-

мації. Крім того, багатовимірна структура важче піддається модифікації; при необхідності вбудувати ще одне вимірювання, потрібно виконати фізичну перебудову всього багатовимірного куба.

На підставі цього можна зробити висновок, що використання систем збереження, в основі яких лежить багатовимірне представлення даних, доцільно в тих випадках, коли об'єм даних порівняно невеликий, а сама багатовимірна модель має стабільний набір вимірів.

В принципі, багатовимірний куб може бути реалізований і за допомогою звичайної реляційної моделі. В цьому випадку має місце емуляція багатовимірного уявлення сукупністю плоских таблиць. Такі системи отримали назву ROLAP - Relational OLAP.

Реляційні сховища даних (РСД).

На початку 70-х років минулого століття американським вченим Богаром Коддом була розроблена реляційна модель організації даних, яка дала початок новому етапу розвитку СУБД. Завдяки простоті та гнучкості реляційна модель стала домінуючою, а реляційні СУБД стали «де факто» промисловим стандартом.

Реляційна база даних (relational data base) - це сукупність відносин, що містять всю інформацію, яка зберігається в базі. Фізично це виражається в тому, що інформація зберігається у вигляді двовимірних таблиць, зв'язаних за допомогою ключових полів.

Застосування реляційної моделі при створенні сховища даних часто дозволяє отримати ряд переваг, особливо щодо ефективності роботи з великими масивами даних. На основі РСД будуються ROLAP-системи (Relational OLAP). В основі технології РСД лежить принцип, відповідно до якого виміри зберігаються в плоских таблицях так само, як і в звичайних реляційних СУБД, а факти (дані, що агрегуються) в окремих спеціальних таблицях цієї ж бази даних. При цьому таблиця фактів є «основою» для пов'язаних з нею таблиць вимірів. Вона містить кількісні характеристики об'єктів і подій, сукупність яких будуть аналізуватися в майбутньому.

На логічному рівні розрізняють дві схеми організації РСД - «зірка» та «сніжинка».

Схеми «зірка». Дана модель даних (рис. 14) складається з двох типів таблиць: однієї таблиці фактів (fact table) - центр «зірки» - та декількох таблиць вимірів (dimension table) - промені «зірки». Таблиця фактів зазвичай містить одну або декілька полів числового типу, що дають числову характеристику певному аспекту предметної області (наприклад, об'єм продаж для торгової компанії або сума платежів для банку), і декілька цілочисельних полів-ключів для доступу до таблиць вимірів. Таблиці вимірів розшифровують ключі, на які посилається таблиця фактів.

Проте розміщення всієї інформації про виміри в одній таблиці не завжди є виправданим. Наприклад, якщо товари, об'єднані в групи, тобто має місце ієрархія, то для кожного товару доведеться показувати, до якої групи він відноситься, що приведе до багаторазового повторення назв груп. Це не тільки викликає надмірність таблиці, але і підвищує вірогідність виникнення суперечностей (якщо, наприклад, один і той же товар помилково віднесуть до різних груп). Тому, для ефективної роботи з ієрархічними

вимірами була розроблена модифікація схеми «зірка», яка отримала назву «сніжинка».

Головною відмінністю схеми «сніжинка» є те, що інформацію про один вимір можна зберігати в декількох пов'язаних таблицях. Тобто, якщо хоч одна з таблиць вимірів має пов'язаних з нею одну чи більше таблиць вимірів, то в цьому випадку матиме місце схема «сніжинка» (рис. 15).

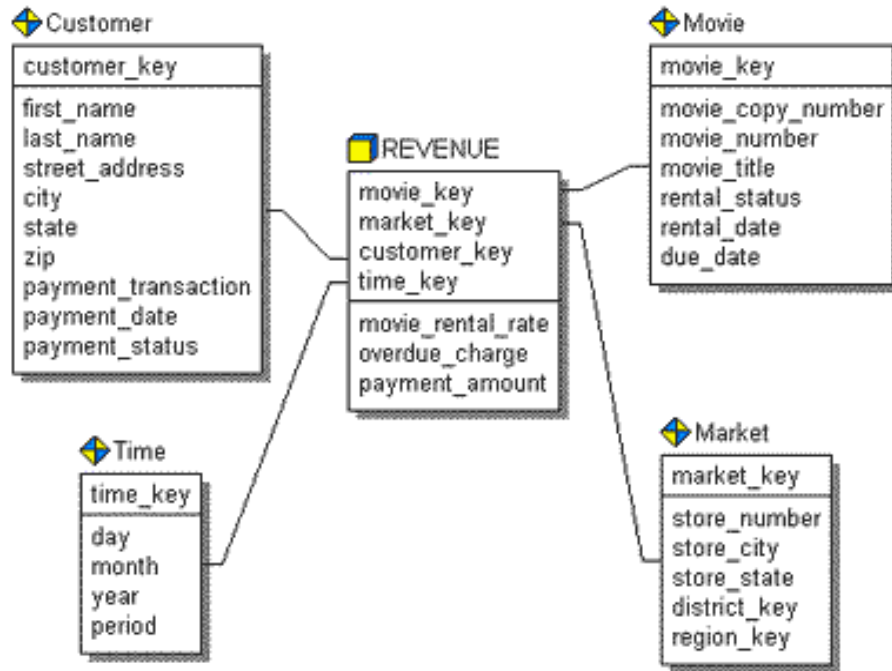


Рисунок. 14. Схема «зірка»

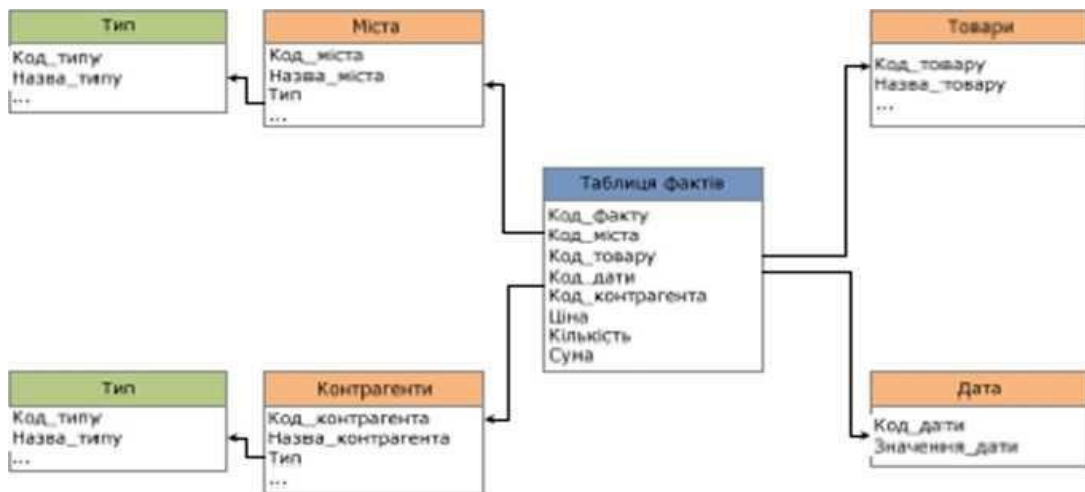


Рис. 15. Схема «сніжинка»

Очевидно, що РСД, побудоване за схемою «сніжинка», МАЄ вищий рівень нормалізації, в порівнянні із «зіркою».

Вибір схеми для побудови РСД залежить від особливостей механізмів збору та обробки даних. Очевидно, що кожна зі схем має свої переваги і недоліки (див. табл.

5), які, проте, можуть проявлятися більшою чи меншою мірою, залежно від особливостей функціонування сховище даних в цілому.

Переваги та недоліки схем «зірка» та «сніжинка»

Таблиця 5.

	Переваги	Недоліки
Схема «Зірка»	<ul style="list-style-type: none"> • Простота і логічна прозорість моделі; • Проста процедура поповнення даних у вимірі, оскільки працюєш лише з однією таблицею. 	<ul style="list-style-type: none"> • Повільна обробка виміру, оскільки одне і те ж значення може зустрічатися декілька раз; • Висока ймовірність виникнення невідповідності даних.
Схема «Сніжинка»	<ul style="list-style-type: none"> • «Сніжинка» більш близька до представлення даних в багатовимірній структурі; • Більш ефективне та просте завантаження даних з РСД в багатовимірні структури, оскільки завантаження відбувається в окремі таблиці; • Низька ймовірність появи помилок невідповідності даних. 	<ul style="list-style-type: none"> • Порівняно складна для реалізації та розуміння структура даних; • ускладнена процедура додавання даних у вимір.

Що ж стосується переваг та недоліків РСД в цілому, то варто відмітити наступне.

До переваг РСД відноситься:

1. Практично необмежений об'єм даних, що зберігаються;
2. Оскільки реляційні СУБД лежать в основі багатьох облікових систем, які зазвичай і є основними джерелами даних для сховище даних, то використання реляційної моделі дозволяє спростити процедуру завантаження та інтеграції даних в сховищі;
3. При необхідності додавання нових вимірів даних немає необхідності виконувати складну фізичну реорганізацію сховища на відмінну, наприклад, від багатовимірних сховище даних;
4. Забезпечується високий рівень захисту даних і хороші можливості обмежування прав доступу.

Головним недоліком РСД є можливість розмноження таблиць вимірів, в разі використання високого рівня ієрархії даних. Крім того, труднощі можуть викликати нетривіальні аналітичні запити, оскільки в цьому випадку доведеться створювати складні SQL-запити.

Таким чином, вибір реляційної моделі при побудові сховище даних доцільний в наступних випадках:

1. Коли обсяг даних для зберігання є дуже великим (тобто коли багатовимірні сховище даних стають неефективними);
2. Якщо ієрархія вимірів не є складною;

3. Коли потрібна часта зміна розмірності даних. В разі використання реляційної моделі можна вийти з положення шляхом додавання нових таблиць, а для багатовимірної моделі доведеться виконувати складну перебудову фізичної структури сховища.

Гібридні сховища даних.

Як MOLAP, так і ROLAP моделі сховищ даних мають свої переваги та недоліки. Наприклад, багатовимірна модель забезпечує високу швидкість реакції на запит, але не дозволяє ефективно управляти такими великими об'ємами даних як реляційна модель. Тоді логічно було б використовувати модель даних, яка була б комбінацією реляційної і багатовимірної моделі і дозволяла б поєднувати високу продуктивність, характерну для багатовимірної моделі, і можливість зберігати великі масиви даних, властиву реляційній моделі. Така модель, що поєднує в собі принципи реляційної і багатовимірної моделей, отримала назву гібридною або HOLAP - Hybrid OLAP. Сховища даних, побудовані на основі HOLAP, називаються гібридними сховищами даних.

Головним принципом побудови гібридного сховища даних є те, що деталізовані дані зберігаються в реляційній структурі (ROLAP), що дозволяє зберігати великі об'єми даних, а агреговані - в багатовимірній (MOLAP), що дозволяє збільшити швидкість виконання запитів (оскільки при виконанні аналітичних запитів вже не потрібно обчислювати агрегати).

Для кращого розуміння сказаного, наведемо наступний приклад. Нехай облікова система, встановлена в супермаркеті, обслуговуючому щодня тисячі покупців. При цьому максимальному рівню деталізації даних відповідає покупка по одному чеку, в якому вказується загальна сума покупки, найменування або коди придбаних товарів і сума по кожному товару. Оперативна інформація, що складається з деталізованих даних, консолідується в реляційній структурі сховища даних. Інтерес з погляду аналізу представляють узагальнені (агреговані) дані, наприклад, по групах товарів, відділах або деяких періодах. Тому початкові деталізовані дані агрегуються і обчислені агрегати зберігаються в багатовимірній структурі гібридного сховища даних.

Якщо дані облікової системи мають великий об'єм (декілька тисяч транзакцій в день і більше) і високий ступінь деталізації, а використовуються для аналізу в основному узагальнені дані, то гібридна архітектура сховища виявляється найбільш підходящою. Однак зрозуміло, що крім позитивних рис, гібридна модель має і недоліки (див. табл.6.)

Переваги та недоліки гібридної моделі сховище даних

Таблиця 6

Переваги	Недоліки
Реляційна структура формує стійкі і несуперечливі таблиці і зв'язки між ними; Оскільки реляційне сховище підтримує актуальність і коректність даних, воно забезпечує надійний транспортний рівень для передачі інформації в багатовимірне сховище.	складне адміністрування, що зумовлено більш складною процедурою поповнення даними такого сховища. Потрібно узгоджувати зміни в реляційній та багатовимірній частині сховища.

Більшість програмних рішень для створення сховищ даних використовують саме гібридну модель. Так, наприклад, корпорація Oracle в своїх розробках використовує ROLAP для збереження даних в сховищі, вітрини даних побудовані на базі MOLAP.

Віртуальні сховища даних.

Невід'ємною проблемою використання сховищ даних, яка рано чи пізно настає, є надмірність. Надмірність призводить до зниження ефективності використання дискового простору і оперативної пам'яті ПК, а іноді і при дуже великих об'ємах інформації - до зниження продуктивності, зростання часу реакції на запит і навіть повної непрацездатності системи. В тій чи іншій мірі це характерно як для реляційних, так і для багатовимірних сховищ. Ситуація ускладнюється ще і тим, що сховище даних зберігають історичну інформацію та реалізує принцип незмінності даних. Це означає, що на відміну від звичайних систем оперативної обробки (OLTP-систем), де зберігаються тільки актуальні дані, а дані, що втратили актуальність, знищуються, сховище даних можуть тільки поповнюватися новими даними, а видалення історичних даних не проводиться. Крім цього, часто потрібно зберігати великі об'єми агрегованих даних. В сукупності всі ці чинники часто приводять до «вибухового» зростання об'ємів сховище даних.

Подолати проблему надмірності і навіть звести її до нуля можна шляхом використання віртуальних сховищ даних (ВСД). В основі концепції ВСД лежить принцип, відповідно до якого дані з локальних джерел, баз даних і облікових систем не консолідується в єдине сховище даних фізично, а витягуються, трансформуються та інтегруються безпосередньо при виконанні запиту в оперативній пам'яті комп'ютера.

Віртуальним сховищем даних називається система, яка, працюючи з розрізненими джерелами даних, емулює роботу звичайного сховища даних, витягуючи, трансформуючи і інтегруючи дані безпосередньо в процесі виконання запиту. Часто віртуальні сховища даних називають динамічними (ДСД) або DOLAP, оскільки формування багатовимірного куба відбувається динамічно на комп'ютері користувача.

При роботі з ВСД користувач фактично має справу з «ілюзією» сховища даних. Віртуальність означає, що ВСД існує тільки до тих пір, поки працює сама програма, якою сховище керується. Як тільки вона закривається, віртуальне сховище припиняє своє існування.

Переваги та недоліки віртуальної моделі сховища даних

Таблиця 7.

Переваги	Недоліки
<ul style="list-style-type: none"> • Мінімізується об'єм дискової і оперативної пам'яті. Це пов'язано з відсутністю необхідності зберігання історичних даних і численних агрегованих даних; • Наявність семантичного слою дозволяє аналітику повністю абстрагуватися від проблем, пов'язаних з процесом витягання даних з різноманітних джерел; • З'являється можливість аналізу даних в OLTP- системі відразу після їх надходження без очікування їх завантаження в сховище. 	<ul style="list-style-type: none"> • Джерела даних, інформація з яких потрапляє у ВСД, можуть бути недоступні, якщо доступ до них здійснюється по мережі. Тимчасова недоступність хоч одного з джерел може привести до неможливості виконання запиту або до спотворення інформації; • Через можливість неузгодженості моментів поповнення джерел даних і відсутності підтримки в них хронології на один і той же запит в різні моменти часу можуть бути отримані різні дані; • Неможлива робота з історичними даними, оскільки у ВОД доступні тільки ті дані, які знаходяться в джерелах в даний момент часу; • Оскільки деякі типи джерел даних не оптимізовані за швидкістю доступу до них, то одержання даних від них займає певний час, що призводить до зниження швидкості виконання запитів віртуальними сховищами.

Таким чином, застосування ВСД є доцільним для підприємств, які не мають технічних засобів і кваліфікованого персоналу для підтримки фізичних сховищ даних. Особливо великі переваги від ВСД можна отримати при необхідності аналізувати «найсвіжішу» інформацію (таб. 7). У ВСД відсутній етап завантаження даних, тому часовий інтервал між появою інформації в OLTP-системі та її готовністю для аналізу даних мінімальний. При цьому слід враховувати, що оскільки ВСД підтримує історичну інформацію тільки за період актуальності OLTP-систем, його застосування виправдане тільки тоді, коли історичні дані для аналізу не потрібні.

Вітрини даних

Важливим поняттям сховищ даних, є вітрини даних (ВД) - data marts. Ситуація, коли для аналізу необхідна вся інформація, що міститься в сховищі даних, виникає рідко. В більшості випадків підрозділи підприємства або організації використовують профільну інформацію, що стосується тільки того напрямку діяльності, який вони обслуговують. Як правило, об'єм такої тематичної інформації невеликий в порівнянні із загальним об'ємом сховища і цілком ефективно може обслуговуватися MOLAP-системою. Якщо виділити профільні дані, що найчастіше використовуються по певному напрямку діяльності, в окремий набір і організувати його зберігання в окремій багатовимірній БДг підключеній де-централізованого сховища даних, то це і буде вітрина даних.

Вітрина даних - це спеціалізоване, локальне, тематичне сховище, підключене до централізованого сховища даних для обслуговування окремого підрозділу організації або певного напрямку його діяльності.

Найчастіше для побудови вітрин даних використовується багатовимірна модель, оскільки для невеликих об'ємів даних вона забезпечує швидшу реакцію на запити, ніж: реляційна, хоча в деяких випадках використовується і реляційна модель.

На рис. 16. зображено схему консолідації з використанням вітрин даних.

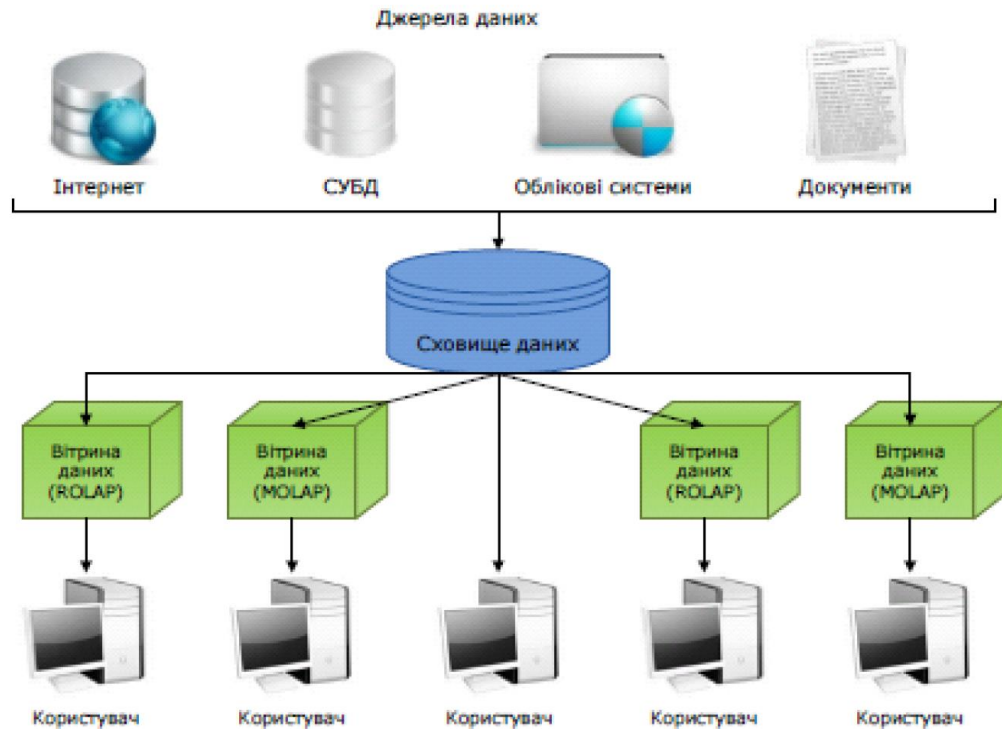


Рисунок 16. Схема консолідації із використанням вітрин даних

Використання вітрин даних має наступні переваги:

- близькість до кінцевого користувача;
- дані тематично орієнтовані на конкретного користувача;
- відносно невеликий об'єм збережених даних, що не вимагає значних витрат на їх організацію і підтримку;
- хороші можливості по розмежуванню прав доступу користувачів, оскільки кожен користувач працює тільки зі своєю вітриною і має доступ тільки до інформації, що відноситься до певного напрямку діяльності.

Використання вітрин даних є найефективнішим у великих організаціях з великою кількістю підрозділів, кожен з яких вирішує власні аналітичні завдання. В цьому випадку вітрини даних можуть використовуватися як самостійно, так і разом з централізованим сховищем даних.

В поточному часі існує три типи організацій вітрин даних:

1. Незалежні вітрини даних;
2. Залежні вітрини даних;
3. Вітрини даних з федеративною системою організації.

Незалежні вітрини даних створюються для користувачів які мають приблизно однакові вимоги до сховища даних. Структура такої системи показана на рис. 17, Така система містить декілька незалежних вітрин які подають дані необхідні конкретному користувачу. Кожна вітрина має свою систему збору, представлення і обробки

даних та свою відповідальність за якість представленої інформації, яка є незалежною від інших вітрин даних .

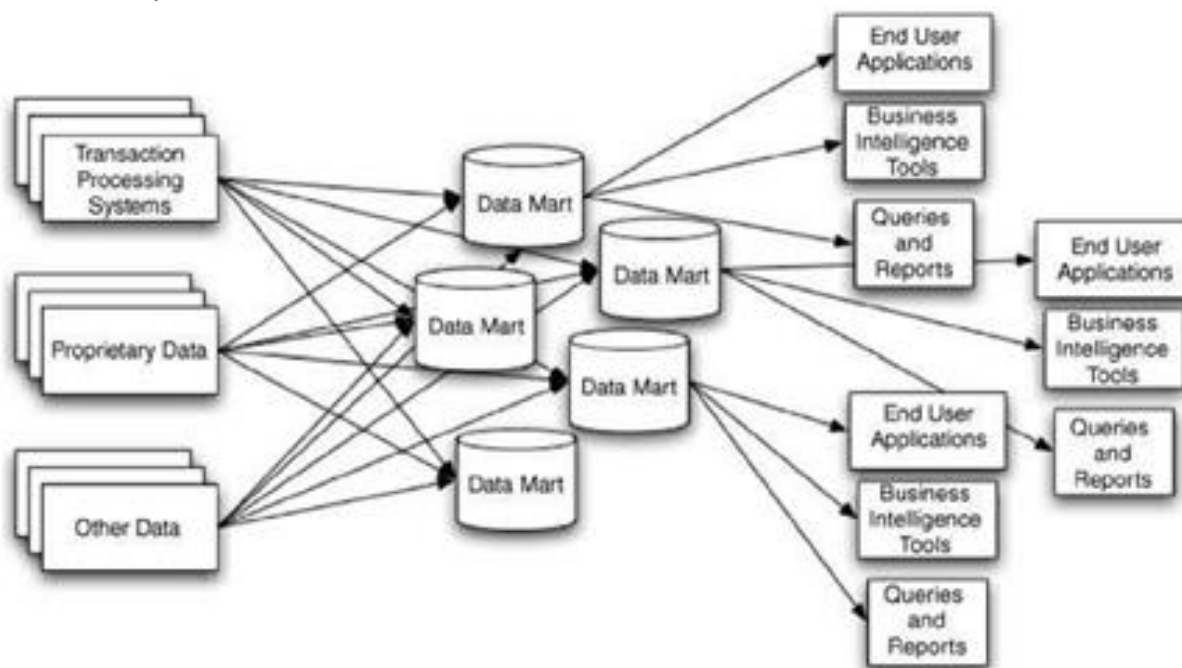


Рисунок 17 Незалежні вітрини даних

Недоліком такої організації є надлишковість в системі обробки вихідної інформації . Вітрини незалежно одна від одної обробляють інформацію і повторюють обробку яку зробили інші вітрини даних.

В залежних вітринах даних на підприємстві розробляється спільне сховище даних підприємства в яке інформація поступає з єдиної системи обробки даних ETL. Окремі вітрини даних отримують інформацію із спільного сховища даних і подають результати користувачам. Така система організації вітрин забезпечує узгодженість і між окремими вітринами даних, що неможливо при незалежній організації. Схему залежних вітрин даних показано на рис. 18

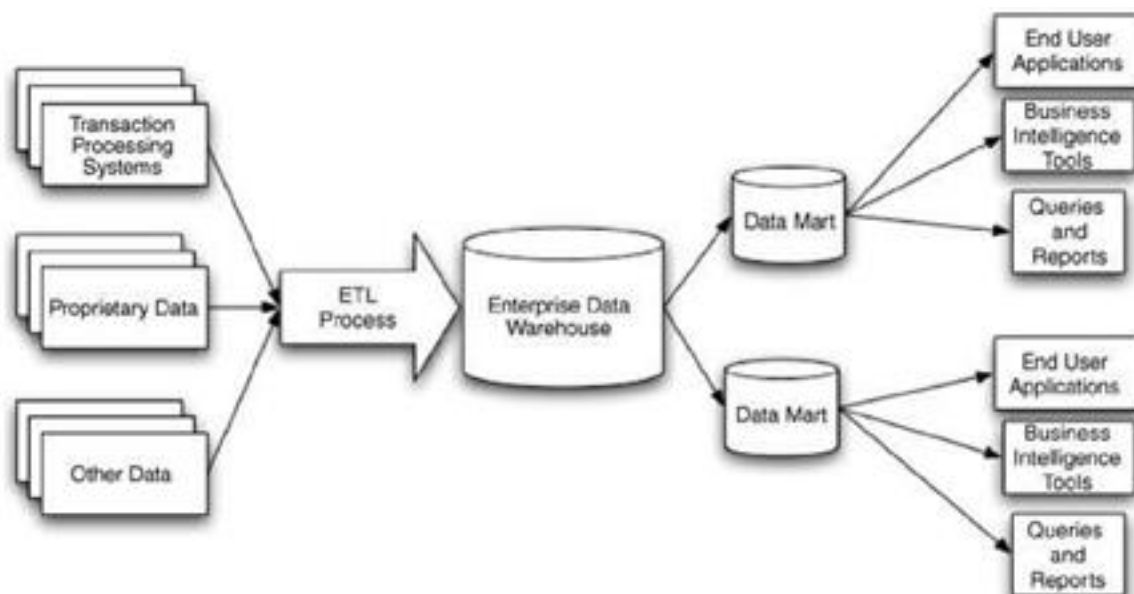


Рисунок 18. Залежні вітрини даних

Недоліком такої системи є те, що необхідно тримати на підприємстві власне сховище даних на фізичних носіях інформації.

