

Інститут проблем реєстрації інформації НАН України

На правах рукопису

Циганок Віталій Володимирович

УДК 519.816, 681.518.2

**Методи отримання та обробки
кардинальних експертних оцінок**

01.05.03 (Математичне та програмне забезпечення обчислювальних машин і систем)

Дисертація на здобуття наукового ступеня

кандидата технічних наук

Науковий керівник

Тоценко Віталій Георгійович,

доктор технічних наук, професор

Київ-2003

Зміст

Перелік умовних скорочень	6
Вступ. Роль і місце кардинальних експертних оцінок. Стан проблеми	7
1. Розробка індивідуальних методів парних порівнянь зі зворотним зв'язком з експертом і послідовною обробкою матриці парних порівнянь	16
1.1. Метод визначення підмножин сумісних об'єктів	16
1.1.1. Два доповнення до алгоритму	21
1.2. Метод парних порівнянь зі зворотним зв'язком з експертом і послідовною обробкою матриці парних порівнянь	23
1.2.1. Сутність методу	25
1.2.2. Стабільність алгоритму	31
1.2.3. Приклад використання довільної шкали	32
1.2.4. Приклад порівняння способом “на скільки”	40
1.3. Способи введення інформації для отримання кардинальних експертних оцінок	41
1.4. Висновки	44
2. Розробка індивідуальних комбінаторних методів парних порівнянь зі зворотним зв'язком	46
2.1. Комбінаторний метод	46
2.1.1. Суть методу	48
2.1.2. Опис алгоритму	51

2.1.3. Деякі особливості реалізації	60
2.1.4. Переваги та недоліки алгоритму	60
2.2. Різновиди методу	61
2.3. Висновки	62
3. Розробка групових методів отримання кардинальних експертних оцінок в умовах неповної визначеності	64
3.1. Постановка задачі. Сутність підходу	66
3.2. Метод безпосередньої оцінки	70
3.3. Метод “лінія”	72
3.4. Метод “трикутник”	75
3.5. Метод “квадрат”	79
3.6. Висновки	85
4. Експериментальне дослідження методів отримання кардинальних експертних оцінок	87
4.1. Вибір оцінюваних об'єктів	88
4.2. Досліджувані методи оцінювання без зворотного зв'язку з експертом	89
4.2.1. Метод безпосереднього оцінювання (#1)	89
4.2.2. Метод “трикутник” мультиплікативних парних порівнянь з виразом дискретних значень переваг у фундаментальній шкалі, словесним введенням даних, попереднім упорядкуванням і послідовним методом обробки матриці порівнянь (#2)	90
4.2.3. Метод “трикутник” мультиплікативних парних порівнянь з вираженням квазібезперервних значень переваг у фундаментальній шкалі, графічним введенням даних, попереднім упорядкуванням і послідовним методом обробки (#3)	91

4.2.4. Метод “трикутник” мультиплікативних парних порівнянь з вираженням дискретних значень переваг у фундаментальній шкалі, словесним введенням даних і паралельним методом обробки (#4)	92
4.2.5. Метод “трикутник” мультиплікативних парних порівнянь з вираженням квазібезперервних значень переваг у фундаментальній шкалі, графічним введенням даних і паралельним методом обробки (#5)	93
4.2.6. Метод “квадрат” адитивних парних порівнянь з вираженням квазібезперервних значень переваг у довільній шкалі, числовим введенням даних і послідовним методом обробки по рядках матриці порівнянь(#6)	94
4.2.7. Метод “квадрат” адитивних парних порівнянь з вираженням квазібезперервних значень переваг у довільній шкалі, числовим введенням даних і послідовним методом обробки по стовпцях рядках матриці порівнянь(#7)	95
4.2.8. Метод “квадрат” мультиплікативних парних порівнянь з вираженням квазібезперервних значень переваг у довільній шкалі, числовим введенням даних і послідовним методом обробки по рядках матриці порівнянь(#8)	96
4.2.9. Метод “квадрат” мультиплікативних парних порівнянь з вираженням квазібезперервних значень переваг у довільній шкалі, числовим введенням даних і послідовним методом обробки по стовпцях матриці порівнянь(#9)	96
4.2.10. Метод “квадрат” мультиплікативних парних порівнянь з вираженням дискретних значень переваг у фундаментальній шкалі, словесним введенням даних і послідовним методом обробки по рядках матриці порівнянь(#10)	97

4.2.11. Метод “квадрат” мультиплікативних парних порівнянь з вираженням дискретних значень переваг у фундаментальній шкалі, словесним введенням даних і послідовним методом обробки по стовицях матриці порівнянь (#11)	98
4.3. Досліджувані методи індивідуального експертного оцінювання зі зворотним зв’язком з експертом	98
4.3.1. Метод парних порівнянь “трикутник” зі зворотним зв’язком з експертом	100
4.3.2. Метод парних порівнянь “квадрат” зі зворотним зв’язком з експертом	104
4.4. Алгоритми обробки даних експериментального дослідження методів експертного оцінювання	108
4.4.1. Визначення спектрального коефіцієнта узгодженості експертних оцінок відносних ваг об’єктів, що забезпечується досліджуваним методом експертного оцінювання	108
4.4.2. Обчислення математичного очікування відносної похибки визначення відносних ваг об’єктів	113
4.4.3. Обчислення математичного очікування тривалості визначення коефіцієнтів відносної ваги об’єктів	114
4.4.4. Визначення відносних показників методів експертного оцінювання	114
4.4.5. Результати обробки експериментальних даних та їх аналіз	115
5. Розробка методу підтримки прийняття рішення при виборі методу отримання кардинальних експертних оцінок	119
Висновки	125
Додаток А.	140
Додаток Б.	141
Додаток В.	144

Перелік умовних скорочень

EO експертні оцінки

IBE інформаційно-вагомий елемент

IУМПП ідеально узгоджена матриця парних порівнянь

KEO кардинальні експертні оцінки

МО математичне очікування

МПП матриця парних порівнянь

ОПР особа, що приймає рішення

СППР системи підтримки прийняття рішень

Вступ. Роль і місце кардинальних експертних оцінок. Стан проблеми

Безперечним є той факт, що для прийняття обґрунтованих рішень у будь-якій сфері діяльності людини потрібно спиратись на знання, досвід та інтуїцію спеціалістів.

Під методами експертних оцінок маються на увазі методи, що включають організацію роботи з фахівцями-експертами та обробку особистих думок експертів, вражених у кількісній і/або якісній формі, з метою підготовки інформації, необхідної для прийняття рішень для осіб, що приймають рішення (ОПР).

На теперішній час існує велике різноманіття методів отримання експертних оцінок. В одних з кожним експертом працюють окремо, він навіть може не знати, хто є експертом, а тому висловлює свою думку досить незалежно. В інших методах експертів збирають разом для підготовки матеріалів для ОПР, при цьому експерти обговорюють проблему один з одним, і невірні думки відкидаються. В одних методах число експертів фіксоване і таке, щоб статистичні методи перевірки узгодженості думок і потім їхнього усереднення дозволяли приймати обґрунтовані рішення. В інших – число експертів росте в процесі проведення експертизи, наприклад, при використанні методу 'сніжної груди' [1].

В даний час не існує науково обґрунтованої класифікації методів експертних оцінок і тим більше – однозначних рекомендацій з їх застосування [2].

Спочатку найбільш поширеними були найпростіші методи експертних оцінок (див., наприклад, [3,4]). Ці тривіальні розуміння довгий час були досить розповсюдженими і навіть стали гальмом на шляху впровадження більш сучасних результатів в області

експертних оцінок, описаних, наприклад, у роботах [5–17, 19].

Виходячи зі зручності для експертів висловлювати свої думки відносно оцінки тих чи інших об'єктів широке застосування серед методів експертних оцінок знайшли методи парних порівнянь. Це відбувалось в більшості завдяки тому, що експертові значно легше порівнювати на кожному кроці тільки два об'єкти ніж, наприклад, зробити ранжування або розбиття множини об'єктів.

Метод парних порівнянь широко застосовується для визначення відносних показників значимості альтернатив при незначних відмінностях порівнюваних об'єктів щодо обраного якісного критерію. Цей метод спочатку використовувався для вирішення задач ранжування об'єктів відповідно до критерію, що відображає деяку спільну для цих об'єктів властивість [18–23]. Однак останнім часом, особливо в зв'язку з розвитком методів багатокритеріальної оптимізації [24, 25], аналітичних ієрархічних процесів (АІП) [26], аналітичних мережевих процесів (АМП) [27] і методів підтримки рішень, що засновані на використанні ієрархії цілей [28–30], парні порівняння стали застосовуватися для визначення відносних пріоритетів w_i , $i = (1, \bar{r})$ об'єктів (критеріїв, цілей, альтернатив). При вирішенні задач цільового програмування [31] ваги (пріоритети, коефіцієнти значимості) критеріїв і альтернатив також є оцінками такого типу. Ці величини являють собою відносні кількісні показники ступеня виразності деякої властивості в кожного об'єкта з заданої множини.

Метод парних порівнянь є методом відносних вимірювань, сутність якого полягає в тому, що шляхом порівнянь у загальному випадку кожного об'єкта з всіма іншими з даної множини визначаються елементи матриці парних порівнянь (МПП) D розмірністю $r \times r$, де елемент d_{ij} є відповідне дійсне число, що визначає результат порівняння об'єкта i з об'єктом j щодо деякої їхньої загальної властивості (критерію). Відносні пріоритети w_i обчислюються як результати обробки матриці порівнянь.

Результати порівнянь об'єктів за якісними критеріями виражают, як правило, у так званій фундаментальній шкалі [26, 27]. Ця шкала має наступні значення ступенів переваги одного об'єкта перед іншим: “еквівалентність (1)”, “слабка перевага (2)”, “помірна перевага (3)”, “більше, ніж помірна (4)”, “сильна перевага (5)”, “досить

сильна перевага (6)" , "дуже сильне перевага (7)" , "дуже, дуже сильна перевага (8)" , "надзвичайно сильна перевага (9)" , де у дужках записані числові еквіваленти ступенів переваги. Таким чином, результати порівнянь об'єктів як за кількісними, так і за якісними критеріями можуть бути виражені у вигляді матриці D відповідних дійсних чисел.

Елементи МПП часто розглядаються як випадкові величини, що задані в деякому інтервалі (інтервальні оцінки) [32–43].

Використання експертної інформації передбачається при застосуванні методів багатокритеріальної оптимізації [23, 24], аналітичних ієрархічних процесів [26], аналітичних мережних процесів [27], цільового програмування [31], цільового оцінювання альтернатив [29, 30, 54, 76], в системах підтримки прийняття рішень (СППР) та експертних системах різноманітного призначення (наприклад, в СППР соціально-економічного спрямування [77], в експертних системах діагностики [85], аналізу суспільно-політичних процесів [86]). Методи її отримання, представлення і обробки є невід'ємною частиною відповідних інформаційних технологій. Крім того, методи експертного оцінювання, зокрема метод парних порівнянь, використовуються як самодостатні, наприклад, при економічних [87, 88], соціальних і медичних дослідженнях [89], спортивні [90].

При підтримці прийняття рішень використовується експертна інформація двох видів: концептуально-понятійна й оцінювальна. Інформація першого виду являє собою формулювання цілей, критеріїв і альтернатив. Вона представляється у текстовій формі на природній мові. До другого виду відноситься інформація про оцінки показників цілей, критеріїв і альтернатив. При цьому розрізняють абсолютні і відносні оцінки. До абсолютних відносять оцінки таких параметрів альтернатив як ймовірність, тривалість і вартість реалізації, час затримки поширення впливу реалізації альтернативи на досягнення цілі. Крім того, використовуються абсолютні оцінки параметрів цілей: ресурс, необхідний для досягнення цілі, ефект від її досягнення, поріг цілі [29,30,54]. У деяких випадках приймають рішення, виходячи з кількісних оцінок ефективності варіантів рішення. Відносні оцінки застосовуються ширше. Серед них

виділяють оцінки двох класів: ординальні і кардинальні.

Ординальні оцінки об'єктів являють собою їхні ранги в ряду переваг по деякому критерію. Кардинальні оцінки об'єктів (цілей, критеріїв, альтернатив) виражаються дійсними числами, що показують ступені вираженості у них властивості, за якою порівнюються об'єкти (наприклад, відносні показники вагомості критеріїв альтернатив відносно деякого критерію, часткові коефіцієнти впливу підцілей на деяку ціль).

Отримання кардинальних оцінок об'єктів є більш складною задачею у порівнянні з визначенням ординальних. Проте такі оцінки не тільки дозволяють формувати ранжування об'єктів (ординальні оцінки), а й свідчать про їх відносну вагомість, що дає змогу вирішувати, наприклад, задачу обґрунтованого розподілу ресурсів між ними. Інакше кажучи, визначення кардинальних оцінок є більш загальною задачею підтримки прийняття рішень, ніж ранжування об'єктів. Ці чинники обумовили розробку СППР, орієнтованих на визначення кардинальних оцінок альтернатив [26, 27, 29, 30, 54].

Методи отримання кардинальних експертних оцінок використовуються в мультикритеріальних методах підтримки прийняття рішень для визначення показників відносної важливості критеріїв і альтернатив [26, 27, 50, 51], у методах цільового оцінювання альтернатив – для визначення коефіцієнтів впливу підцілей на досягнення безпосередніх надцілей [29, 30, 54], у методах цільового програмування [31] – для знаходження коефіцієнтів відносної важливості цілей. Крім методу Saatі [26, 27], який вже став класичним, і його узагальнення Харкером на не повністю визначені МПП [79] останнім часом розроблено цілу гамму методів отримання кардинальних експертних оцінок [26–28, 44–46, 82].

Всі ці методи відрізняються за декількома ознаками: алгоритмом отримання інформації від експерта (безпосередня оцінка, парні порівняння), типом оцінок переваг (безперервні, дискретні), класом функцій залежності ваг об'єктів від ступенів переваг (мультиплікативні, адитивні), способом обробки первісної експертної інформації (паралельний, послідовний, комбінаторний), способом формування множини первісних експертних оцінок (“лінія”, “трикутник”, “квадрат”), способом уводу первісної

експертної інформації (вербальний, числовий, графічний), наявністю чи відсутністю зворотного зв'язку з експертом. Природно, що така відмінність ознак методів отримання і обробки кардинальних експертних оцінок (CEO) породжує різноманітність значень показників методів, які є суттєвими для користувача, зокрема для особи, що приймає рішення (ОПР). Ці показники доречно назвати показниками застосування методів. До них відносяться: показник точності оцінок (математичне очікування відносної похибки); коефіцієнт узгодженості CEO однакових об'єктів, даних різними експертами; показник тривалості визначення CEO (математичне очікування необхідного часу або кількості дій виконаних експертами).

Методи отримання CEO часто використовують парні порівняння для підвищення достовірності отримуваних оцінок, але існуючі методи обробки МПП не дозволяють виявити джерело можливих протиріч, що виникають під час порівнянь об'єктів, і порекомендувати експертові спосіб узгодити ці порівняння і тим самим підвищити достовірність результатів. У зв'язку з цим існує нагальна необхідність у створенні нових методів обробки МПП зі зворотним зв'язком з експертом, які б були позбавлені вказаного недоліку. Крім того всі методи EO характеризуються рядом параметрів, і актуальною науковою проблемою є вибір методу EO, за допомогою якого оцінки можливо знайти з найбільшою ефективністю в конкретній ситуації. Дисертаційна робота присвячена дослідженню існуючих методів EO, розробці нових методів, які б відповідали поставленим вимогам, та розробці методики вибору конкретного методу EO серед існуючих для застосування в конкретній ситуації.

Оскільки напрямленість дослідження тісно пов'язана з розробкою СППР, то дисертаційна робота виконувалась у рамках плану фундаментальних наукових досліджень Інституту проблем реєстрації інформації НАН України за темами "Розробка елементів теорії ієархій як основи побудови прогнозуючих систем підтримки рішень" (1999р., N держреєстрації 0197U016092) [94], "Система підтримки рішень при комплексному цільовому плануванні НДДКР та закупівлі озброєння і військової техніки для Збройних Сил України" (1999р., N держреєстрації 0199U003984) [95], "Розробка системи інформатизації діяльності науково-технічної ради Міністерства Обо-

рони України” (1999р., № держреєстрації 0199U003985) [96] та ”Розробка елементів теорії прийняття колективних рішень” (2002р., № держреєстрації 0100U002077) [97].

Метою досліджень є підвищення ефективності компонент СППР шляхом розробки методів ЕО, критеріїв їхнього оцінювання та вирішення задачі вибору конкретного методу ЕО в конкретній ситуації. Відповідно до поставленої мети сформульовано наступні завдання, що стоять перед дисертаційним дослідженням:

- розробка індивідуальних методів парних порівнянь зі зворотним зв’язком з експертом;
- розробка індивідуальних комбінаторних методів парних порівнянь;
- розробка групових методів отримання кардинальних ЕО;
- розробка методики експериментального дослідження та проведення експерименту з метою визначення основних показників, що характеризують той чи інший метод ЕО;
- розробка методики та алгоритму вибору конкретного методу ЕО на основі отриманих експериментальним шляхом характеристик методів та вимог до методу, що має застосовуватись в конкретній ситуації.

При вирішенні задач дисертаційного дослідження застосовувалися методи булевої алгебри та теорії множин для опису властивостей новоуведеної операції сумісності об’єктів та при вирішенні задачі виділення підмножин сумісних об’єктів. Для знаходження інформаційно-вагомих елементів в комбінаторних методах ЕО та для визначення конструктивності множин елементів в групових методах ЕО були застосовані методи теорії графів. Крім того, для обробки даних, отриманих під час експерименту, застосовувався апарат математичної статистики.

В результаті даного дослідження отримано ряд наукових результатів. Серед них – розробка методу визначення в множині об’єктів підмножин взаємно сумісних об’єктів, який має значно меншу трудоємність у порівнянні з раніше застосовуваними для цього методами, а саме методами, які базувались на знаходженні множини простих шляхів максимальної довжини в графі [54].

В межах цього наукового дослідження розроблено методи парних порівнянь з використанням зворотного зв'язку з експертом для покращення внутрішньої узгодженості думок експерта шляхом виявлення та вказівки експертові на джерело можливих протиріч. Ці методи використовують так звану послідовну обробку МПП, яка на відміну від паралельної обробки [26] дає можливість виявити джерела протиріч, які виникли при формуванні даної МПП, і за допомогою зворотного зв'язку з експертом усунути або, принаймні, зменшити цю неузгодженість і, тим самим, поліпшити достовірність отримуваних результатів. Розроблені методи відрізняються ще й тим, що рівень узгодженості в МПП вимірюється при допомозі спектральних коефіцієнтів узгодженості, які дозволяють визначити достатність рівня узгодженості, при якому застосування результатів парних порівнянь для розрахунків відносних ваг об'єктів є коректним.

Серед різновидів методів парних порівнянь крім методів, які використовують мультиплікативні порівняння (коли експертові пропонується визначити, *у скільки разів один об'єкт переважає інший*), розроблено ще й методи, що використовують адитивні порівняння (коли експертові пропонується визначити *на скільки умовних одиниць один об'єкт переважає інший*).

Розроблено методи парних порівнянь комбінаторного типу, які дозволяють звести кількість звернень за думкою експерта при обробці МПП до мінімуму, що важливо для методів зі зворотним зв'язком. Для даних методів, як і для попередніх, підтримуються як мультиплікативні, так і адитивні парні порівняння. Крім того, що ці методи мають достатньо високу точність знаходження оцінок об'єктів, вони позитивно відрізняються тим, що є нечутливими до наявних в МПП багатократних неузгодженостей та до протиріч в ранжуванні об'єктів.

Розроблено групові методи отримання кардинальних ЕО, які дозволяють прийти до єдиної думки групі експертів з різною компетентністю в даному питанні при відмові з деяких причин в оцінюванні деякими експертами деяких об'єктів. На відміну від існуючих методів, які застосовуються для обробки неповних МПП [79] для методу парних порівнянь з паралельною обробкою МПП, цей підхід розповсюдженено на метод

безпосередньої оцінки, а також на три методи парних порівнянь, що відрізняються трудомісткістю та потенційною точністю: ”лінія”, ”трикутник” та ”квадрат”.

Характерним для цих методів є принципова можливість обчислення оцінок по частині матриці парних порівнянь, а також спосіб формального визначення діалогів з експертами у ході покращення узгодженості їхніх оцінок. Методи розроблені з урахуванням двох видів неповноти матриці парних порівнянь: неповноти, породженої неможливістю для експерта виконати порівняння пар об'єктів по деякій причині (наприклад, недостатності для цього необхідної інформації) та неповноти, що породжується прагненням отримати оцінки за мінімальну кількість кроків. Розглянуто організацію роботи експертів з метою отримання потрібної вірогідності оцінок за мінімальний час.

Розроблено метод прийняття рішення при виборі методу ЕО, який на основі отриманих експериментальним шляхом середньостатистичних параметрах кожного з методів та по отриманих від експерта оцінках переваг цих параметрів в конкретній ситуації дає змогу обґрунтованого вибору методу ЕО.

Таким чином, розроблені методи отримання та обробки КЕО зайняли достойне місце серед методів ЕО і, як показали результати експериментального дослідження, мають кращі показники достовірності отримуваних результатів ніж існуючі методи. Це дає змогу більш ефективно використовувати знання експертів у конкретній прикладній сфері життєдіяльності людини і дозволяє приймати більш обґрунтовані рішення.

Запропоновані у дисертаційній роботі методи ЕО (крім методів групового експертного оцінювання) реалізовані в програмній системі підтримки прийняття рішень ”Солон-2“ [47] (Додаток А). Запропоновані методи, включаючи і методи групового експертного оцінювання, знайшли свою реалізацію в СППР ”Солон-3“. Результати дисертаційного дослідження впроваджені в Головному управлінні розробок та закупівлі озброєння і військової техніки озброєння Міністерства Оборони України (акт від 29.04.2003р. - Додаток Б)

Основні положення та результати роботи доповідались та обговорювалися на 9-й

міжнародній школі-семінарі ”Інформаціонно-управляючі системи на железнодорожном транспорті”(Алушта, 1996) [48, 49] та науково-технічних конференціях Інституту проблем реєстрації інформації НАН України в 2002 - 2003 роках.

Основні результати дисертації опубліковані в 6 статтях у наукових фахових журналах та в одному свідоцтві про державну реєстрацію прав автора на твір.

Усі основні результати дисертаційного дослідження отримані автором самостійно, а саме: розроблено метод виділення у множині об'єктів підмножин взаємно сумісних об'єктів [83], розроблено комбінаторний метод парних порівнянь зі зворотним зв'язком з експертом [82]. У роботах, опублікованих у співавторстві, дисертанту належить: спосіб досягнення стабільності методу парних порівнянь зі зворотним зв'язком з експертом та спосіб уникнення повторних звернень до експерта з одним і тим самим запитанням [28], визначення кардинальних групових оцінок значимості об'єктів з урахуванням відносної компетентності експертів, урахування неповноти інформації, спосіб перевірки на предмет конструктивності множин оцінок, отриманих від різних експертів [97], вибір об'єкта для оцінювання при проведенні експерименту, розробка алгоритму проведення експерименту, розробка системи збору та обробки експериментальних даних [91, 92], формалізація задачі прийняття рішення щодо вибору конкретного методу експертного оцінювання, програмна реалізація запропонованого алгоритму та опис конкретних прикладів [93].

Розділ 1.

Розробка індивідуальних методів парних порівнянь зі зворотним зв'язком з експертом і послідовною обробкою матриці парних порівнянь

1.1. Метод визначення підмножин сумісних об'єктів

В СППР перед тим, як застосовувати методи парних порівнянь для визначення кардинальних експертних оцінок деякої множини альтернатив, виникає необхідність у визначенні таких підмножин альтернатив, які є сенс порівнювати в даній конкретній ситуації. Тобто виникає задача виділення в множині об'єктів мінімальної кількості підмножин взаємно сумісних між собою об'єктів. Об'єднання отриманих підмножин повинне складати вихідну множину. З мінімальноті кількості підмножин випливає, що результуючі підмножини будуть мати максимально можливу потужність.

Для вирішення цієї задачі раніше був застосований метод, що базується на використанні механізму теорії графів, а саме, пошуку простих шляхів максимальної

довжини в графі [29, 30]. Велика трудомісткість цього алгоритму стала причиною розробки методу, що описується нижче.

Перш ніж перейти до формулювання задачі, дамо означення поняттю сумісності об'єктів.

Означення 1.1.1. *Два об'єкти є сумісними між собою, якщо їхне взаємне існування або функціонування можливе або доцільне.*

Прикладом сумісних об'єктів на практиці можуть бути: радіоелектронні засоби, що працюють у різних діапазонах електромагнітних хвиль; люди, що можуть спілкуватися між собою на якій-небудь мові; психологічно сумісні члени колективу; події, коли настання однієї не виключає можливість настання інших і т.п.

Грунтуючись на означенні 1.1.1, введемо поняття сумісної множини об'єктів.

Означення 1.1.2. *Множина об'єктів є сумісною тоді і тільки тоді, коли будь-яка пара об'єктів з цієї множини сумісна між собою.*

Для зручності формального опису завдання введемо *операцію сумісності* над елементами множини об'єктів. Ця операція є бінарною булевою операцією, позначимо її символом (\odot).

Сформулюємо властивості новоуведені операції, що випливають з означення сумісної множини. Нехай задана множина об'єктів $X = \{x_i\}, i \in I$, де $I = \{1, 2, \dots, n\}$ – множина індексів, тоді властивості операції сумісності можна сформулювати у такий спосіб:

- 1) $\forall i, j : i, j \in I [x_i \odot x_j = \text{"істина"}]$, якщо об'єкти x_i і x_j – сумісні між собою, і $[x_i \odot x_j = \text{"хіба"}]$ – у протилежному випадку. Поширивши цю властивість на множину об'єктів довільної потужності, одержимо, що $\bigodot_{i \in I} x_i = \text{"істина"}$, якщо всі об'єкти $x_i[i \in I]$ – сумісні між собою, і $\bigodot_{i \in I} x_i = \text{"хіба"}$, якщо в множині знайдеться хоча б одна несумісна пара об'єктів.
- 2) $\forall i, j : i, j \in I [x_i \odot x_j = x_j \odot x_i]$ (властивість симетричності).

$$3) \bigodot_{i \in I} x_i = \text{"істина"} \Rightarrow \bigodot_{\substack{j \in J \\ J \subset I}} x_j = \text{"істина"}$$

$$4) \forall k : k \in I \bigodot_{i \in I} x_i = x_k \odot \left(\bigodot_{\substack{i \in I \\ i \neq k}} x_i \right) = \left(\bigodot_{i \in I} x_i \right) \wedge \left(\bigwedge_{\substack{j \in I \\ j \neq k}} (x_j \odot x_k) \right).$$

Зазначимо, що оскільки операція сумісності визначена над елементами множини, то вираз $x_i \odot x_i$ – не має змісту, тому що в множині, по визначення, не може бути нерозрізних (однакових) елементів [78].

Тепер задачу можна сформулювати в такий спосіб:

Дано: множина об'єктів $A = \{a_i\}, i \in I$, де $I = \{1, 2, \dots, n\}$ – множина індексів;

P – множина пар сумісних об'єктів, тобто пар $\langle a_i; a_j \rangle [a_i, a_j \in A]$, для яких $a_i \odot a_j = \text{"істина"}$.

Знайти: $A_s \subseteq A (s \in I)$, такі, що

$$\left(\forall m : \left[a_m \in A_s \wedge \bigodot_{m \in I} a_m = \text{"істина"} \right] \wedge \forall k : \left[a_k \in A \wedge a_k \notin A_s \right] \right) \text{ справедлива}$$

рівність: $a_k \odot \left(\bigodot_{\substack{a_m \in A_s \\ m \in I}} a_m \right) = \text{"хіба"}$.

Нехай P^- множина пар несумісних об'єктів, тобто пар $\langle a_i; a_j \rangle [a_i, a_j \in A]$, для яких $a_i \odot a_j = \text{"хіба"}$. Тоді $P \cap P^- = \emptyset$ і $P \cup P^- = P_0$, де P_0 – множина усіх пар. При цьому зручніше викладати алгоритм, спираючись на множину P^- , а не на множину P .

Пропонується наступний метод знаходження підмножин A_s у множині A , який полягає у визначенні цих підмножин у процесі послідовного виключення пар з множини P^- .

Метод знаходження підмножин A_s у множині A є ітераційним, і кількість ітерацій дорівнює кількості елементів у множині P^- . Перед початком виконання алгоритму вважаємо, що усі пари об'єктів – сумісні, при цьому, природно, множина сумісних об'єктів співпадає з вихідною множиною ($A_s = A$). Нехай у результаті деякого експерименту встановлена несумісність пари $\langle a_k; a_l \rangle$. Знайдемо сумісні підмножини множини об'єктів при цій умові.

Дано: множина $A = \{a_i\}$ сумісних об'єктів, $i \in I$, де $I = \{1, 2, \dots, n\}$ – множина індексів, тобто $\bigodot_{i \in I} a_i = \text{"істина"}$.

Знайти: підмножини сумісних об'єктів максимальної потужності, при умові надходження інформації, що $a_k \odot a_l = \text{"хиба"} [k, l \in I]$.

Вирішення: до отримання інформації про несумісність пари $\langle a_k; a_l \rangle$ перетворимо вихідний вираз згідно зластивості 4 у кон'юнктивну форму:

$$\bigodot_{i \in I} a_i = a_k \odot \left(\bigodot_{\substack{i \in I \\ i \neq k}} a_i \right) = \left(\bigodot_{\substack{i \in I \\ i \neq k}} a_i \right) \wedge \left(\bigwedge_{\substack{i \in I \\ i \neq k}} (a_k \odot a_i) \right) = \text{"істина"}$$

Після отримання інформації про несумісність пари $\langle a_k; a_l \rangle$, тобто після того, як $a_k \odot a_l$ стане дорівнювати "хиба", для збереження істинності кон'юнктивної форми виключимо з неї $(a_k \odot a_l)$ і одержимо вираз:

$$\left(\bigodot_{\substack{i \in I \\ i \neq k}} a_i \right) \wedge \left(\bigwedge_{\substack{i \in I \\ i \neq k \neq l}} (a_k \odot a_i) \right) = \text{"істина"}$$

Згідно з властивістю 3, якщо $\bigodot_{\substack{i \in I \\ i \neq k}} a_i = \text{"істина"}$, то і $\bigodot_{\substack{i \in I \\ i \neq k \neq l}} a_i = \text{"істина"}$.

Тоді при доповненні кон'юнктивної форми останнім виразом її істинність не порушиться, її одержимо:

$$\left(\bigodot_{\substack{i \in I \\ i \neq k}} a_i \right) \wedge \left(\bigwedge_{\substack{i \in I \\ i \neq k \neq l}} (a_k \odot a_i) \right) \wedge \left(\bigodot_{\substack{i \in I \\ i \neq k \neq l}} a_i \right) = \text{"істина"},$$

і застосувавши властивість 4 до заключної частини кон'юнктивної форми одержимо:

$$\left(\bigodot_{\substack{i \in I \\ i \neq k}} a_i \right) \wedge \left(\bigodot_{\substack{i \in I \\ i \neq l}} a_i \right) = \text{"істина"}$$

Отже, для множини A сумісних об'єктів потужності n при виключенні пари $\langle a_k; a_l \rangle$ з множини сумісних пар, отримано 2 підмножини: $A_1 = \{a_i | i \in I, i \neq k\}$, і $A_2 = \{a_i | i \in I, i \neq l\}$ потужності $(n - 1)$ кожна. Очевидно, що отримана потужність множин – $(n - 1)$ – максимальна, оскільки на одиницю більшої потужності, а саме n , одержати неможливо, тому що одна з результатуючих підмножин у цьому випадку була б рівна

вихідній множині A , але це неможливо, бо A не може бути сумісною через $(a_k \odot a_l) = "хіба"$. Analogічне міркування можна застосувати, говорячи і про мінімальність кількості отриманих підмножин (їх 2).

Таким чином, у результаті проведеного кроку алгоритму, формально одержуємо виділення в множині сумісних об'єктів дві сумісні підмножини для випадку, коли надійшла інформація про несумісність яких-небудь двох об'єктів з множини.

Наслідуючи запропонований метод, будемо робити вищевикладену процедуру над усіма підмножинами, отриманими на попередньому кроці алгоритму, виключаючи із P^- відповідні пари несумісних об'єктів після кожного кроку алгоритму.

Крім того, з метою мінімізації кількості отриманих підмножин сумісних об'єктів, післяожної ітерації алгоритму необхідно зробити всі можливі поглинання отриманими множинами тих множин, що є підмножинами перших.

Визначимо *операцію поглинання* над множинами. Вона є бінарною операцією і буде позначена символом (\ominus) . Якщо K, L – довільні множини, то операція поглинання має сенс при виконанні умови, що $L \subseteq K$, і тоді $K \ominus L = K$.

Таким чином, в результаті поглинання виключаються множини меншої потужності і множини, що повторюються. Іншими словами, якщо розглянути кон'юнктивну форму, що відповідає підмножинам, отриманим у результаті роботи запропонованого алгоритму, то виконання всіх можливих операцій поглинання над підмножинами відповідає мінімізації цієї кон'юнктивної форми.

Повертаючись до алгоритму відмітимо, що ітеративний процес припиняється, коли множина P^- стане пустою ($P^- = \emptyset$).

Продемонструємо роботу алгоритму на конкретному прикладі. Нехай дана множина $A = \{a_i\}$ потужністю 5. Всі елементи a_i попарно сумісні між собою за виключенням пар елементів із множини $P^- = \{\langle a_2; a_3 \rangle, \langle a_3; a_5 \rangle\}$.

Для визначення підмножин сумісних об'єктів виконаємо наступні кроки:

- 1) розглянемо першу пару з множини $P^- = \langle a_2; a_3 \rangle$, і оскільки обидва елементи пари входять в множину $A = \{a_1, a_2, a_3, a_4, a_5\}$, то розділимо цю множину на дві підмножини $A_1 = \{a_1, a_3, a_4, a_5\}$ та $A_2 = \{a_1, a_2, a_4, a_5\}$;

- 2) виключимо першу пару з множини P^- ;
- 3) перевіряємо множини A_1 та A_2 на предмет можливих поглинань одна одною – таких не виявлено;
- 4) розглянемо другу пару з множини $P^- - \langle a_3; a_5 \rangle$ і розділимо множини, до яких входять одночасно обидва елементи цієї пари a_3 та a_5 . Такою виявилась множина A_1 , яку і розділимо на дві підмножини $A_{1/1} = \{a_1, a_4, a_5\}$ та $A_{1/2} = \{a_1, a_3, a_4\}$;
- 5) виключимо другу пару з множини P^- ;
- 6) перевіряємо множини $A_{1/1}, A_{1/2}$ та A_2 на предмет можливих поглинань – виявлено, що $A_{1/1} \subset A_2$, отже $A_{1/1}$ поглинається множиною A_2 ;
- 7) оскільки $P^- = \emptyset$, то алгоритм завершено.

В результаті роботи алгоритму визначили дві підмножини сумісних об'єктів $A_{1/2} = \{a_1, a_3, a_4\}$ та $A_2 = \{a_1, a_2, a_4, a_5\}$.

1.1.1. Два доповнення до алгоритму

У тому випадку, коли вже маємо результати роботи описаного вище алгоритму над деякою множиною об'єктів A' , тобто вже знайдені всі підмножини сумісних об'єктів, і при цьому виникає необхідність включити в A' новий додатковий елемент, то для знаходження сумісних підмножин можна значно скоротити кількість ітерацій в алгоритмічному процесі. Тобто пропонується не виконувати весь описаний алгоритм спочатку, а скористатися наступним алгоритмом:

- 1) включаємо новий елемент в усі отримані сумісні підмножини;
- 2) послідовно розглядаємо пари об'єктів, що не є сумісними (одним із елементів цих пар є новий елемент), і проводимо всі можливі поглинання;
- 3) закінчення алгоритму аналогічне.

Приклад: Припустимо, що маємо результати роботи алгоритму, що отримані в передньому прикладі, – підмножини сумісних об'єктів $A_{1/2} = \{a_1, a_3, a_4\}$ та $A_2 = \{a_1, a_2, a_4, a_5\}$ з множини A . Нехай в A додано елемент a_6 , який сумісний з усіма іншими елементами A за виключенням a_2 та a_4 , тобто $P^- = \{\langle a_2; a_6 \rangle, \langle a_4; a_6 \rangle\}$.

Виконаємо наступні кроки:

- 1) включимо новий елемент a_6 в підмножини $A_{1/2}$ та A_2 і отримаємо $A_{1/2} = \{a_1, a_3, a_4, a_6\}$ та $A_2 = \{a_1, a_2, a_4, a_5, a_6\}$;
- 2) розглянемо першу пару з множини $P^- - \langle a_2; a_6 \rangle$ і розділимо множини, до яких входять одночасно обидва елементи цієї пари. В нашому випадку – це єдина множина A_2 , яку і розділимо на дві підмножини $A_{2/1} = \{a_1, a_4, a_5, a_6\}$ та $A_{2/2} = \{a_1, a_2, a_4, a_5\}$.
- 3) виключимо розглянуту пару з множини P^- ;
- 4) перевіряємо всі підмножини $A_{1/2}, A_{2/1}$ та $A_{2/2}$ на предмет можливих поглинань – таких не виявлено;
- 5) розглянемо чергову пару з множини $P^- - \langle a_4; a_6 \rangle$ і розділимо множини до яких входять одночасно обидва елементи цієї пари. В нашому випадку – це множини $A_{1/2}$ та $A_{2/1}$, кожну з яких розділимо на дві підмножини і отримаємо: $A_{1/2/1} = \{a_1, a_3, a_6\}$ та $A_{1/2/2} = \{a_1, a_3, a_4\}$, $A_{2/1/1} = \{a_1, a_5, a_6\}$ та $A_{2/1/2} = \{a_1, a_4, a_5\}$.
- 6) виключимо розглянуту пару з множини P^- ;
- 7) перевіряємо всі підмножини $A_{1/2/1}, A_{1/2/2}, A_{2/1/1}, A_{2/1/2}$ та $A_{2/2}$ на предмет можливих поглинань – виявлено, що $A_{2/1/2} \subset A_{2/2}$, отже $A_{2/1/2}$ поглинається множиною $A_{2/2}$;
- 8) оскільки $P^- = \emptyset$, то алгоритм завершено.

Таким чином, в результаті роботи алгоритму визначено підмножини сумісних об'єктів: $A_{1/2/1} = \{a_1, a_3, a_6\}$, $A_{1/2/2} = \{a_1, a_3, a_4\}$, $A_{2/1/1} = \{a_1, a_5, a_6\}$ та $A_{2/2} = \{a_1, a_2, a_4, a_5\}$.

І ще один додатковий алгоритм корисний, коли є необхідність виключити елемент із вихідної множини об'єктів. У цьому випадку робимо наступні дії:

- 1) виключаємо даний елемент зі всіх отриманих сумісних підмножин;
- 2) робимо поглинання, якщо такі можливі.