Іншим підходом для створення сховища даних, який зберігає залежні вітрини без додаткових витрат на власні сховища даних на підприємстві є вітрини з федеративною системою організації. В такому підході організації вихідні дані обробляються і зберігаються централізованою системою ETL, але не передаються в сховища даних підприємства. Замість цього дані передаються системою безпосередньо на вітрини даних. Для передачі даних з вітрини створюються тимчасове зберігання даних яке можна представити як віртуальне сховище даних. В федеративній системі організації передача інформації забезпечується за допомогою схеми мета-даних і єдиних стандартів передачі та збереження даних для всіх вітрин. Така система забезпечує організацію сховища даних з невеликими витратами на створення і підтримку вітрин. Схему вітрин із федеративною системою організації показано на рис. 19

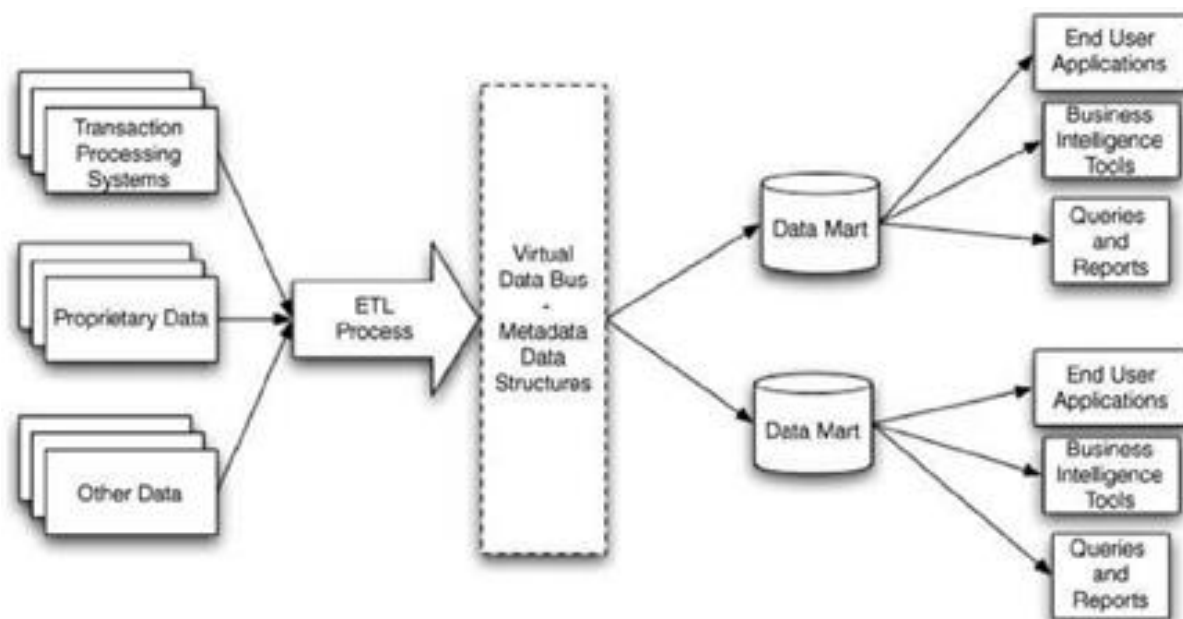


Рисунок 19 Федеративна система організації сховища даних

Поточний стан і майбутній розвиток сховищ даних

OLAP-сервери, або сервери багатовимірних БД, зберігають багатовимірні дані по-різному. В сховищі даних - і в звичайному, і в багатовимірному - поряд з детальними даними, додатково, зберігаються сумарні, узагальнені показники (агреговані показники, агрегати), такі, як сумарні обсяги продажів по місяцях, за категоріями товарів, тощо. Агрегати зберігаються в явному вигляді з метою прискорити виконання запитів. Тому при завантаженні даних в багатовимірну БД обчислюються і зберігаються всі сумарні показники або їх частини. Користувачі отримують в такому випадку швидше результати, тому що вони тільки зчитуються з бази даних, а не збираються із різних джерел і не обчислюються.

Розглянемо сховища даних, які розроблено різними виробниками.

Компанія IBM

Рішення компанії IBM називається A Data Warehouse Plus. Метою компанії є забезпечення інтегрованого набору програмних продуктів і сервісів, заснованих на єдиній архітектурі. Основою сховищ даних є сімейство СКБД DB2. Пропонуються три рішення для сховищ даних:

- *ізольована вітрина даних* - призначена для вирішення окремих завдань поза зв'язком із загальним сховищем корпорації;
- *залежна вітрина даних* - аналогічна ізольованій вітрині даних, але джерела даних знаходяться під централізованим контролем;
- *глобальне сховище даних* - корпоративне сховище даних, яке повністю централізовано контролюється і керується. Глобальне сховище даних може

зберігатися централізовано або складатися з декількох розподілених в мережі ринків даних.

Oracle

Рішення компанії [Oracle](#) в області сховищ даних ґрунтується на двох чинниках: широкий асортимент продуктів самої компанії і діяльність партнерів в рамках програми Warehouse Technology Initiative. Можливості Oracle в області сховищ даних базуються на таких складових:

- наявність реляційної СКБД Oracle 11, яка постійно вдосконалюється для кращого задоволення потреб сховищ даних;
- існування набору готових додатків, які забезпечують можливості розробки сховища даних;
- високий технологічний потенціал компанії в галузі аналізу даних;
- доступність низки продуктів, вироблених іншими компаніями.

Hewlett Packard

Роботи, пов'язані з OLAP, виконуються в рамках програми OpenWarehouse. Виконання цієї програми повинно забезпечити можливість побудови сховищ даних на основі потужних комп'ютерів HP, апаратури інших виробників і програмних компонентів. Основою підходу HP є Unix-платформи і програмний продукт Intelligent Warehouse, який призначений для керування сховищами даних. Основа побудови сховищ даних, пропонована HP, залишає свободу вибору реляційної СКБД, засобів реінжинірингу тощо.

NCR

Рішення компанії NCR спрямоване на вирішення проблем корпорацій, у яких однаково сильні потреби і в системах підтримки прийняття рішень, і в системах оперативної аналітичної обробки даних. Пропонована архітектура називається Enterprise Information Factory і ґрунтується на досвіді використання системи керування базами даних Teradata і пов'язаних з нею методах паралельної обробки.

Informix Software

Стратегія компанії по відношенню до сховищ даних спрямована на розширення ринку для її продукту On-Line Dinamic Parallel Server. Запропонована архітектура сховища даних базується на чотирьох технологіях: реляційній бази даних, програмному забезпеченні для керування сховищем даних, засобах доступу до даних і платформі відкритих систем. Три останні компонента розробляються партнерами компанії. Після

виходу Універсального Сервера, розробленого компанією на об'єктно-реляційному підході, і він буде використовуватися для побудови OLAP.

SAS Institute

Компанія є постачальником повного рішення для організації сховища даних.

Підхід заснований на наступному:

- забезпечення доступу до даних з можливістю їх витягання з найрізноманітніших сховищ даних (реляційних і нереляційних);
- перетворення даних і маніпулювання ними з використанням [4GL](#) (fourth-generation programming language);
- наявність сервера багатовимірних баз даних;
- великий набір методів і засобів для аналітичної обробки та статистичного аналізу.

Sybase

Стратегія компанії в області сховищ даних ґрунтується на розробленій їй архітектурі Warehouse WORKS. В основі підходу перебуває реляційна СКБД Sybase Adaptive Server Enterprise 15.5, засіб для підключення та доступу до баз даних OmniCONNECT і засіб розробки додатків PowerBuilder. Компанія продовжує удосконалювати свою СКБД для кращого задоволення потреб сховищ даних (наприклад, введено побітну індексацію).

Software AG

Діяльність компанії в області сховищ даних відбувається в рамках програми Open Data Warehouse Initiative. Програма базується на основних продуктах компанії ADABAS і Natural 4GL, власних і придбаних засобах вилучення та аналізу даних, засобі керування сховищем даних SourcePoint, який дозволяє автоматизувати процес вилучення і пересилання даних, а також їх завантаження в сховище даних.

Тенденції розвитку сховищ даних

Важливими тенденціями розвитку сховищ даних є збільшення обсягів сховищ даних підвищення потужності процесорів і швидкості передачі даних. У подальшому об'єм інформації сховище даних будуть зростати в експоненційній залежності.

У поточному часі і в майбутньому розробляються сховища даних які працюють у реальному часі, тобто система ETL обробляє вихідні дані в такому темпі як вони виходять з виробничих або бізнес процесів. Проблемою таких систем в тому, що під час обробки великої кількості нової інформації та запитів, виникає перевантаження системи яке може призвести до блокування системи і неправильного функціонування.

Ця проблема вирішується за рахунок розробки нових методів обробки інформації в системі ETL. Структура і архітектура сховищ даних відрізняється від архітектури бази даних, тому їх розробка виділяється в окремий напрямок розробки інформаційних систем .

Майбутній розвиток сховищ даних визначається новими тенденціями в розробці програмного забезпечення і виробництва технічного забезпечення. У програмному забезпеченні є тенденція передачі розробки програм на аутсорсинг, іншим компаніям. Компанії можуть розробляти як вільне програмне забезпечення так і своє авторське програмне забезпечення. Такий підхід робить розробку програмного забезпечення більш гнучким і пристосованим до вимог ринку. В розробці технічного забезпечення передбачається, що до баз даних може бути підключено велика кількість різноманітних приладів, датчиків, спеціального обладнання які використовують Інтернет та сучасні методи і засоби передачі інформації .

Висновки

Менеджери які приймають бізнес-рішення мають потребу у високоякісних даних, з якими вони працюють. Дані зібрані на основі щоденної обробки запитів страждають від поганої якості даних їх фрагментарності і відсутності історичної точки зору на зміну даних з плином часу. Сховища даних вирішують проблеми управління даними, інтеграцією і якістю даних. Без сховищ даних розв'язок цих проблем був би пов'язаний з розробкою окремих систем обробки інформації . В результаті сховища даних знижують вартість підтримки прийняття рішень і забезпечують кращу основу для прийняття рішень на базі інтегрованих високоякісний даних .

Дизайн, розробка та експлуатація сховищ даних відрізняється від проектування розробки та експлуатації систем баз даних. Сховища даних не є великою базою даних але є процесом отримання потрібної інформації, в потрібному форматі для конкретних людей, щоби вони прийняли ефективні бізнес-рішення .

Контрольні запитання

1. Дайте визначення поняттю консолідації даних.
2. Як виглядає схема зберігання даних в СППР?
3. Як виглядає схема сховища даних підприємства?
4. Які переваги і недоліки має багатовимірне сховище даних?
5. Які переваги і недоліки має реляційне сховище даних?
6. Поясніть логічні схеми організації реляційних сховищ даних “Зірка” і “Сніжинка”.
7. Як функціонують гібридні і віртуальні сховища даних?
8. Для чого потрібні які переваги надають вітрини даних?
9. Якими є перспективи розвитку СППР?

ТЕМА 4. Моделі в системах СППР

Для отримання науково обґрунтованих управлінських рішень системи підтримки прийняття рішень використовують математичні моделі. Серед них найбільш популярними у використанні є такі типи моделей: оптимізаційні, статистичного аналізу, прогнози, нейромереві.

Впровадження моделей в системах підтримки прийняття рішення містить наступні етапи:

- Формалізація вихідної проблеми.
- Побудова математичної моделі.
- Розв'язання моделі.
- Перевірка адекватності моделі.
- Впровадження результатів розв'язку .

Формалізація проблеми представляє собою осмислення і формулювання в чіткій письмовій формі суті економічної чи організаційної проблеми. На цьому етапі мають бути визначені критерії за якими буде виконуватися моделювання а також

На цьому етапі має бути визначено економічний зміст критеріїв і обмежень задачі оптимізації, або мета статистичного аналізу, критерії прогнозування та навчання моделей нейронних мереж.

Після того, як економічний зміст задачі чітко сформульовано переходять до етапу перекладу задачі на математичну мову, тобто побудову математичної моделі .

Розглянемо побудову оптимізаційних моделей . Оптимізаційна модель містить критерій оптимізації який називається цільовою функцією та обмеження.

Критерій оптимізації це показник ефективності функціонування системи управління. Цільова функція є математичним формулюванням критерія оптимізації .

Цільова функція максимізується або мінімізується. На цільову функцію накладаються обмеження Зазвичай , вони формуються у вигляді нерівностей. Загальний вигляд математичної моделі може бути представлений наступним чином .

$$p = \min f_0(x)$$

$$\text{обмеження } f_i(x) \leq b_i \quad i=1, \dots, m$$

де x - змінна задачі оптимізації;

- f_0 - цільова функція;
- f_i - обмеження;
- p - оптимальне значення цільової функції/

Методи розв'язку таких моделей показано на рис. 20

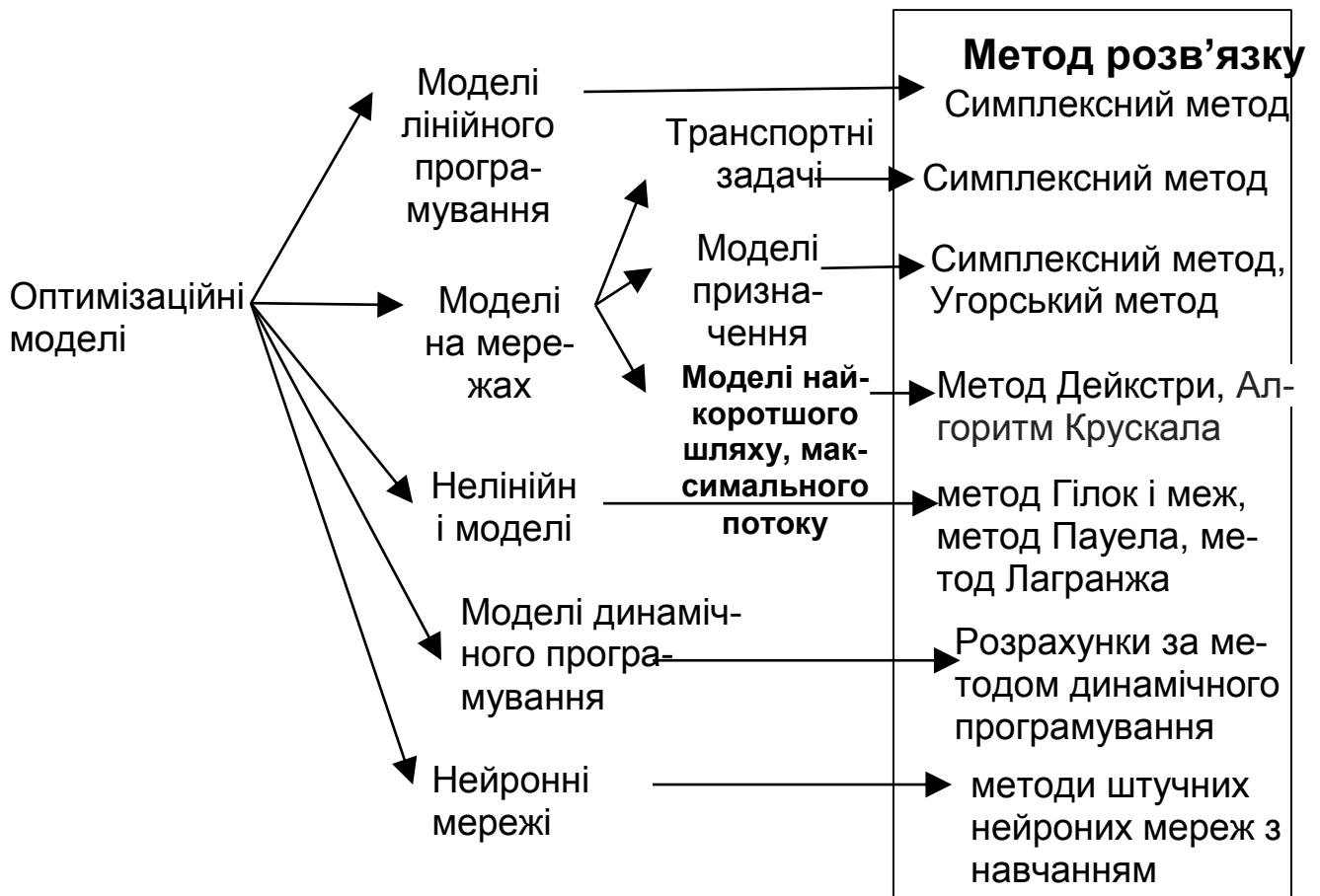


Рисунок 20. Моделі в задачах СППР та методи їх математичного розв'язку

Вибір критеріїв в оптимізаційні моделях систем СППР

Дуже важливим для побудови математичні моделі є формулювання критерія оптимізації. Існують задачі оптимізації з одним критерієм і багатокритеріальні задачі. Більшість задач оптимізації вирішується з одним критерієм. Якщо критеріїв декілька необхідно застосувати відповідно методику, яка дозволить оптимізувати задачу фактично по одному критерію. Розглянемо чому необхідно оптимізувати задачу за одним критерієм. Припустимо, що необхідно змодельювати оптимізацію виробничого процесу на підприємстві за двома критеріями прибуток і собівартість як видно з рис.21 максимум прибутку маємо при обсязі виробництва, який дорівнює x_1 , а мінімум собівартості досягається при обсязі x_2 . Якщо ми застосуємо обидва критерія для прийняття рішення, то будемо вимушені одночасно випускати обсяг продукції який дорівнює x_1 і x_2 , але це неможливо. Тому з двох критеріїв ми маємо вибрати той який є більш важливим для управління, а інший зробити обмеженням.

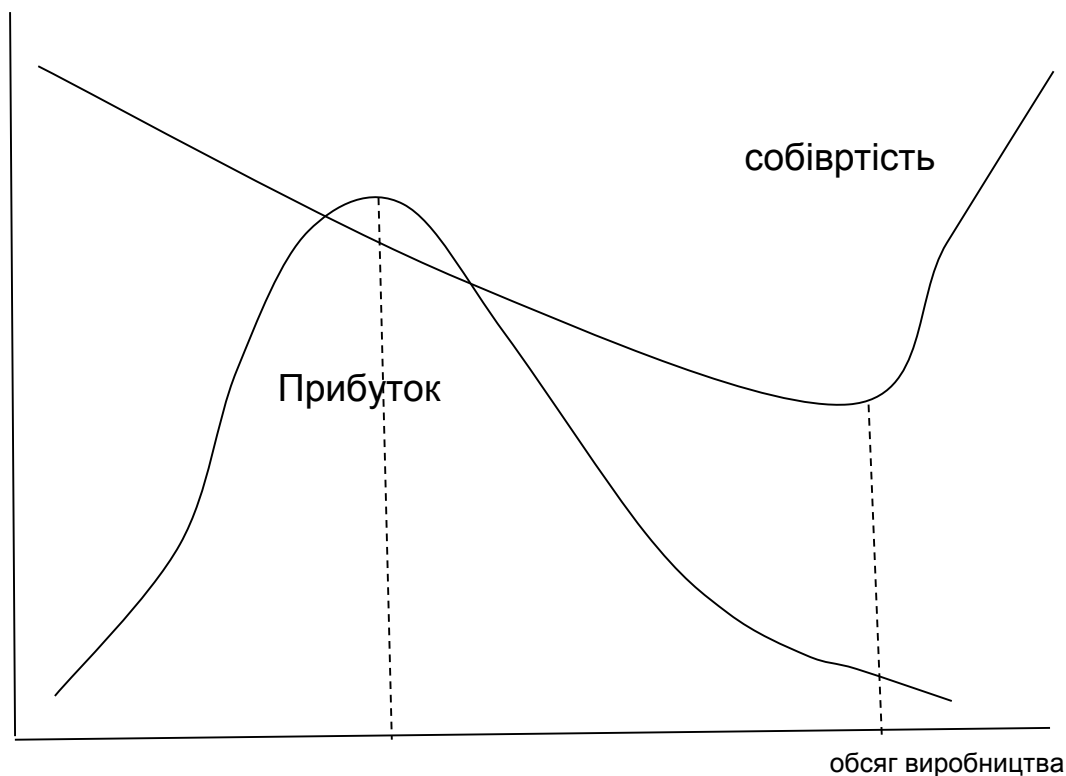


Рисунок 21 Задача з двома критеріями

В іншому підході усім критеріям надаються ваги. Більш важливому критерію надається більша вага, менше важливому - менша вага. Сумарна вага усіх критеріїв має дорівнювати одиниці. З цих критеріїв формується один мультиплікативний чи адитивний критерій. В адитивному критерії часткові критерії додаються між собою, а в мультиплікативному перемножуються між собою.

Для розв'язання оптимізаційної задачі з багатьма критеріями можна застосувати метод послідовних поступок.

Процедура вирішення багатокритеріальної задачі методом послідовних поступок полягає в тому, що всі часткові критерії розташовують і нумерують у порядку їх відносної важливості; максимізують (або мінімузують) за першим, найбільш важливим критерієм; потім призначають величину допустимого зниження значення цього критерію і максимізують другий за важливістю частковий критерій за умови, що значення першого критерію не повинно відрізнятись від максимального більш ніж на величину встановленого зниження (поступки); знову призначають величину поступки, але вже за другим критерієм і знаходять максимум третього за важливістю критерію за умови, щоб значення перших двох критеріїв не відрізнялися від раніше знайдених максимальних значень більше ніж на величини відповідних поступок; далі подібним же чином по черзі використовуються всі інші часткові критерії; оптимальною буде та стратегія, яка отримана при вирішенні задачі пошуку умовного максимуму останнього за важливістю критерію. Таким чином, при використанні методу послідовних поступок багатокритеріальна задача зводиться до почергової максимізації часткових критеріїв і вибору величин поступок.

Розглянемо декілька типів оптимізаційних моделей. Серед найбільш уживаних математичних моделей для оптимізації економічних процесів є моделі лінійного програмування.

Економіко-математичні моделі лінійного програмування

Задачами лінійного програмування називають задачі оптимізації, що мають такі особливості:

- критерій оптимізації є лінійною функцією від невідомих задачі x_1, x_2, \dots, x_n ;
- обмеження, що накладаються на можливі розв'язки мають тип лінійних рівностей або нерівностей;
- змінні приймають не від'ємні значення.

Математична постановка задачі лінійного програмування в загальному випадку сформулюється таким чином.

Нехай x_j - невідомі задачі, a_{ij} - коефіцієнти при невідомих, b_i - обмеження, Z - критерій оптимізації, c_j - коефіцієнти при невідомих у математичному формулюванні критерію оптимізації.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m \end{cases} \quad (I)$$

$$Z = c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \begin{matrix} \max \\ \min \end{matrix} \rightarrow opt$$

$$x_{ij} \geq 0$$

Необхідно визначити такі невід'ємні значення змінних $x_1, x_2, x_3, \dots, x_n$, що задовольняють умовам (I), при яких лінійна функція Z перетвориться в оптимум (мінімум або максимум у залежності від економічного змісту задачі).

Для розв'язання задач лінійного програмування при довільному числі змінних x_j застосовуються обчислювальні методи, найбільш універсальним, серед яких є симплексний метод.

Цілочислові задачі лінійного програмування

Оптимізаційна задача, змінні якої за фізичним змістом можуть приймати лише цілочислові значення, називається задачею цілочислового програмування. Обмежимося розглядом випадку, коли цільова функція й система обмежень задачі є лінійними. При цьому математична модель задачі має вигляд

$$F = \sum_{j=1}^n c_j x_j \rightarrow \max; \quad (1)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = \overline{1, m}; \quad (2)$$

$$x_j \geq 0, \quad j = \overline{1, n}; \quad (3)$$

$$x_j \text{ цілочислові,} \quad j = \overline{1, n} \quad (4)$$

Розв'язок задачі (1) - (4) можна знайти одним із розглянутих раніше методів. Проте розв'язання може виявитися як цілочисловим, так і ні. Тому у загальному випадку для визначення цілочислового розв'язку задачі (1) - (4) потрібно застосування спеціальних методів. Одним із найбільше поширених методів розв'язку задачі (1) - (4) є метод Гоморі або метод Гілок і меж.

Задачі дробово-лінійного програмування

Для певного класу практичних задач, цільова функція може являти собою відношення двох лінійних функцій. Обмеження при цьому є лінійними. Це є задача дробово-лінійного програмування, що має вигляд

$$F = \frac{\sum_{j=1}^n c_j x_j}{\sum_{j=1}^n d_j x_j} \rightarrow \max; \quad (5)$$

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = \overline{1, m}; \quad (6)$$

$$x_j \geq 0, \quad j = \overline{1, n}. \quad (7)$$

Тут c_j, d_j, a_{ij}, b_i - задані константи. Крім того, зазначимо, що $\sum_{j=1}^n d_j x_j \neq 0$.

У загальному випадку задача дробово-лінійного програмування (5) – (7) може бути зведена до задачі лінійного програмування. Для цього введемо нові змінні

$$y_0 = \left(\sum_{j=1}^n d_j x_j \right)^{-1}; \quad (8)$$

$$y_j = y_0 x_j, \quad j = \overline{1, n}. \quad (9)$$

Використовуючи введені позначення (8), (9), початкову задачу (5) – (7) зводимо до наступної задачі:

$$F' = \sum_{j=1}^n c_j y_j \rightarrow \max; \quad (10)$$

$$\sum_{j=1}^n a_{ij} y_j - b_i y_0 = 0, \quad i = \overline{1, m}; \quad (11)$$

$$\sum_{j=1}^n d_j y_j = 1; \quad (12)$$

$$y_j \geq 0, \quad j = \overline{0, n}. \quad (13)$$

Задача (10) - (13) розв'язується симплексним методом.

Нелінійне моделювання. Загальна постановка задач нелінійного програмування

У загальному випадку задача нелінійного програмування сформулюється так: визначити екстремальне значення функції $f(x_1, x_2, \dots, x_n)$,

$$f(x_1, x_2, \dots, x_n) \rightarrow \text{extr} \quad (14)$$

за умови, що змінні x_1, x_2, \dots, x_n задовольняють обмеженням

$$\left\{ \begin{array}{ll} q_i(x_1, x_2, \dots, x_n) \leq b_i, & i = \overline{1, k} \\ q_j(x_1, x_2, \dots, x_n) \geq b_j, & j = \overline{k+1, m} \end{array} \right. \quad (15)$$

$$q_s(x_1, x_2, \dots, x_n) = b_s, \quad s = \overline{m+1, n},$$

f, q - функції n змінних, b_s - задані числа. ($s = \overline{i, j, l}$).

Значення змінних x_i ($i = \overline{1, n}$), що задовольняють (15) називаються припустимими розв'язаннями, а припустиме розв'язання $x^* = (x_1^*, x_2^*, \dots, x_n^*)$, яке доставляє в екстремум функцію f називається оптимальним розв'язанням задачі (14),(15). Якщо хоча б одна з функцій є нелінійною, то задача (14), (15) представляє собою задачу нелінійного програмування (НП). Універсальних методів розв'язання задач НП не існує. У залежності від вигляду функції (14) і обмежень (15) застосовується той або інший метод розв'язання.

Якщо система обмежень на змінні x_1, x_2, \dots, x_n містить тільки рівняння, то це є окремий випадок задачі НП:

$$f(x_1, x_2, \dots, x_n) \rightarrow \text{extr}, \quad (16)$$

$$q_i(x_1, x_2, \dots, x_n) = b_i, \quad i = \overline{1, m}.$$

(17)

Функції f, q_i ($i = \overline{1, m}$) є такими, що диференціюються.

Алгоритм розв'язання задачі методом множників Лагранжа

Задача (16), (17) на умовний екстремум може бути розв'язана методом множників Лагранжа, що налічує такі основні етапи:

1. Уводиться набір допоміжних змінних $\lambda_1, \lambda_2, \dots, \lambda_m$ за кількістю обмежень (8.2), що називаються невизначеними множниками Лагранжа.
2. Укладається функція Лагранжа

$$L(x_1, x_2, \dots, x_n, \lambda_1, \lambda_2, \dots, \lambda_m) = f(x_1, x_2, \dots, x_n) + \sum_{i=1}^m \lambda_i [q_i(x_1, x_2, \dots, x_n) - b_i] \quad (18)$$

3. Складається система рівнянь:

$$\begin{aligned} \frac{\partial L}{\partial x_j} &= 0; & j &= \overline{1, n} \\ &\dots & & \\ \frac{\partial L}{\partial \lambda_i} &= 0; & i &= \overline{1, m} \end{aligned} \quad (19)$$

4. Система (7.4) містить $(n+m)$ рівнянь, розв'язуючи їх визначають невідомі $x_1^*, x_2^*, \dots, x_n^*, \lambda_1^*, \lambda_2^*, \dots, \lambda_m^*$. Необхідною умовою того, що точка $x_1^*, x_2^*, \dots, x_n^*$ є розв'язком задачі, має бути виконання рівностей (19). Система (19) має декілька

розв'язань і вони по черзі безпосередньо підставляються в (16) для визначення виду екстремуму.

Моделі, які застосовують математичний апарат теорії ігор

Гра — це дійсний або формальний конфлікт, у якому є принаймні два учасники (гравці), кожний із яких прагне досягнення власних цілей.

Однозначний опис вибору, який здійснює гравець у кожній з можливих ситуацій, називається стратегією гравця.

Стратегія гравця називається оптимальною, якщо при багатократному повторенні гри вона забезпечує гравцю максимальний середній виграш (мінімальний середній програш).

Постановка задачі

Нехай є два гравці, один із яких може вибрати i -ю стратегію з m -можливих ($i=1, m$), а другий, не знаючи вибору першого, вибирає j -ю стратегію з n -можливих ($j=1, n$). У результаті перший гравець виграє кількість $a(ij)$, а другий програє цю кількість. Матриця

$$A = a(ij) = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \quad (20)$$

називається матрицею гри. Рядки матриці A відповідають стратегіям першого гравця, а стовпчикам-стратегіям другого. Ці стратегії називаються чистими.

Число $\alpha = \max_i (\min_j a_{ij})$ називається нижньою ціною гри, а відповідна α стратегія максиміною.

Число $\beta = \min_j (\max_i a_{ij})$ називається верхньою ціною гри, а відповідна β стратегія мінімаксною. Нижня ціна гри завжди не перевершує верхню, тобто $\alpha \leq \beta$.

Розв'язок гри із сідловою точкою

Якщо $\alpha = \beta = V$, то число V називається ціною гри, а сама гра є гра із сідловою точкою. Для гри із сідловою точкою її розв'язок складається у визначенні ціни гри, а також мінімальною й максимальною стратегіями гравців, що і є оптимальними. Таким чином. розв'язок у такому випадку шукається у чистих стратегіях.

Приклад 1. Знайти розв'язок гри заданою матрицею A .

Перший гравець має три можливі стратегії, а другий - чотири. Знайдемо нижню й верхню ціну гри.

$$\alpha = \max\{\min(3;4; 1; 5); \min(6;3; 2; 4); \min(3;1; 0; 7)\} = \max\{1; 2; 0\} = 2;$$

$$\beta = \min\{\max(3;6; 3); \max(4;3; 1); \max(1;2; 0); \max(5;4; 7)\} = \min\{6; 4; 2; 7\} = 2.$$

$$A = \begin{pmatrix} 3 & 4 & 1 & 5 \\ 6 & 3 & 2 & 4 \\ 3 & 1 & 0 & 7 \end{pmatrix} \leftarrow$$

Таким чином $\alpha = \beta = 2$, і маємо гру із сідловою точкою. Ціна гри дорівнює 2. Оптимальна стратегія першого гравця - третя, а оптимальна стратегія другого гравця - друга. Ці стратегії забезпечують максимальний гарантований вигреш першому гравцю (дорівнює 2) і такий само гарантований мінімальний програш другому. Відхилення від оптимальної стратегії одним із гравців

призводить до додаткових утрат.

Розв'язок гри, що не має сідлової точки

Коли гра, яка задана матрицею, не має сідлової точки, то $\alpha < V < \beta$, а для знаходження її розв'язку використовують змішані стратегії.

Вектор, кожна з компонент якого показує відносну частоту використання гравцем відповідної стратегії, називається змішаною стратегією даного гравця.

Позначимо: $U = (u_1, u_2, \dots, u_m)$ $u_i \geq 0$ ($i = \overline{1, m}$) - змішана стратегія першого

гравця; $Z = (z_1, z_2, \dots, z_n)$ $z_j \geq 0$ ($j = \overline{1, n}$) - змішана стратегія другого гравця.

Так як u_i і z_j - відносні частоти використання гравцями чистих стратегій, то

$$\sum_{i=1}^m u_i = 1 \text{ и } \sum_{j=1}^n z_j = 1$$

Якщо U^* і Z^* -оптимальні змішані стратегії першого й другого гравців, то ціна гри обчислюється за формулою:

$$V = \sum_{j=1}^n \sum_{i=1}^m a_{ij} u_i^* z_j^* \quad (21)$$

Визначення оптимальних змішаних стратегій V^* , Z^* і ціни гри складає процес розв'язку гри без сідлової точки.

Теорема 1. Якщо один із гравців застосовує оптимальну змішану стратегію, то його вигреш дорівнює ціні гри V незалежно від того, з якими частотами буде застосовувати другий гравець стратегії, що ввійшли в оптимальну (у тому числі і чисті стратегії).

За допомогою теореми 1 можна знайти розв'язок ігор без сідлової точки розміром 2×2 ; $2 \times n$; $n \times 2$.

Приклад 2. Знайти розв'язок гри, заданої матрицею

$$A = \begin{pmatrix} -1 & 3 \\ 4 & 2 \end{pmatrix}.$$

$\alpha = \max(-1; 2) = 2$ - нижня ціна гри;

$\beta = \min(4; 3) = 3$ - верхня ціна гри.

Значить маємо гру без сідлової точки, причому $2 < V < 3$.

$U = (u_1 \ u_2)$ - змішана стратегія першого гравця;

$Z = (z_1 \ z_2)$ - змішана стратегія другого гравця.

Якщо перший гравець застосовує оптимальну змішану стратегію, то за теоремою 1 можна скласти рівняння

$-1u_1^* + 4u_2^* = V$ - при застосуванні другим гравцем першої стратегії.

$3u_1^* + 2u_2^* = V$ - при застосуванні другим гравцем другої стратегії.

З урахуванням умови $u_1^* + u_2^* = 1$, маємо систему

$$\begin{cases} -u_1^* + 4u_2^* = V \\ 3u_1^* + 2u_2^* = V \\ u_1^* + u_2^* = 1 \end{cases} \quad (21)$$

Розв'язуючи її, знаходимо $u_1^* = 1/3$, $u_2^* = 2/3$, $V = 7/3$.

Аналогічно, застосовуючи теорему 1 для другого гравця, одержимо систему

$$\begin{cases} -z_1^* + 3z_2^* = V \\ 4z_1^* + 2z_2^* = V \\ z_1^* + z_2^* = 1 \end{cases} \quad (22)$$

Звідки $z_1^* = 1/6$, $z_2^* = 5/6$, $V = 7/3$. Таким чином, $U^* = (1/3; 2/3)$; $Z^* = (1/6; 5/6)$; $V = 7/3$.

Для першого гравця оптимальної буде стратегія, коли друга чиста стратегія застосовується у два рази частіше ніж перша, а для другого - коли друга застосовується в п'ять разів частіше, чим перша.

Розв'язок задач теорії ігор за допомогою методів лінійного програмування

Теорема 2. Для того щоб число V було ціною гри, а u^* і z^* - оптимальними стратегіями, необхідно й достатньо виконання нерівностей

$$\sum_{i=1}^m a_{ij} u_i^* \geq V, \quad j = \overline{1, n}; \quad (23)$$

$$\sum_{j=1}^n a_{ij} z_j^* \leq V, \quad i = \overline{1, m} \quad (24)$$

За допомогою цієї теореми можна розв'язати задачу теорії ігор будь-якого розміру $m \times n$, попередньо претворивши її до задачі ЛП.

Розглянемо гру $m \times n$, обумовлену матрицею

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

Розділимо обидві частини нерівності (23) на V

$$\sum_{i=1}^m a_{ij} \frac{u_i^*}{V} \geq 1 \quad j = \overline{1, n}$$

Позначимо $\frac{u_i^*}{V} = y_i^* \quad i = \overline{1, m}$. Тоді

$$\sum_{i=1}^m a_{ij} y_i^* \geq 1 \quad j = \overline{1, n} \quad y_i \geq 0, i = \overline{1, m}$$

$$\sum_{i=1}^m u_i^* = 1$$

Використовуючи введене позначення, перепишемо умову $\sum_{i=1}^m u_i^* = 1$ у вигляді

$$\sum_{i=1}^m y_i^* = \frac{1}{V}$$

. Перший гравець прагне одержати максимальний виграш, то він повинен

прагнути забезпечити мінімум $\frac{1}{V}$. Маємо задачу лінійного програмування

$$\sum_{i=1}^m y_i^* \rightarrow \min \quad ; \quad (25)$$

$$\sum_{i=1}^m a_{ij} y_i^* \geq 1 \quad j = \overline{1, n} \quad ; \quad (26)$$

$$y_i^* \geq 0, \quad i = \overline{1, m} \quad . \quad (27)$$

Міркуючи аналогічно у відношенні другого гравця, можна скласти задачу, двоїсту стосовно (23), (24), (25):

$$\sum_{j=1}^n x_j^* \rightarrow \max \quad ; \quad (28)$$

$$\sum_{j=1}^n a_{ij} x_j^* \leq 1 \quad i = \overline{1, m} \quad ; \quad (29)$$

$$x_j^* \geq 0, \quad j = \overline{1, n} \quad . \quad (30)$$

Використовуючи розв'язок пари двоїстих задач, одержуємо вираз для визначення стратегій і ціни гри:

$$V = \frac{1}{\sum_{j=1}^n x_j^*} = \frac{1}{\sum_{i=1}^m y_i^*} \quad ; \quad (31)$$

$$u_i^* = Vy_i^*, \quad i = \overline{1, m}; \quad z_j^* = Vx_j^*, \quad j = \overline{1, n} \quad . \quad (32)$$

Таким чином розв'язуються задачі теорії ігор. Практична реалізація розв'язку на комп'ютері виконується із застосуванням програмного забезпечення яке показано у таблицях нижче .

Моделі які використовують нейронні мережі

Штучні нейромережі є електронними моделями нейронної структури мозку, які здатні навчатись, використовувати накопичений досвід. Багато проблем, які поки що не підвладні розв'язуванню сучасними комп'ютерами, можуть бути ефективно вирішені блоками нейромереж. Наведемо декілька з них.

- **Класифікація образів.** Наприклад розпізнавання букв, розпізнавання мови, класифікація сигналу електрокардіограми, класифікація кліток крові.
- **Кластеризація/категоризація.** Алгоритм кластеризації розміщує близькі образи в один кластер. Відомі випадки застосування кластеризації для видобутку знань, стиснення даних і дослідження властивостей даних.
- **Апроксимація функцій.** Завдання апроксимації полягає в знаходженні невідомої функції F . Апроксимація функцій необхідна при рішенні численних інженерних і наукових задач моделювання.
- **Прогноз.** Визначення значень економічних показників, які вони будуть мати у майбутньому.
- **Оптимізація.** Численні проблеми в математиці, статистиці, техніці, науці, медицині й економіці можуть розглядатися як проблеми оптимізації. Задачею алгоритму оптимізації є знаходження такого розв'язку, що задовольняє системі обмежень і максимізує чи мінімізує цільову функцію.
- **Керування.** Оптимальне управління підприємством

Нейронні мережі мають наступні характеристики:

- розподілене представлення інформації і паралельні обчислення;
- здатність до навчання і здатність до узагальнення;
- адаптивність;

- толерантність до помилок
- низьке енергоспоживання.

СППР, побудовані на принципах біологічних нейронів мають перелічені характеристики, що можна вважати суттєвим здобутком в обробці даних інформаційними системами.

Базовий модуль нейронних мереж штучний нейрон моделює основні функції природного нейрона (рис. 22).

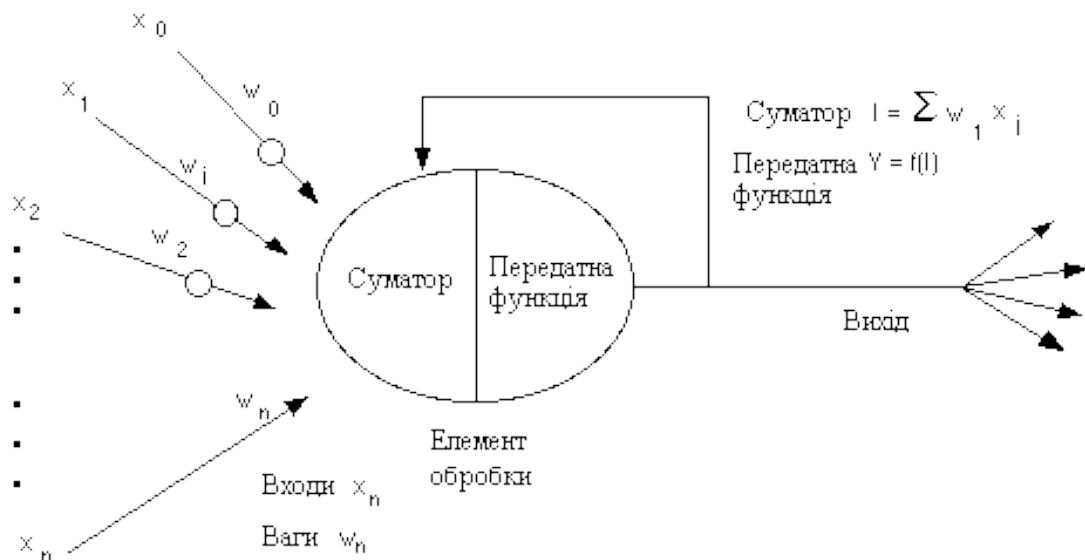


Рис. 22. Базовий штучний нейрон.

Вхідні сигнали x_n зважені ваговими коефіцієнтами w_n підсумовуються суматором, проходять через передатну функцію, генерують результат і виводяться.

В сучасних пакетах програм штучні нейрони називаються "елементами обробки" і мають набагато більше можливостей, ніж штучний нейрон, описаний вище. На рис. 23 зображена детальна схема спрощеного штучного нейрону.

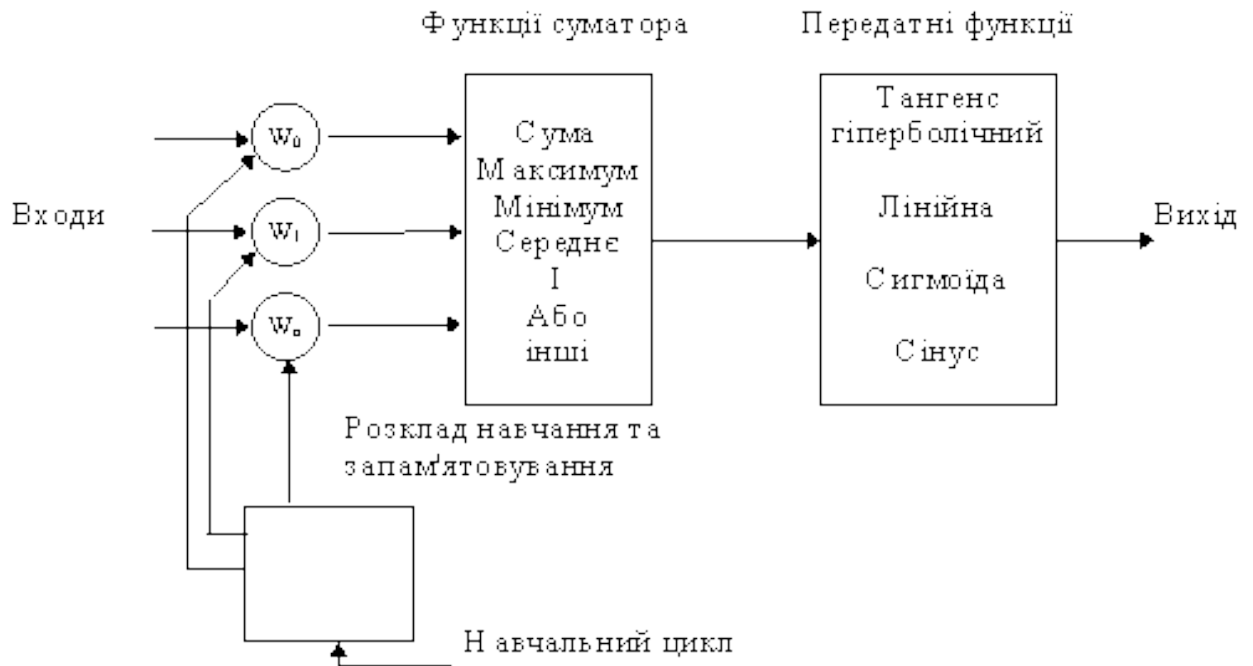


Рис. 23. Модель "елементу обробки"

Модифіковані входи передаються на суматор, який може виконувати в загальному випадку не тільки додавання, але і пошук середнього, максимуму, мінімуму, обчислення логічних функцій OR, AND, тощо. Окрім того, більшість комерційних програм дозволяють інженерам-програмістам створювати власні функції сумування за допомогою підпрограм, закодованих на мові високого рівня (C, C++, TurboPascal). Інколи функція сумування ускладнюється додаванням функції активації, яка дозволяє функції сумування оперувати в часі.

Далі з виходу суматора результат надсилається у передатну функцію, яка за конкретною математичною формулою обліковує вихідний сигнал нейрону. На виході нейрону за допомогою певного алгоритму отримуємо 0 або 1, -1 або 1, або яке-небудь інше число. В існуючих нейромережах в якості передатних функцій можуть бути використані сигмоїда, синус, гіперболічний тангенс та ін. Приклад того, як працює передатна функція показаний на рис. 24. Як видно з рисунку для сигмоїдної передатної функції нейромережа видає сигнал від 0 до 0,5 для значень входу -1, 0 і сигнал 0,5, 1 для входів у діапазоні від 0 до 1.



Рис. 24. Сигмоїдна передатна функція.

Отриманий сигнал з виходу нейрона надходить на входи інших нейронів або до зовнішнього з'єднання, як це передбачає структура нейромережі.

Нейрони можуть бути об'єднані шари і мережі. Об'єднані в мережу нейрони можуть розв'язувати математичні задачі, розпізнавати об'єкти, прогнозувати.

Штучні нейронні мережі

Ефективне використання нейронних мереж згідно з розробкою схеми зв'язків ніж окремими нейронами. Від того як вони пов'язані між собою залежить здатність системи навчатися і правильно знаходити розв'язок задач.

Групування нейронів у мозку людини забезпечує обробку інформації динамічним, інтерактивним та самоорганізуючим шляхом. Біологічні нейронні мережі створені у тривимірному просторі з мікроскопічних компонент і здатні до різноманітних з'єднань. Але для створеної штучної мережі існують фізичні обмеження.

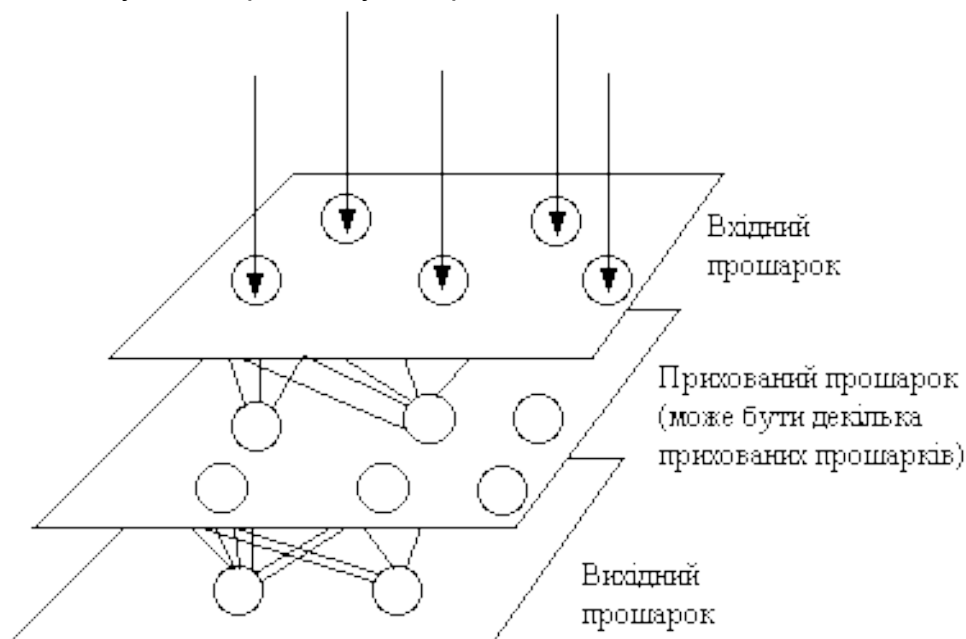


Рис. 25. Діаграма простої нейронної мережі

На рис. 25 показана типова структура штучних нейромереж. В практичних дослідженнях іноді застосовуються мережі, які містять лише один прошарок, або навіть один елемент. Більшість нейронних мереж в системах СППР, містять як мінімум три типи прошарків - вхідний, прихований та вихідний.

Прошарок вхідних нейронів отримує дані або з вхідних файлів, або безпосередньо з електронних датчиків. Вихідний прошарок пересилає інформацію безпосередньо до зовнішнього середовища, до вторинного комп'ютерного процесу, або до інших пристроїв. Між цими двома прошарками може бути багато прихованих прошарків, які

містять багато нейронів у різноманітних зв'язаних структурах. Входи та виходи кожного з прихованих нейронів йдуть до інших нейронів.