Приклад: За вихідні дані візьмемо результат з попереднього прикладу. Нехай виключено елемент a_5 з вихідної множини A , тоді:

- 1) виключаємо елемент a_5 з усіх підмножин і отримуємо: $A_{1/2/1} = \{a_1, a_3, a_6\}$, $A_{1/2/2} = \{a_1, a_3, a_4\}$, $A_{2/1/1} = \{a_1, a_6\}$ та $A_{2/2} = \{a_1, a_2, a_4\}$;
- 2) перевіряємо всі підмножини $A_{1/2/1}$, $A_{1/2/2}$, $A_{2/1/1}$ та $A_{2/2}$ на предмет можливих поглинань – виявлено, що $A_{2/1/1} \subset A_{1/2/1}$, отже $A_{2/1/1}$ поглинається множиною $A_{1/2/1}$.

Результат: визначені підмножини сумісних об'єктів: $A_{1/2/1} = \{a_1, a_3, a_6\}$, $A_{1/2/2} = \{a_1, a_3, a_4\}$ та $A_{2/2} = \{a_1, a_2, a_4\}$.

На практиці описаний у цьому розділі алгоритм був використаний у СППР “Солон-2” [47] та “Солон-3” при виділенні в множині підцілей деякої цілі груп повністю сумісних між собою підцілей.

1.2. Метод парних порівнянь зі зворотним зв'язком з експертом і послідовною обробкою матриці парних порівнянь

Метод парних порівнянь широко застосовується при підтримці прийняття рішень для упорядкування альтернатив відповідно до деякого критерію [19–22]. Крім того, при використанні методів аналітичних ієрархічних процесів [26] або методу ієрархічних мережевих процесів [27] необхідно визначати кількісні показники ступеня пріоритетності альтернатив і критеріїв. Ці показники обчислюються з використанням множини результатів парних порівнянь. Метод динамічного цільового оцінювання [29,30,53,54] також потребує використання методів парних порівнянь.

При виконанні парних порівнянь, як правило, виникають протиріччя думок експерта, висловлених ним при порівнянні різних пар альтернатив. Якщо ці протиріччя великі, не можна використовувати результати обробки такої множини порівнянь. Для кількісної оцінки ступеня суперечливості матриці парних порівнянь можна використовувати показник узгодженості (вірніше, його варто називати показником неузгодженості, тому що при відсутності протиріч він дорівнює 0), запропонований Saatі [26,27]. При цьому результати обчислень величин пріоритетів альтернатив можуть бути використані, якщо величина цього показника менше деякого граничного значення. У протилежному випадку експерту пропонується повторити процедуру порівнянь усіх пар альтернатив. Ця спроба знову може виявитися невдалою, тому що експерт не знає джерела протиріч. Інакше кажучи, він не знає, результати порівняння яких пар викликають протиріччя. Тому бажано запровадити зворотний зв'язок з експертом, за допомогою якого йому можна було б показувати джерела протиріч.

Слід зазначити, що протиріччя оцінок експерта можуть бути викликані як природною суперечливістю проявів властивостей альтернатив, так і впливом на експерта різних психофізіологічних факторів. Протиріччя першого типу виникають, наприклад, при порівнянні сили трьох команд за результатами їхніх ігор між собою, коли команда A виграла в команди B , команда B виграла у команди C , а команда C виграла в команди A). Другий тип протиріч викликається, наприклад, утомою, неуважністю експерта, впливом на нього зовнішніх подразників і т.д. Зрозуміло, що для підвищення вірогідності результатів парних порівнянь необхідно вказати експерту джерело протиріч, що винikли, і надати йому можливість відкоригувати результати, якщо на його думку протиріччя викликані факторами, що заважають.

У даному розділі пропонується метод парних порівнянь зі зворотним зв'язком з експертом, який має названі властивості.

Друга проблема, що досліджується, зв'язана зі способом визначення переваги однієї альтернативи над іншою щодо деякої загальної властивості або критерію. У дослідженнях Saatі [26,27] і ряді ініційованих ними робіт запропоновано використовувати при проведенні парних порівнянь так звану фундаментальну шкалу, що має

наступні значення ступеня переваги: “еквівалентність (1)”, “слабка перевага (2)”, “помірна перевага (3)”, “більше, ніж помірна (4)”, “сильна перевага (5)”, “досить сильна перевага (6)”, “дуже сильне перевага (7)”, “дуже, дуже сильна перевага (8)”, “надзвичайно сильна перевага (9)”. У дужках наведені числові еквіваленти ступенів переваги. Рекомендується визначати ці числа в такий спосіб [27]: “числа показують, *у скільки разів* один об'єкт перевершує інший щодо загальної властивості або критерію”.

Такий підхід зручний, коли розходження між альтернативами у відношенні до якої загальної властивості досить велики (наприклад, коли дід і онук порівнюються за віком). Однак, якщо розходження між альтернативами – невеликі (наприклад, при порівнянні за віком учнів одного класу), для представлення результатів порівнянь зручніше використовувати числа, що показують на скільки одна альтернатива перевершує іншу. У цьому випадку матриця порівнянь уже не має чудової властивості, відповідно до якої $a_{ij} = 1/a_{ji}$. Тому методи обробки цієї матриці, засновані на використанні такої властивості [26, 27], виявляються непридатними.

У даній роботі пропонується новий метод обробки результатів парних порівнянь спочатку для загального випадку, а потім для двох важливих у практичному відношенні окремих випадків.

1.2.1. Сутність методу

Нехай експерт здійснює парні порівняння альтернатив A_1, A_2, \dots, A_m з метою визначення нормалізованих значень відносних пріоритетів цих альтернатив за деяким критерієм C . Результат порівняння альтернатив A_i і A_h експерт виражає у вигляді ступеня d_{ih} переваги альтернативи A_i над альтернативою A_h відносно цього критерію. Визначення d_{ih} здійснюється шляхом порівняння абсолютноного значення, що *інтуїтивно* відчувається експертом, v_i деякої “ваги” (відносно критерію C) альтернативи A_i зі значенням v_h аналогічної “ваги” альтернативи A_h . Тому можна припустити, що існує деяка функція

$$d_{ij} = \varphi(v_i, v_j)$$

і зворотна їй функція

$$v_j = f(v_i, d_{ij}). \quad (1.1)$$

Ці функції можна визначити, виходячи зі зручності вираження експертом своїх переваг альтернатив. Наприклад, якщо відома одиниця виміру ступеня виразності властивості, що задається критерієм, і розходження між альтернативами відносно цього критерію невеликі, то зручно використовувати d_{ih} для вираження того, наскільки більше виражена ця властивість в одній альтернативі в порівнянні з іншою.

Запропонуємо експерту виконати порівняння i -ої альтернативи зі всіма іншими $(r-1)$ альтернативами відповідно до обраної функції (1.1). Результати цих порівнянь утворюють деяку підмножину $DR_i = \{d_{ih}\}, h = (1, r)$, що сформує i -й рядок матриці парних порівнянь (МПП). Повторюючи цей процес для всіх $i, 1 \leq i \leq r$, одержимо повну матрицю парних порівнянь.

Визначимо найменший елемент d_{ih} i -го рядка МПП. Нехай функція $f(v_i, d_{ih})$ така, що при зменшенні d_{ih} вага альтернативи A_h збільшується. Тоді з мінімальності d_{ih} випливає, що альтернатива $A_h (1 \leq h \leq r)$ має найбільшу серед всіх альтернатив вагу v_h^i , обчислену по i -му рядку матриці. Назовемо цю альтернативу домінуючою по i -му рядку і позначимо її через A_{ph} , а її вагу – через v_{ph}^i . Далі запропонуємо експерту визначити вагу v_{ph}^i методом безпосереднього оцінювання альтернативи A_h .

Використовуючи вираз

$$v_i^i = f(v_{ph}^i, d_{ih}), \quad (1.2)$$

визначимо по елементах i -го рядка ненормалізоване значення ваги i -ої альтернативи. Будемо називати v_i^i власною вагою альтернативи A_i .

З виразу

$$v_h^i = f(v_i^i, d_{ih}) \quad (1.3)$$

визначимо ваги v_h^i всіх інших альтернатив. Ці значення надалі будемо називати транзитивними.

Повторюючи цей процес для всіх рядків МПП, для кожної альтернативи A_h одержимо множину $V_h = v_h^i, i = (1, r)$ ненормалізованих значень ваг.

Округлимо кожне з них до найближчої поділки шкали з границями $(0,1)$ і n поділками ($n = 1/2 \cdot \varepsilon + 1$, де $\varepsilon \in$ припустима похибка). Отримані в такий спосіб величини $v_h^i \in V_h$ зручно представити n -компонентним вектором $S_h = \{s_{hh}\}, h = (0, n)$, де s_{hh} дорівнює кількості значень v_h^i , округлених до h -ої поділки шкали.

Назвемо цей вектор *спектром оцінок* ваги v_h альтернативи A_h .

Розглянемо конкретний приклад. Для МПП, заданої табл.1.1,

Таблиця 1.1.

	A_1	A_2	A_3	A_4
A_1	0	0.6	-0.1	0.5
A_2	-0.8	0	-0.6	-0.1
A_3	0	0.4	0	0.4
A_4	-0.5	0.2	-0.3	0

і функції 1.3, що має вигляд:

$$v_h^i = v_i^i - d_{ih};$$

при $n = 11$ маємо: $V_1 = \{0.8; 1.0; 0.9; 0.9\}$, $V_2 = \{0.2; 0.2; 0.5; 0.2\}$, $V_3 = \{0.9; 0.8; 0.9; 0.7\}$, $V_4 = \{0.3; 0.3; 0.5; 0.4\}$, $S_1 = (0; 0; 0; 0; 0; 0; 0; 1; 2; 1)$, $S_2 = (0; 0; 3; 0; 0; 1; 0; 0; 0; 0; 0)$, $S_3 = (0; 0; 0; 0; 0; 0; 0; 1; 1; 2; 0)$, $S_4 = (0; 0; 0; 2; 1; 1; 0; 0; 0; 0; 0)$.

Для наочності зручно представляти спектр S_h діаграмою. Вона являє собою вісь h , на якій у точках w, r, \dots, t встановлені перпендикуляри довжиною sw_h, sr_h, \dots, st_h (рис. 1.1).

Наявність у спектрі оцінок ваги альтернативи декількох компонент, що знаходяться на різних поділках шкали, свідчить про наявність протиріч у результатах парних порівнянь. Розглянемо компоненти s_1, s_2, \dots, s_r спектра S_h як оцінки ваги альтернативи A_h , дані r різними експертами з однаковою компетентністю. Для кількісної оцінки ступеня цих протиріч можна використовувати коефіцієнт узгодженості, запропонований у [52]. Значення цього коефіцієнта визначається наступним виразом:

$$k(S_i) = \left(1 - \frac{1/r \sum_{z=1}^n s_z \left|z/n - 1/r \sum_{z=1}^n z/n s_z\right| - \sum_{z=1}^n s_z/r \ln(s_z/r)}{G \sum_{z=1}^n \left|z/n - \frac{n+1}{2n}\right| + \ln(n)}\right)Y, \quad (1.4)$$

де r – кількість альтернатив;

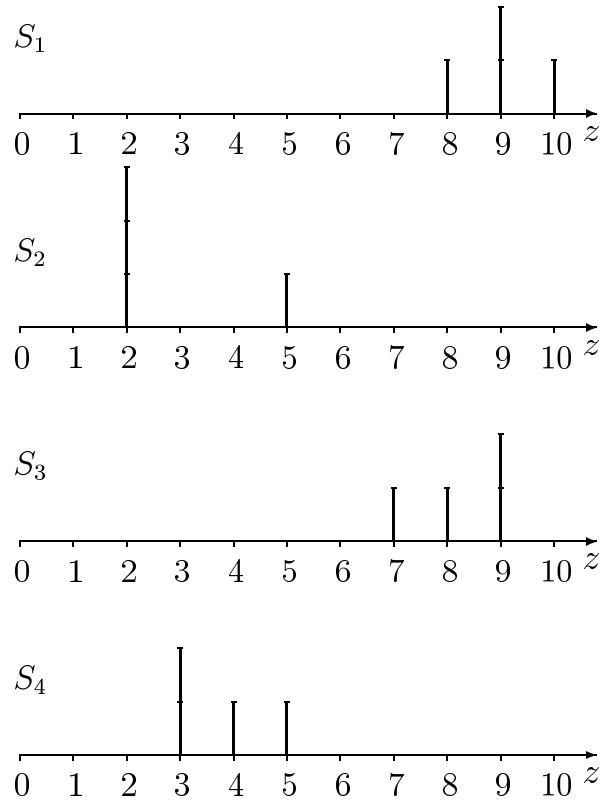


Рис. 1.1. Спектри оцінок ваг альтернатив. Через $S_i, i = (1, 4)$ позначений спектр оцінок ваги i -ої альтернативи.

n – число поділок шкали;

s_z – число значень ваги, округлених до z -ої поділки шкали;

$$G = \frac{n}{\ln(n)r \ln(r)} \text{ – масштабний коефіцієнт;}$$

$$Y = \begin{cases} 1, & \text{якщо } Y^* = \text{"истина"} \\ 0, & \text{якщо } Y^* = \text{"хіба"} \end{cases}$$

$$Y^* = \overline{(s_1 = s_n = r/k)}_{b=1}^{k-1} \overline{(s_{z_b} = s_{z_{b+1}})}_{b=1}^{k-1} \overline{(z_{b+1} - z_b)} = const;$$

де z_b – номер поділки шкали, до якої заокруглено значення ваги $b = (1, k)$;

k – число підмножин однакових значень ваги.

Використовуючи (1.4), обчислимо $k(S_h)$ для $h = (1, r)$. Для визначення достатності ступеня узгодженості спектра S_h обчислимо значення порога виявлення T_o і порога застосування T_u [52].

Поріг виявлення визначається виразом: $T_o = k(S_o)$, де S_o являє собою спектр значень ваги, у якому: $r = n$; $s_o = 0$; $s_g = 2$; $s_{y \neq g \neq 0} = 1$; $g = [n/2 + 1]$; поріг застосування T_u дорівнює коефіцієнту узгодженості $k(S_u)$ спектра S_u , у якому $r = 2$, $s_i = s_{i+1} = 1$ і $s_{h \neq i \wedge h \neq i+1} = 0$.

Якщо

$$\exists x : 1 \leq x \leq r [k(S_x) < T_o], \quad (1.5)$$

то спектр S_x не несе корисної інформації, тобто являє собою “інформаційний шум”. У цьому випадку експерту пропонується повторити парні порівняння всіх альтернатив з альтернативою A_x , тобто визначити нові значення всіх d_{ix} , $i = (1, r)$, $i \neq x$, що утворюють стовпець матриці порівнянь, позначений альтернативою A_x . Потім необхідно повторювати процес, описаний вище, поки буде виконуватися умова (1.5).

Припустимо, що $\forall x : (1 \leq x \leq r) [k(S_x) \geq T_o]$, але існує підмножина $S^* \subset S, S^* \neq \emptyset$ спектрів, для якої з $S_h \in S^*$ випливає, що

$$k(S_h) < T_u. \quad (1.6)$$

У цьому випадку всі спектри S_x містять корисну інформацію, але точність визначення ваг альтернатив, обчислених по спектрах з підмножини S^* недостатня. Використовуючи припустимі операції для шкали, що розглядається (див. [22]), обчислимо середнє значення a_h ваг альтернативи A_h для спектра $S_h \in S^*$ і визначимо множину D_h значень ваг v_h^i , найбільш віддалених від a_h . Цю процедуру необхідно виконати для всіх $S_h \in S^*$.

Розділімо множину $D = \bigcup_{j=1}^r D_j$ на три підмножини D^p, D^o, D^t . Значення v_{px}^i ваг домінуючих альтернатив включаються в підмножину D^p ; D^o – це підмножина власних ваг альтернатив. Компоненти спектра, що є транзитивними вагами, включаються в підмножину D^t . Упорядкуємо підмножини D^p, D^o, D^t відповідно до зменшення їхніх відмінностей від середніх значень ваг.

З (1.4) випливає, що збільшити коефіцієнт узгодженості будь-якого спектра можна за рахунок зменшення відмінностей його компонент від середнього значення. Для зміни значень v_j^i або v_i^i експерт повинен змінити величини d_{ij} або d_{ip} відповідно.

Зміна значення v_p^i досягається переглядом оцінки ваги домінуючої альтернативи, даної експертом.

Обумовимо алгоритм, що дозволяє визначити послідовність, у якій експерту варто пропонувати виконати ці зміни. Цей алгоритм повинен бути умовним, тому що експерт може відмовитися змінити деякі значення. Він повинен дозволяти за кінцеве число кроків визначити з прийнятною точністю значення ваг альтернатив, або зробити висновок про те, що таку множину ваг по оцінках, даним цим експертом, не можна визначити.

Процес підвищення коефіцієнта узгодженості починається з елементів підмножини D^p , тому що зміна v_p^i приводить до змін значень v_i^i і v_p^i . Визначимо спектр V_x , якому відповідає перший елемент v_{px}^i підмножини D^p . Тому що в цьому випадку підвищення значення $k(V_x)$ може бути досягнуто зменшенням різниці між a_x і v_{px}^i , запропонуємо експерту змінити оцінку ваги домінуючої альтернативи A_x . Напрямок пропонованої зміни визначається знаком різниці між a_x і v_{px}^i . Після цього відповідно до нового значення v_{px}^i для всіх спектрів обчислюються нові значення компонент, що мають верхній індекс i . Для цього використовуються описані вище процедури. Якщо експерт відмовився змінити значення v_{px}^i , то значення v_{px}^i виключається з підмножини D^p , і якщо $D^p \neq \emptyset$, то ця операція виконується для другого елемента підмножини D^p і т.д.; у протилежному випадку переходять до перетворення підмножини D^o .

Нехай $v_y^y \in D^o$ – перший елемент цієї підмножини. У цьому випадку збільшення значення $k(V_y)$ може бути досягнуто зменшенням різниці між a_y і v_y^y . Для цього варто запропонувати експерту змінити ступінь переваги d_{yp} домінуючої альтернативи A_p над альтернативою A_y . Напрямок цієї зміни (збільшення або зменшення) визначається типом функції (1.2) і знаком різниці між a_y і v_y^y . Після цього відповідно до нового значення v_y^y для всіх спектрів обчислюються нові значення компонент, що мають верхній індекс y , і описана вище процедура повторюється. Якщо експерт відмовився змінити значення d_{yp} , то компонента v_y^y виключається з підмножини D^o . Якщо $D^o \neq \emptyset$, то описана операція виконується для другого елемента підмножини D^o і т.д., інакше починається перетворення підмножини D_t . Відзначимо, що якщо

компоненти v_{px}^i, v_y^y аналізувалися на q -ом кроці алгоритму, то на $(q + k)$ -му ($k \geq 1$) кроці вони не будуть аналізуватися, тому що до початку цього кроку вони будуть виключені з множини D .

Операція перетворення підмножини D^t аналогічна описаній операції перетворення підмножини D^o за тим тільки виключенням, що в цьому випадку експерту пропонується змінити ступінь переваги d_{iz} альтернативи A_i над альтернативою A_z , якщо v_{zi} – перший елемент підмножини D^t . Якщо експерт відмовився змінити d_{iz} , то значення v_z^i виключається з підмножини D^t . Якщо $D^t \neq \emptyset$, то ця операція виконується для другого елемента підмножини D^t і т.д., інакше діалог з експертом припиняється. Якщо при цьому хоча б для одного спектра виконується умова (1.6), потрібно зробити висновок про те, що результати парних порівнянь, виконаних цим експертом, не можуть бути використані, інакше величини a_j ($1 \leq j \leq r$) приймаються як ненормалізовані значення ваг альтернатив. Нормалізовані значення цих ваг визначаються зі співвідношення:

$$w_x = a_x / \sum_{j=1}^r a_j.$$

Зазначимо, що після кожної корекції значень v_{px}^i або v_y^y або v_z^i необхідно обчислити нові значення ваг всіх альтернатив, використовуючи функції (1.2), (1.3) і відкоригувати всі спектри S_j для всіх $1 \leq j \leq r$. Якщо умова (1.6) виконується хоча б для одного спектра, необхідно повторно виконати описаний вище алгоритм.