Напрямок зв'язку від одного нейрону до іншого є важливим аспектом нейромереж. У більшості мереж кожен нейрон прихованого прошарку отримує сигнали від всіх нейронів попереднього прошарку та звичайно від нейронів вхідного прошарку. Після виконання операцій над сигналами, нейрон передає свій вихід до всіх нейронів наступних прошарків, забезпечуючи шлях передачі вперед (feed forward) на вихід.

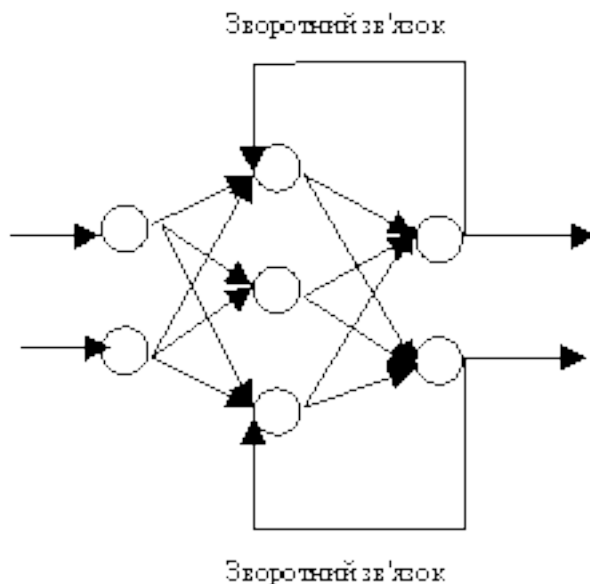


Рис. 26. Мережа зі зворотнім зв'язком

При зворотному зв'язку, вихід нейронів прошарку скеровується до нейронів попереднього прошарку (рис. 26).

Шлях, яким нейрони з'єднуються між собою має значний вплив на роботу мережі. Більшість пакетів професіональної розробки програмного забезпечення дозволяють користувачу додавати, вилучати та керувати з'єднаннями як завгодно. Постійно корегуючи параметри, зв'язки можна робити як збуджуючими так і гальмуючими.

Навчання штучної нейронної мережі

Здатність до навчання є фундаментальною властивістю мозку. Процес навчання може розглядатися як визначення архітектури мережі і налаштування ваг зв'язків для ефективного виконання спеціальної задачі. Під час навчання нейромережа налаштовує ваги зв'язків між нейронами.

Для процесу навчання необхідно мати:

- Модель зовнішнього середовища, у якій функціонує нейронна мережа;
- Алгоритми за якими будуть визначатися вагові параметри мережі.

Алгоритм навчання це процедура, в якій визначено правила налаштування ваг для навчання мережі.

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

Методи і моделі прогнозу які застосовуються в системах підтримки прийняття рішень .

В СППР застосовуються методи короткострокового, середньострокового і довгострокового прогнозування. Найбільш поширеними методами є методи короткострокового і середньострокового прогнозування . Практика показує, що чим більший проміжок часу на який ми прогнозуємо тим більша похибка прогнозу . Методологічно, задачі довгострокового прогнозу мають найбільшу похибку, тому і використовуються рідше.

При виконанні прогнозування необхідно відповісти на ряд питань.

- **Який прогноз нам потрібен?**

Необхідно визначити тип прогнозу: довгостроковий, короткостроковий. Вказати кількість кроків на які буде зроблено прогноз

- **Яка очікувана точність прогнозу?**

Необхідно вказати відносну і абсолютні похибки моделі прогнозу. При цьому потрібно враховувати горизонт прогнозування, дані з найбільшою кількістю кроків вперед мають максимальну похибку.

- **Де знаходиться інформація, необхідна для формування прогнозів?**

Необхідно точно вказати з яких документів повинна бути взята інформація і за який період часу.

- **Хто і як буде використовувати прогноз?**

Необхідно точно оцінити вимоги до прогнозу. Необхідно знати для чого і як буде застосовано прогноз, оцінити наслідки використання прогнозу (прибутки, збитки), врахувати людський фактор.

- **Чи буде прогноз зроблений вчасно, щоб допомогти процесу прийняття рішення?**

Оцінити час отримання прогнозу (створення моделі, настройка, налагодження, підгонка моделі, розрахунки та аналіз результатів). Якщо всі зазначені етапи укладаються в прийнятні терміни прогнозування, то модель приймають, інакше переглядають модель і методи організації прогнозування

Якість і точність прогнозу залежать від вихідних даних, які мають бути ретельно підготовлені за спеціальними методиками.

Розглянемо підготовку даних для прогнозування.

Підготовка вихідних даних для прогнозування

Підготовка даних передбачає відновлення пропущених даних, коригування неправильних даних, перетворення нестационарних даних до стаціонарного виду.

Відновлення пропущених даних

При проведенні статистичного аналізу на практиці обмежуються аналізом не всієї генеральної сукупності в цілому, а лише деякого вибіркового числа спостережень. Вибірка що аналізується повинна відповідати критеріям якості і повноти. У реальності доводиться стикатися з ситуацією, коли деякі з властивостей одного або декількох об'єктів відсутні - виникає ситуація даних з пропусками, що значно ускладнює математичну обробку. Похибки основних статистичних характеристик, таких як математичне очікування або дисперсія, зростають прямо пропорційно обсягам пропущених даних.

Нині в математичній статистиці існує декілька шляхів вирішення проблеми неповних даних

1. Виключення неправильних даних з вихідної вибірки. Даний підхід до проблеми можна охарактеризувати як некоректний, оскільки неповні дані несуть в собі нову інформацію, необхідну для дослідження, і тому їх важливо включати в аналіз;
2. Застосування спеціально розроблених математичних методів аналізу неповних даних, таких як метод зважування або метод максимальної вірогідності і EM-алгоритм (Expectation Maximization), при цьому значно зростає складність проведення аналізу;
3. Відновлення пропусків (найбільш поширені методи заповнення по середньому і по регресії). У більшості випадків саме цей підхід вважається найбільш ефективним і зручним рішенням проблеми.

Методи аналізу неповних даних можна умовно розподілити на наступні групи.

1. **Метод виключення некомплектних об'єктів.** За відсутності у ряду об'єктів значень яких-небудь змінних некомплектні об'єкти видаляються з аналізу. Підхід

легко реалізується і може бути задовільним при малому числі пропусків. Головний недолік такого підходу обумовлений втратою інформації при виключенні неповних спостережень.

2. **Методи із заповненням.** При цьому підході пропущені значення вихідної вибірки заповнюються і отримані «повні» дані обробляються звичайними методами. Найбільш часто використовуються наступні процедури заповнення пропусків.
 - а. **Заповнення середніми.** Підставляються середні присутніх значень. Метод безумовного середнього - найпростіший вид заповнення. Він полягає в заміні відсутніх значень y_{ij} середнім $\bar{y}_j^{(i)}$ значенням.
 - б. **Заповнення з упередженим підбором.** Пропуски заповнюються значеннями, отриманими для іншого подібного об'єкта вибірки.

Найбільш часто використовувані методи з упередженим підбором: підстановка з підбором всередині груп і підбір найближчого сусіда.

У **першому підході** до проблеми формуються групи, і пропуски в кожній групі заповнюються присутніми значеннями з неї ж. Заповнення з підбором широко поширене. Воно може включати дуже складні схеми відбору об'єктів. Хоча практика підтвердила переваги цього методу, літератури, присвяченої його теоретичним властивостям, явно недостатньо.

Другий підхід заснований на введенні метрики d для вимірювання відстані між об'єктами, визначеного в просторі супутніх змінних, і виборі підстановки по об'єкту з присутнім значенням, найближчому до об'єкта з пропуском. Наприклад, позначимо x_{i1}, \dots, x_{iJ} - значення J змінних, в об'єкта i з пропуском y_i . Визначимо відстань $d(i, k)$ між об'єктами i і k .

$$d(i, k) = \max_j |x_{ij} - x_{kj}|$$

Ми можемо вибрати підстановку для y_i з тих x_k -об'єктів, у яких:

- 1) спостерігаються $y_k, x_{k1}, \dots, x_{kJ}$;
- 2) $d(i, k)$ менше деякого порога d_0 .

Число «кандидатів» - підходящих x_k об'єктів - можна вибрати, змінюючи d_0 . Схеми найближчого сусіда вимагають значних обчислювальних витрат. Вони стали застосовуватися порівняно недавно.

Заповнення пропущених значень за допомогою регресії.

До цієї групи відноситься метод заповнення умовними середніми або так званий метод Бака. У цьому методі складається регресійне рівняння для декількох змінних $Y_k = f(Y_i)$, в це рівняння підставляються реальні значення змінних і отримують пропущені

значення Y_k . Метод є більш перспективним способом заповнення пропусків у порівнянні з попередніми методами і відноситься до модельних методів.

EM-алгоритм ([англ.](#) *Expectation-maximization (EM) algorithm*) — алгоритм, для знаходження оцінок максимальної схожості параметрів моделей з пропущеними(прихованими) даними. Кожна ітерація алгоритму складається з двох кроків. На Е-кроці (expectation) вираховується очікуване значення функції правдоподібності, і знаходиться приблизне значення пропущених даних. На М-кроці (maximization) розраховується оцінка максимальної схожості, і уточнюються значення пропущених даних знайдених на Е-кроці. Потім це значення використовується для Е-кроку на наступній ітерації. Ітерації повторюються доти, поки буде знайдено значення пропущеної (втраченої) змінної із заданою точністю.

Аналіз викидів

Викидами називаються спостереження значення яких різко відрізняються від інших. Такі викиди часто вважають помилковими даними і тому вони можуть бути відкинуті. Для цього виконується візуальний аналіз даних або діаграми розсіювання на яких виявляються викиди. Дані, які різко виділяються можуть бути відкинуті (рис. 27)

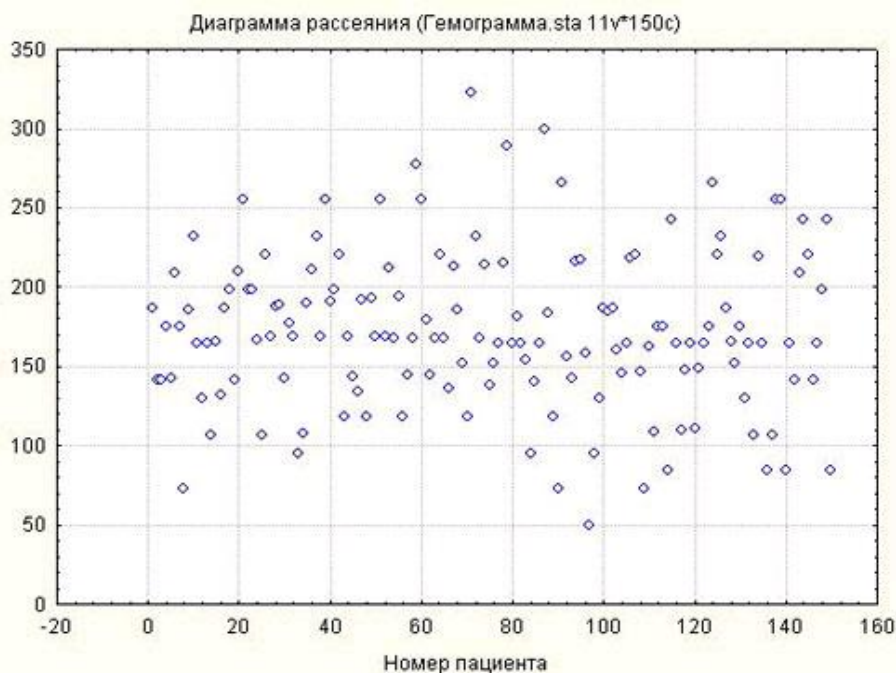


Рисунок 27. Діаграма розсіювання яка містить викиди.

Видалення нестационарності у вихідних даних

Більшість моделей прогнозування мають алгоритми, які використовують стаціонарні дані. На практиці дані які надходять до СППР не є стаціонарним. Тому, дані треба привести до стаціонарного виду із застосуванням спеціальних методів.

Нестаціонарності бувають двох видів . Дані нестаціонарні за математичним очікуванням і нестаціонарні за дисперсією . Данні нестаціонарні за математичним очікуванням - такі які мають тренд. Дані нестаціонарні за дисперсією значно змінюють амплітуду своїх коливань.

Для виключення не стаціонарності за математичним очікуванням застосовують методи обчислення кінцевих різниць. Для компенсації не стаціонарності за дисперсією застосовують логарифмування вихідних даних.

Розглянемо усунення не стаціонарності з математичним очікуванням.

Перша кінцева різниця для вихідних даних X_i обчислюється за формулою $\Delta X_{i+1} = X_{i+1} - X_i$

Друга кінцева різниця знаходиться з формули $\Delta^2 X_{i+1} = \Delta X_{i+1} - \Delta X_i$.

Для нестаціонарних даних обчислюється перша кінцева різниця. Якщо в результаті розрахунків тренд усунено, то ці дані використовуються як вихідні для прогнозу . Якщо тренд не усунено виконують розрахунок другої кінцевої різниці , отриманий результат використовують як вихідні данні для прогнозу .

Розглянемо приклад . Вихідні дані подано у таб. 8 , результати обчислень перших і других кінцевих різниць також містяться у цій таблиці. Як видно з графіків перша і друга кінцеві різниці видаляють тренд із даних.

Відновлення даних з кінцевої різниці виконується за оберненою формулою яка наведена в таб. 8

Таблиця 8

X	$\Delta X = X_{i+1} - X_i$	$\Delta^2 X = \Delta X_{i+1} - \Delta X_i$	Відновлені дані $X_i = \Delta X_i + X_{i-1}$
15			
25	10		25
22	-3	-13	22
28	6	9	28
32	4	-2	32
28	-4	-8	28
25	-3	1	25
34	9	12	34
40	6	-3	40
42	2	-4	42

Результати розрахунків таблиці 9 показано на рис. 28 — рис.30

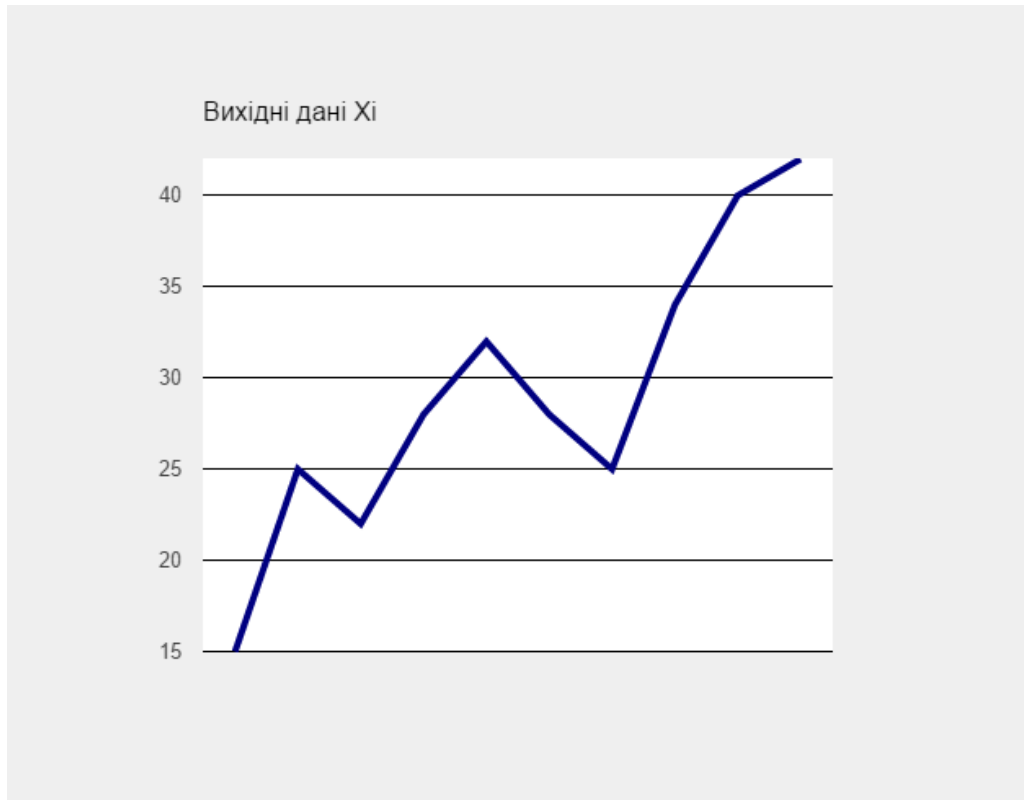


Рисунок 28. Вихідні дані з трендом

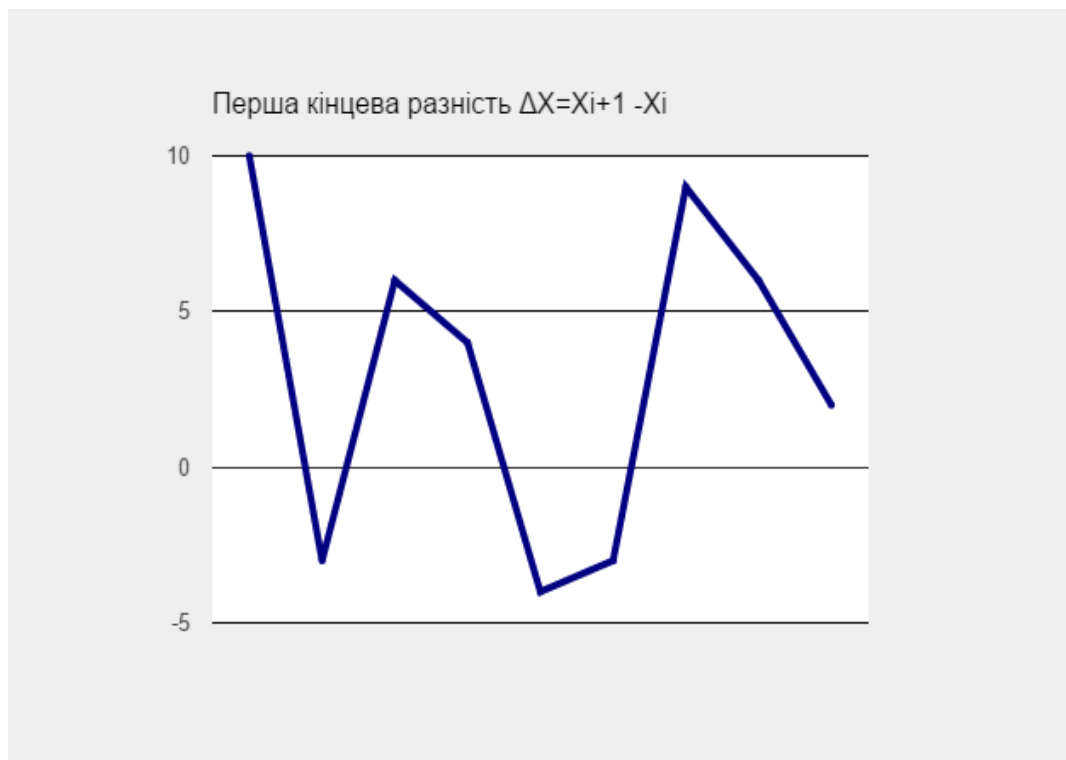


Рисунок 29 . Перша кінцева різниця . Тренду немає.

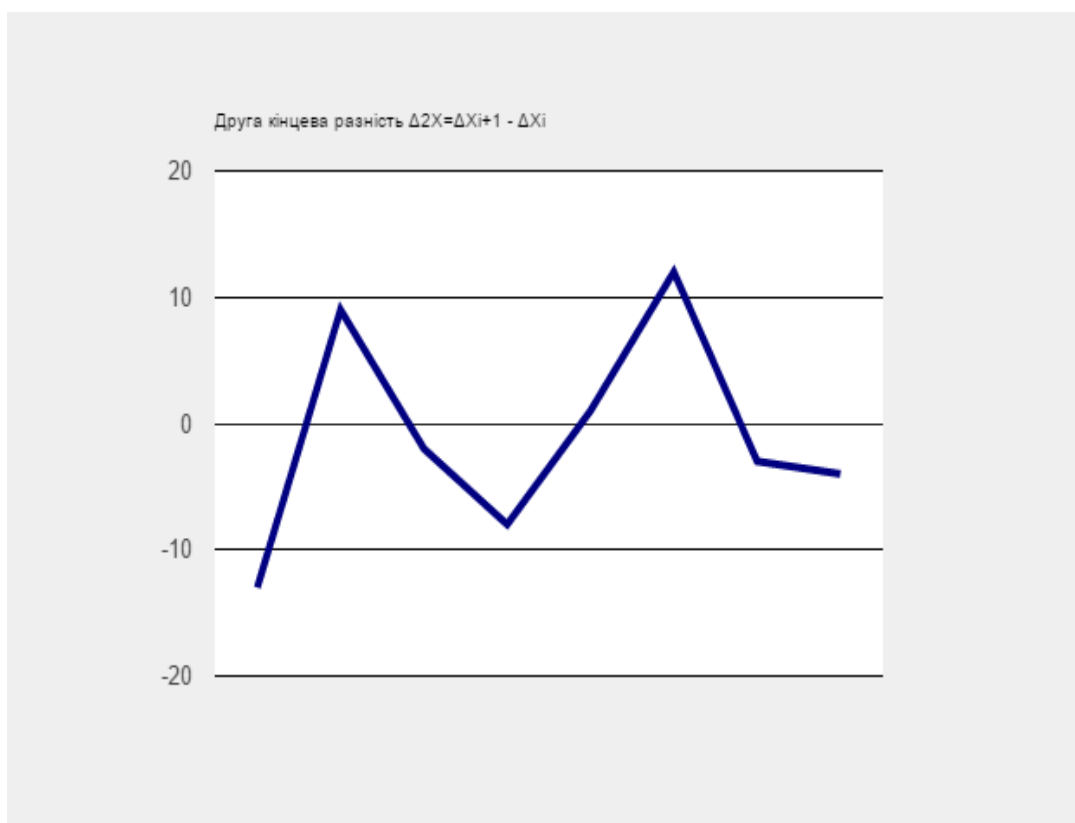


Рисунок 30 . Друга кінцева різниця

Видалення нестационарності за дисперсією проілюстровано у таблиці 9 і малюнках рис. 31 і 32

Таблиця 9

X	LN(X)
150	5,010635294
25	3,218875825
22	3,091042453
28	3,33220451
320	5,768320996
28	3,33220451
205	5,323009979
64	4,158883083
140	4,941642423
42	3,737669618

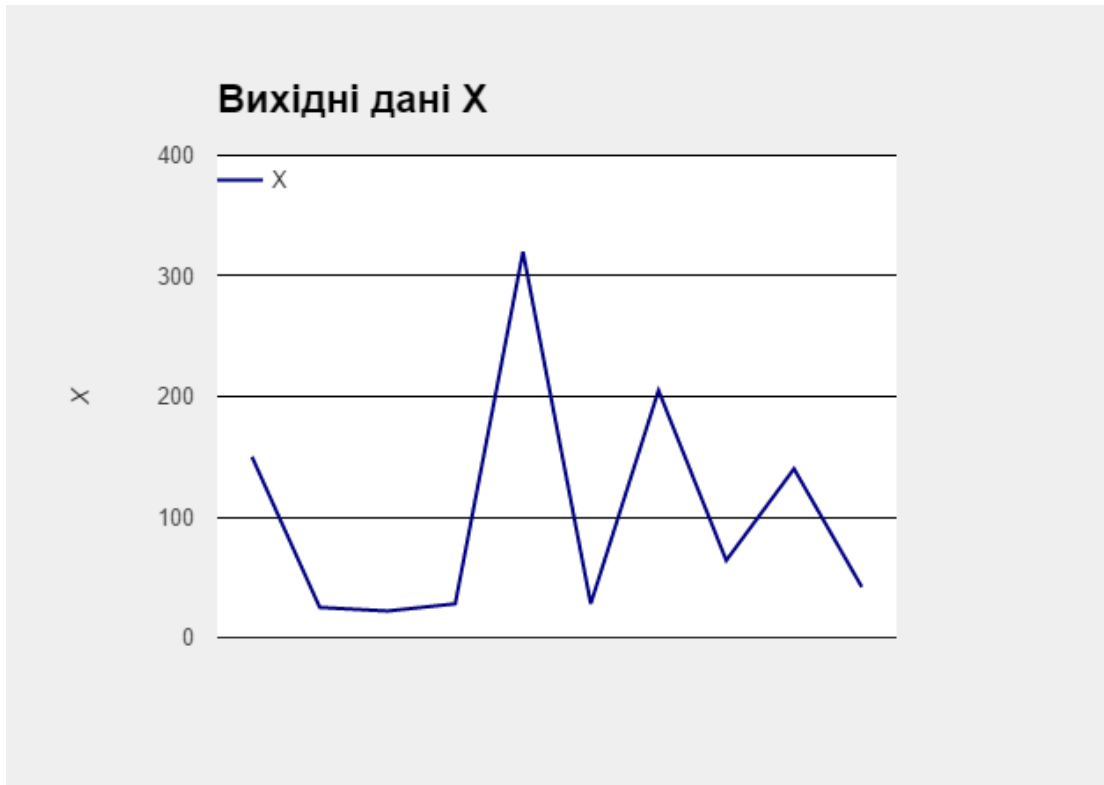


Рисунок 31 Вихідні дані нестационарні за дисперсією



Рисунок 32 . Дані з виділеною нестаціонарністю за дисперсією

Як видно з рис. 32 дані коливаються у малому діапазоні від 6 до 2,8. У вихідних даних коливання були від 420 до 22. Таким чином нестаціонарність була усунена.

Регресійні моделі прогнозування

Регресійними називають моделі, засновані на рівнянні регресії чи системі регресійних рівнянь, що зв'язують величини ендогенних (залежних) і екзогенних (незалежних) змінних [4].

Розрізняють рівняння (моделі) парної і множинної регресії. Якщо для позначення залежних змінних використовувати букву y , а для незалежних змінних букву x , то рівняння парної регресії має вигляд: $y = f(x) + \varepsilon$, а рівняння множинної регресії: $y_t = f(x_{1t}, x_{2t}, \dots, x_{pt}) + \varepsilon_t$. де, ε_t і $\varepsilon_{,i}$ - похибки моделей регресії.

Для знаходження параметрів-коефіцієнтів при незалежних змінних, цих моделей використовують метод найменших квадратів. Метод забезпечує мінімум квадрату відхилень ε^2 , значень розрахованих за регресійною моделлю y_i від реальних даних, які досліджуються (рис.32).

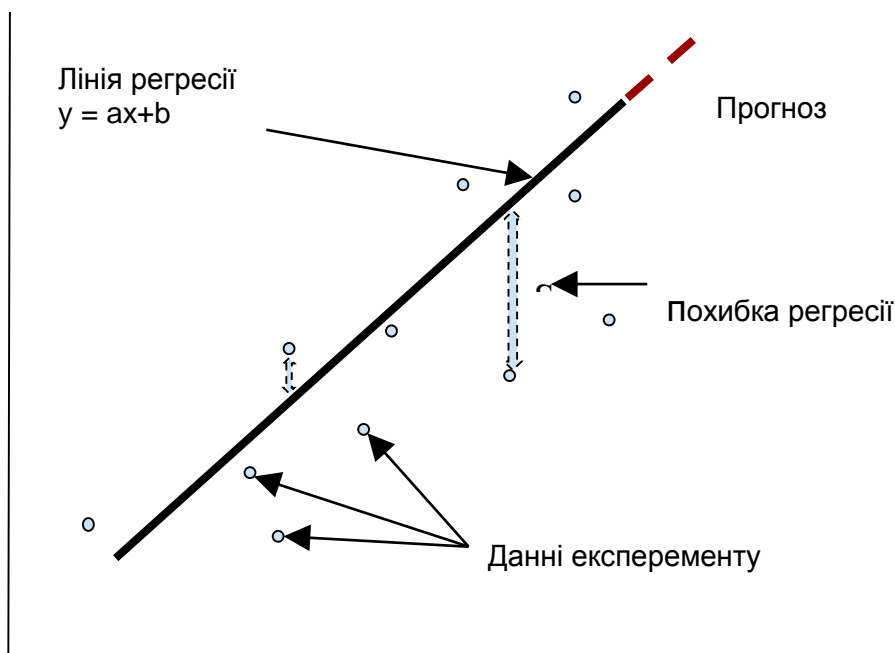


Рисунок 33. Регресійна модель прогнозування

Для отримання прогнозу продовжують отриману регресійну залежність у майбутнє на задану кількість інтервалів прогнозу (рис. 33).

Регресійні моделі застосовують для прогнозування даних які містять тренди. Якість прогнозу залежить від того, як правильно було обрано математичну формулу для

апроксимації експериментальних даних. Моделі такого типу передбачають, що виявлена функціональна залежність буде не змінною у майбутньому. Тому, такі моделі можна застосовувати для процесів в яких є сталий тренд. Якщо тренд не є сталим і може змінювати свою математичну формулу, то використання регресійних моделей може призвести до значних похибок прогнозування.

Метод ковзної середньої, метод експоненціального згладжування

З концептуальної точки зору метод ковзного середнього досить простий. Термін "ковзне середнє" використовується тому, що нове середнє обчислюється і слугує в якості прогнозу. [8].

Для ряду y_1, y_2, \dots, y_n визначається інтервал згладжування m , де $m < n$. Якщо необхідно згладити несуттєві хаотичні коливання, то за можливості інтервал згладжування обирають значним. Величину m зменшують, коли необхідно зберегти незначні коливання. Інтервал згладжування пропонується обирати непарним. Непарне значення m дозволяє отримати розраховану змінну середню у центрі інтервалу згладжування.

Далі для перших m значень часового ряду обчислюється їхнє середнє арифметичне. Воно приймається за згладжене значення рівня ряду, яке знаходиться в середині інтервалу згладжування. Потім інтервал згладжування зсувається на m значень праворуч, операції згладжування повторюються і т. д. Метод ковзного середнього застосовують, переважно, для рядів, які мають лінійний тренд

Експоненційне згладжування — це різновид методу ковзного середнього. Однак замість того, щоб при складанні прогнозу призначати однакові вагові коефіцієнти всім спостереженнями в згладжувальній послідовності значень m , метод експоненціального згладжування надає найбільші вагові коефіцієнти найостаннім спостереженням, тому що вони містять найсвіжішу інформацію про зміни які відбулися в процесі, що прогнозується.

На ефективність використання методу експоненційного згладжування, впливає вибір константи згладжування, яка в алгоритмі обчислення експоненційного згладжування позначається як α і знаходиться в діапазоні від 0 до 1. Значення α , які є близькими до одиниці надають більшу вагу останнім спостереженням і близькими до нуля — давнішим. Якщо процес, що прогнозується повільно змінюється у часі, то доцільно використовувати значення в діапазоні $0,01 \leq \alpha \leq 0,25$. Однак, коли процес швидко змінюється, слід використовувати значення в діапазоні $0,7 \leq \alpha \leq 1$, в результаті чого прогнозований ряд буде швидко реагувати на ці зміни. Найбільш відповідне значення α для конкретного прогнозу визначається емпіричним шляхом: перевіряють різні значення α і беруть те з них, яке забезпечує найменшу похибку прогнозування. [4].

Особливість даного методу полягає в тому, що в процедурі знаходження згладженого рівня використовуються тільки попередні рівні ряду, взяті з певною вагою. Причому вага спостереження стає тим меншою, чим далі вона знаходиться від рівня, для якого визначається згладжене значення.

Вихідний динамічний ряд можна записати у вигляді полінома:

$$y_t = a_0 + a_1 t + \frac{a_2}{2} t^2 + \dots + \frac{a_p}{p!} t^p + \varepsilon^t$$

Експоненційне згладжування здійснюється за формулою

$$S_t^{[k]}(y) = \alpha \sum_{i=0}^n (1-\alpha)^i S_{t-1}^{[k-1]}(y)$$

де $S_{t(\varphi)}^{[k]}$ — експоненціальна середня k -го порядку для ряду y ;
 α — параметр згладжування: $\alpha = 1 - \beta$.

Для розрахунку експоненційної і середньої використовують рекурентну формулу

$$S_t^{[k]}(y) = \alpha S_t^{[k-1]}(y) + (1-\alpha) S_{t-1}^{[k]}(y)$$

На практиці прогноз виконується за двома формулами:

Прогноз за лінійною моделлю

$$y_{t+1} = a_0 + a_1 \tau;$$

де

$$a_0 = 2S_t^{[1]} - S_t^{[2]};$$

$$a_1 = \frac{\alpha}{\beta} (S_t^{[1]} - S_t^{[2]});$$

Прогноз за квадратичною моделлю

$$y_{t+1} = a_0 + a_1 \tau + \frac{a_2}{2} \tau^2 /$$

$$a_0 = 3(S_t^{[1]} - S_t^{[2]}) + S_t^{[3]};$$

$$a_1 = \frac{\alpha}{\beta^2} [(6-5\alpha)S_t^{[1]} - 2(5-4\alpha)S_t^{[2]} + (4-3\alpha)S_t^{[3]}];$$

$$a_2 = \frac{\alpha^2}{\beta^2} [S_t^{[1]} - 2S_t^{[2]} + S_t^{[3]}]$$

Тип моделі обирають відповідно до вихідних даних. Якщо тенденція в даних близька до лінійної моделі обирають першу формулу. Для нелінійних даних обирають квадратичну модель.

Моделі експоненційного згладжування достатньо гнучкі і можуть описувати значний спектр часових рядів, які зустрічаються на практиці. Формальна процедура перевірки адекватності моделі є простою і доступною.

Але метод має ряд недоліків:

- Не існує простого і швидкого способу корекції параметрів моделі в тих випадках коли з'являються нові дані. Виникає необхідність заново підібрати всі коефіцієнти для моделі прогнозування;

- Побудова ефективної моделі експоненційного заклад жування вимагає значних витрати часу та ресурсів.

Метод статистичної екстраполяції

Екстраполяція – це визначення по ряду даних функції інших її значень поза цим рядом. Екстраполяція використовується для короткострокового прогнозування на 1 - 5 кроків уперед.

Нехай маємо ряд спостережень x_1, x_2, \dots, x_n . Ряд має бути стаціонарним. Розраховуємо для нього кореляційну функцію за формулою

$$k_x(\tau) = \frac{1}{(n-\tau)} \sum_{i=1}^{n-\tau} (x_i - \bar{x})(x_{i+\tau} - \bar{x}) \tau = 0, 1..n, \quad m=n/4$$

Метод статистичної екстраполяції дозволяє виконати прогноз маючи всього одну або дві точки передісторії. В формулу прогнозування включено коефіцієнти кореляційної функції, які в агрегованій формі дозволяють враховувати зміни у часовому ряді у минулому.

Прогноз по одній точці перед історії виконується за формулою

$$x_{t+1} = (k_x(1))/(k_x(0))(x_t - \bar{x}) + \bar{x}, \quad \text{де, } \bar{x} - \text{середнє арифметичне.}$$

Коефіцієнти кореляційних функцій обчислюється за формулами

$$\left. \begin{aligned} k_x(1) &= \frac{1}{n-1} \sum_{i=1}^{n-1} (x_i - \bar{x})(x_{i+1} - \bar{x}) \\ k_x(0) &= D_x - \text{дисперсія} \end{aligned} \right\}$$

Прогнозування за двома точками передісторії розраховується так

$$x_{t+1} = \frac{k_x(0)k_x(1) - k_x(2)k_x(1)}{k_x^2(0) - k_x^2(1)} x_t + \frac{k_x(0)k_x(2) - k_x^2(1)}{k_x^2(0) - k_x^2(1)} x_{t-1} - \bar{x} \left[\frac{k_x(1) - k_x(2)}{k_x(0) - k_x(1)} - 1 \right]$$

Метод статистичної екстраполяції дає точні результати, якщо коефіцієнти кореляційної функції процесу, який прогнозується задовольняють нерівностям

$$R(1) = \frac{k_x(1)}{k_x(0)} > 0.5, \quad R(2) = \frac{k_x(2)}{k_x(0)} > 0.4$$

Перевагою методу є достатньо прості розрахунки і невелика кількість точок передісторії для прогнозування.

Програмне забезпечення для моделювання в системах СППР.

Вільне (безкоштовне) оптимізаційне програмне забезпечення

Розглянуті вище математичні методи моделювання реалізовані в програмному забезпеченні комп'ютерів. В поточний час маємо безкоштовне або вільне програмне забезпечення, ліцензійне програмне забезпечення і програмне забезпечення, що надається вищим навчальним закладам за зниженою ціною або безкоштовно.

В таблицях подано перелік програмного забезпечення, яке може бути застосовано в системах підтримки прийняття рішень. Таблиця містить посилання на відповідне програмне забезпечення.

Назва програми	Ліцензія	Інформація
ADMB	BSD	Нелінійне методи оптимізації з використанням автоматичного диференціювання .
ALGENCAN	GPL	Fortran програми нелінійної оптимізації яка працює з програмами на AMPL, C/C++, CUTEr, Matlab, Python, Octave and R.
APMonitor	BSD	Змішане цілочислове програмування з використанням MATLAB Toolbox і Python APIs .
ASCEND	GPL	Математичне моделювання систем.
BOBYQA	LGPL	Алгоритм нелінійної оптимізації з обмеженнями. Метод Пауела Michael J. D. Powell CCPForge or here .
COBYLA	LGPL	Алгоритм пошуку мінімуму нелінійної функції з нелійними обмеженнями типу нерівностей. Алгоритми відшукує мінімум без застосування похідних Michael J. D. Powell . CCPForge or here .
CONDOR	GPL	Оптимізація нелінійної безперервної цільової функції з кількістю обмежень менше 20 .
COIN-OR SYMPHONY	Eclipse v.1	Цілочислове програмування

Назва програми	Ліцензія	Інформація
dlib	Boost	Бібліотека програм на C++ для розв'язання лінійних та нелінійних оптимізаційних задач.
EvA2	GPL	Генетичні алгоритми на мові програмування Java.
GLPK	GPL	Лінійне програмування.
IPOPT	CPL	Лінійне програмування для задач з великими обсягами змінних і обмежень.
JOptimizer	Apache License	Java бібліотека для нелінійної оптимізації.
L-BFGS	BSD	Нелінійна оптимізація за методом Ньютона для задач з великою кількістю змінних і обмежень. limited-memory quasi-Newton method .
Liger	LGPL	Розв'язок багатокритеріальних оптимізаційних задач.
LINCOA	LGPL	Програма для розв'язку задач з нелінійною функцією цілі і лінійними обмеженнями типу нерівностей.
MIDACO	BY-NC-ND	Глобальна оптимізація із використанням пакетів програм і розподіленого обчислення.
MINUIT/MINUIT2	(L)GPL	Мінімізація за методом градієнтів
MLPACK	BSD	Пакет програм який дозволяє виконувати оптимізацію, кластеризацію і здійснювати регресійний аналіз.
NEWUOA	LGPL	Метод оптимізації для цільової функції без обмежень
NLopt	LGPL, MIT	Метод глобальної та локальної оптимізації. Пошук екстремуму без використання похідних.
NOMAD	LGPL	Генетичні алгоритми оптимізації.

Назва програми	Ліцензія	Інформація
OpenMDAO	ASL	Багатопрофільний дизайн, аналіз і оптимізація систем управління написана на мові програмування Python. Розроблено NASA Glenn Research Center, за підтримки науково-дослідного центру НАСА Ленглі.
Opt4J	MIT	Оптимізація з використанням еволюційних алгоритмів на мові програмування Java
PPL	GPLv3	Програма для розв'язку цілочислових задач оптимізації.
Scilab	CeCILL	Пакет програм які можуть застосовуватися на різних операційних системах і язык програмування високого рівня орієнтований на розв'язання математичних і оптимізаційних задач.
TAO	BSD	Оптимізаційний алгоритми для розв'язку великих задач з використанням паралельних обчислень на багатьох комп'ютерах.
TOLMIN	LGPL	Програма мінімізації нелінійної диференційованої функції з лінійними обмеженнями.
UOBYQA	LGPL	Алгоритм оптимізації для задачі без обмежень , який не використовує похідних.

Ліцензійне програмне забезпечення для розв'язання задач оптимізації

- **AIMMS** — система моделювання оптимізації з можливостями графічного інтерфейсу;
- **Altair HyperStudy** — багатопрофільний дизайн експериментів, оптимізація та програмне забезпечення стохастичних досліджень;
- **AMPL** — мова моделювання для великомасштабних задач лінійної, цілочисельної і нелінійної оптимізації;
- **APMonitor** — пакет програм для великомасштабних задач лінійної, цілочисельної і нелінійної оптимізації;
- **BARON** — для глобальної оптимізації нелінійних , змішано-цілочисельних нелінійних задач;
- **CPLEX** — цілочислове лінійне і квадратичне програмування;
- **EASY-FIT** — Програма для регресійного аналізу;
- **FortMP** — цілочислове лінійне і квадратичне програмування;
- **FortSP** — стохастичне програмування;
- **Gurobi** — цілочислове, лінійне і квадратичне програмування;
- **HillStormer** — Нелдера-Міда нелінійна оптимізація з лінійними обмеженнями;

- **HEEDS MDO** — оптимізація із використанням адаптивного алгоритму;
- **IMSL Numerical Libraries Linear**, — лінійне, квадратичне нелінійне програмування із застосуванням алгоритмів оптимізації, які реалізовані в стандартних мовах програмування C, Java, C# .NET, Fortran, and Python;
- **Kimeme** — відкрита платформа для багатоцільової оптимізації та міждисциплінарної оптимізації;
- **Lingo** — лінійна, цілочислова, нелінійна та стохастична оптимізація;
- **LIONSolver** — інтегрована програма для аналізу даних, моделювання, навчання та інтелектуальної оптимізації і бізнес-аналітики;
- **modeFRONTIER** — modeFRONTIER® інтеграційна платформа для багатоцільової і мульти-дисциплінарної оптимізації, дозволяє автоматизувати процес моделювання і підтримує прийняття рішень;
- **Maple (software)** — лінійна, квадратична і нелінійна, безперервного і ціле оптимізації;
- **MATLAB** — масштабна система для математичного моделювання. Дозволяє вирішувати лінійні, нелінійні, цілочислові, стохастичні, нейромережеві задачі. **Mathematica** — Математична система програма, яка дозволяє розв'язувати освіти усі типи задачі оптимізації;
- **MIDACO** — глобальна оптимізація на основі еволюційних обчислень;
- **MISQP** — пакет для нелінійного рівняння змішаного цілочисельного програмування;
- **MOSEK** — лінійна, квадратична, нелінійна, безперервна, цілочислова оптимізація
- **NAG** — лінійна, квадратична, нелінійна оптимізація, регресійне моделювання
- **Nexus** — міждисциплінарний і багатоцільова оптимізація. Nexus забезпечує пряму інтеграцію з багатьма пакетами CAD ;
- **NMath** — лінійна, цілочислова, нелінійна оптимізація;
- **optiSLang** — програмні рішення для на основі аналізу чутливості, оптимізації та оцінки робастності;
- **OptiStruct** — технологія CAE для концептуального проектування і синтезу структурної оптимізації;
- **PHX ModelCenter** — графічна середовище для автоматизації, інтеграції та оптимізації;
- **PottersWheel** — оцінка параметрів в звичайних диференціальних рівнянь;
- **pSeven** — програмна платформа для автоматизації моделювання та аналізу, оптимізації та міждисциплінарного аналізу даних. Працює за допомогою макросів;
- **SAS/OR** — Набор програм які дозволяють розв'язувати лінійні, нелінійні, цілочислові, комбінаторні оптимізаційні задачі;
- **SmartDO** — багатопрофільна глобальна оптимізація, спеціалізується на автоматизованого проектуванні(CAE);
- **SNOPT** — для масштабних завдань оптимізації;
- **TOMLAB** — підтримує глобальну оптимізацію, цілочислове програмування, всі види метода найменших квадратів, лінійне, квадратичне і безумовне програмування;
- **VisSim** — візуальний блок-схемна мова моделювання та оптимізації динамічних систем;
- **XPRESS** — цілочислове, лінійне, квадратичне, всі види нелінійного програмування;
- **XTREME** — багатопрофільна глобальна оптимізація на основі генетичних алгоритмів прискорених за допомогою нейронних мереж, мульти-критеріальні завдання (Парето), графічний користувальницький інтерфейс.

Безкоштовне програмне забезпечення для університетів

- **AIMMS** — оптимізація, безкоштовно для академічного використання;
- **AMPL** — оптимізація, безкоштовно для академічного використання;
- **ASTOS** — оптимізація, безкоштовно для академічного використання;
- **CPLEX** — оптимізація, безкоштовно для академічного використання;
- **Galahad library** — оптимізація, безкоштовно для академічного використання;

- **GUROBI** — оптимізація, безкоштовно для академічного використання;
- **LIONsolver** — оптимізація, безкоштовно для академічного використання;
- **MINTO** — цілочислове програмування, програма безкоштовна для персонального використання;
- **XPRESS** — оптимізація, безкоштовно для академічного використання.

Висновки

В системах підтримки прийняття рішень застосовується велика кількість економіко-математичних моделей. Важливим етапом в розробці моделей є правильний вибір критеріїв оптимізації. Існує ряд методів лінійного, цілочислового, нелінійного програмування які дозволяють розв'язати широкий спектр задач. Нейромережеві моделі дозволяють побудувати системи зі штучним інтелектом які можуть навчатися. Апарат теорії ігор дозволяє змодельовувати конфліктні ситуації, конкуренцію на ринку, знайти оптимальне рішення для розв'язання таких проблем.

В системах підтримки прийняття рішення широко застосовуються методи прогнозування. Найбільш ефективними з точки зору точності прогнозу є методи короткострокового і середньострокового прогнозування. Для прогнозування часових рядів у яких спостерігається тренд є доцільним використовувати прогнозування із використанням регресійних моделей. Для стаціонарних процесів доцільно застосовувати метод статистичної екстраполяції або ковзного середнього чи експоненційного згладжування. Для приведення нестационарних процесів до стаціонарного вигляду існує ряд ефективних методів.

Ефективне прогнозування передбачає наявність у користувача СППР феоахового володіння математичним апаратом методів прогнозу та досвіду застосування моделей.

Контрольні запитання

1. Назвіть основні етапи побудови економіко математичної моделі .
2. Як треба вибирати критерій оптимізації в умовах, коли ми маємо декілька критеріїв які є у протиріччі один з одним?
3. Які типи задач можна вирішувати з лінійними і цілочисловими моделями?
4. Поясніть методи розв'язання нелінійних задач оптимізації.
5. Які моделі відносяться до моделей теорії ігор?
6. Що таке гра з сідловою точкою?
7. Які типи задач можна розв'язати із застосуванням нейронних мереж?
8. Поясніть принцип роботи нейронної мережі.
9. Як правильно підготувати дані для прогнозування?
10. Чому необхідно приводити вихідні дані до стаціонарного вигляду?
11. Коли доцільно використовувати прогнозування із застосуванням регресійних моделей?
12. Які переваги і недоліки мають методи ковзного середнього і експоненціального ковзного середнього?
13. Коли доцільно використовувати для прогнозування метод статистичної екстраполяції?

ТЕМА 5. Методи обробки даних в системах підтримки прийняття рішень

Online Analytical Processing (OLAP) он-лайніві аналітична обробка в системах підтримки прийняття рішень

Переглянемо методи он-лайніві обробки інформації в сучасних системах підтримки прийняття рішень.

Он-лайніві аналітична обробка інформації дозволяє отримувати дані із баз та сховищ даних, обробляти їх за математичними алгоритмами і представляти результати у вигляді аналітичних звітів.

Існують три основних методи он-лайніві обробки інформації (OLAP), які регулярно використовуються аналітиками, до них відносять:

- Метод розбивки на сегменти;
- Метод обертання;
- Метод консолідації та деталізації.

Метод розбивки на сегменти визначається як стратегія розбивки на елементи, візуалізацію і осмислення даних які отримані з баз даних. Користувач системи підтримки прийняття рішень розбиває великий масив даних на сегменти які потім розбиваються на невеликі частки. Процес повторюється до тих пір, поки ми отримуємо рівень деталізації необхідний для аналізу .

На рис.34 показано обробку за цим принципом. В базі даних ми маємо інформацію по продажу товарів по роках і по містах. Далі ми робимо запит: “Проаналізувати інформацію продажів товару № 1 протягом 2001 - 2007 років у місті Шостка”.

Продажі

Товар №1	9	11	26	18	14	41	39
Товар №2	13	7	33	14	26	16	14
Товар №3	42	10	15	9	25	21	25
Товар №4	22	15	66	38	16	51	55
Товар №5	10	11	12	15	22	24	28
Товар №6	14	22	32	27	41	15	33
Товар №7	16	33	29	9	13	12	7
Товар №8	12	8	26	15	32	19	11
	2001	2002	2003	2004	2005	2006	2007

Час

Продажі

Товар №1	12	8	26	15	32	19	11
	2001	2002	2003	2004	2005	2006	2007

Час

Рисунок 34. Метод розбивки на сегменти

За цим методом спочатку із бази даних вирізається великий сегмент-прошарок з даними продажів у Шостці усіх товарів протягом 2001 - 2007 років. Далі робимо наступний крок і з отриманого прошарку вирізаємо інформацію про продаж товару №1 протягом вказаного проміжку часу.

Обертання - перетворення стовпців таблиці на рядки і навпаки. Обертання дозволяє аналітику обертати куб з даними у просторі так, щоби бачити його різні сторони. Наприклад міста могли бути показані вертикально або горизонтально відносно товарів для заданого проміжку часу.



Рисунок 35. Метод обертання даних

Консолідація та деталізація — операції, що визначають перехід від детального представлення даних до агрегованого (вгору) і навпаки — від агрегованого до детального (вниз).

До даних, як були оброблені за методами розбивки на сегменти, обертання, консолідації та деталізації аналітики у подальшому застосовують, розглянуті вище, математичні моделі для виявлення прихованих залежностей або оптимізації управлінських рішень.

Дейтамайнінг в СППР

Дейтамайнінг (Data mining) видобування даних — це процес виявлення в великих масивах даних раніше невідомих, нетривіальних, практично корисних і доступних для інтерпретації знань, необхідних для прийняття рішень у різних галузях людської діяльності.

Інструментальні засоби добування даних використовують різ-номанітні методи, включаючи доказову аргументацію, візуалізацію даних, методи нечіткої логіки й аналізу, нейромережі та інші.

Доказову аргументацію (міркування за прецеден-тами) застосовують для пошуку записів, подібних до якогось певного запису чи низки записів. Ці інструментальні засоби дають змогу користувачеві конкретизувати ознаки подібності підібраних записів.

За допомогою візуалізації даних можна легко і швидко оглядати графічні відображення інформації в різних аспектах (ракурсах).