Слід відзначити, що при використанні цього методу “тиск” на експерта не відбувається, тому що в нього зберігається вільний вибір прийняття рішень на кожному кроці діалогу. Метод дозволяє знайти наявність протиріч в оцінках експерта і вказати напрямок їхнього зменшення.

1.2.2. Стабільність алгоритму

Для доведення стабільності описаного вище алгоритму досить переконатися в тім, що він за кінцеве число кроків дозволяє з прийнятною точністю визначити множину ваг альтернатив або зробити висновок про її відсутність.

Як випливає з описаного вище алгоритму, збільшення коефіцієнта узгодженості

$k(V_j)$ може бути досягнуто відповідно зміною домінуючої (v_{pj}^i) і/або власної (v_j^j) і/або транзитивних (v_j^i) компонент ваги альтернативи A_j . З (1.2), (1.3) випливає, що зміна домінуючої або власної ваги однієї альтернативи приводить до змін транзитивних ваг всіх альтернатив, тобто до зміни по одній компоненті в спектрі ваг кожної альтернативи. Тому зміна, наприклад, v_{pj}^i може викликати збільшення $k(V_j)$ і, одночасно, зменшення $k(V_{h \neq j})$. Однак, як випливає з алгоритму, змінити домінуючу або власну вагу альтернативи експерту може бути запропоновано тільки один раз. Тому для доведення стабільності алгоритму досить переконатися в тім, що $k(V_h)$ не зменшується, якщо $k(V_j) \neq V_h$ збільшується за рахунок зміни експертом транзитивних ваг.

Для збільшення $k(V_j)$ необхідно і достатньо збільшити значення v_j^i для деяких i і зменшити v_j^i для інших. Розглянемо випадок, коли $\partial v_j^i / d_{ij} < 0$. При цьому для зменшення v_j^i потрібно запропонувати експерту збільшити d_{ij} . При збільшенні d_{ij} кількість m домінуючих альтернатив у i -му рядку МПП не може змінитися і, як випливає з (1.3), не може змінитися жодна з компонент v_h^k для $k \neq i, h \neq j$. Тому при такій зміні d_{ij} може змінитися тільки значення $k(V_j)$.

Якщо для збільшення $k(V_j)$ необхідно збільшити v_j^i шляхом зменшення d_{ij} , то

$$d_{ij} \geq d_{ip} \quad (1.7)$$

є достатньою умовою того, щоб v_i^i не змінилося. Тому (1.7) – достатня умова того, що усі компоненти $v_h^k, k \neq i, h \neq j$ будуть незмінні, а зміниться тільки значення $k(V_j)$.

Доведення при $\partial v_j^i / d_{ij} > 0$ аналогічне.

Описаний вище алгоритм може бути використаний для різних функцій (1.2), (1.3). Розглянемо два важливі особливих випадки. Перший пов'язаний з використанням довільної шкали. В другому випадку експерт відповідає на питання “на скільки” одна альтернатива перевершує іншу щодо деякого критерію.

1.2.3. Приклад використання довільної шкали

У цьому випадку d_{ij} є число, що визначає ступінь переваги однієї альтернативи над іншою; $d_{ij} > 0$, якщо альтернатива A_i – більш значима, ніж A_j , і $d_{ij} < 0$ – у проти-

лежному випадку; $d_{ij} = 1$, якщо альтернативи – еквівалентні.

Визначимо функцію $f(v_j, d_{ji})$. З означення поняття “еквівалентність” випливає необхідність виконання умови:

$$f(v_i, 1) = v_i. \quad (1.8)$$

Задамо також умову сталості відносної похибки обчислення значення v_i у вигляді:

$$f(v_i, d_m) = v_i \cdot \varepsilon, \quad (1.9)$$

де $\varepsilon > 0$ – відносна похибка обчислення значення v_i ;

d_m – максимальне значення ступеня переваги.

Будемо шукати вираз для функції $f(v_i, d_{ij})$ у формі:

$$f(v_i, d_{ij}) = z \cdot v_i + u. \quad (1.10)$$

З (1.8) - (1.10) після нескладних перетворень одержуємо:

$$v_j = v_i \frac{d_m - \varepsilon - d_{ij} (1 - \varepsilon)}{d_m - 1}. \quad (1.11)$$

При цьому функції (1.2), (1.3) мають вигляд:

$$v_i^i = v_{pj}^i \frac{d_m - 1}{d_m - \varepsilon - d_{ij} (1 - \varepsilon)}, \quad (1.12)$$

$$v_h^i = v_i^i \frac{d_m - \varepsilon - d_{ih} (1 - \varepsilon)}{d_m - 1}. \quad (1.13)$$

Використовуючи (1.11), одержуємо:

$$(v_j - v_a)/(v_b - v_c) = (d_{ia} - d_{ij})/(d_{ic} - d_{ib}). \quad (1.14)$$

З (1.14) випливає, що ваги альтернатив, обчислені відповідно до виразу (1.11), представлені в шкалі інтервалів. Тому для знаходження середньої оцінки ваги може бути застосована операція знаходження середнього арифметичного [21]. Нехай табл. 1.2 задає МПП. Тоді результати виконання кроків алгоритму в стислому вигляді наведені в табл. 1.3.

Опишемо докладніше дані, що розміщені в цій таблиці. Отже, спочатку, на першому кроці алгоритму розглядаємо 1-й рядок МПП (див. табл. 1.2) і знаходимо в

Таблиця 1.2.

	A_1	A_2	A_3	A_4
A_1	1	-2	5	-8
A_2	3	1	6	-2
A_3	-2	-4	1	-6
A_4	7	3	6	1

ньому домінуючий (мінімальний) елемент. Таким елементом є четвертий елемент рядка. Присвоїмо вазі 4-ї альтернативи значення 1.0 ($v_4^1 = 1.0$). Скориставшись (1.12), знаходимо значення v_1^1 і, скориставшись (1.13), знаходимо v_2^1 та v_3^1 .

Переходимо до другого рядка матриці. Знаходимо в ньому домінуючий елемент – четвертий, присвоюємо вазі відповідної альтернативи значення 1.0 ($v_4^2 = 1.0$). Підставивши це значення в (1.12), знаходимо v_2^2 і з (1.13) – v_1^2 та v_3^2 .

По аналогії, розглядаючи 3-й і 4-й рядки матриці, знаходимо значення v_4^3 , v_3^3 , v_1^3 , v_2^3 , v_4^4 , v_1^4 , v_2^4 та v_3^4 .

Сформуємо вектор V_1 , елементами якого є округлені до найближчої поділки шкали значення v_1^1 , v_1^2 , v_1^3 та v_1^4 ($V_1 = \{0.5; 0.6; 0.7; 0.3\}$). Спектр S_1 у цьому випадку буде мати чотири складові одиничної довжини на відповідних поділках – 0.3; 0.5; 0.6; 0.7, тобто $S_1 = \{0; 0; 0; 1; 0; 1; 1; 1; 0; 0; 0\}$.

Далі, використовуючи вираз (1.4), знайдемо значення коефіцієнта узгодженості для цього спектра – $k(S_1) = 0.536$ і порівняємо з обчисленими на початку алгоритму порогом виявлення ($T_o = 0.398$) та порогом застосування ($T_u = 0.79$) (пороги обчислюються один раз на початку роботи алгоритму, бо їхні значення залежать лише від кількості поділок шкали в спектрі – n і не залежать від виду конкретного спектра). В прикладі, що розглядається, $T_o < k(S_1) \leq T_u$ – узгодженість недостатня, тому обчислимо деякі величини, які будемо використовувати в процесі покращення узгодженості. Це a_1 – середнє арифметичне точних (не округлених) значень ваг v_1^1 , v_1^2 , v_1^3 та v_1^4 ($a_1 = 0.431$) та δ_{max1} – максимальне з відхилень точних значень ваг від середнього значення a_1 (в даному прикладі максимальним є відхилення ваги v_1^4 і $\delta_{max1} = 0.231$).

З ваг, які максимально відхиляються від середнього значення кожного з недоста-

Таблиця 1.3.

$d_m = 9$	$\varepsilon = 0.05$	$T_o = 0.398$	$T_u = 0.790$	при $n = 11$
крок 1				
1 рядок	$v_p = v_4^1 = 1.000$	$v_1^1 = 0.483$	$v_2^1 = 0.656$	$v_3^1 = 0.254$
2 рядок	$v_p = v_4^2 = 1.000$	$v_2^2 = 0.737$	$v_1^2 = 0.562$	$v_3^2 = 0.300$
3 рядок	$v_p = v_4^3 = 1.000$	$v_3^3 = 0.546$	$v_1^3 = 0.741$	$v_2^3 = 0.870$
4 рядок	$v_p = v_4^4 = 1.000$	$v_1^4 = 0.288$	$v_2^4 = 0.763$	$v_3^4 = 0.406$
$V_1 = \{$	$\boxed{0.5} \quad 0.6 \quad 0.7 \quad 0.3 \}$		$S_1 = \{0; 0; 0; 1; 0; 1; 1; 0; 0; 0\}$	
$k(S_1) =$	0.536	$a_1 = 0.518$	$\delta_{max1} = a_1 - v_1^4 = 0.231$	
$V_2 = \{$	$0.7 \quad \boxed{0.7} \quad 0.9 \quad 0.8 \}$		$S_2 = \{0; 0; 0; 0; 0; 0; 2; 1; 1; 0\}$	
$k(S_2) =$	0.685	$a_2 = 0.756$	$\delta_{max2} = a_2 - v_2^3 = 0.114$	
$V_3 = \{$	$0.3 \quad 0.3 \quad \boxed{0.5} \quad 0.4 \}$		$S_3 = \{0; 0; 0; 2; 1; 1; 0; 0; 0; 0\}$	
$k(S_3) =$	0.685	$a_3 = 0.376$	$\delta_{max3} = a_3 - v_3^3 = 0.170$	
$V_4 = \{$	$\boxed{1.0} \quad \boxed{1.0} \quad \boxed{1.0} \quad \boxed{1.0} \}$		$S_4 = \{0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 4\}$	
$k(S_4) =$	$1.000 > T_u$	$\Rightarrow a_i$	δ_{max}	не обчислюються
$D = \{v_1^4, v_2^3, v_3^3\};$				
$D^p = \{\emptyset\}; D^o = \{v_3^3\}; D^t = \{v_1^4, v_2^3\};$				
v_3^3 потрібно $\downarrow \Rightarrow d_{34} - \downarrow$; Згодні? Так. $d_{34} := d_{34} - \Delta^* = -9$				
крок 2				
1 рядок	$v_p = v_4^1 = 1.000$	$v_1^1 = 0.483$	$v_2^1 = 0.656$	$v_3^1 = 0.254$
2 рядок	$v_p = v_4^2 = 1.000$	$v_2^2 = 0.737$	$v_1^2 = 0.562$	$v_3^2 = 0.300$
3 рядок	$v_p = v_4^3 = 1.000$	$v_3^3 = 0.457$	$v_1^3 = 0.620$	$v_2^3 = 0.729$
4 рядок	$v_p = v_4^4 = 1.000$	$v_1^4 = 0.288$	$v_2^4 = 0.763$	$v_3^4 = 0.406$
$V_1 = \{$	$\boxed{0.5} \quad 0.6 \quad 0.6 \quad 0.3 \}$		$S_1 = \{0; 0; 0; 1; 0; 1; 2; 0; 0; 0\}$	
$k(S_1) =$	0.641	$a_1 = 0.488$	$\delta_{max1} = a_1 - v_1^4 = 0.201$	
$V_2 = \{$	$0.7 \quad \boxed{0.7} \quad 0.7 \quad 0.8 \}$		$S_2 = \{0; 0; 0; 0; 0; 0; 3; 1; 0; 0\}$	
$k(S_2) =$	$0.835 > T_u$			
$V_3 = \{$	$0.3 \quad 0.3 \quad \boxed{0.5} \quad 0.4 \}$		$S_3 = \{0; 0; 0; 2; 1; 1; 0; 0; 0; 0\}$	
$k(S_3) =$	0.685	$a_3 = 0.354$	$\delta_{max3} = a_3 - v_3^3 = 0.103$	
$V_4 = \{$	$\boxed{1.0} \quad \boxed{1.0} \quad \boxed{1.0} \quad \boxed{1.0} \}$		$S_4 = \{0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 4\}$	
$k(S_4) =$	$1.000 > T_u$			
$D = \{v_1^4, v_3^3\};$				
$D^p = \{\emptyset\}; D^o = \{v_3^3\}; D^t = \{v_1^4\};$				
v_3^3 потрібно $\downarrow \Rightarrow d_{34} - \downarrow$;				
d_{34} має бути $\geq -9 \Rightarrow$ зменшення d_{34} неможливе $\Rightarrow v_3^3$ виключається з D ;				

* Δ – величина кроку зміни переваг $\boxed{\square}$ – ваги домінуючих альтернатив; \square – власні ваги альтернатив;

Таблиця 1.3. продовження

крок 3

1 рядок	$v_p = v_4^1 = 1.000$	$v_1^1 = 0.483$	$v_2^1 = 0.656$	$v_3^1 = 0.254$
2 рядок	$v_p = v_4^2 = 1.000$	$v_2^2 = 0.737$	$v_1^2 = 0.562$	$v_3^2 = 0.300$
3 рядок	$v_p = v_4^3 = 1.000$	$v_3^3 = 0.457$	$v_1^3 = 0.620$	$v_2^3 = 0.729$
4 рядок	$v_p = v_4^4 = 1.000$	$v_1^4 = 0.288$	$v_2^4 = 0.763$	$v_3^4 = 0.406$
$V_1 = \{$	$\boxed{0.5}$	0.6	0.6	$\} S_1 = \{0; 0; 0; 1; 0; 1; 2; 0; 0; 0; 0\}$
$k(S_1) =$	0.641	$a_1 =$	0.488	$\delta_{max1} = a_1 - v_1^4 = 0.201$
$V_2 = \{$	0.7	$\boxed{0.7}$	0.7	$\} S_2 = \{0; 0; 0; 0; 0; 0; 3; 1; 0; 0\}$
$k(S_2) =$	0.835	$> T_u$		
$V_3 = \{$	0.3	0.3	$\boxed{0.5}$	$\} S_3 = \{0; 0; 0; 2; 1; 1; 0; 0; 0; 0; 0\}$
$k(S_3) =$	0.685	$a_3 =$	0.354	$\delta_{max3} = a_3 - v_3^1 = 0.100$
$V_4 = \{$	$\boxed{1.0}$	$\boxed{1.0}$	$\boxed{1.0}$	$\} S_4 = \{0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 4\}$
$k(S_4) =$	1.000	$> T_u$		
$D = \{v_1^4, v_3^1\};$				
$D^p = \{\emptyset\}; D^o = \{\emptyset\}; D^t = \{v_1^4, v_3^1\};$				
v_1^4 потрібно $\uparrow \Rightarrow d_{41} - \downarrow$; Згодні? Так. $d_{41} := d_{41} - \Delta = 6$				

крок 4

1 рядок	$v_p = v_4^1 = 1.000$	$v_1^1 = 0.483$	$v_2^1 = 0.656$	$v_3^1 = 0.254$
2 рядок	$v_p = v_4^2 = 1.000$	$v_2^2 = 0.737$	$v_1^2 = 0.562$	$v_3^2 = 0.300$
3 рядок	$v_p = v_4^3 = 1.000$	$v_3^3 = 0.457$	$v_1^3 = 0.620$	$v_2^3 = 0.729$
4 рядок	$v_p = v_4^4 = 1.000$	$v_1^4 = 0.406$	$v_2^4 = 0.763$	$v_3^4 = 0.406$
$V_1 = \{$	$\boxed{0.5}$	0.6	0.4	$\} S_1 = \{0; 0; 0; 1; 1; 2; 0; 0; 0; 0\}$
$k(S_1) =$	0.685	$a_1 =$	0.518	$\delta_{max1} = a_1 - v_1^4 = 0.112$
$V_2 = \{$	0.7	$\boxed{0.7}$	0.7	$\} S_2 = \{0; 0; 0; 0; 0; 0; 3; 1; 0; 0\}$
$k(S_2) =$	0.835	$> T_u$		
$V_3 = \{$	0.3	0.3	$\boxed{0.5}$	$\} S_3 = \{0; 0; 0; 2; 1; 1; 0; 0; 0; 0; 0\}$
$k(S_3) =$	0.685	$a_3 =$	0.354	$\delta_{max3} = a_3 - v_3^1 = 0.100$
$V_4 = \{$	$\boxed{1.0}$	$\boxed{1.0}$	$\boxed{1.0}$	$\} S_4 = \{0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 4\}$
$k(S_4) =$	1.000	$> T_u$		
$D = \{v_1^4, v_3^1\};$				
$D^p = \{\emptyset\}; D^o = \{\emptyset\}; D^t = \{v_1^4, v_3^1\};$				
v_1^4 потрібно $\uparrow \Rightarrow d_{41} - \downarrow$; Згодні? Так. $d_{41} := d_{41} - \Delta = 5$				

Таблиця 1.3. продовження

крок 5

1 рядок	$v_p = v_4^1 = 1.000$	$v_1^1 = 0.483$	$v_2^1 = 0.656$	$v_3^1 = 0.254$
2 рядок	$v_p = v_4^2 = 1.000$	$v_2^2 = 0.737$	$v_1^2 = 0.562$	$v_3^2 = 0.300$
3 рядок	$v_p = v_4^3 = 1.000$	$v_3^3 = 0.457$	$v_1^3 = 0.620$	$v_2^3 = 0.729$
4 рядок	$v_p = v_4^4 = 1.000$	$v_1^4 = 0.525$	$v_2^4 = 0.763$	$v_3^4 = 0.406$
$V_1 = \{$	$\boxed{0.5}$	0.6	0.6	$\} \quad S_1 = \{0; 0; 0; 0; 0; 2; 2; 0; 0; 0; 0\}$
$k(S_1) =$	0.790	$= T_u$		
$V_2 = \{$	0.7	$\boxed{0.7}$	0.7	0.8
$k(S_2) =$	0.835	$> T_u$		
$V_3 = \{$	0.3	0.3	$\boxed{0.5}$	0.4
$k(S_3) =$	0.685	$a_3 = 0.354$	$\delta_{max3} = a_3 - v_3^1 = 0.100$	
$V_4 = \{$	$\boxed{1.0}$	$\boxed{1.0}$	$\boxed{1.0}$	$\boxed{1.0}$
$k(S_4) =$	1.000	$> T_u$		
$D = \{v_3^1\};$				
$D^p = \{\emptyset\}; D^o = \{\emptyset\}; D^t = \{v_3^1\};$				
v_3^1 потрібно $\uparrow \Rightarrow d_{13} - \downarrow$; Згодні? Так. $d_{13} := d_{13} - \Delta = 3$				