Відомі п'ять загальних типів інформації, що можуть бути одержані засобами дейтамайнінгу:

- *класифікація*: дозволяє робити висновок щодо визначення характеристик конкретної групи (наприклад, споживачі, які були втрачені через дії конкурентів);
- *кластеризація*: ототожнює групи елементів, які мають показники близькі до параметру за яким їх об'єднують у групу (кластеризація відрізняється від класифікації, бо не вимагається наперед визначена характеристика);
- *асоціація*: ідентифікує зв'язки або відношення між подіями, які відбувались у минулому (наприклад, вподобання клієнтів магазину, які відвідували його протягом року).
- *упорядковування*: подібно асоціації, крім того, установлюється зв'язок в часовому вимірі (наприклад, повторний візит до супермаркету або фінансове планування виготовлення продукту);
- *прогнозування*: оцінює майбутні значення параметрів і показників економічних процесів, які обчислюються з використанням визначених тенденцій, трендів в передісторії цих процесів із використанням математичних методів.

Методи добування даних в СППР

Методи добування даних, класифікація і технології їх використання показано на рис. 36. Розглянемо складові дерева методів дейтамайнінгу. На першому рівні дерева рішень визначаються, чи будуть видобуті дані збережені в системі або будуть відфільтровані для подальшої обробки.

Збереження даних

Збережені дані знаходяться в системі далі аналізуються із застосуванням статистичних і математичних методів. Аналіз виконується за допомогою зіставлення даних з взірцем (шаблоном). Відповідно до того, співпадають чи не співпадають дані з шаблоном, приймають рішення щодо класифікації даних та їх використання. Широко застосовується статистичний метод кластерного аналізу

Основна мета *Кластерного аналізу* - виділити в початкових багатовимірних даних такі однорідні підмножини, щоб об'єкти всередині груп були схожі у відомому зна-

ченні один на одного, а об'єкти з різних груп не схожі. Під "схожими" розуміється близькість об'єктів в багатовимірному просторі ознак, і тоді задача зводиться до виділення в цій множині об'єктів, тих які і вважаються однорідними групами. В кластерному аналізі використовуються різні алгоритми але найбільш поширеним серед них є метод К-середніх([K-Means](#)). Цей метод реалізовано в пакеті [Statistica](#). Суть метода у наступному. Маємо велику вибірку, яку досліджуємо. Робимо гіпотезу, що в цій вибірці є m груп споріднених даних. Далі обчислюються середні значення для кожної із m груп. Наступним кроком, вибірка аналізується і дані близькі до відповідних середніх, згідно критерію, об'єднуються в групи. В результаті отримуємо m масивів чисел, в кожному з яких дані незначно відрізняються від середнього по групі.

Метод "найближчого сусіда" ("[nearest neighbor](#)") - добре відомий приклад підходу, який ґрунтується на збереженні даних. При цьому набір даних тримається в пам'яті для порівняння з новими елементами даних. Коли аналізується новий набір даних то знаходяться "відхилення" між ним і подібними наборами даних, і найбільш подібний (або найближче сусідній) ідентифікується.

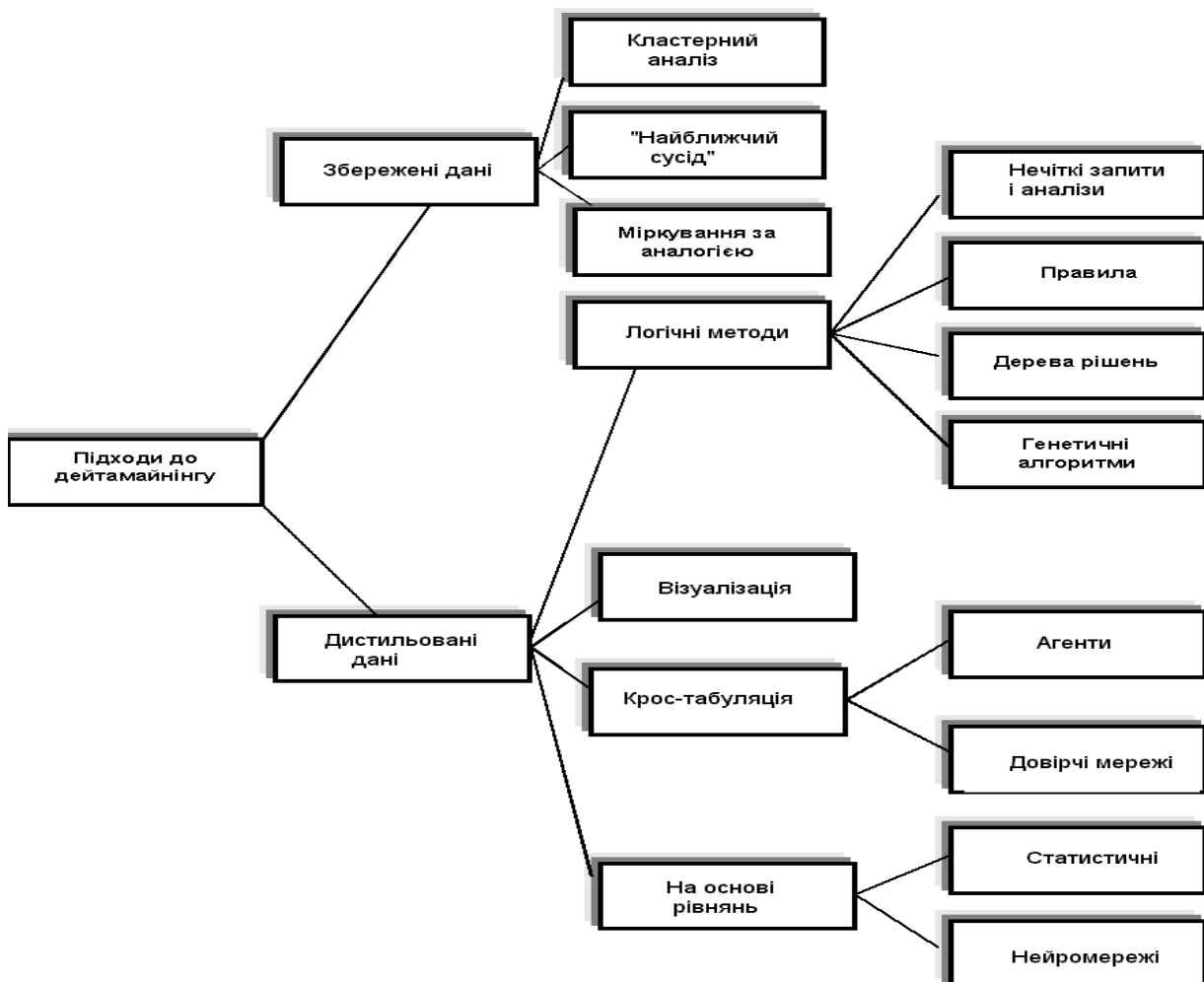


Рисунок 36. Методи дейтамайнінгу

Наприклад, якщо розглядається новий споживач банківських послуг, то атрибути пропонованого клієнта порівнюються з всіма існуючими банківськими клієнтами

(наприклад, вік і прибуток перспективного порівняно з віком і прибутком існуючих клієнтів). Потім множина найближчих “сусідів” для перспективного клієнта вибирається на підставі найближчого значення прибутку, віку тощо. Наступне порівняння виконується, для того щоб вибрати серед послуг банку такі, що найбільше відповідають інтересам клієнта, з урахуванням тих продуктів, які використовуються найближчими до нього К сусідами.

Міркування за аналогією (case-based reasoning - CBR) або міркування за прецедентами (аналогічними випадками). Даний метод має дуже просту ідею – щоб зробити прогноз на майбутнє або вибрати правильне рішення, система підтримки прийняття рішень знаходить близькі аналогії в минулому, при різних ситуаціях і відбирає ту відповідь, яка за схожими ознаками була правильною. Інструментальні засоби міркування за прецедентами знаходять записи в базі даних, які подібні до описаних записів. Ця категорія інструментальних засобів також має назву *міркування на основі пам'яті (memory-based reasoning)*.

Програмне забезпечення CBR виміряє “відхилення (дистанцію)”, одного запису по відношенню до інших записів і згрупує записи за подібністю. Ця методика мала успіх при аналізованні зв'язків в текстах вільного формату.

Ефект від застосування цих методів залежить від правильності вибору типових даних з якими зіставляються інші дані.

Дистиляція даних

Дистиляція даних передбачає вибір взірця або шаблону з набору даних, і його використання для фільтрації даних із застосуванням чотирьох наступних методів: *логічні методи, візуалізація, крос-табуляції, методи і на основі рівнянь*.

Логічні методи (підходи). Методи логічного підходу в системах дейтамайнінгу можуть бути розділені на чотири групи: нечіткі запити і аналізи, правила, дерева рішень, генетичні алгоритми.

Нечіткі запити і аналізи (Fuzzy Query and Analysis). Ця категорія інструментальних засобів дейтамайнінгу ґрунтується на відгалуженні математики, що називається нечіткою логікою (fuzzy logic), або логікою невпевненості і розмитості (fuzziness). Вона надає рамку для виявлення розмитості і ранжування результатів запитів.

Правила. Розглянемо основні різновиди правил та особливості їх застосування в дейтамайнінгу.

Логічні зв'язки між елементами ділових процесів звичайно представляють правила. Найпростіші типи правил виражаються *умовними* або *афінними (асоціативними)* зв'язками (відношеннями).

Умовне правило є твердження типу: *Якщо умова 1 -- Тоді умова 2.*

Афінна логіка є чіткою як в термінах мови вираження, так і в термінах структури даних, які використовуються. *Афінний аналіз (або асоціативний аналіз)* є пошуком взірців і умов, які описують як різні данні “групуються разом” або “зіставляються між собою”. Афінне правило має форму: *Коли елемент (позиція) 1 тоді також елемент (позиція) 2.* Приклад цього є “Коли фарба, тоді також пензель фарби”. Система

афінного аналізу в СППР застосовує таблицю транзакцій, щоб ідентифікувати схожі елементи. Номер транзакції використовується, щоб створити групу схожих елементів, які групуються. Правила *Афінної* логіки широко застосовується для багатовимірних систем OLAP.

Правила індукції. Правила індукції — це процес перегляду набору даних за допомогою логічного мислення для створення взірців. Правила індукції можуть бути запрограмовані і автоматично застосовуватися на комп'ютері в системі СППР для отримання рішення. В системі реалізується процес подібній до того, як людина-аналітик проводить дослідницький аналіз.

Дерева Рішень. Метод дерев рішень (decision trees) є одним з найбільш популярних методів вирішення завдань класифікації і аналізу. Іноді цей метод також називають деревами вирішальних правил, деревами класифікації та регресії. Як видно з останньої назви, за допомогою даного методу вирішуються задачі класифікації і прогнозування. Якщо залежна, тобто цільова змінна набуває дискретних значень, за допомогою методу дерева рішень вирішується задачі класифікації. Якщо ж залежна змінна набуває безперервних значень, то дерево рішень встановлює залежність цієї змінної від незалежних змінних, тобто вирішує задачу чисельного прогнозування. Отже, дерева рішень – це спосіб представлення правил в ієрархічній, послідовній структурі, де кожному об'єкту відповідає єдиний вузол, що дає рішення. Під правилом розуміється логічна конструкція, що представлена у вигляді «якщо ... то ...». На рис. 37 наведено приклад дерева рішень, завдання якого – відповісти на питання: «Чи йти на першу пару?» Щоб вирішити задачу, тобто ухвалити рішення, чи йти на першу пару, слід віднести поточну ситуацію до одного з відомих класів (в даному випадку – «йти» або «не йти»). Для цього потрібно відповісти на ряд питань, які знаходяться у вузлах цього дерева, починаючи з його кореня.

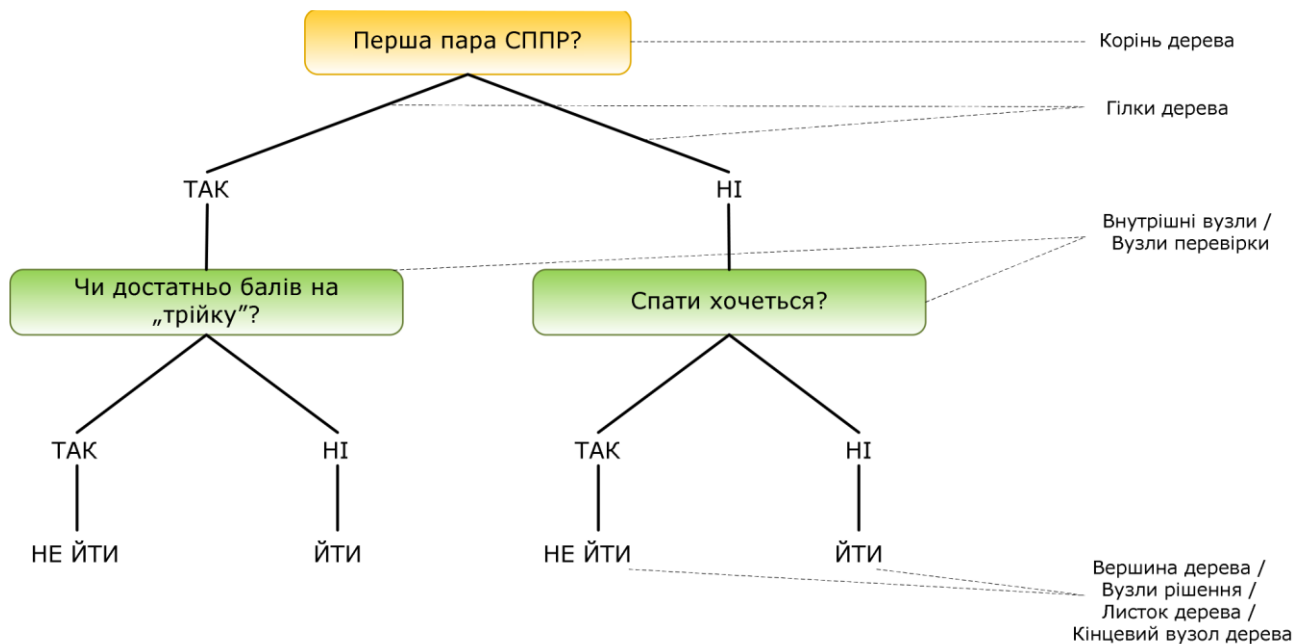


Рисунок 37 Приклад дерева рішень

Перший вузол нашого дерева «Перша пара СППР?» є вузлом перевірки, тобто умовою. При позитивній відповіді на питання здійснюється перехід до лівої частини дерева – лівої гілки, при негативному – до правої частини дерева (правої гілки). Таким чином, внутрішній вузол дерева є вузлом перевірки певної умови. Далі йде наступне питання і так далі, поки не буде досягнутий кінцевий вузол дерева, що є вузлом рішення, – листок дерева. Для нашого дерева існує два типи листків: «йти» і «не йти» на першу пару. В результаті проходження від кореня дерева до його вершини (листоків) вирішується завдання класифікації, тобто вибирається один з класів – «йти» і «не йти» на першу пару. В розглянутому прикладі вирішується завдання бінарної класифікації, тобто створюється дихотомічна класифікаційна модель. Приклад демонструє роботу так званих бінарних дерев. У вузлах бінарних дерев розгалуження може вестися тільки в двох напрямках, тобто існує можливість тільки двох відповідей на поставлене питання («так або ні»). Бінарні дерева є найпростішим прикладом дерев рішень. В решті випадків відповідей і, відповідно, гілок дерева, що виходять з його внутрішнього вузла, може бути більше двох. Розглянемо складніший приклад. База даних, на основі якої повинна здійснюватися класифікація, містить наступні дані про клієнтів банку, що одночасно є її атрибутами: вік, наявність нерухомості, освіта, середньомісячний дохід, чи повернув клієнт вчасно кредит. Завдання полягає в тому, щоб на підставі перерахованих вище даних (окрім останнього атрибуту) визначити, чи варто видавати кредит новому клієнтові. Вирішення даного завдання вирішується в 2 етапи: побудова моделі та її використання. На етапі побудови моделі будується дерево рішень або створюється набір певних правил. На етапі використання моделі побудоване дерево або шлях від його кореня до однієї з вершин використовується для відповіді на поставлене питання «Чи видавати кредит?». На рис.38. приведений приклад дерева класифікації, за допомогою якого вирішується задача «Видавати кредит клієнтові?».

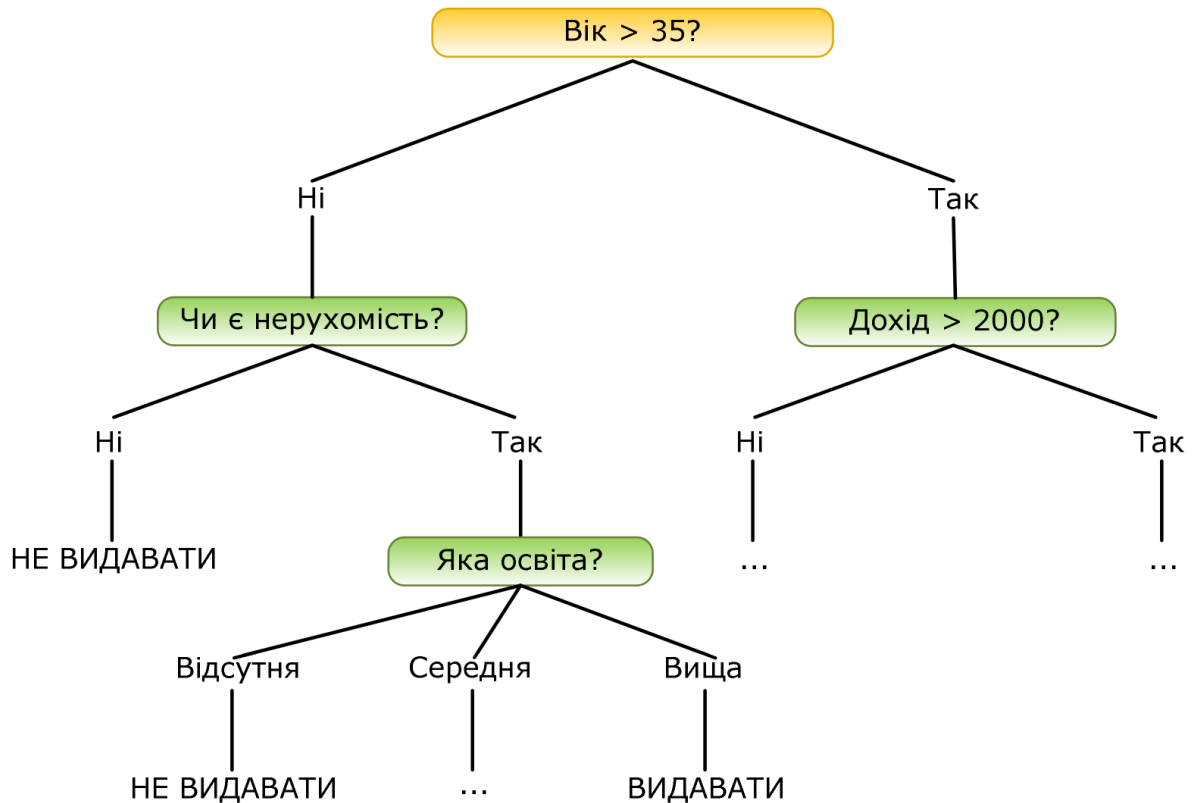


Рисунок 38. Дерево рішень для надання кредиту

Як бачимо, внутрішні вузли дерева (вік, наявність нерухомості, дохід та освіта) є атрибутами описаної вище бази даних. Ці атрибути називають атрибутами розщеплення. Кожна гілка дерева, що йде від внутрішнього вузла, відмічена предикатом розщеплення. Останній може відноситися лише до одного атрибуту розщеплення даного вузла. Характерна особливість предикатів розщеплювання: кожен запис використовує унікальний шлях від кореня дерева тільки до одного вузла-рішення. Об'єднана інформація про атрибути розщеплення і предикати розщеплення у вузлі називається критерієм розщеплення (splitting criterion).

На рис. 38. зображене одне з можливих дерев рішень для даної бази даних. Наприклад, критерій розщеплення «Яка освіта?», міг би мати два предикати розщеплення і виглядати інакше: освіта «вища» і «не вища». Тоді дерево рішень мало б інший вигляд. Таким чином, для даного завдання (як і для будь-якої іншої) може бути побудоване декілька дерев рішень різної якості, з різною прогнозуючою точністю. Якість побудованого дерева рішень дуже залежить від правильного вибору критерію розщеплення. Метод дерев рішень часто називають «наївним» підходом. Але завдяки цілому ряду переваг, даний метод є одним з найбільш популярних для вирішення завдань класифікації.

Процес конструювання дерева рішень.

Алгоритми конструювання дерев рішень складаються з двох етапів:

1. Побудови або створення дерева – на даному етапі вирішуються питання вибору критерію розщеплювання і зупинки навчання (якщо це передбачено алгоритмом).
2. Скорочення дерева – дозволяє відсікти деякі гілки дерева.

Розглянемо особливості кожного з етапів. Процес створення дерева відбувається зверху вниз, тобто є низхідним. В ході процесу алгоритм повинен знайти такий критерій розщеплення, щоб розбити множину на підмножини, які б асоціювалися з даним вузлом перевірки. Кожен вузол перевірки повинен бути помічений певним атрибутом. Існує правило вибору атрибуту: він повинен розбивати початкову множина даних так, щоб об'єкти підмножин, що отримуються в результаті розбиття, були представниками одного класу або ж були максимально наближені до такого розбиття. Остання фраза означає, що кількість об'єктів з інших класів, так званих «домішок», в кожному класі повинно бути як найменше. Існують різні критерії розщеплювання.

Найбільш відомі:

- міра ентропії – в деяких методах для вибору атрибуту розщеплення використовується так звана міра інформативності підпросторів атрибутів, яка ґрунтується на ентропійному підході і відома під назвою «міра інформаційного виграшу» (information gain measure) або міра ентропії;
- індекс Джині (Gini) – критерій розщеплення, запропонований Брейманом (Breiman) та реалізований в алгоритмі CART. За допомогою цього індексу атрибут вибирається на підставі відстаней між розподіленням класів. Якщо дана множина T , що включає приклади об'єктів з n класів, то індекс Джині, тобто $gini(T)$, визначається по формулі:

$$gini(T) = 1 - \sum_{j=1}^n p_j^2$$

де T – поточний вузол дерева, p_j – ймовірність класу j у вузлі T , n – кількість класів. Важливим моментом в процесі створення дерева є зупинка процесу навчання дерева (фактично це є завершення процесу розгалуження дерева). В процесі побудови дерева з метою недопущення неконтрольованого збільшення його розмірів використовують спеціальні процедури, які дозволяють створювати оптимальні дерева, так звані дерева «відповідних розмірів». Який же розмір дерева може вважатися оптимальним? Дерево повинне бути достатнє складним, щоб враховувати інформацію з досліджуваного набору даних, але одночасно воно повинне бути достатнє простим. Іншими словами, дерево повинне використовувати інформацію, що покращує якість моделі, та ігнорувати ту інформацію, яка її погіршує.

Переваги алгоритму дерева рішень:

1. Інтуїтивна зрозумілість та наочність дерева рішень – результат роботи алгоритмів конструювання дерев рішень, на відміну від нейронних мереж, що є «чорними ящиками», легко інтерпретується користувачем. Ця властивість дерев рішень не тільки важлива при віднесенні до певного класу нового об'єкту, але і при інтерпретації моделі в цілому. Дерево рішень дозволяє зрозуміти і пояснити, чому конкретний об'єкт відноситься до того чи іншого класу.

2. Довільні вхідні атрибути – алгоритм конструювання дерева рішень не вимагає від користувача вибору вхідних атрибутів. На вхід алгоритму можна подавати всі існуючі атрибути, алгоритм сам вибере найбільш значущі серед них і лише вони будуть використані для побудови дерева.

3. Точність моделей виявляється достатньо високою.

4. Масштабованість алгоритму – означає, що із зростанням числа атрибутів або записів бази даних, час, що витрачається на побудову дерева рішень, зростає лінійно.

5. Обробка пропущених значень – більшість алгоритмів конструювання дерев вирішень мають можливість спеціальної обробки пропущених значень.

6. Універсальність щодо типів даних, які оброблюються – дерева рішень працюють і з числовими, і з категоріальними типами даних.

7. Непараметральність моделей дерев рішень – дерева рішень здатні вирішувати такі завдання, в яких відсутня апріорна інформація про вид залежності між досліджуваними даними, тобто аналітику не потрібно знати вид моделі, характеру розподілу даних тощо.

Генетичні алгоритми — це тип алгоритмів, які моделюють механізмами еволюції живої природи, і застосовуються, головню, до задач оптимізації. Ідея генетичних полягає в моделюванні еволюційного процесу створення, модифікації і відбору кращих розв'язків проблеми.

При описі генетичних алгоритмів використовуються визначення, запозичені з генетики. Наприклад, мова йде про популяцію особин, а в якості базових понять застосовуються ген, хромосома, генотип, фенотип, алель. Також використовуються відповідні цим термінам визначення з технічного лексикону, зокрема, ланцюг, двійкова послідовність, структура.

Популяція — це кінцева множина особин.

Особини, що входять в популяцію, у генетичних алгоритмах представляються хромосомами з закодованими в них множинами параметрів задачі, тобто рішень, які інакше називаються точками в просторі пошуку (search points). У деяких роботах особини називаються організмами.

Хромосоми (інші назви — ланцюжки або кодові послідовності) — це впорядковані послідовності генів.

Ген (який також називається властивістю, знаком чи детектором) — це атомарний елемент генотипу, зокрема, хромосоми.

Генотип або структура — це набір хромосом даної особини. Отже, особинами популяції можуть бути генотипи або одиничні хромосоми (в досить поширеному випадку, коли генотип складається з однієї хромосоми).

Фенотип — це набір значень, які відповідає даному генотипу, тобто декодована структура або безліч параметрів задачі (розв'язок, точка простору пошуку).

Алель — це значення конкретного гена, також визначається як значення властивості або варіант властивості.

Локус чи позиція вказує місце розміщення даного гена в хромосомі (ланцюжку). Множина позицій генів — це локи.

Дуже важливим поняттям у генетичних алгоритмах вважається **функція пристосованості**, яка інакше називається функцією оцінки. Вона представляє міру пристосованості даної особини в популяції. Ця функція відіграє найважливішу роль, оскільки дозволяє оцінити ступінь пристосованості конкретних особин у популяції і вибрати з них найбільш пристосовані (тобто мають найбільші значення функції пристосованості) відповідно з еволюційним принципом виживання «найсильніших» (які найкраще пристосувалися).

Функція пристосованості також отримала свою назву безпосередньо із генетики. Вона надає сильний вплив на функціонування генетичних алгоритмів і повинна мати точне і коректне визначення. У задачах оптимізації функція пристосованості, як правило, оптимізується (точніше кажучи, максимізується) і називається цільовою функцією.

На кожній ітерації генетичного алгоритму пристосованість кожної особини даної популяції оцінюється за допомогою функції пристосованості, і на цій основі створюється наступна популяція особин, що складають безліч потенційних рішень проблеми, наприклад, задачі оптимізації. Чергова популяція в генетичному алгоритмі називається поколінням, а до новостворюваної популяції особин застосовується термін «нове покоління» або «покоління нащадків».

Основний (класичний) генетичний алгоритм складається з наступних кроків:

1. ініціалізація, або вибір вихідної популяції хромосом;
2. оцінка пристосованості хромосом в популяції;
3. перевірка умови зупинки алгоритму;
4. селекція хромосом;
5. застосування генетичних операторів;
6. формування нової популяції;
7. вибір «найкращої» хромосоми.

Блок — схема основного генетичного алгоритму зображена на рис. 39..

Ініціалізація, тобто формування вихідної популяції, полягає у випадковому виборі заданої кількості хромосом (особин), що представляються двійковими послідовностями фіксованої довжини.

Оцінювання пристосованості хромосом в популяції полягає в розрахунку функції пристосованості для кожної хромосоми цієї популяції. Чим більше значення цієї функції, тим вище «якість» хромосоми. Форма функції пристосованості залежить від характеру розв'язуваної задачі. Передбачається, що функція пристосованості завжди приймає невід'ємні значення і, крім того, що для вирішення оптимізаційної задачі потрібно

максимізувати цю функцію. Якщо вихідна форма функції пристосованості не задовольняє цим умовам, то виконується відповідне перетворення (наприклад, завдання мінімізації функції можна легко звести до задачі максимізації).

Перевірка умови зупинки алгоритму. Визначення умови зупинки генетичного алгоритму залежить від його конкретного застосування. У оптимізаційних задачах, якщо відомо максимальне (або мінімальне) значення функції пристосованості, то зупинка алгоритму може відбутися після досягнення очікуваного оптимального значення, можливо — з заданою точністю.

Зупинка алгоритму також може статися у разі, коли його виконання не приводить до поліпшення вже досягнутого значення. Алгоритм може бути зупинений після закінчення певного часу виконання або після виконання заданої кількості ітерацій. Якщо умова зупинки виконана, то проводиться перехід до завершального етапу вибору «найкращої» хромосоми. В іншому випадку на наступному кроці виконується селекція.

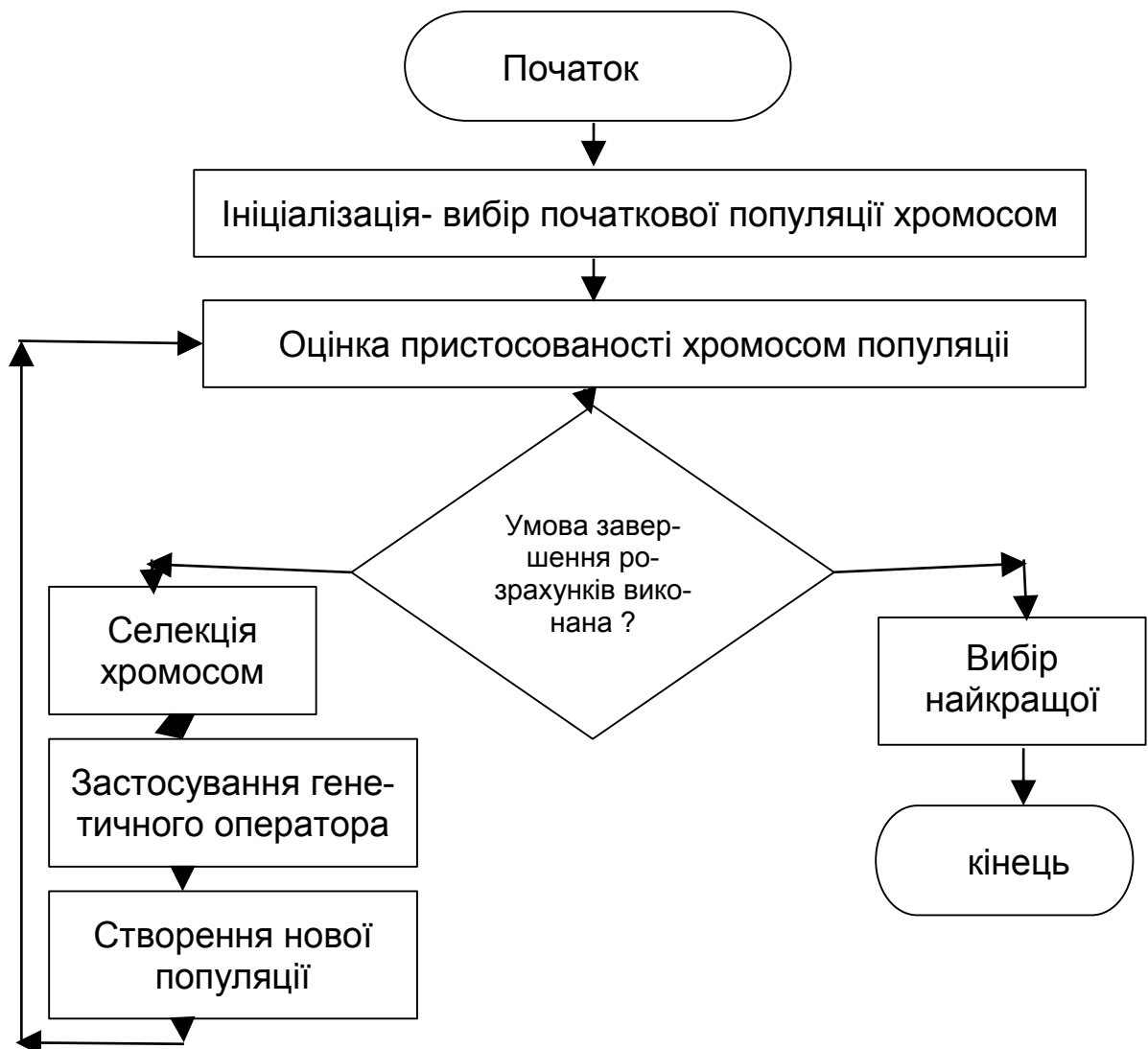


Рис. 39. Блок-схема генетичного алгоритму

Селекція хромосом полягає у виборі (по розрахованим на другому етапі значеннями функції пристосованості) тих хромосом, які братимуть участь у створенні нащадків для наступної популяції, тобто для чергового покоління. Такий вибір здійснюється згідно з принципом природного добору, за яким найбільші шанси на участь у створенні нових особин мають хромосоми з найбільшими значеннями функції пристосованості.

Існують різні методи селекції. Найбільш популярним вважається так званий метод рулетки, який свою назву отримав за аналогією з відомою азартною грою. Кожній хромосомі може бути зіставлений сектор колеса рулетки, величина якого встановлюється пропорційною значенню функції пристосованості даної хромосоми. Тому чим більше значення функції пристосованості, тим більше сектор на колесі рулетки.

Селекція хромосоми може бути представлена як результат повороту колеса рулетки, оскільки хромосома «яка виграла» (тобто обрана) відноситься до сектору цього колеса, що випав.

Очевидно, що чим більше сектор, тим більше вірогідність «перемоги» відповідної хромосоми. Тому ймовірність вибору даної хромосоми виявляється пропорційною значенню її функції пристосованості

Все коло рулетки відповідає сумі значень функції пристосованості всіх хромосом розглянутої популяції. Кожній хромосомі, що позначається ch_i для $i = 1, 2, \dots, N$ (де N позначає чисельність популяції) відповідає сектор кола $V(ch_i)$, виражений у відсотках згідно з формулою

$$V(ch_i) = P_s(ch_i)100\%$$

$$P_s(ch_i) = \frac{F(ch_i)}{\sum_{i=1}^n F(ch_i)}$$

Де

причому

$F(ch_i)$ — значення функції пристосованості хромосоми

$P_s(ch_i)$ — вірогідність селекції хромосоми (ch_i).

В результаті процесу селекції створюється батьківська популяція, також звана батьківським пулом з чисельністю N , що дорівнює чисельності поточної популяції. Застосування генетичних операторів до хромосом, відібраних за допомогою селекції, призводить до формування нової популяції нащадків від створеної на попередньому кроці батьківської популяції.

Приклад застосування генетичного алгоритму

Розглянемо спрощений приклад, що складається в знаходженні хромосоми з максимальною кількістю одиниць. Припустимо, що хромосоми складаються з 12 генів,

а популяція налічує 8 хромосом. Зрозуміло, що найкращою буде хромосома, що складається з 12 одиниць. Подивимось, як протікає процес вирішення цієї вельми тривіальної задачі за допомогою генетичного алгоритму.

Ініціалізація, або вибір вихідної популяції хромосом. Необхідно випадковим чином згенерувати 8 двійкових послідовностей довжиною 12 бітів. Це можна досягти, наприклад, підкиданням монети (96 разів, при випаданні «орла» приписується значення 1, а у разі «решки» — 0). Таким чином можна сформувати вихідну популяцію

ch1 = [111001100101] ch5 = [010001100100]

ch2 = [001100111010] ch6 = [010011000101]

ch3 = [011101110011] ch7 = [101011011011]

ch4 = [001000101000] ch8 = [000010111100]

Оцінка пристосованості хромосом в популяції. У спрощеному прикладі, що розглядається, вирішується задача знаходження такої хромосоми, яка містить найбільшу кількість одиниць. Тому функція пристосованості визначає кількість одиниць у хромосомі. Позначимо функцію пристосованості символом F . Тоді її значення для кожної хромосоми з вихідної популяції будуть такі:

$F(\text{ch1}) = 7$ $F(\text{ch5}) = 4$

$F(\text{ch2}) = 6$ $F(\text{ch6}) = 5$

$F(\text{ch3}) = 8$ $F(\text{ch7}) = 8$

$F(\text{ch4}) = 3$ $F(\text{ch8}) = 5$

Хромосоми ch3 і ch7 характеризуються найбільшими значеннями функції приналежності. У цій популяції вони вважаються найкращими кандидатами на рішення задачі. Якщо згідно з блок-схемою генетичного алгоритму (рис. 39) умова зупинки алгоритму не виконується, то на наступному кроці проводиться селекція хромосом з поточної популяції.

Селекція хромосом. Селекція проводиться методом рулетки. На підставі формул розрахунку секторів колеса для кожної з 8 хромосом поточної популяції (у нашому випадку — вихідної популяції, для якої $N = 8$) отримуємо сектори колеса рулетки, виражені у відсотках (рис. 40)

$v(\text{ch1}) = 15,22$ $v(\text{ch2}) = 13,04$ $v(\text{ch3}) = 17,39$ $v(\text{ch4}) = 6,52$

$v(\text{ch5}) = 8,70$ $v(\text{ch6}) = 10,87$ $v(\text{ch7}) = 17,39$ $v(\text{ch8}) = 10,87$

Розіграш за допомогою колеса рулетки зводиться до випадкового вибору числа з інтервалу $[0, 100]$, що вказує на відповідний сектор на колесі, тобто на конкретну хромосому. Припустимо, що розіграні наступні 8 чисел:

79 44 9 74 44 86 48 23

Це означає вибір хромосом

ch7 ch3 ch1 ch7 ch3 ch7 ch4 ch2

Як видно, хромосома ch7 була обрана тричі, а хромосома ch3 — двічі. Зауважимо, що саме ці хромосоми мають найбільше значення функції пристосованості. Проте обрана й хромосома ch4 з найменшим значенням функції пристосованості. Всі вибрані таким чином хромосоми включаються в так званий батьківський пул.

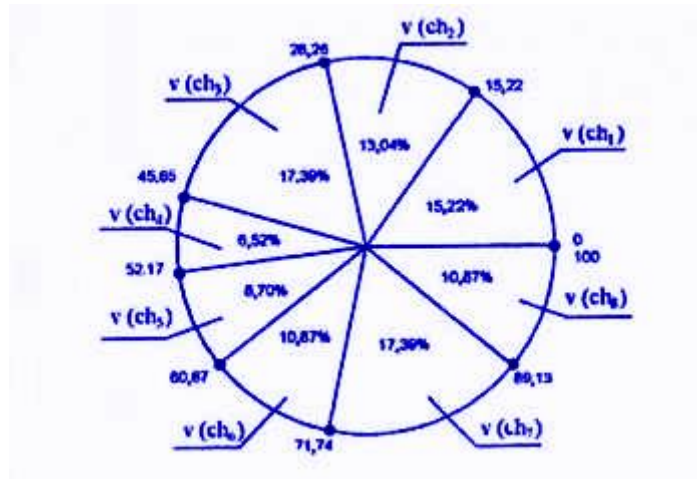


Рис. 40. Колесо рулетки для селекції в прикладі.

Застосування генетичних операторів. Припустимо, що ні одна з відібраних у процесі селекції хромосом не піддається мутації, і всі вони складають популяцію хромосом, призначених для схрещування. Це означає, що ймовірність схрещування $P_c = 1$, а ймовірність мутації $p_m = 0$. Припустимо, що з цих хромосом випадковим чином сформовані пари батьків

ch2 и ch7 ch1 и ch7 ch3 и ch4 ch3 и ch7

Для першої пари випадковим чином вибрана точка схрещування $l_k = 4$, для другої $l_k = 3$, для третьої $l_k = 11$, для четвертої $l_k = 5$. При цьому процес схрещування протікає так, як показано на рис. 41. В результаті виконання оператора схрещування виходять 4 пари нащадків.

Якщо б при випадковому підборі пар хромосом для схрещування були об'єднані, наприклад, ch3 с ch3 и ch4 с ch7 замість ch3 с ch4 и ch3 с ch7, а інші пари залишились без змін, то схрещування ch3 с ch3 дало б дві такі ж хромосоми незалежно від розіграної точки схрещування. Це означало б одержання двох нащадків, ідентичних своїм батькам. Зауважимо, що така ситуація найбільш імовірна для хромосом з найбільшим значенням функції пристосованості, тобто саме такі хромосоми отримують найбільші шанси на перехід до нової популяції.



Рис. 41. Процес схрещування хромосом.

Формування нової популяції. Після виконання операції схрещування ми отримуємо (згідно рис. 41) наступну популяцію нащадків:

Ch1 = [001111011011] Ch5 = [011101110010]

Ch2 = [101000111010] Ch6 = [001000101001]

Ch3 = [111011011011] Ch7 = [011101011011]

Ch4 = [101001100101] Ch8 = [101011110011]

Щоб відрізнити від хромосом попередньої популяції позначення знову сформованих хромосом починаються з великої літери С.

Згідно блок-схемі генетичного алгоритму (рис. 39) провадиться повернення до другого етапу, тобто до оцінки пристосованості хромосом з новосформованої популяції, яка стає поточною. Значення функцій пристосованості хромосом цієї популяції складають:

$F(\text{Ch1}) = 8$ $F(\text{Ch5}) = 7$

$F(\text{Ch2}) = 6$ $F(\text{Ch6}) = 4$

$F(\text{Ch3}) = 9$ $F(\text{Ch7}) = 8$

$F(\text{Ch4}) = 6$ $F(\text{Ch8}) = 8$

Помітно, що популяція нащадків характеризується набагато більш високим середнім значенням функції пристосованості, ніж популяція батьків. Звернемо увагу, що в результаті схрещування отримана хромосома Ch3 з найбільшим значенням функції пристосованості, яким не володіла ні одна хромосома з батьківської популяції.

Проте могло статися і зворотне, оскільки після схрещування на першій ітерації хромосома, яка в батьківській популяції характеризувалася найбільшим значенням

функції пристосованості, могла просто «загубитися». Крім цього «середня» пристосованість нової популяції все одно виявилася б вищою від попередньої, а хромосоми з великими значеннями функції пристосованості мали б шанси з'явитися в наступних поколіннях.

Після завершення роботи генетичного алгоритму з кінцевої популяції вибирається та особина, яка дає максимальне (або мінімальне) значення цільової функції і, отже, є результатом здійснення генетичного алгоритму. Кінцева популяція краща, ніж початкова за вибраним критерієм і отриманий результат є оптимальним рішенням.

Візуалізація даних

Візуалізація даних – це інструментальні засоби графічного зображення комплексних зв'язків в багатовимірних даних з різних перспектив або точок зору, представлення даних і узагальнюючої інформації з використанням графіки, анімації, 3-D дисплеїв та інших мультимедійних засобів. Графічне подання інформації засобами візуалізації має на меті забезпечення спостерігача якісним розумінням контексту інформації.

Візуалізація даних відноситься до інструментальних засобів дейтамайнінгу, які трансформують комплексні формули, математичні зв'язки або інформацію сховища даних в діаграми або інші легко зрозумілі моделі. Статистичні інструментальні засоби подібно кластерному аналізу або дереву класифікації і регресії CART часто є компонентами інструментальних засобів візуалізації даних. Аналітики можуть візуалізувати кластери або досліджують бінарне дерево, яке створюється за допомогою класифікування записів.

Крос-табуляція

Крос-табуляція або перехресна табуляція (перехресні табличні дані) є основна і дуже проста форма аналізу даних, добре відома в статистиці і широко використовується для створення звітів. На практиці використовують двовимірні таблиці. Заголовки стовпчиків такої таблиці відповідають значенням однієї змінної, а заголовки строк значенням другої змінної. Комірки таблиці представляють собою агреговані дані, які зустрічаються разом для двох змінних. Графічна крос-таблиця може бути представлена за допомогою трьохвимірної стовпчикової діаграми. Наприклад таблиця і діаграма які показують рахунки за однакові товари в двох різних супермаркетах для різних вікових груп населення .

В двох супермаркетах зібрано статистичний матеріал про обсяги товарів які придбали різні вікові групи клієнтів. Була зроблена крос-таблиця табл. 10 в якій проаналізовано обсяг продажів товарів в двох супермаркетах за віковими категоріями. Заголовками стовпців в крос-таблиці є вікові групи. Заголовками стовпчиків є товари, які були одночасно продані в першому і другому супермаркетах. В клітинках таблиці містяться агреговані дані обсягів продажу відповідних товарів для вказаної вікової групи (кількість проданих одиниць товару в двох супермаркетах).

Таблиця 10

	Вікові групи клієнтів які придбали товари (роки)				
Товари Супермаркетів 1, 2		18-24	25-35	36-50	51-85
Товар 1		1524	5557	12	23
Товар 2		15	45	14	365
Товар 2		78	800	600	20
Товар 4		9000	2500	1000	2500

Таблиці крос таблиці використовується для знаходження у мовних логічних зв'язків між різними факторами які досліджуються.

Крос-таблиці доцільно використовувати для таких проблем: коли маємо декілька факторів, які треба проаналізувати; коли треба виявити взаємозв'язок між факторами по статистичним даним.

Програмні агенти

Програмні агенти — є автономними програмними модулями, які призначені для автоматичного виконання задач пошуку інформації і моніторингу комп'ютерних систем, діють від імені користувача для забезпечення бажаних результатів, так само як людина-агент діє в інтересах замовника, щоб розширити його можливості (звідси й запозичений термін «агент»). Термін «програмні агенти» має низку синонімів: «інтелектуальні (розумні) агенти», «інтелектуальні інтерфейси», «персональні агенти», «програмні роботи», «аглети» (aglets) — так називаються програмні агенти в продукті IBM Aglets Workbench; часто вживаються скорочені терміни: «агент», «робот» тощо. Програмні агенти вбудовуються у програмне забезпечення, щоб зробити дії користувачів ефективнішими та результативнішими.

Сучасні програмні агенти, котрі постійно еволюціонують, не тільки проводять спостереження і виконують різні вимірювання, але й розв'язують завдання щодо управління мережами. Зокрема, інтелектуальні агенти здатні автоматизувати численні операції керування мережами, наприклад, вибір оптимального трафіка, контроль за завантаженням, поновлення даних за спотворень у процесі обміну тощо. Крім того, інтелектуальні агенти можуть застосовуватися і в інших галузях: для передавання повідомлень, вибирання інформації, автоматизації ділових процедур.

Існує багато типів програмних агентів (стаціонарні й мобільні, та ін.), котрі розроблені з застосуванням результатів досліджень у нейронних мережах, нечіткої логіки, інтерпретації текстів природною мовою, колаборативної фільтрації.

З метою глибшого розуміння суті поняття «програмний агент» потрібно описати, яким він має бути.

- **Функції.** Агент виконує низку завдань за дорученнями користувача (або іншого агента).
- **Можливості щодо обміну інформацією.** Агент мусить мати можливість обмінюватися інформацією з користувачем (а також іноді з іншими агентами), щоб отримувати від нього інструкції, повідомляти йому про хід і завершення виконання завдань і передавати отримані результати.
- **Автономність.** Агент працює без прямого втручання користувача (наприклад, як фоновий процес у той час, коли комп'ютер виконує інші завдання). Завдання, що виконуються агентом, можуть бути найрізноманітнішими — від щонічного резервного копіювання даних до пошуку (за дорученням користувача) продавця, що пропонує зазначений продукт за найнижчою ціною.
- **Моніторинг.** Щоб мати можливість виконувати свої завдання в автономному режимі, агент має бути здатним контролювати середовище, в якому він діє.
- **Активізація.** Щоб мати можливість працювати в автономному режимі, агент має бути здатним впливати на своє робоче середовище за допомогою механізму активізації (самостійного переходу у робочий стан).
- **«Розумність».** Агент має бути здатним інтерпретувати події, що контролюються ним, щоб ухвалювати належні рішення.
- **Безперервність роботи.** Агенти мають виконувати свої завдання постійно у часі.
- **«Індивідуальність».** Деякі агенти можуть мати добре виражений індивідуальний «характер».
- **Адаптивність.** Деякі агенти, ґрунтуючись на нагромадженому досвіді, автоматично пристосовуються до звичок і переваг своїх користувачів і можуть автоматично пристосовуватися до змін у навколишньому середовищі.
- **Мобільність.** Деякі агенти мають допускати можливість переміщення їх в інші комп'ютери, у тому числі й на системи іншої архітектури та інші платформи.

Програмні агенти значно різняться за їх складністю та функціями. Як простий приклад розглядають системи електронної пошти, які містять агентів, що допомагають оперувати великою кількістю повідомлень, які люди отримують кожного дня. Агент фільтрує пошту, попереджує про небезпеку, про наявність пріоритетних повідомлень, перенаправляє повідомлення у разі відсутності користувача і відкидає повідомлення за його вказівками.

Програмні агенти, що самі навчаються, спостерігають за тим, як користувач реально використовує програму, і пропонують виконувати це самі автоматично. Наприклад, якщо користувач читає всі повідомлення спершу від керівника (шефа), то агент міг би запропонувати помістити всі його повідомлення на початку списку.

Агент може керуватися часом, подією або алгоритмом чи деякою їх комбінацією. Наприклад, агент міг би бути запрограмованим так, щоб попередити користувача про

небезпеку, коли ціна на сировину перевищить заданий заздалегідь рівень. Або (приклад *керування за часом*), агент в кінці дня перевіряє, рівень запасів на складі, формує заявку на закупівлю у випадку якщо запасів недостатньо.

Попереджуючі агенти постійно переглядають середовище з метою пошуків певної інформації. Наприклад, агент може постійно шукати інформацію про клієнтів в схожих даних і Інтернеті та надсилати попереджати менеджера, що інформація знайдена. Керований користувачем агент має шукати інформацію тільки тоді, коли йому дана на це вказівка.

В інформаційних системах, зокрема в СППР, програмні агенти можуть застосовуватися для пошуку в базах даних потрібної для користувача інформації, для її аналізу з метою виявлення тенденцій або розробки моделей. Крім того, інтелектуальні агенти можуть швидко отримувати інформацію для ідентифікації незвичайних ситуацій, що дасть змогу користувачеві негайно на них зреагувати.

Користувачі інформаційних систем можуть знаходити інформацію з використанням розумних агентів на конкретну дату або здійснювати пошук за подіями. Наприклад, менеджер може виконувати регулярну перевірку браку або пропуску звітних даних для висвітлення через індикатори тих проблем, на які необхідно звернути увагу. Менеджер за допомогою розумного агента може також знаходити інформацію про просування товарів вище запланованого рівня або після досягнення конкретним показником точно визначеного рівня.

Результати роботи агентів можуть поєднуватися з іншими блоками інформаційної системи. Наприклад, агент може знайти інформацію, яку слід автоматично імпортувати до задачі з формування прогнозу для визначення майбутнього попиту. За бажання інший агент може бути переключений на результати розв'язання деяких задач для автоматичного інформування управлінського персоналу.