крок 6

1 рядок	$v_p = v_4^1 = 1.000$	$v_1^1 = 0.483$	$v_2^1 = 0.656$	$v_3^1 = 0.369$
2 рядок	$v_p = v_4^2 = 1.000$	$v_2^2 = 0.737$	$v_1^2 = 0.562$	$v_3^2 = 0.300$
3 рядок	$v_p = v_4^3 = 1.000$	$v_3^3 = 0.457$	$v_1^3 = 0.620$	$v_2^3 = 0.729$
4 рядок	$v_p = v_4^4 = 1.000$	$v_1^4 = 0.525$	$v_2^4 = 0.763$	$v_3^4 = 0.406$
$V_1 = \{$	$\boxed{0.5}$	0.6	0.6	$\} \quad S_1 = \{0; 0; 0; 0; 0; 2; 2; 0; 0; 0; 0\}$
$k(S_1) =$	0.790	$= T_u$		
$V_2 = \{$	0.7	$\boxed{0.7}$	0.7	0.8
$k(S_2) =$	0.835	$> T_u$		
$V_3 = \{$	0.4	0.3	$\boxed{0.5}$	0.4
$k(S_3) =$	0.729	$a_3 = 0.383$	$\delta_{max3} = a_3 - v_3^2 = 0.083$	
$V_4 = \{$	$\boxed{1.0}$	$\boxed{1.0}$	$\boxed{1.0}$	$\boxed{1.0}$
$k(S_4) =$	1.000	$> T_u$		
$D = \{v_3^2\};$				
$D^p = \{\emptyset\}; D^o = \{\emptyset\}; D^t = \{v_3^2\};$				
v_3^2 потрібно $\uparrow \Rightarrow d_{23} - \downarrow$; Згодні? Так. $d_{23} := d_{23} - \Delta = 5$				

Таблиця 1.3. продовження

крок 6						
1 рядок	$v_p = v_4^1 =$	1.000	$v_1^1 = 0.483$	$v_2^1 = 0.656$	$v_3^1 = 0.369$	
2 рядок	$v_p = v_4^2 =$	1.000	$v_2^2 = 0.737$	$v_1^2 = 0.562$	$v_3^2 = 0.387$	
3 рядок	$v_p = v_4^3 =$	1.000	$v_3^3 = 0.457$	$v_1^3 = 0.620$	$v_2^3 = 0.729$	
4 рядок	$v_p = v_4^4 =$	1.000	$v_1^4 = 0.525$	$v_2^4 = 0.763$	$v_3^4 = 0.406$	
$V_1 = \{$	$\boxed{0.5}$	0.6	0.6	0.5 } $S_1 = \{0; 0; 0; 0; 0; 2; 2; 0; 0; 0\}$		
$k(S_1) =$	0.790	$= T_u$				
$V_2 = \{$	0.7	$\boxed{0.7}$	0.7	0.8 }	$S_2 = \{0; 0; 0; 0; 0; 0; 3; 1; 0; 0\}$	
$k(S_2) =$	0.835	$> T_u$				
$V_3 = \{$	0.4	0.4	$\boxed{0.5}$	0.4 }	$S_3 = \{0; 0; 0; 0; 3; 1; 0; 0; 0; 0\}$	
$k(S_3) =$	0.835	$> T_u$				
$V_4 = \{$	$\boxed{1.0}$	$\boxed{1.0}$	$\boxed{1.0}$	$\boxed{1.0}$ }	$S_4 = \{0; 0; 0; 0; 0; 0; 0; 0; 0; 4\}$	
$k(S_4) =$	1.000	$> T_u$				
$D = \{\emptyset\};$						
$a_1 = 0.548$	$a_2 = 0.721$	$a_3 = 0.405$	$a_4 = 1.000$			
$w_1 = 0.205$	$w_2 = 0.270$	$w_3 = 0.151$	$w_4 = 0.374$			

тньо узгоджених спектрів будемо формувати множину D , до якої на даний момент і включається v_1^4 .

Аналогічним чином будуються спектри S_2, S_3, S_4 та обчислюються значення a_2 , δ_{max2} , a_3 і δ_{max3} , а значення v_2^3 та v_3^3 включаються до множини D . Зазначимо, що оскільки узгодженість спектра S_4 є достатньою ($k(S_4) > T_u$), то обчислювати a_4 і δ_{max4} немає потреби, і при цьому жодний додатковий елемент не включається до множини D .

Після цього множину D розділяємо на три непересічні підмножини: D^p – включає ваги, що відповідають домінуючим альтернативам, D^o – ваги, яким відповідають власні значення і D^t – транзитивні. На кроці алгоритму, що розглядається, маємо: $D^p = \emptyset$, $D^o = \{v_3^3\}$ і $D^t = \{v_1^4, v_2^3\}$.

Далі аналізуємо значення ваг, що входять до перелічених підмножин на предмет зміни цих ваг з метою покращення узгодженості. Підмножини розглядаються у за-значеній вище послідовності, тобто поки існують нерозглянуті елементи підмножини D^p , ми не переходимо до аналізу елементів підмножини D^o , і поки існують нерозглянуті елементи D^o – не переходимо до D^t . Отже, оскільки $D^p = \emptyset$, то розглядаємо елементи з підмножини D^o (це вага v_3^3).

Для того, щоб підвищити рівень узгодженості спектра S_3 до якого входить складова v_3^3 , округлена до найближчої поділки шкали, потрібно перемістити цю складову спектра у напрямку до середнього значення, тобто зменшити значення v_3^3 . З (1.12) випливає, що для зменшення v_3^3 достатньо зменшити елемент МПП d_{34} , що і пропонується експертovі, поставивши йому запитання типу: "Чи не погодились би Ви зменшити ступінь переваги 3-ї альтернативи над 4-ю ?". Припустимо, що отримана від експерта відповідь: "Так", тоді значення елемента МПП d_{34} зменшуємо на величину Δ . Значення Δ обираємо в даному випадку виходячи з (1.12) та виду застосованої для парних порівнянь альтернатив шкали таким чином, щоб наслідком зміни d_{34} була б зміна відповідної складової спектру S_3 хоча б на одну поділку. У даному разі $\Delta = 3$ і d_{34} стає рівним -9 .

Крім того зазначимо, що у випадку незгоди експерта змінити ступінь переваги деякої альтернативи над іншою в якомусь конкретному напрямку (збільшення або зменшення) – відповідна складова виключається з множини D на всіх подальших кроках алгоритму. Це робиться для виключення повторних звернень до експерта з тим самим запитанням, на яке вже отримано негативну відповідь. Переходимо до наступного кроку.

На кроці 2 процедура розрахунків ваг є аналогічною описаній для кроку 1. Використовуючи округлені значення ваг будуємо спектри і розраховуємо коефіцієнти узгодженості кожного з них. Відмітимо, що узгодженість спектрів S_2 та S_4 на другому кроці є достатньою ($k(S_2) > T_u$, $k(S_4) > T_u$), тому середні арифметичні a_2 і a_4 та відхилення δ_{max2} і δ_{max4} не обчислюються.

Одна особливість алгоритму проявляється на цьому кроці, а саме: перед тим як змінити той чи інший елемент МПП, проводиться перевірка на відповідність нового значення шкалі переваг, що використовувалась при формуванні цієї МПП. Так на даному кроці для покращення узгодженості потрібно зменшити елемент d_3^4 , що неможливо, бо він і так має мінімальне для выбраної шкали переваг значення (-9) . У такому випадку пов'язана з d_3^4 складова спектра S_3 – v_3^3 в подальшому не розглядається як найбільш віддалена від середнього значення і на подальших кроках

виключається з множини D . Найбільш віддаленою в подальшому будемо вважати наступну по віддаленості від середнього значення складову спектра. В такому випадку з питаннями до експерта не звертаємося і переходимо до наступного кроку.

Кроки 3-7 виконуємо по аналогії з попередніми. Процес покращення узгодженості припиняється, якщо в множині D не виявиться елементів. Це може статися в двох випадках: перший – це коли коефіцієнти узгодженості всіх без виключення спектрів більші або рівні порогу застосування (випадок як у прикладі, що розглядається – крок 7); другий – може статися, коли експерт відмовився переглядати деякі ступені переваги одних альтернатив над іншими, і в результаті чого можливий висновок про неуспішність процесу покращення узгодженості до достатнього рівня і надання рекомендацій провести експертне оцінювання з початку.

У прикладі, що розглядається, процес покращення узгодженості пройшов успішно, і в цьому випадку середні значення ваг компонент спектрів (a_1, \dots, a_4) нормуються згідно з формулою:

$$w_i = a_i / \sum_j a_j.$$

Отримані значення w_i приймаються як відносні значення ваг альтернатив.

1.2.4. Приклад порівняння способом “на скільки”

У цьому випадку функції (1.1),(1.2),(1.3) мають вигляд:

$$v_j = v_i + d_{ij}; \quad (1.15)$$

$$v_i^i = v_{pj}^i + d_{ij}; \quad (1.16)$$

$$v_j^i = v_i^i - d_{ij}. \quad (1.17)$$

Використовуючи (1.15), неважко переконатися в тому, що й у цьому випадку виконується умова (1.14), тобто ваги альтернатив, обчислені відповідно до (1.15), представлені в шкалі інтервалів, для якої операція знаходження середнього арифметичного є припустимою [21]. Результати виконання кроків алгоритму для випадку, коли МПП задана табл.1.4, приведені в табл.1.5.

Таблиця 1.4.

	A_1	A_2	A_3	A_4
A_1	0	0.2	-0.2	0.4
A_2	0.1	0	0.3	-0.3
A_3	-0.3	-0.3	0	-0.6
A_4	0.4	0.2	0.5	0

Представлена в стислому вигляді робота алгоритму (табл. 1.5) абсолютно аналогічна докладно описаній для попереднього прикладу (пункт 1.2.3). Відмінність полягає в тому, що замість виразів (1.12) та (1.13) відповідно використовуються вирази (1.16) та (1.17).

Крім того, в даному випадку значення домінуючим альтернативам не присвоюються рівним 1.0, а пропонується експертові оцінити ваги відповідних альтернатив методом безпосередньої оцінки. Як видно з табл. 1.5 на кроці 1 експерт оцінив відповідні альтернативи, і таким чином $v_4^1 = 0.8$, $v_4^2 = 0.85$, $v_4^3 = 0.85$, $v_4^4 = 0.8$. Також слід зазначити, що на подальших кроках алгоритму оцінка ваг домінуючих альтернатив не повторюється знову, а ці значення залишаються тими ж самими, що на кроці 1.

1.3. Способи введення інформації для отримання кардинальних експертних оцінок

На теперішній час на практиці застосовуються декілька способів введення інформації, одержаної від експерта, яка в подальшому використовується для отримання кардинальних експертних оцінок. Серед цих способів можна виділити найбільш поширені, це: числовий, словесний та графічний.

При числовому способі введення інформації експертові пропонується ввести конкретне значення оцінки об'єкта, використовуючи цифрові клавіші комп'ютера і, можливо, клавіші розділових знаків. Зазвичай діапазон можливих для введення експертом значень обмежується для зменшення кількості помилок при введенні. На рис. 1.2 наведено приклад однієї із форм для введення інформації цим способом, що застосовується у системі "Солон-2". Цей спосіб введення інформації не потребує

Таблиця 1.5.

	$\varepsilon = 0.05$	$T_o = 0.398$	$T_u = 0.790$	при $n = 11$
крок 1				
1 рядок	$v_p = v_4^1 \xrightarrow{\text{експерт}} 0.8$	$v_1^1 = 0.400$	$v_2^1 = 0.600$	$v_3^1 = 0.200$
2 рядок	$v_p = v_4^2 \xrightarrow{\text{експерт}} 0.85$	$v_2^2 = 0.550$	$v_1^2 = 0.450$	$v_3^2 = 0.250$
3 рядок	$v_p = v_4^3 \xrightarrow{\text{експерт}} 0.85$	$v_3^3 = 0.250$	$v_1^3 = 0.550$	$v_2^3 = 0.550$
4 рядок	$v_p = v_4^4 \xrightarrow{\text{експерт}} 0.8$	$v_1^4 = 0.400$	$v_2^4 = 0.600$	$v_3^4 = 0.300$
$V_1 = \{$	$\boxed{0.4} \quad 0.5 \quad 0.6 \quad 0.4 \quad \}$		$S_1 = \{0; 0; 0; 0; 2; 1; 1; 0; 0; 0; 0\}$	
$k(S_1) =$	$0.685 \quad a_1 = 0.450$	$\delta_{max1} =$	$ a_1 - v_1^3 = 0.100$	
$V_2 = \{$	$0.6 \quad \boxed{0.6} \quad 0.6 \quad 0.6 \quad \}$		$S_2 = \{0; 0; 0; 0; 0; 0; 4; 0; 0; 0; 0\}$	
$k(S_2) =$	$1.000 > T_u \Rightarrow a \text{ и } \delta_{max}$	не	обчислються	
$V_3 = \{$	$0.2 \quad 0.3 \quad \boxed{0.3} \quad 0.3 \quad \}$		$S_3 = \{0; 0; 1; 3; 0; 0; 0; 0; 0; 0; 0\}$	
$k(S_3) =$	$0.835 > T_u$			
$V_4 = \{$	$\boxed{0.8} \quad \boxed{0.9} \quad \boxed{0.9} \quad \boxed{0.8} \quad \}$		$S_4 = \{0; 0; 0; 0; 0; 0; 0; 0; 2; 2; 0\}$	
$k(S_4) =$	$0.790 = T_u$			
$D = \{v_1^3\};$				
$D^p = \{\emptyset\}; D^o = \{\emptyset\}; D^t = \{v_1^3\};$				
v_1^3 потрібно $\downarrow \Rightarrow d_{31} - \uparrow$; Згодні? Так. $d_{31} := d_{31} + \Delta = -0.2$				
крок 2				
1 рядок	$v_p = v_4^1 = 0.8 \quad v_1^1 = 0.400 \quad v_2^1 = 0.600 \quad v_3^1 = 0.200$			
2 рядок	$v_p = v_4^2 = 0.85 \quad v_2^2 = 0.550 \quad v_1^2 = 0.450 \quad v_3^2 = 0.250$			
3 рядок	$v_p = v_4^3 = 0.85 \quad v_3^3 = 0.250 \quad v_1^3 = 0.450 \quad v_2^3 = 0.550$			
4 рядок	$v_p = v_4^4 = 0.8 \quad v_1^4 = 0.400 \quad v_2^4 = 0.600 \quad v_3^4 = 0.300$			
$V_1 = \{$	$\boxed{0.4} \quad 0.5 \quad 0.5 \quad 0.4 \quad \}$	$S_1 = \{0; 0; 0; 0; 2; 2; 0; 0; 0; 0\}$		
$k(S_1) =$	$0.790 = T_u$			
$V_2 = \{$	$0.6 \quad \boxed{0.6} \quad 0.6 \quad 0.6 \quad \}$	$S_2 = \{0; 0; 0; 0; 0; 0; 4; 0; 0; 0\}$		
$k(S_2) =$	$1.000 > T_u$			
$V_3 = \{$	$0.2 \quad 0.3 \quad \boxed{0.3} \quad 0.3 \quad \}$	$S_3 = \{0; 0; 1; 3; 0; 0; 0; 0; 0; 0\}$		
$k(S_3) =$	$0.835 > T_u$			
$V_4 = \{$	$\boxed{0.8} \quad \boxed{0.9} \quad \boxed{0.9} \quad \boxed{0.8} \quad \}$	$S_4 = \{0; 0; 0; 0; 0; 0; 0; 0; 2; 2; 0\}$		
$k(S_4) =$	$0.790 = T_u$			
$D = \{\emptyset\};$				
<hr/>				
$a_1 = 0.453 \quad a_2 = 0.595 \quad a_3 = 0.316 \quad a_4 = 0.825$				
$w_1 = 0.207 \quad w_2 = 0.272 \quad w_3 = 0.144 \quad w_4 = 0.377$				
<hr/>				
$\boxed{\square}$ – ваги домінуючих альтернатив; \square – власні ваги альтернатив;				

Безпосередня оцінка впливу підцілей

В лінійній шкалі від 0 до **100** оцініть вплив кожної підцілі на ціль:

Зменшити рівень шумів

Назва підцілі	Оцінка впливу
Збільшити площі зелених насаджень	40
Зменшити відсоток трамвайних перевезень	30
Видалити транзитний транспорт з міста	40
Виховання населення	40
Інші фактори	30
Заборонити проліт літаків над містом	20

Скасувати **Підтвердити**

Рис. 1.2. Числовий спосіб введення інформації

додаткової програмної обробки введених даних, але є найменш зручним для середньостатистичного експерта і потребує більше часу на одне введення в порівнянні з іншими способами введення інформації.

Словесний (або його ще називають вербалний) спосіб введення інформації найбільш доцільно застосовувати при оцінюванні об'єктів, яким важко поставити у відповідність якийсь числовий еквівалент. У таких випадках експертам пропонується вибрати з декількох словесних характеристик об'єкта або із словесних характеристик його переваги над іншим об'єктом ту, що, на думку експерта, найбільш точно відповідає реальній ситуації. На рис. 1.3 наведено приклад однієї із форм для введення інформації цим способом. При словесному способі введення інформації кожному словесному варіантові вибраної характеристики числовий варіант ставиться у відповідність програмно. Знаходження цих співвідношень вирішується в рамках психологічних експериментів. При словесному введенні даних, яке в деяких випадках є найбільш зручним для експерта, інколи втрачається можливість більш точно задати значення характеристики через занадто обмежену кількість варіантів словесного опису оцінок.

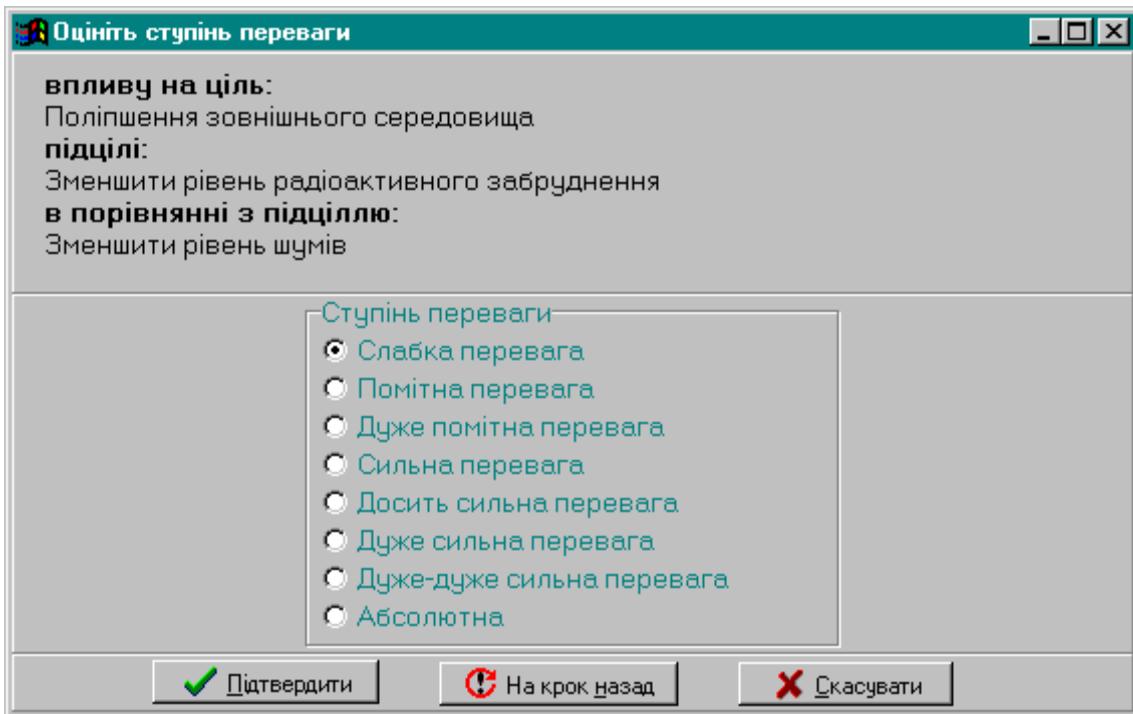


Рис. 1.3. Словесний спосіб введення інформації

Графічний спосіб введення інформації є найбільш наочним і зручним для експерта. Інформація при такому способі введення може бути представлена у вигляді різного плану зображень діаграм, лінійок, гістограм і т.ін. Приклад однієї із форм для введення інформації, що застосовується у системі ”Солон-3”, наведено на рис. 1.4.

Хоча цей спосіб і потребує більш значних затрат на розробку інтерфейсу з користувачем та на обробку введених даних, але він має незаперечні переваги в порівняльних можливостях при оцінюванні об'єктів та в можливостях більш точно задати значення характеристик.

1.4. Висновки

Можливості методу парних порівнянь при його використанні для обчислення ваг альтернатив можуть бути розширені за рахунок надання експерту можливості вибору способу представлення ступеня переваги альтернатив, найбільш зручного для вираження його думки. Наступним напрямком удосконалування методу парних порівнянь є організація зворотного зв'язку з експертом. Вона дозволяє повідомляти екс-

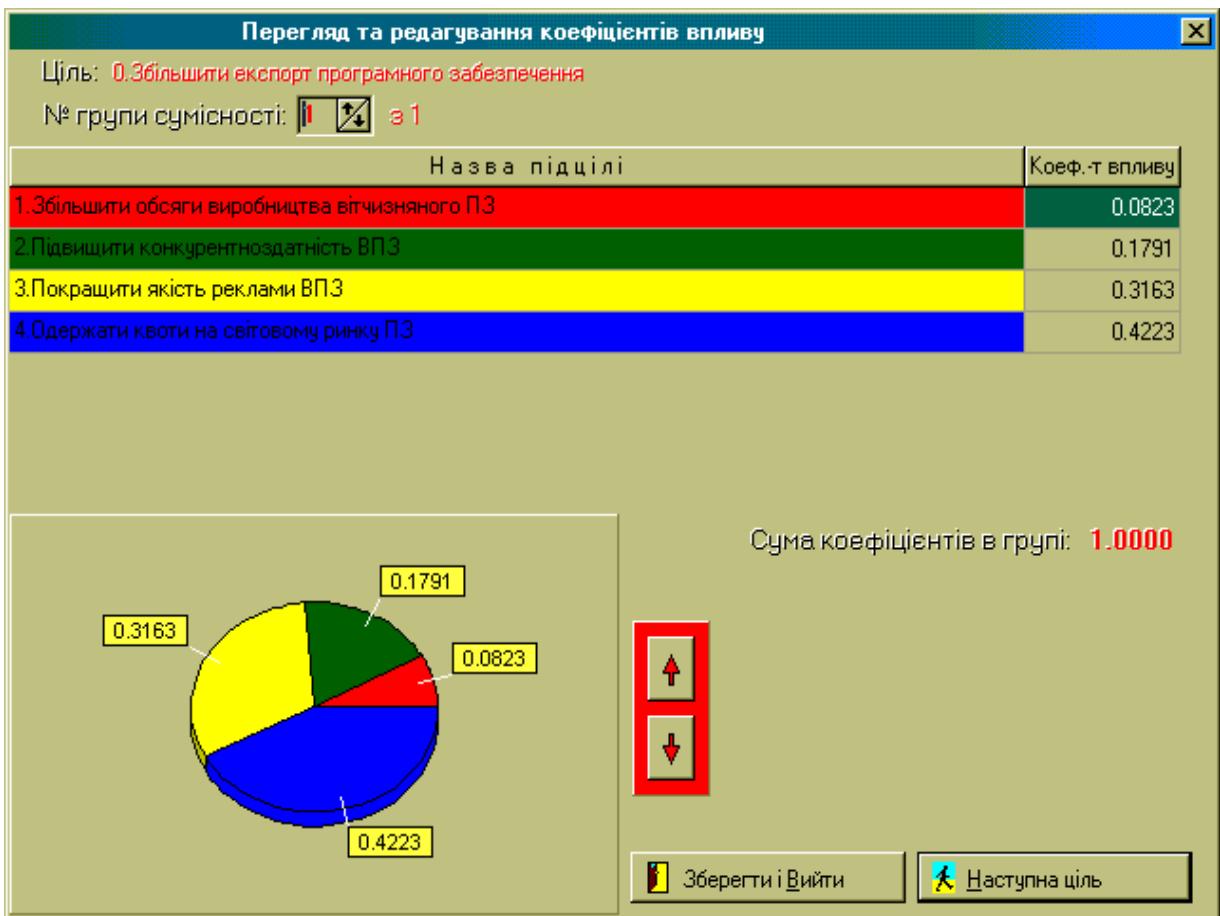


Рис. 1.4. Графічний спосіб введення інформації

перту про джерела внутрішніх протиріч його оцінок і шляхах, що рекомендуються, для їх усуненъ. Це забезпечується використанням запропонованого підходу до обчислення ступеня внутрішньої узгодженості результатів парних порівнянь. Визначено умови достатності рівня внутрішньої узгодженості і спосіб формування рекомендацій експерту про напрямки зменшення внутрішніх протиріч.

Основні наукові результати розділу опубліковані в працях [28, 83].

Розділ 2.

Розробка індивідуальних комбінаторних методів парних порівнянь зі зворотним зв'язком

2.1. Комбінаторний метод

В процесі аналізу та визначення кількісних і якісних показників методів експертної оцінки інформації, зокрема методів зі зворотним зв'язком з експертом, були відзначенні деякі недоліки цих методів. Одним із основних недоліків є значна кількість ітерацій алгоритмів і, як наслідок, – велика кількість запитань експертові у випадках, коли узгодженість вхідної МПП невисока, або/та у випадках, коли у вхідній матриці зустрічаються протиріччя в ранжуванні деяких альтернатив.

Цей недолік методів спонукав замислитися над створенням нового методу, який би прагнув звести кількість запитань експертові до мінімуму.

Перед тим як розпочати опис запропонованого методу, звернемо увагу на поняття *ідеально узгодженої матриці парних порівнянь* (ІУМПП), яке введено в [44]. Скористаємося цим означенням і сформулюємо деякі властивості ІУМПП.

Перед тим як сформулювати першу властивість, будемо розмірковувати наступним чином: припустимо, що значення ненормованих ваг кожної альтернативи $A_i, i \in$

$\{1, 2, \dots, n\}$ із множини альтернатив A апріорі відомі і відповідно рівні $\langle v_1; v_2; \dots; v_n \rangle$. Тоді для того, щоб МПП була ідеально узгодженою, потрібно, щоб кожний елемент цієї матриці був однозначно визначенім через значення цих ваг альтернатив.

Властивість 2.1.1. *Матриця парних порівнянь D розмірністю $(n \times n)$ є ідеально узгодженою тоді і тільки тоді, якщо знаходиться такий кортеж $\langle v_1; v_2; \dots; v_n \rangle$, що для $\forall i, j : i, j \in \{1, 2, \dots, n\}$ виконується рівність*

$$d_{ij} = f(v_i; v_j), \quad (2.1)$$

де d – елементи матриці D , f – функція, вид якої залежить від вибраних системи та шкали оцінки альтернатив.

Наприклад, коли при формуванні МПП експертові задається питання ”Як на вашу думку, *у скільки разів* значення A_i перевищує значення A_j ?” (мультиплікативні порівняння [44]), то рівність (2.1) має вигляд:

$$d_{ij} = v_i/v_j, \quad (2.2)$$

а в випадку запитання ”на скільки значення A_i перевищує значення A_j ?” (адитивні порівняння [44]) - (2.1) приймає вигляд:

$$d_{ij} = v_i - v_j. \quad (2.3)$$

Інколи застосовуються й інші види функції f [45].

Із (2.1), шляхом простих перетворень, можна отримати залежність між елементами ГУМПП, яку відобразимо у властивості 2.1.2.

Властивість 2.1.2. *Якщо матриця парних порівнянь D – ідеально узгоджена, то для $\forall i, j, k \in \{1, 2, \dots, n\}$ виконується рівність*

$$d_{ij} = f(d_{ik}; d_{jk}). \quad (2.4)$$

Для мультиплікативних порівнянь ця рівність має вигляд:

$$d_{ij} = d_{ik}/d_{jk}, \quad (2.5)$$

а для адитивних:

$$d_{ij} = d_{ik} - d_{jk}. \quad (2.6)$$

Для прикладу покажемо перетворення, які приводять до рівності (2.5). Із властивості 2.1.1 та виразу (2.2) випливає, що для $\forall k \in \{1, 2, \dots, n\} : [v_i = d_{ik} \cdot v_k \wedge v_j = d_{jk} \cdot v_k]$. Шляхом підстановки v_i та v_j в (2.2) отримаємо, що для $\forall k \in \{1, 2, \dots, n\}$ в ІУМПП виконується рівність $d_{ij} = d_{ik}/d_{jk}$. Аналогічними перетвореннями можливо також вивести і вираз (2.6).

2.1.1. Суть методу

Оскільки в ІУМПП частина її елементів пов'язані між собою у відповідності до (2.4), то існує можливість знайти таку множину елементів ІУМПП мінімальної потужності, використовуючи яку можна визначити решту елементів ІУМПП. Будемо далі називати підмножину елементів матриці, по якій однозначно будеться повна МПП, підмножиною інформаційно-вагомих елементів (ІВЕ).

Зазначимо, що в будь-якій МПП елементи d_{ii} визначаються однозначно, виходячи з (2.5) $d_{ii} = 1$, або з (2.6) $d_{ii} = 0$. Фактично, вони не несуть інформації, тому що порівнювати альтернативу A_i саму з собою немає сенсу.

Крім того, у значній кількості випадків, якщо є інформація про порівняння деякої альтернативи A_i з деякою альтернативою A_j в будь-якій МПП, то немає рації повторно порівнювати A_j з A_i , а припустити, що ці порівняння взаємно узгоджені. Тоді для визначення симетричного відносно головної діагоналі елемента МПП можна скористатись співвідношенням для ІУМПП. Тобто, якщо елемент d_{ij} – відомий, то d_{ji} можна визначити з (2.5): $d_{ji} = 1/d_{ij}$; або з (2.6): $d_{ji} = -d_{ij}$. Ці співвідношення співпадають зі співвідношеннями, отриманими в [45].

Із зазначеного вище випливає, що в будь-якій МПП розміром $(n \times n)$ кількість ІВЕ є $(n^2 - n)/2$. Далі приділимо увагу знаходженню кількості ІВЕ в ІУМПП.

Скористаємося графовою інтерпретацією МПП. Для цього поставимо у відповідність ІУМПП граф Γ . Із попередніх міркувань випливає, що це неоріентований граф і не має петель. Вершини графа Γ позначимо номерами альтернатив, а ребра мають

ваги, що відповідають IBE в ГУМПП.

Отже, для знаходження кількості IBE в ГУМПП потрібно визначити:

- 1) Яким властивостям повинен відповідати граф Γ , щоб була можливість по вагах ребер, що наявні в графі, знайти решту ваг ребер, яких немає в наявності в цьому графі.
- 2) Яка мінімальна кількість ребер (з їхніми вагами) є достатньою для знаходження ваг решти ребер, яких немає в наявності в Γ .

На рис. 2.1 зображено графічне представлення виразу (2.4).

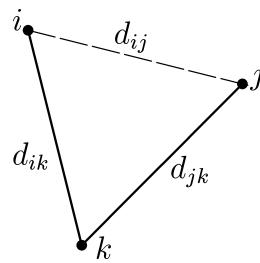


Рис. 2.1.

З цього зображення видно, що для визначення невідомої ваги d_{ij} необхідна наявність в графі двох дуг з їхніми вагами, які з'єднують вершини i та j з деякою вершиною k . Іншими словами, між двома вершинами i та j необхідно мати шлях довжиною 2.

Тепер узагальнимо це твердження на граф Γ і сформулюємо властивість графа, яка використовується в алгоритмі.

Для знаходження ваги відсутнього ребра в графі між деякими вершинами i та j необхідна наявність шляху будь-якої довжини між цими вершинами (якщо довжина шляху більше ніж 2, то шукана вага знаходиться за декілька кроків). Наприклад, на рис. 2.2 зображено частину графа Γ , де між вершинами i та j є шлях деякої довжини, то для знаходження d_{ij} можна спочатку знайти $d_{il} = f(d_{ik}; d_{lk})$, потім знаходимо $d_{ir} = f(d_{il}; d_{rl}), \dots, d_{ip}$ і, нарешті, $d_{ij} = f(d_{ip}; d_{jp})$.

Отже, можна зробити висновок, що умова знаходження усіх відсутніх в графі Γ ваг ребер – це наявність шляху між будь-якими двома вершинами графа (іншими

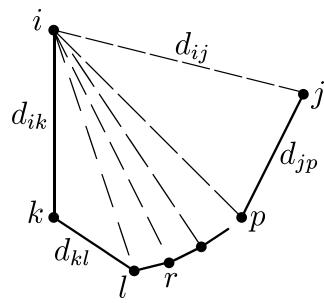


Рис. 2.2.

словами, це сильна зв'язність графа Γ).

У монографії Литвака [19] показано, що саме сильна зв'язність графа є необхідною і достатньою умовою, при якій підмножина ваг ребер є підмножиною IBE.

Оскільки Γ має n вершин (по кількості альтернатив, що оцінюються), то мінімальна кількість ребер, яка може з'єднати n вершин, є $(n - 1)$. Отже, відповідь на друге питання: кількість ребер в Γ рівна $(n - 1)$.

Таким чином, будуючи графи, які мають вищезгадані властивості, ми можемо сформувати множини IBE ГУМПП мінімальної потужності, тобто використовуючи елементи, що входять в такі множини, можливо однозначно обчислити решту елементів ГУМПП.

Тепер, коли маємо множини IBE, будемо по черзі для кожної такої множини ставити у відповідність ГУМПП, в якій значення IBE запозичені з вхідної (реальної) МПП, а решта елементів обчислені. Всі отримані таким чином ГУМПП зберігаємо в наборі даних для подальшого використання.

Слід зазначити, що цей набір даних формуємо виключаючи повтори ГУМПП і упорядковуючи по кількості елементів в ГУМПП, що не співпадають з відповідними елементами реальної МПП.

Наступний етап пов'язаний з аналізом сформованого набору ГУМПП і з пошуком в ньому такої ГУМПП, в яку б перетворювалась реальна МПП шляхом зміни (збільшення або зменшення) частини її елементів, погоджуючи ці зміни з думкою експерта.

У випадку, коли при перегляді набору буде знайдено ГУМПП, яка відповідає по-

ставленим вимогам, то ця матриця і стане результатом роботи алгоритму (по цій ГУМПП будуть обчислені ваги альтернатив).

Тепер звернемо увагу на те, яким чином запропонований метод прагне мінімізувати кількість питань до експерта. Це відбувається за рахунок того, що в першу чергу в наборі даних розглядаються ГУМПП, в котрих кількість елементів, якими вони відрізняються від реальної МПП – мінімальна. Зазначимо, що кількість питань експертів була б мінімальною тільки у випадку, якщо б експерт завжди погоджувався на пропозиції про зміну деяких переваг значень альтернатив над іншими. Але, оскільки експерт на запит про зміну переваги може давати також і негативні відповіді, а передбачити, в яких випадках це станеться, та мінімізувати кількість цих випадків практично неможливо, то спроба досягти абсолютної мінімізації кількості питань до експерта не завжди буде вдалою.

Тепер, для більш докладного опису, покажемо роботу алгоритму на конкретному прикладі.

2.1.2. Опис алгоритму

Отже, нехай маємо реальну матрицю D_0 (табл.2.1), отриману при парному порів-

	A_1	A_2	A_3	A_4
A_1	0	-5	-2	-2
A_2	5	0	3	4
A_3	2	-3	0	0
A_4	2	-4	0	0

нянні деяких чотирьох альтернатив ($n = 4$). Причому, при формуванні D_0 експертові задавались питання типу: "На скільки деяких умовних одиниць одна альтернатива переважає іншу?". Будемо брати до уваги, наприклад, тільки елементи матриці, розташовані праворуч від головної діагоналі, а решта елементів D_0 не несуть інформації і можуть бути обчислені на основі тих, що розглядаються.

Таким чином, на підготовчому етапі алгоритму розглядаємо множину елементів D_0 , що знаходяться справа від головної діагоналі: $\{d_{ij} | i < j\}$. В нашому прикладі – це

множина елементів $\{d_{12}; d_{13}; d_{14}; d_{23}; d_{24}; d_{34}\}$, з якої, перебираючи всі можливі варіанти, будемо формувати підмножини ІВЕ, що ляжуть в основу формування ГУМПП. Кількість елементів в цих підмножинах рівна $(n - 1)$, тобто 3.

Зупинимося більш докладно на алгоритмі перевірки підмножини на предмет її інформаційної вагомості, тобто на алгоритмі перевірки графа, що відповідає цій підмножині, на предмет наявності шляху між будь-якими двома його вершинами.

Позначимо через x_{ab} – індекс елементу d з вищезгаданої підмножини, де $a \in \{1; 2; \dots; (n - 1)\}$ – порядковий номер елементу у цій підмножині, $b \in \{1; 2\}$ – номер індексу в парі. Нехай S – множина індексів x ; i, j – лічильники в циклах, тоді алгоритм матиме наступний вигляд:

- 1) $S = \{x_{11}; x_{12}\}, i = 2;$
- 2) $j = 2;$
- 3) якщо $x_{j1} \in S$, то додаємо елемент x_{j2} до множини S , і переходимо до п.5;
- 4) якщо $x_{j2} \in S$, то додаємо елемент x_{j1} до множини S ;
- 5) $j = j + 1;$
- 6) якщо $j < n$, то до п.3;
- 7) $i = i + 1;$
- 8) якщо $i < n$, то до п.2;
- 9) якщо $S = \{1, 2, \dots, n\}$, то підмножина є інформаційно-вагомою, інакше робимо висновок, що підмножина не є інформаційно-вагомою;
- 10) кінець.

Проілюструємо роботу цього алгоритму на прикладі перевірки підмножини $\tilde{D} = \{d_{14}; d_{23}; d_{34}\}$ на предмет її інформаційної вагомості. Для вибраного прикладу $x_{11} = 1$, $x_{12} = 4$, $x_{21} = 2$, $x_{22} = 3$, $x_{31} = 3$ і $x_{32} = 4$.

Отже після виконання п.1 та п.2 маємо $S = \{1; 4\}, i = 2, j = 2$.

Далі в п.3 робимо перевірку приналежності $x_{21} = 2$ до множини S , і оскільки $x_{21} \notin S$, то переходимо до наступного пункту.

В п.4 робимо перевірку приналежності $x_{22} = 3$ до множини S , і оскільки $x_{22} \notin S$, то переходимо до наступного пункту.

Після п.5 $j = 3$.

В п.6 перевіряємо умову виходу з циклу $j < n$, і оскільки $3 < 4$, переходимо до п.3.

В п.3 робимо перевірку приналежності $x_{31} = 3$ до множини S , і оскільки $x_{31} \notin S$, то переходимо до наступного пункту.

В п.4 робимо перевірку приналежності $x_{32} = 4$ до множини S , і оскільки $x_{32} \in S$, то додаємо елемент $x_{31} = 3$ до S , тобто виконуємо об'єднання $S := S \cup x_{31}$. В даному прикладі після виконання п.3 маємо $S = \{1; 3; 4\}$ і переходимо до п.5.

Після п.5 $j = 4$.

В п.6 перевіряємо умову $j < n$, і оскільки вона не виконується, то виходимо з циклу по j і переходимо до п.7.

Після п.7 $i = 3$.

В п.8 перевіряємо умову виходу з циклу по i — $i < n$, і оскільки $3 < 4$, переходимо до п.2.

Після п.2 $j = 2$.

Далі в п.3 робимо перевірку приналежності $x_{21} = 2$ до множини S , і оскільки $x_{21} \notin S$, то переходимо до наступного пункту.