Програмні агенти включаються в програмне забезпечення додатків. Менеджери та інші працівники, які використовують знання, перенавантажені інформацією. Тому, проблема полягає у впорядкуванні існуючої інформації та виявленні тієї, яка є найважливішою. Програмні агенти застосовуються у, щоб допомогти менеджерам охопити великі масиви даних і інформації, які зберігаються в електронному вигляді. Програмний агент функціонує в СППР як фоновий, базовий процес, який застосовує низку логічних правил для того, щоб автоматично спостерігати за певною сукупністю даних, з метою пошуку зразка, що відповідає цим правилам, і інформувати зацікавлених користувачів, коли такі зразки виявляються. Цей вид агентів також називають «*програмним агентом-фільтром*» чи «*програмним агентом-спостерігачем*».

Програмні агенти є цінними інструментальними засобами для допомоги користувачам систем підтримки прийняття рішень та виконавчих інформаційних систем в аналізі великих баз даних на безперервній основі.

Довірчі мережі

Довірча мережа є орієнтованим графом, що складається з вершин (змінні) і дуг - імовірнісних залежностей між вершинами. Приклад довірчої мережі зображений на рис.42. Рисунок відображає частину крос-таблиці, наведеної раніше.

Вершини відображають вікові групи і види товарів (змінні) стрілки показують імовірність зв'язків між вершинами тобто вірогідність придбання даною віковою групою клієнтів відповідного товару.

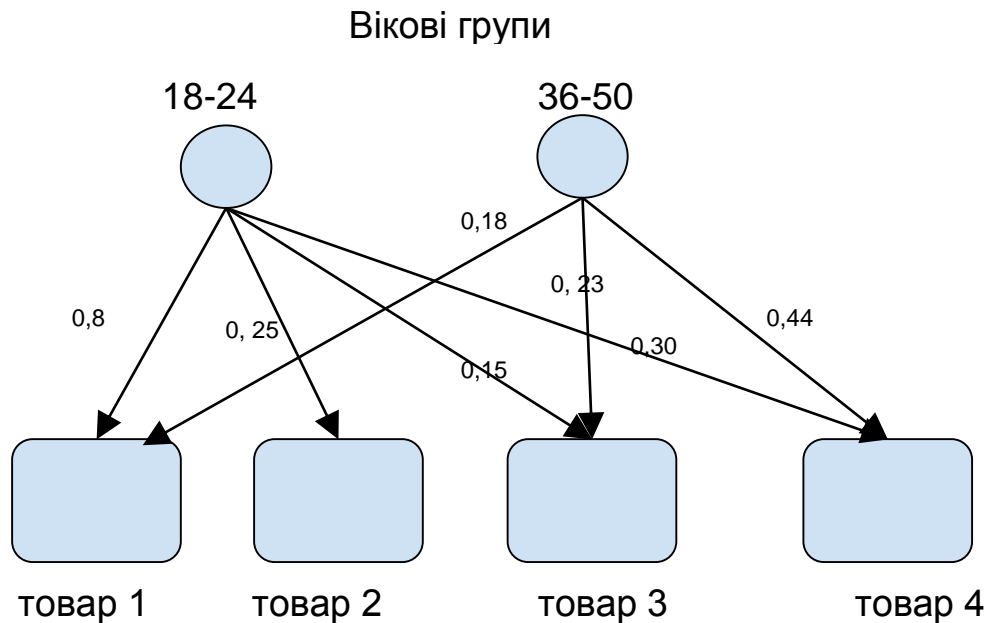


Рисунок 42 Орієнтований граф довірчої мережі

Довірчий мережі використовуються для виявлення ймовірнісних залежностей між різними чинниками процесу управління.

Висновки

1. Поясніть методи он-лайн обробки інформації в сучасних системах підтримки прийняття рішень.
2. Дайте визначення поняттю датамайнинг.
3. Поясніть методи видобування даних, які будуються на збереженні і дистиляції даних.
4. Що таке генетичні алгоритми?
5. Поясніть блок схему генетичного алгоритму.
6. Що таке візуалізація і крос-табуляція?
7. Для чого використовуються програмні агенти?
8. Що таке довірчі мережі?

ТЕМА 6. Ефективність запровадження систем підтримки прийняття рішень

Системи підтримки прийняття рішень дозволяють отримати кращі рішення, поліпшити процес прийняття оптимальних рішень. Ефективність управління вимірюється такими економічними показниками, як прибутковість підприємства, собівартість продукції, обсяги виробництва продукції. Системи СППР впливають на економічні показники опосередковано, тому кількісний вимір ефективності запровадження систем підтримки прийняття рішень ускладнюється, а інколи неможливий.

Системи підтримки прийняття рішень мають наступні переваги:

- покращують обробку та пошук знань;
- дозволяють ефективно розв'язувати складні проблеми;
- скорочують час прийняття рішень;
- скорочують витрати на розробку оптимального рішення;
- збільшують можливості по виявленню нових знань та інформації про процес яким керують;
- покращують обґрунтованість прийнятих рішень;
- підвищують надійність прийнятих рішень;
- поліпшують комунікації та координацію між працівниками в процесі прийняття рішень
- розширюють можливості і надають конкурентні переваги підприємству, яке застосовує систему підтримки прийняття рішень.

Оцінка розрахункової економічної ефективності системи на етапі проектування

Оцінка економічної ефективності системи СППР виконується до її впровадження в промислову експлуатацію і після провадження.

На етапі проектування системи оцінюється розрахункова економічна ефективність системи, ця оцінка базується на припущеннях і тому має бути перевірена вже на етапі промислової експлуатації системи.

Розглянемо як можна оцінити ефективність системи СППР на етапі проектування, тобто до впровадження системи в експлуатацію.

СППР яка використовується на ранній фазі процесу вирішення проблеми може забезпечити ряд переваг на всіх етапах управління. Системи СППР дають більше інформації керівникам і це дозволяють їм зменшити невизначеність при прийнятті рішень. Використання СППР стандартизує процес управління та його документування.

Система СППР поліпшує процес прийняття рішень і ми повинні оцінити економічні переваги які ми отримуємо. Для такої оцінки можна скористатися вищенаведеними запитаннями:

- Якщо система СППР дає кращу обробку інформації і знань, який економічний ефект від цього покращення?

- Якщо система СППР забезпечує краще розуміння проблеми, як можна оцінити втрати від неповного розуміння цієї проблеми?
- Скільки коштує здатність справлятися з більшим і складнішим обсягом проблем?
- Якщо ми можемо прийняти рішення швидше, скільки часу будуть збережено? Як збережений час відбився на економії заробітної плати та інших економічних ресурсів?
- Якщо вартість управлінського рішення знижується, яка буде економія для кожного рішення і як часто таке рішення приймається щоб заощадити кошти?
- Якщо система СППР дозволить більш глибоке вивчення процесів управління, скільки може коштувати таке вивчення ?
- Якщо система забезпечує кращий зв'язок або кращу координацію співпрацівників, як це відіб'ється на їх продуктивності ?
- Якщо система забезпечує більше ефективне управління персоналом та клієнтами, чи приводить це до зниження плинності кадрів або в зменшенню відтоку клієнтів? Як оцінити обсяги цих витрат?
- Якщо система СППР покращує систему прийняття рішень, то у чому виражається для працівників це покращення, вони більш мотивовані, більше задоволені роботою, пропонують більше інновацій і т. д. ?
- Якщо система СППР забезпечує конкурентні переваги, в чому виражаються ці переваги (наприклад збільшення частки ринку зменшення або збільшення ринкової вартості ?

Відповіді на наведені запитання мають бути доповнені розрахунками згідно стандартів оцінки економічної ефективності СППР. При цьому маємо брати до уваги, те що не всі показники можна оцінити кількісно.

Оцінка економічної ефективності системи на етапі впровадження в експлуатацію

Ефективність впровадження системи підтримки прийняття рішень можна оцінити через певний час її експлуатації. Переважно, в якості такого проміжку часу обирають один рік . Система експлуатується протягом року, визначаються результати її експлуатації і розраховуються показники економічної ефективності

Ефективність СППР характеризується системою показників, які відображають співвідношення прибутків і витрат.

Виділяють такі **показники ефективності проекту**:

- **Показники комерційної ефективності**, які враховують фінансові наслідки реалізації проекту СППР для його безпосередніх учасників;
- **Показники економічної ефективності**, які враховують господарські вигоди й витрати проекту, включаючи оцінку еколого-гічних та соціальних наслідків, і допускають грошовий вимір;

- **Показники бюджетної ефективності**, які відображають фінансові наслідки здійснення проекту СППР для державного та місцевого бюджетів.

Для розрахунку цих показників можуть використовуватись однакові формули, але значення вихідних показників для розрахунків істотно відрізняться.

Залежно від тривалості циклу проекту оцінка показників ефективності може бути різною. Показники комерційної ефективності можуть розраховуватися не тільки на весь цикл проекту, а й на місяць, квартал, рік.

Розрізняють три основні *методи визначення ефективності проектів СППР*:

- Порівняння витрат;
- Порівняння прибутку;
- Порівняння рентабельності.

До *показників ефективності проектів СППР* відносять:

1. Капіталовіддачу (річні продажі, поділені на капітальні витрати);
2. Оборотність товарних запасів (річні продажі, поділені на середньорічний обсяг товарних запасів);
3. Трудовіддачу (річні продажі, поділені на середньорічну кількість зайнятих робітників і службовців).

Однак ці показники належать до числа показників моментного, статичного стану і не враховують динамічних процесів у їх взаємозв'язку.

Для оцінки ефективності проектів доцільно використовувати такі показники, які дають змогу розрахувати значення критеріїв ефективності проектів, беручи до уваги комплексну оцінку доходності і витрат, зміну вартості грошей у часі та інші чинники. Правильне визначення обсягу початкових витрат на проект є запорукою якості розрахунків окупності проекту.

При аналізі ефективності проекту *СППР* використовують такі показники:

Сума інвестицій — це вартість початкових грошових вкладень у проект, без яких він не може здійснюватися. Ці витрати мають довгостроковий характер. За період функціонування проекту протягом його "життєвого циклу" капітал, вкладений у такі активи, повертається у вигляді амортизаційних відрахувань як частина грошового потоку, а капітал, вкладений в оборотні активи, в тому числі в грошові активи, по закінченню "життєвого циклу" проекту має залишатися у інвестора у незмінному вигляді й розмірі. Сума інвестицій у фінансові активи представляє собою номінальну суму витрат на створення цих активів;

Грошовий потік — дискontований або недискontований дохід від здійснення проекту, який включає чистий прибуток та амортизаційні відрахування, які надходять у складі виручки від реалізації продукції. Якщо у завершальний період "життєвого циклу" проекту підприємство-інвестор одержує кошти у вигляді недоамортизованої вартості основних засобів і нематеріальних активів та має вкладення капіталу в оборотні активи, вони враховуються як грошовий потік за останній період;

Чиста поточна вартість проекту — *Net Present Value (NPV)*. Це найвідоміший і найуживаніший критерій. У літературі зустрічаються й інші його назви: чиста приведена вартість, чиста приведена цінність, дискontовані чисті вигоди.

NPV являє собою дисконтовану цінність проекту (поточну вартість доходів або вигід від зроблених інвестицій). Чиста поточна вартість проекту — це різниця між величиною грошового потоку, дисконтованого за прийнятною ставкою доходності і сумою інвестицій. Для розрахунку NPV проекту необхідно визначити ставку дисконту, використати її для дисконтування потоків витрат та вигід і підсумувати дисконтовані вигоди й витрати (витрати зі знаком мінус). При проведенні фінансового аналізу ставка дисконту, звичайно, є ціною капіталу для фірми. В економічному аналізі ставка дисконту являє собою закладену вартість капіталу, тобто прибуток, який міг би бути одержаний при інвестуванні найприбутковіших альтернативних проектів. Якщо NPV позитивна, то проект можна рекомендувати для фінансування. Якщо NPV дорівнює нулю, то надходжень від проекту вистачить лише для відновлення вкладеного капіталу. Якщо NPV менша нуля — проект не приймається.

Розрахунок NPV робиться за такими формулами:

$$NPV = \sum_{t=1}^n \frac{B_t - C_t}{(1+i)^t},$$

або

$$NPV = \sum_{t=1}^n \frac{B_t}{(1+i)^t} - \sum_{t=1}^n \frac{C_t}{(1+i)^t},$$

де — B_t — доход від проекту в рік t ; C_t — витрати на проект у рік t ; i — ставка дисконту; n — строк експлуатації системи.

Основна перевага NPV полягає в тому, що всі розрахунки проводяться на основі грошових потоків, а не чистих доходів. Крім того, ефективність проекту СППР можна оцінити шляхом підсумовування NPV його окремих під-проектів. Це дуже важлива властивість, яка дає змогу використовувати NPV як основний критерій при аналізі проекту.

Основним недоліком NPV є те, що цей розрахунок вимагає детального прогнозу грошових потоків на термін життя проекту. На практиці застосовується припущення про постійність ставки дисконту.

Термін окупності інвестицій — час, протягом якого грошовий потік, одержаний інвестором від втілення проекту, досягає величини вкладених у проект фінансових ресурсів. У господарській практиці його можуть визначати без урахування необхідності грошових потоків у часі або з урахуванням такої необхідності. **Термін окупності проекту** — використовується переважно в промисловості. Один із найбільш часто вживаних показників оцінки ефективності капітальних вкладень. На відміну від показників, які використовуються у вітчизняній практиці, показник «термін окупності капітальних вкладень» базується не на прибутку, а на

грошо-вому потоці з приведенням коштів, які інвестуються в інновації та суми грошового потоку до теперішньої вартості. Критерій прямо пов'язаний із відшкодуванням капітальних витрат у найкоротший період часу і не сприяє проектам, які дають великі вигоди лише згодом. Він не може слугувати мірою прибутковості, оскільки грошові потоки після терміну окупності не враховуються.

Критерій найменших витрат (НВ) використовується тоді, коли оцінка вигід проекту складна й ненадійна. При цьому порівнюються наведені витрати по різних варіантах проекту і вибирають той, який при найменших витратах забезпечує найкращі результати. Критерій прибутку в перший рік експлуатації дає змогу перевірити, чи забезпечують вигоди за перший рік експлуатації проекту "достатню" доходність. При цьому порівнюється чистий дохід за перший рік експлуатації з капітальними витратами проекту, включаючи процентний дохід у період робіт по будівництву (береться накопичена сума процентів, а не наведені проценти). Якщо відношення вигід до витрат менше ціни капіталу, то проект є збитковим, а при більшому відношенні можна він є прибутковим.

До оцінки ефективності системи підтримки прийняття рішень мають бути додані показники, які неможливо оцінити кількісно.

Висновки

Системи підтримки прийняття рішень використовуються в різноманітних організаціях і підприємствах, для підтримки широкого діапазону рішень, для менеджерів що приймають рішення і для різних частин процесу прийняття рішень. Тому вигоди від цих систем можуть бути оцінені кількісними і якісними показниками.

Системи підтримки прийняття рішень дозволяють отримати кращі рішення, поліпшити процес прийняття оптимальних рішень. Ефективність управління вимірюється такими економічними показниками, як прибутковість підприємства, собівартість випускаємої продукції, обсяги випускаємої продукції. Системи СППР впливають на економічні показники опосередковано, тому кількісний вимір ефективності запровадження систем підтримки прийняття рішень ускладнюється.

Оцінка економічної ефективності системи СППР виконується до її впровадження в промислову експлуатацію і після провадження.

На етапі проектування системи оцінюється розрахункова економічна ефективність системи, ця оцінка базується на припущеннях і тому має бути перевірена вже на етапі промислової експлуатації системи.

Після запровадження системи підтримки прийняття рішень через певний час приблизно один рік виконуються розрахунки реальної економічної ефективності системи.

Розрахунок реального економічного ефекту забезпечує економічну основу для обґрунтування подальших вдосконалень і поліпшень в системі підтримки прийняття рішень. Позитивні результати економічної експлуатації СППР можуть бути підставою для розробки майбутніх подібних систем.

Контрольні запитання

1. Які переваги надають системи підтримки прийняття рішень?
2. Які критерії оцінки ефективності СППР на етапі розробки системи коли вона ще не експлуатується?
3. Чи всі показники ефективності СППР можуть бути обчислені в кількісному аспекті?
4. Які оцінити економічну ефективність СППР після впровадження в експлуатацію?
5. Поясніть поняття терміну окупності СППР.
6. Які показники ефективності СППР вам відомі?

Список рекомендованої та використаної літератури

1. Ситник В. Ф. Системи підтримки прийняття рішень: Навч. посіб. — К.: КНЕУ, 2004. — 614 с
2. Демиденко М.А. Математичне програмування: Навч. посібник. — Дніпропетровськ: Національний гірничий університет, 2005.— 110 с.
3. Демиденко М.А. Управління проектами інформатизації. /Навч. посібн. — Дніпропетровськ: НГУ, 2014. — 118 с.
4. Демиденко М. А. Економіко-математична модель багатокритеріальної оптимізації із застосуванням еволюційних алгоритмів / Михайло Андрійович Демиденко. // Економічний вісник Національного гірничого університету. — 2003. — №4. — С. 120–124.
5. Демиденко М.А. Підвищення прибутковості електронної комерції з використанням моделей спліт аналізу.. /“Проблеми і перспективи інноваційного розвитку економіки України” — Дніпропетровськ: НГУ, 2014.
6. Демиденко М. А. Економічно-математична модель підвищення ефективності управління кар’єрним транспортом / Михайло Андрійович Демиденко. // Економічний вісник Національного гірничого університету. — 2003. — №1. — С. 92–98.
7. Кочура Є.В. Пістунов І.М., Демиденко і др. Інформатика для економістів .Навч. посібник. — Дніпропетровськ: Наука і освіта, 2007.— 323 с.
8. Орлов А. И. Теория принятия решений: учебник. — М.: Экзамен, 2006. — 573 с. Орлов А. И. Принятие решений. Теория и методы разработки управленческих решений. Учебное пособие. — М.: MapT, 2005. — 496 с
9. Литвак Б. Г. Разработка управленческого решения — М.: Издательство «Дело», 2004 г. — 392 с.
10. Г. С. Кильдишев, А. А. Френкель. Анализ временных рядов и прогнозирование. М.: «Статистика» 1973.
11. Хемди А. Таха Глава 14. Теория игр и принятия решений // Введение в исследование операций. 7-е изд. — М.: «Вильямс», 2007.
12. Г. Тейл. Экономические прогнозы и принятие решений. М.: «Прогресс» 1970.
13. К. Д. Льюис. Методы прогнозирования экономических показателей. М.: «Финансы и статистика» 1986.
14. Айвазян С.А. Прикладная статистика. Том 2. Основы эконометрики Учебник для вузов: В 2 т. — 2-е изд., испр. — М.: ЮНИТИ-ДАНА, 2001. — 432 с.
15. Потрашкова Л.В., Комп’ютерна підтримка прийняття рішень у сфері дизайну поліграфічної продукції.: Системи обробки інформації, 2010, випуск 7 (88)
16. Архіпова, Т. Л., and В. В. Крініцин. "Економічна оцінка ефективності інформаційних систем." Науковий вісник КУЕІТУ 2 (2011): 32.
17. George M. Marakas, Decision Support Systems in the 21st Century, 2nd Edition, Prentice Hall, 2002
18. Alex Berson, Stephen J. Smith Data Warehousing, Data Mining, and OLAP (Data Warehousing/Data Management), McGraw-Hill, 2010
19. Power D.J. A Brief History of Decision Support Systems. DSSResources.COM, World Wide Web, <http://DSSResources.COM/history/dsshhistory.html>, version 2.8, May 31, 2003.
20. Marius Cioca, Florin Filip (2015). Decision Support Systems - A Bibliography 1947-2007
21. Bonczek, R., C. Holsapple and A. Whinston, Foundations of Decision Support Systems. New York: Academic, 1981.

Предметний покажчик

EM-алгоритм	57	Метод дерев рішень.....	77
HOLAP.....	28	метод кластерного аналізу	74
IBM	34	Метод статистичної екстраполяції.....	65
Intelligent Warehouse	35	Методи аналізу неповних даних	55
MOLAP	28	Методи добування даних	74
OLAP-сервери	34	методи навчання.....	54
Oracle	35	методи онлайнової обробки	71
ROLAP.....	28	нтовані на моделі	15
Web орієнтовані СППР	14	організація багатостороннього прий- няття рішень	7
Алгоритм отримання знань	8	Оцінка економічної ефективності.....	93
атрибути знань	9	Переваги багатовимірного підходу	23
Багатовимірна модель даних	22	Переваги та недоліки гібридної моделі	28
Багатовимірний куб.....	23	передатна функція	51
Взаємодія.....	6	Підготовка даних	55
Взаємозв'язок систем	2	побудова сховищ даних.....	21
Видалення нестаціонарності	60	прийняття рішень	5
викиди	57	прогнозування	54
Виміри	22	програмне забезпечення	66
Візуалізація даних.....	88	<i>Програмні агенти</i>	89
Віртуальним сховищем даних	29	регресії.....	62
Вітрина даних	30	Регресійні моделі	62
Генетичні алгоритми	81	Реляційна база даних.....	25
Гібридні СППР	16	розв'язання задач лінійного програму- вання	42
Гра.....	45	система обробки знань	10
Групова система.....	17	СППР	4
Групування нейронів	52	Структура систем підтримки прийняття рішень.....	12
Дейтамайнінг	73	Схеми «зірка».....	25
Дистиляція даних	76	схеми «сніжинка»	25
Довірча мережа	92	Сховище даних.....	14
Експоненційне згладжування	63	Табличні СППР.....	15
Екстраполяція.....	65	Текстові системи	13
Ефективність впровадження	94	тенденціями розвитку сховищ даних .	36
задача дробово-лінійного програму- вання	42	Традиційний метод	3
задача нелінійного програмування	43	функція Лагранжа.....	44
Збережені дані.....	74	Штучний інтелект	16
ковзне середнє	63	Штучні нейромережі.....	49
Консолідація	18		
Критерій оптимізації	38		
Крос-табуляція	88		
Метод “найближчого сусіда”	74		

ВИТЯГ З ПРОТОКОЛУ № 5
засідання Вченої ради
Державного вищого навчального закладу
«Національний гірничий університет»

22.12.2015 р.

м. Дніпропетровськ

Голова Вченої ради

Г.Г. Півняк

Вчений секретар

О.А. Данилова

Склад Вченої рада затверджений рішенням Конференції трудового
колективу від 26.10.2015 (протокол № 1).

На засіданні були присутні 69 з 90 членів Вченої ради

1. Розгляд результатів експертизи рукопису навчальної книги “Системи підтримки прийняття рішень” (автор: Демиденко М.А.) на предмет надання грифу «Затверджено Вченою радою як підручник».

СЛУХАЛИ: завідувача кафедри Економічної кібернетики та інформаційних технологій проф. Кочури Є.В., який відзначив, що:

1. На розгляд подано матеріали:

- рукопис навчальної книги “Системи підтримки прийняття рішень”;
- дві рецензії фахівців у сфері економічної кібернетики та комп’ютерних наук;
- програма дисципліни “Системи підтримки прийняття рішень”;
- обґрунтування доцільності видання навчальної книги.

2. Рецензії на навчальну книгу “Системи підтримки прийняття рішень” позитивні.

3. Експертизою встановлено:

- зміст навчальної книги синтезує та узагальнює відомості про інформаційні системи підтримки прийняття рішень і відповідає нормативній компоненті освітньо-професійної програми підготовки та програмі дисципліни “Системи підтримки прийняття рішень” для бакалаврів напрямку 6.050102, “Економічна кібернетика”;

- за побудовою, методами структурування навчального матеріалу, дидактичним опрацюванням змісту, обсягом навчальна книга відповідає чинним вимогам;

- вказані вище фактори, є підставою для надання навчальній книзі “Системи підтримки прийняття рішень” грифу «Затверджено вченою радою як підручник»;

За участю автора навчального посібника, питання систем підтримки прийняття рішень розглянуто у такій навчальній літературі, створеній за участю автора зазначеного рукопису:

○ Інформатика для менеджерів. Навч. Посібник / Є.В. Кочура, Р.І. Лепа, І.М. Пістунів, В.А. Нецветаєв, Т.В. Борщ, – Дніпропетровськ: Освіта і наука, 2009. – 324 с.

○ Управління проектами інформатизації : навч. посіб. / М.А. Демиденко ; Нац. гірн. ун-т. – Електрон. текст. дані. – Д. : 2014. – 114 с.

Апробація матеріалів, викладених у посібнику, складає чотири роки, що є підставою для того, щоб навчальний посібник “Системи підтримки прийняття рішень”, претендував на гриф «Затверджено вченою радою як підручник».

УХВАЛИЛИ: Надати навчальній книзі “Системи підтримки прийняття рішень” (автор: Демиденко М.А.) гриф “Затверджено Вченою радою як навчальний посібник” для бакалаврів напрямку 6.050102, “Економічна кібернетика”;

Вчений секретар



О.А. Данилова

Навчальне видання

Демиденко Михайло Андрійович,

СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

Навчальний посібник

У редакції автора

Підписано до друку . Формат 30 × 42/4.
Папір офсетний. Ризографія. Авт. Арк. 4,28.
Обл.–вид. арк. 13,7. Тираж 100 прим. Зам. № 96/12.

Підготовлено до друку та видруковано
у Державному вищому навчальному закладі
Свідоцтво про внесення до Державного реєстру ДК No 1842.
49600, м. Дніпропетровськ, просп. К. Маркса, 19.