В п.4 робимо перевірку приналежності $x_{22} = 3$ до множини S , і оскільки $x_{22} \in S$, то додаємо елемент $x_{21} = 2$ до S , тобто виконуємо об'єднання $S := S \cup x_{21}$. В даному прикладі після виконання п.4 маємо $S = \{1; 2; 3; 4\}$ і переходимо до п.5.

Після п.5 $j = 3$.

В п.6 перевіряємо умову виходу з циклу $j < n$, і оскільки $3 < 4$, переходимо до п.3.

В п.3 робимо перевірку приналежності $x_{31} = 3$ до множини S , і оскільки $x_{31} \in S$, додаємо елемент $x_{32} = 4$ до S , тобто виконуємо об'єднання $S := S \cup x_{32}$. В даному

випадку S не змінилась, і після виконання п.3 маємо $S = \{1; 2; 3; 4\}$ і переходимо до п.5.

Після п.5 $j = 4$.

В п.6 перевіряємо умову $j < n$, і оскільки вона не виконується, то виходимо з циклу по j і переходимо до п.7.

Після п.7 $i = 4$.

В п.8 перевіряємо умову виходу з циклу по i — $i < n$, і оскільки вона не виконується, то виходимо з циклу по i і переходимо до наступного пункту.

В п.9 перевіряємо виконання рівності $S = \{1; 2; 3; 4\}$, і оскільки вона виконується, то робимо висновок про інформаційну вагомість підмножини $\tilde{D} = \{d_{14}; d_{23}; d_{34}\}$.

П.10 – кінець алгоритму.

Отже, із всіх можливих варіантів підмножин (до речі, їх кількість в загальному випадку дорівнює $C_{n(n-1)/2}^{(n-1)}$) за допомогою щойно показаного алгоритму відбираємо тільки інформаційно-вагомі підмножини. Далі, на основі цих підмножин, сформуємо ідеально узгоджені матриці. Їх формування полягає у запозиченні IBE з реальної матриці і у обчисленні решти (в прикладі, що розглядається – трьох) невідомих елементів матриці (беремо до уваги елементи, що розміщені праворуч від головної діагоналі). Ці невідомі елементи обчислюються, виходячи із співвідношення (2.4), в нашому конкретному прикладі – із співвідношення (2.6).

В результаті цих обчислень отримаємо 16 ГУМПП, які зображені в табл.2.2.

Тепер виключимо повтори серед ГУМПП і упорядкуємо решту цих матриць за кількістю елементів, що не співпадають з відповідними елементами реальної матриці. Отже, в нашому прикладі маємо наступні множини номерів матриць, що співпадають: $\{1; 3; 4; 5; 7; 9; 12; 15\}$, $\{2; 6; 11\}$, $\{8\}$, $\{10; 13; 16\}$, $\{14\}$. І в результаті проведених дій отримаємо упорядковану послідовність з п'яти ГУМПП, яку наведено в табл.2.3.

Зазначимо, щоб мати повну інформацію про кожну ГУМПП, достатньо зберігати тільки відмінності кожної ГУМПП від реальної МПП. Це може виглядати так, як показано в табл.2.4 (крок 1). Тут кожній ГУМПП ставиться у відповідність множина триелементних кортежів, кожен кортеж відповідає одному елементу матриці. Пер-

Таблиця 2.2.

Номер ІУМПП	Вигляд ІУМПП	Вигляд графу Γ	Номер ІУМПП	Вигляд ІУМПП	Вигляд графу Γ																																
1	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-5</td><td>-2</td><td>-2</td></tr> <tr><td>5</td><td>0</td><td>3</td><td>3</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> </table>	0	-5	-2	-2	5	0	3	3	2	-3	0	0	2	-3	0	0		9	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-5</td><td>-2</td><td>-2</td></tr> <tr><td>5</td><td>0</td><td>3</td><td>3</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> </table>	0	-5	-2	-2	5	0	3	3	2	-3	0	0	2	-3	0	0	
0	-5	-2	-2																																		
5	0	3	3																																		
2	-3	0	0																																		
2	-3	0	0																																		
0	-5	-2	-2																																		
5	0	3	3																																		
2	-3	0	0																																		
2	-3	0	0																																		
2	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-5</td><td>-2</td><td>-1</td></tr> <tr><td>5</td><td>0</td><td>3</td><td>4</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>-4</td><td>-1</td><td>0</td></tr> </table>	0	-5	-2	-1	5	0	3	4	2	-3	0	1	1	-4	-1	0		10	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-6</td><td>-2</td><td>-2</td></tr> <tr><td>6</td><td>0</td><td>4</td><td>4</td></tr> <tr><td>2</td><td>-4</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>-4</td><td>0</td><td>0</td></tr> </table>	0	-6	-2	-2	6	0	4	4	2	-4	0	0	2	-4	0	0	
0	-5	-2	-1																																		
5	0	3	4																																		
2	-3	0	1																																		
1	-4	-1	0																																		
0	-6	-2	-2																																		
6	0	4	4																																		
2	-4	0	0																																		
2	-4	0	0																																		
3	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-5</td><td>-2</td><td>-2</td></tr> <tr><td>5</td><td>0</td><td>3</td><td>3</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> </table>	0	-5	-2	-2	5	0	3	3	2	-3	0	0	2	-3	0	0		11	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-5</td><td>-2</td><td>-1</td></tr> <tr><td>5</td><td>0</td><td>3</td><td>4</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>-4</td><td>-1</td><td>0</td></tr> </table>	0	-5	-2	-1	5	0	3	4	2	-3	0	1	1	-4	-1	0	
0	-5	-2	-2																																		
5	0	3	3																																		
2	-3	0	0																																		
2	-3	0	0																																		
0	-5	-2	-1																																		
5	0	3	4																																		
2	-3	0	1																																		
1	-4	-1	0																																		
4	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-5</td><td>-2</td><td>-2</td></tr> <tr><td>5</td><td>0</td><td>3</td><td>3</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> </table>	0	-5	-2	-2	5	0	3	3	2	-3	0	0	2	-3	0	0		12	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-5</td><td>-2</td><td>-2</td></tr> <tr><td>5</td><td>0</td><td>3</td><td>3</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> </table>	0	-5	-2	-2	5	0	3	3	2	-3	0	0	2	-3	0	0	
0	-5	-2	-2																																		
5	0	3	3																																		
2	-3	0	0																																		
2	-3	0	0																																		
0	-5	-2	-2																																		
5	0	3	3																																		
2	-3	0	0																																		
2	-3	0	0																																		
5	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-5</td><td>-2</td><td>-2</td></tr> <tr><td>5</td><td>0</td><td>3</td><td>3</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> </table>	0	-5	-2	-2	5	0	3	3	2	-3	0	0	2	-3	0	0		13	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-6</td><td>-2</td><td>-2</td></tr> <tr><td>6</td><td>0</td><td>4</td><td>4</td></tr> <tr><td>2</td><td>-4</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>-4</td><td>0</td><td>0</td></tr> </table>	0	-6	-2	-2	6	0	4	4	2	-4	0	0	2	-4	0	0	
0	-5	-2	-2																																		
5	0	3	3																																		
2	-3	0	0																																		
2	-3	0	0																																		
0	-6	-2	-2																																		
6	0	4	4																																		
2	-4	0	0																																		
2	-4	0	0																																		
6	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-5</td><td>-2</td><td>-1</td></tr> <tr><td>5</td><td>0</td><td>3</td><td>4</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>-4</td><td>-1</td><td>0</td></tr> </table>	0	-5	-2	-1	5	0	3	4	2	-3	0	1	1	-4	-1	0		14	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-6</td><td>-3</td><td>-2</td></tr> <tr><td>6</td><td>0</td><td>3</td><td>4</td></tr> <tr><td>3</td><td>-3</td><td>0</td><td>1</td></tr> <tr><td>2</td><td>-4</td><td>-1</td><td>0</td></tr> </table>	0	-6	-3	-2	6	0	3	4	3	-3	0	1	2	-4	-1	0	
0	-5	-2	-1																																		
5	0	3	4																																		
2	-3	0	1																																		
1	-4	-1	0																																		
0	-6	-3	-2																																		
6	0	3	4																																		
3	-3	0	1																																		
2	-4	-1	0																																		
7	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-5</td><td>-2</td><td>-2</td></tr> <tr><td>5</td><td>0</td><td>3</td><td>3</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> </table>	0	-5	-2	-2	5	0	3	3	2	-3	0	0	2	-3	0	0		15	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-5</td><td>-2</td><td>-2</td></tr> <tr><td>5</td><td>0</td><td>3</td><td>3</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>-3</td><td>0</td><td>0</td></tr> </table>	0	-5	-2	-2	5	0	3	3	2	-3	0	0	2	-3	0	0	
0	-5	-2	-2																																		
5	0	3	3																																		
2	-3	0	0																																		
2	-3	0	0																																		
0	-5	-2	-2																																		
5	0	3	3																																		
2	-3	0	0																																		
2	-3	0	0																																		
8	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-5</td><td>-1</td><td>-1</td></tr> <tr><td>5</td><td>0</td><td>4</td><td>4</td></tr> <tr><td>1</td><td>-4</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>-4</td><td>0</td><td>0</td></tr> </table>	0	-5	-1	-1	5	0	4	4	1	-4	0	0	1	-4	0	0		16	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0</td><td>-6</td><td>-2</td><td>-2</td></tr> <tr><td>6</td><td>0</td><td>4</td><td>4</td></tr> <tr><td>2</td><td>-4</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>-4</td><td>0</td><td>0</td></tr> </table>	0	-6	-2	-2	6	0	4	4	2	-4	0	0	2	-4	0	0	
0	-5	-1	-1																																		
5	0	4	4																																		
1	-4	0	0																																		
1	-4	0	0																																		
0	-6	-2	-2																																		
6	0	4	4																																		
2	-4	0	0																																		
2	-4	0	0																																		

— елементи, що запозичено з реальної МПП

Таблиця 2.3.

Номер ГУМПП	Вигляд ГУМПП	Кількість відмінностей від реальної МПП
1	$\begin{matrix} 0 & -5 & -2 & -2 \\ 5 & 0 & 3 & \boxed{3} \\ 2 & -3 & 0 & 0 \\ 2 & -3 & 0 & 0 \end{matrix}$	1
2	$\begin{matrix} 0 & -5 & -2 & \boxed{-1} \\ 5 & 0 & 3 & 4 \\ 2 & -3 & 0 & \boxed{1} \\ 1 & -4 & -1 & 0 \end{matrix}$	2
3	$\begin{matrix} 0 & \boxed{-6} & -2 & -2 \\ 6 & 0 & \boxed{4} & 4 \\ 2 & -4 & 0 & 0 \\ 2 & -4 & 0 & 0 \end{matrix}$	2
4	$\begin{matrix} 0 & \boxed{-6} & \boxed{-3} & -2 \\ 6 & 0 & 3 & 4 \\ 2 & -3 & 0 & \boxed{1} \\ 2 & -4 & -1 & 0 \end{matrix}$	3
5	$\begin{matrix} 0 & -5 & \boxed{-1} & \boxed{-1} \\ 5 & 0 & \boxed{4} & 4 \\ 1 & -4 & 0 & 0 \\ 1 & -4 & 0 & 0 \end{matrix}$	3

$\boxed{}$ – елементи, відмінні від елементів реальної МПП

ший елемент кортежу – номер рядка, другий – номер стовпчика, а третій – значення елемента ГУМПП, що відрізняється від відповідного елемента реальної МПП.

Варто відмітити, що формування послідовності, її упорядкування відбуваються в процесі генерації матриць, тому всі 16 варіантів ГУМПП, що зображені в табл.2.2, ніколи одночасно не будуть присутні в пам'яті комп'ютера.

На цьому підготовчий етап алгоритму закінчується.

На наступному етапі реалізується зворотний зв'язок із експертом. Сформована на попередньому етапі послідовність ГУМПП аналізується, починаючи з першої матриці (у котрої найменше відмінностей від реальної матриці). У цьому випадку кожний

Таблиця 2.4.

Номер кроку	Номер п/п	Упорядкована послідовність множин кортежів
1	1	$\{\langle 2; 4; 3 \rangle\}$
	2	$\{\langle 1; 4; -1 \rangle, \langle 3; 4; 1 \rangle\}$
	3	$\{\langle 1; 2; -6 \rangle, \langle 2; 3; 4 \rangle\}$
	4	$\{\langle 1; 2; -6 \rangle, \langle 1; 3; -3 \rangle, \langle 3; 4; 1 \rangle\}$
	5	$\{\langle 1; 3; -1 \rangle, \langle 1; 4; -1 \rangle, \langle 2; 3; 4 \rangle\}$
2	1	$\{\langle 1; 4; -1 \rangle, \langle 3; 4; 1 \rangle\}$
	2	$\{\langle 1; 2; -6 \rangle, \langle 2; 3; 4 \rangle\}$
	3	$\{\langle 1; 2; -6 \rangle, \langle 1; 3; -3 \rangle, \langle 3; 4; 1 \rangle\}$
	4	$\{\langle 1; 3; -1 \rangle, \langle 1; 4; -1 \rangle, \langle 2; 3; 4 \rangle\}$
3	1	$\{\langle 1; 2; -6 \rangle, \langle 2; 3; 4 \rangle\}$
	2	$\{\langle 1; 3; -1 \rangle, \langle 1; 4; -1 \rangle, \langle 2; 3; 4 \rangle\}$
4	1	$\{\langle 1; 3; -1 \rangle, \langle 1; 4; -1 \rangle, \langle 2; 3; 4 \rangle\}$

кортеж виду $\langle \alpha; \beta; \gamma \rangle$ відповідає поставленому експертові запитанню типу: "Чи не погодились би Ви збільшити (зменшити) перевагу альтернативи α над альтернативою β ?". "Збільшити" чи "зменшити" залежить від γ та значення елемента $d_{\alpha\beta}$ реальної матриці. Таким чином, позитивна відповідь експерта на таке запитання дає право в подальшому змінити елемент $d_{\alpha\beta}$ в визначену сторону (збільшення або зменшення), тим самим ця ж відповідь забороняє зміну цього ж елемента в іншу сторону. Негативна ж відповідь забороняє зміну $d_{\alpha\beta}$ в визначену запитанням сторону. Тому післяожної відповіді експерта будемо проводити аналіз решти кортежів, в яких першим елементом є саме α і другим саме β , на рахунок того, чи вони не суперечать думці експерта. Далі ми виключаємо з розгляду всю множину кортежів, що відповідають одній і тій самій ГУМПП, якщо елемент γ кортежу, що аналізується, перебуває в "забороненому" діапазоні.

Очевидно, що порядок, за яким ставляться запитання експертові, має вплив на швидкість завершення алгоритму. Тому, можливо, потрібно ставити першими запитання, ймовірність позитивної відповіді на які – більша. Є припущення, що для більш швидкого завершення алгоритму в першу чергу потрібно ставити саме те запитання, де абсолютна різниця між γ і $d_{\alpha\beta}$ – мінімальна, тобто потрібно упорядкувати кортежі

$\langle \alpha; \beta; \gamma \rangle$ в кожній із множин по збільшенню значення $|\gamma - d_{\alpha\beta}|$. Це питання потребує подальшого комплексного вивчення.

Повернемося до конкретного прикладу. Розглянемо перший кортеж (табл.2.4 крок 1) і згідно з інформацією, яку він містить, сформулюємо запитання до експерта: "Чи не погодились би Ви зменшити перевагу 2-ї альтернативи над 4-ю?". При позитивній відповіді (експерт погоджується зменшити перевагу) змінюємо елемент d_{24} реальної матриці з 4 на 3. Таким чином, реальна матриця стає ідеально узгодженою, і алгоритм завершує роботу.

Для більш детального пояснення роботи алгоритму припустимо, що на попереднє запитання експерт дав негативну відповідь. У цьому випадку з упорядкованої послідовності (табл.2.4 крок 1) видаляємо всі множини, до яких належать кортежі $\langle 2; 4; \gamma \rangle$, де $\gamma < d_{24}$ ($\gamma < 4$). Такою множиною виявляється тільки перша множина, яка і видаляється. Послідовність приймає вигляд, як показано в табл.2.4 крок 2.

Формулюємо наступне запитання експертові: "Чи не погодились би Ви збільшити (з алгебраїчної точки зору) перевагу 1-ї альтернативи над 4-ю?" і отримуємо від експерта позитивну відповідь. Після цього видаляємо з послідовності множини (звичайно, якщо вони є), які містять в собі кортежі $\langle 1; 4; \gamma \rangle$, де $\gamma < d_{14}$ ($\gamma < -2$). Крім того – помічаємо всі кортежі $\langle 1; 4; \gamma \rangle$, де $\gamma > d_{14}$ ($\gamma > -2$), щоб в подальшому, якщо буде потрібно, вже без додаткової згоди експерта мати можливість збільшити перевагу 1-ї альтернативи над 4-ю (в таблиці ці помітки зроблено шляхом підкреслення). Формулюємо наступне запитання: "Чи не погодились би Ви збільшити перевагу 3-ї альтернативи над 4-ю?" і отримуємо, наприклад, негативну відповідь. Видаляємо всі множини, до яких належать кортежі $\langle 3; 4; \gamma \rangle$, де $\gamma > 0$. Послідовність приймає вигляд, як показано в табл.2.4 крок 3.

Припустимо, що на запитання "Чи не погодились би Ви зменшити перевагу 1-ї альтернативи над 2-ю?" отримуємо теж негативну відповідь. Тоді послідовність прийме вигляд, як в табл.2.4 крок 4.

Далі експертові ставиться запитання "Чи не погодились би Ви збільшити перевагу 1-ї альтернативи над 3-ю?". Наприклад, відповідь – "так". Проводиться перевірка

на видалення множин, що містять "недопустимі" кортежі, і пошук кортежів для підмітки, але таких не знаходимо. Запитання про збільшення переваги 1-ї альтернативи над 4-ю не задаємо експертові (відповідь відома – "так"), а формулюємо запитання: "Чи не погодились би Ви збільшити перевагу 2-ї альтернативи над 3-ю?". Якщо відповідь експерта – "ні", то в цьому випадку експертові видається повідомлення, що протиріччя, які мають місце в реальній МПП, погоджуючись з його думкою, виправити не вдалося. В іншому випадку, коли на останнє запитання отримуємо позитивну відповідь експерта, то рішення по узгодженню реальної МПП знайдено. Це рішення полягає в зміні трьох елементів реальної МПП, а сама МПП при цьому стає ідеально узгодженою і приймає вигляд матриці 5 з табл.2.3.

На цьому представлений алгоритм завершує роботу.

Таким чином, у ході виконання алгоритму експерт погоджується привести реальну МПП до ІУМПП шляхом зміни деякої множини елементів першої матриці. В іншому випадку може відбутися так, що на деякому кроці алгоритму в наборі не залишиться кортежів, тоді робиться висновок про те, що спроба експерта узгодити реальну матрицю (привести її до ІУМПП) є невдалою.

При успішному завершенні алгоритму ненормовані ваги об'єктів знаходяться по будь-якому i -ому рядку отриманої ІУМПП у такий спосіб. У i -ому рядку знаходимо мінімальний елемент d_{ij} . Оскільки d_{ij} – результат порівняння a_i і a_j , то при цьому a_j – домінуючий по i -ому рядку елемент.

Якщо експерт виконував адитивні порівняння, то йому пропонується методом безпосередньої оцінки визначити ненормоване абсолютне значення $v_{j\text{дом}}$ домінуючого в цьому рядку об'єкта a_j . Потім визначається власна вага v_i^i елемента a_i :

$$v_i^i = v_{j\text{дом}} + d_{ij}.$$

Після цього визначаються ваги інших об'єктів:

$$v_h^i = v_i^i - d_{ih}.$$

У випадку мультиплікативних порівнянь для визначення ваг об'єктів немає необхідності в безпосередній оцінці ненормованого абсолютноного значення домінуючого

по рядку об'єкта, а ненормовані значення ваг об'єктів знаходяться в такий спосіб:

$$v_{j\text{дом}} = 1; v_i^i = v_{j\text{дом}} \cdot d_{ij} = d_{ij}; v_h^i = v_i^i / d_{ih}.$$

Оскільки сформована в результаті виконання алгоритму матриця є ГУМПП, то ваги, визначені по будь-якому рядку, збігаються.

2.1.3. Деякі особливості реалізації

Попередньо зазначимо, що представлений алгоритм є алгоритмом комбінаторного типу. Він потребує значних обчислювальних ресурсів, як швидкодії процесора, так і об'єму оперативної пам'яті комп'ютера. У зв'язку з можливістю використання реалізації алгоритму на комп'ютерах, які, наприклад, не мають достатнього об'єму доступної оперативної пам'яті, в алгоритмі, перед тим, як зберегти чергову множину кортежів (яка відповідає деякій ГУМПП), передбачено перевірку на наявність необхідного об'єму оперативної пам'яті. У випадку її недостатності проводиться видалення множини кортежів, яка на той момент має максимальну потужність (знаходитьться останньою в упорядкованій послідовності), і вже після цієї дії чергова множина кортежів зберігається у пам'яті.

До особливостей реалізації алгоритму слід віднести можливість брати до розгляду не всі ГУМПП, а тільки ті, що відрізняються від реальної МПП попередньо визначеною кількістю елементів, наприклад, ГУМПП, що відрізняються, не більше ніж половиною загальної кількості інформаційно-вагомих елементів в матриці. Іншими словами, зберігати і розглядати тільки ті множини, потужність яких не перевищує деякого значення. Ця особливість дає змогу значно скоротити необхідні для виконання алгоритму ресурси комп'ютера, хоча, як і попередня особливість реалізації, звичайно, впливає на точність та ймовірність вдалого завершення алгоритму.

2.1.4. Переваги та недоліки алгоритму

Переваги:

- при позитивному виході алгоритму результатом його роботи завжди є ідеально узгоджена матриця;

- алгоритм дозволяє прийти до ідеально узгодженої матриці, зводячи до мінімуму кількість запитань після кожної відповіді експерта;
- алгоритм дозволяє виправляти багатократні помилки та помилки зі значним відхиленням значення від узгодженого, а також помилки, пов'язані з невірним ранжуванням альтернатив;
- можливість пристосування до різних типів шкал;

недоліки:

- алгоритм потребує значних обчислювальних ресурсів комп'ютера, а саме, важливими є продуктивність процесора та об'єм доступної оперативної пам'яті (значим по часу є підготовчий етап, при якому в оперативній пам'яті формулюються дані для подального використання);
- завершення алгоритму з позитивним результатом відбувається тільки при досягненні стовідсоткової узгодженості;
- враховуючи потужність сучасних ПК, використовувати алгоритм є сенс при розмірах вхідної (реальної) матриці парних порівнянь не більше як 7×7^1 .

2.2. Різновиди методу

При аналізі щойно описаного методу та при його тестуванні виникла ідея, що не завжди є оправданим вести процес діалогу з експертом аж до тих пір, поки реальна МПП буде повністю співпадати з однією із згенерованих ІУМПП, тобто до моменту повної узгодженості МПП. Ідея полягає в тому, щоб припинити процес пошуку ІУМПП, або навіть його не розпочинати, якщо на цей момент реальна МПП є достатньо узгодженою. Тим самим можливе додаткове скорочення кількості звернень до

¹ Цей недолік відносний, тому що враховуючи психофізіологічні можливості людини-експерта [84] та досить швидкий ріст кількості парних порівнянь при збільшенні кількості альтернатив, максимальний розмір матриці, що обробляється є 7×7 . Якщо є необхідність оцінки більше, ніж семи альтернатив, то альтернативи групуються, де кількість в групі не перевищує семи [50].

експерта, чого ми і прагнули розробляючи комбінаторні методи. Так виник різновид комбінаторного методу парних порівнянь – з додатковою перевіркою узгодженості МПП.

Цей метод цілком аналогічний описаному в підрозділі 2.1, з тією особливістю, що на пошуковому етапі алгоритму перед кожним звертанням до експерта перевіряється узгодженість МПП, отриманої з вихідної реальної МПП шляхом заміни тих її елементів на відповідні елементи ГУМПП, зміну яких було погоджено з експертом на попередньому кроці. Узгодженість перевіряється методом, описаним у [80], і у випадку, якщо розрахований коефіцієнт узгодженості більше або дорівнює порогові застосування – процес пошуку ГУМПП зупиняється і проводиться обчислення ваг по частково узгодженні матриці.

При використанні адитивних порівнянь для визначення ваг експерту пропонується методом безпосередньої оцінки визначити ненормоване абсолютноне значення $v_{i,\text{dom}}$ – домінуючої по i -ому рядку альтернативи. На підставі цього визначаються ненормовані абсолютноні значення ваг альтернатив:

$$v_h^* = \frac{1}{k} \sum_{i=1}^k (v_{i,\text{dom}} - d_{ih} + d_{h,\text{dom}}).$$

У випадку мультиплікативних порівнянь для визначення ваг альтернатив немає необхідності в безпосередній оцінці ненормованого абсолютноного значення домінуючої по кожному рядку альтернативи, а ненормовані значення ваг альтернатив знаходяться як середнє геометричне кожного рядка отриманої матриці.

2.3. Висновки

Запропонований метод парних порівнянь зі зворотним зв'язком з експертом, який можна віднести до методів комбінаторного типу, прагне звести кількість звертань за думкою експерта до мінімуму. В цьому методі запропоновано оригінальний підхід до вирішення проблеми знаходження експертних оцінок за результатами парних порівнянь альтернатив. Цей підхід полягає в генерації на базі реальної сформованої експертом матриці і впорядкуванні всіх можливих варіантів ідеально узгоджених ма-

триць парних порівнянь з подальшим пошуком серед цієї послідовності матриць саме тієї з них, до якої, погоджуючись з думкою експерта, можливо привести вихідну реальну матрицю парних порівнянь. Запропоновано декілька різновидів методу, котрі відрізняються характером парних порівнянь, що застосовуються (мультиплікативні або адитивні) та наявністю перевірки узгодженості матриці парних порівнянь для визначення моменту можливого більш швидкого завершення алгоритму.

Реалізація такого виду алгоритмів на персональних комп'ютерах ще декілька років тому через велику трудомісткість обчислень була неможливою, але зараз алгоритми такого плану безперечно заслуговують на увагу.

Основні наукові результати розділу опубліковані в [82].

Розділ 3.

Розробка групових методів отримання кардинальних експертних оцінок в умовах неповної визначеності

Різноманітні модифікації методів парних порівнянь широко використовуються в технологіях підтримки прийняття рішень як інструмент для побудови ієархій критеріїв [26, 27, 29, 30, 54] і, як спрощений інструмент, – для ранжування об'єктів по деяких критеріях. В більшості випадків для підвищення достовірності отримуваних результатів, якщо є така можливість, слід враховувати думку не одного окремого експерта, а групи експертів. Дослідження в області визначення групових рішень на основі індивідуальних є досить поширеними [19, 21, 52, 55–75, 80, 81]. В загальному випадку матриця парних порівнянь, яка сформована окремим експертом, є неповною, тобто результати деяких порівнянь пар об'єктів (альтернатив, критеріїв, цілей і т.п.) відсутні. Причини неповноти матриць можуть бути різні. Однією з найбільш розповсюджених причин є відсутність в експерта достатньої інформації або знань для того, щоб виконати порівняння деякої пари об'єктів за заданим критерієм. Наприклад, якщо експерт залучається для порівняння компетентності своїх колег у деякій

області, причиною відмовлення експерта A від порівняння експертів B і C за цим критерієм може бути недостатнє знання ним ступеня ерудованості експерта C через те, що цей експерт лише недавно ввійшов до складу групи.

Harker [79] першим розглянув задачу обробки неповної матриці парних порівнянь, виконаних у фундаментальній шкалі [26], за умови застосування методу власного вектора для обчислення відносних ваг альтернатив. Метод запропоновано використовувати для скорочення числа парних порівнянь, що мають метою отримання відносних ваг альтернатив. Запропонований ним метод зводиться до відновлення повної матриці порівнянь по наявній неповній і застосуванні до сформованої в такий спосіб повній матриці стандартного алгоритму обчислення ваг. Відновлення повної матриці ґрунтуються на використанні властивостей цілком узгодженої позитивної зворотної матриці парних порівнянь. Останнім часом були запропоновані ряд нових методів парних порівнянь у різних шкалах [28, 44, 45], застосування до яких методу [79] не передбачалося. Крім того, дуже актуальною стала задача отримання узгоджених агрегованих групових оцінок альтернатив з урахуванням різної компетентності експертів – членів групи. У даному розділі представлено метод отримання таких оцінок, придатний для методів парних порівнянь [28, 44, 45], за умови неповноти матриць порівнянь, виконуваних експертами. При цьому використовується спектральний метод обчислення кількісних показників ступеня узгодженості результатів парних порівнянь, запропонований у [80], а також ідея використання зворотного зв'язку з експертами в ході виконання ними парних порівнянь [28, 44, 45].

У ході розробки такого методу необхідно вирішити ряд окремих задач. Перша полягає у визначенні необхідних і достатніх умов, яким повинна задовольняти множина неповних матриць парних порівнянь для того, щоб вона була конструктивною, тобто дозволяла обчислити узгоджені групові оцінки об'єктів. Друге завдання полягає в розробці методу обчислення узагальнених оцінок об'єктів по конструктивній множині неповних часткових матриць порівнянь. Сутність наступної задачі полягає в розробці методу організації діалогу з експертами, що має метою досягти достатнього ступеня узгодженості оцінок, даних різними експертами, з урахуванням їх компетен-

тності. Рішення цих задач досить для отримання узгоджених агрегованих групових оцінок об'єктів по неповних часткових матрицях порівнянь за умови, що експертам не пропонується виконати порівняння додаткових пар об'єктів з метою поліпшення якості оцінок. Нижче приводяться строгі постановки названих задач і методи їхнього вирішення.

3.1. Постановка задачі. Сутність підходу

Означення 3.1.1. *Первинною оцінкою за критерієм K множини O об'єктів, даної h -м експертом, $h = (1, m)$, називається матриця парних порівнянь $D_h = \{d_{ijh}\}, i, j = (1, t)$, де d_{ijh} – ступінь переваги за критерієм K об'єкта o_i над об'єктом o_j , визначена h -м експертом.*

Означення 3.1.2. *Вторинною експертною оцінкою v_{jh} об'єкта $o_j \in O$, за критерієм K , визначеної по первинній оцінці D_h , даної h -им експертом, називається показник ступеня виразності властивостей цього об'єкта, обумовлених критерієм K , обчислений по оцінці D_h .*

Надалі будемо розрізняти ненормовані і нормовані до одиниці вторинні оцінки. Останні є показниками відносної значимості об'єктів за критерієм K . Для скорочення надалі будемо називати їх відносними вагами об'єктів.

Елементами множини D_h є дійсні числа, семантика яких залежить від обраного методу парних порівнянь оцінювання об'єктів. Так при використанні методу мультиплікативних парних порівнянь d_{ijh} показує в скільки разів на думку h -го експерта об'єкт o_i перевершує об'єкт o_j у відношенні критерію K . Якщо застосовується метод адитивного порівняння, d_{ijh} показує, на скільки об'єкт o_i перевершує об'єкт o_j [28, 44]. При використанні парних порівнянь у фундаментальній шкалі [26, 27] d_{ijh} є числовий еквівалент словесно вираженого ступеня переваги. При використанні методів парних порівнянь вторинні оцінки обчислюються по відповідному алгоритму на підставі первинної оцінки множини об'єктів.

Метод безпосередньої оцінки можна розглядати як вироджений випадок методу парних порівнянь, при якому всі об'єкти порівнюються з одним еталоном. При цьому первинна оцінка множини об'єктів є множина їх ненормованих вторинних оцінок.

Задача групового оцінювання відносної значимості об'єктів за критерієм K формулюється в такий спосіб. Є множина $E = \{e_h\}$ експертів, $h = (1, m)$, множина $O = \{o_j\}, j = (1, t)$ об'єктів, множина $D = \{D_h\}$ первинних оцінок об'єктів, даних експертами, причому $\exists h : |D_h| < p_{\max}$, де p_{\max} – максимальна для обраного методу оцінювання кількість елементів (ступенів переваги) первинної оцінки множини об'єктів; h -й експерт характеризується нормованим відносним коефіцієнтом компетентності c_h . Задано алгоритм визначення ненормованих вторинних експертних оцінок $v_j, j = (1, t)$ значимості об'єктів щодо критерію K (надалі для скорочення – “ваг”) по первинній оцінці об'єктів. Потрібно знайти агреговані узгоджені нормовані вторинні оцінки w_j відносної значимості об'єктів за критерієм K .

Визначимо більш строго поняття “агреговані” та “узгоджені” оцінки. Не втрачаючи спільності, будемо вважати, що вторинні ненормовані експертні оцінки v_j являють собою номери поділок деякої шкали з n поділками. Якщо v_j представлена дійсним числом, то, вибравши припустиму похибку ε , неважко перейти до задання цієї оцінки у вигляді номера поділки шкали з $n = 1 + 1/2\varepsilon$ поділками. Тип шкали визначається алгоритмами визначення вторинних експертних оцінок (ЕО). Передбачається, що шкала є шкалою відношень, інтервалів, різностей або абсолютною [52], тому що саме в шкалах цього типу представляються ненормовані значення показників відносної значимості об'єктів, отримані при використанні відомих і пропонованих методів.

Якщо число членів групи більше одиниці, то при будь-якому методі оцінювання кожен об'єкт одержує деяку множину вторинних оцінок, потужність якої визначається кількістю членів групи і потужностями первинних оцінок. Множину $V_j = \{v_{hj}\}$, де v_{hj} – вторинна оцінка ваги об'єкта A_j , що дана h -им експертом зручно задавати спектром R_{vj} .

Означення 3.1.3. Спектром R_{vj} множини V_j експертних оцінок, представлених у шкалі з n поділками, називається n -позиційний вектор, на i -й позиції якого за-

писане число експертних оцінок, округлених до i -ої поділки шкали.

При визначенні агрегованих оцінок з урахуванням компетентності експертів і неповноти їхніх первинних оцінок множину V_j оцінок j -го об'єкта будемо задавати зваженим спектром.

Означення 3.1.4. Зваженим спектром S_{vj} множини V_j експертних оцінок j -го об'єкта, представлених у шкалі з n поділками, називається n -позиційний вектор, на i -й позиції якого записана сума коефіцієнтів відносної компетентності експертів у групі, що оцінили j -й об'єкт, оцінки яких округлені до i -ої поділки шкали.

Означення 3.1.5. Узгодженими агрегованими відносними оцінками $w_j, j = (1, t)$ важливості об'єктів за критерієм K називаються нормовані до одиниці коректно обчислені середні достатньо узгоджених множин ненормованих вторинних оцінок важливості, даних t експертами.

Коректність обчислення середнього визначається відповідністю операції знаходження середнього типу шкали, у якій представлені експертні оцінки. Для шкал названих типів припустимою операцією визначення середнього є операція обчислення середнього арифметичного [19].

Зважений спектр S_{vj} множини V_j експертних оцінок вважається достатньо узгодженим, якщо його коефіцієнт узгодженості перевищує деякий поріг застосування. Для кількісної оцінки ступеня узгодженості спектра S_{vj} будемо використовувати спектральний коефіцієнт узгодженості [52], що визначається виразом:

$$k(S_{vj}) = \left(1 - \frac{\sum_{i=1}^n \sigma_i \left| i - \sum_{i=1}^n i\sigma_i \right| - \sum_{i=1}^n \sigma_i \ln \sigma_i}{G \sum_{i=1}^n |i - (n+1)/2| + \ln n} \right) z \quad (3.1)$$

де

$$G = \frac{m_j}{\ln m_j \cdot n \cdot \ln n} - \text{масштабний коефіцієнт};$$

$$z = \begin{cases} 1, & \text{якщо } z^* = \text{"істина"} \\ 0, & \text{якщо } z^* = \text{"хиба"} \end{cases}$$

$$z^* = \overline{[i(1) = 1]} \vee \overline{[i(n) = 1]} \bigvee_{d=1}^{q_j-1} \overline{[\sigma_i(d) = \sigma_i(d+1)]} \bigvee_{d=1}^{q_j-1} \overline{[i(d) - i(d+1)]} = const \quad (3.2)$$

де m_j – кількість експертів, що оцінили j -й об'єкт;

q_j – кількість груп експертів, що дали однакові оцінки j -го об'єкта;

$i(d)$ – номер поділки шкали, до якого округлені оцінки, дані експертами d -ої групи,

$d = (1, q_j)$;

$\sigma_i(d)$ – сума коефіцієнтів компетентності експертів, оцінки яких округлені до поділки з номером $i(d)$;

z^* – булева функція, що задає необхідні і достатні умови рівності 0 коефіцієнта узгодженості $k(S_{vj})$.

З (3.1),(3.2) випливає, що значення коефіцієнта узгодженості знаходяться в межах $[0,1]$, причому цілком узгоджених множині ЕО відповідає значення 1, а найменш узгодженному – 0.

Для визначення достатності ступеня узгодженості спектра S_{vj} використовуються значення порога виявлення T_{oj} і порога застосування T_{uj} [80].

Означення 3.1.6. Порогом виявлення T_{oj} називається коефіцієнт узгодженості множини експертних оцінок, даних m_j експертами, що містить мінімальну кількість інформації, яку можливо зареєструвати.

Поріг виявлення визначається виразом [80]: $T_{oj} = k(S_{oj})$, де S_{oj} являє собою зважений спектр, у якому $m_j = n$; $\sigma_1 = 0$; $\sigma_g = 2/n$; $\sigma_{y \neq g \neq 0} = 1/n$; $g = [n/2 + 1]$.

Означення 3.1.7. Порогом застосування T_{uj} називається коефіцієнт узгодженості множини експертних оцінок, що забезпечує обчислення узагальненої експертної оцінки з припустимою точністю.

Поріг застосування T_u дорівнює коефіцієнту узгодженості $k(S_{uj})$ зваженого спектра S_{uj} , у якому $m_j = 2$, $\sigma_i = \sigma_{i+1} = 1/2$; $\sigma_{h \neq i} = 0$; $\sigma_{h \neq (i+1)} = 0$ [80].

Означення 3.1.8. Конструктивною щодо заданого методу визначення відносної значимості об'єктів називається множина первинних оцінок множини об'єктів, даних експертами, по якій, використовуючи цей метод, можна визначити вторинні агреговані узгоджені оцінки відносної значимості об'єктів.

Розглянемо методи узгодження й агрегування експертних оцінок відносної значимості об'єктів.

3.2. Метод безпосередньої оцінки

При використанні цього методу h -й експерт $h = (1, m)$ дає незалежні ненормовані оцінки v_{jh} j -ому об'єкту $j = (1, t)$ з деякої підмножини O_h , причому оцінки, що даються h -им експертом, знаходяться в межах $q_{h\min} - q_{h\max}$. У загальному випадку $h \neq k \rightarrow q_{h\min} \neq q_{k\min}; q_{h\max} \neq q_{k\max}$. Тому виникає необхідність перерахунку оцінок об'єктів у шкалах з різними межами до єдиної шкали, що має, як правило, найбільшу кількість поділок. Оцінку v_{jh} у шкалі з межами $q_{h\min}, q_{h\max}$ через оцінку v_{jk} з межами $q_{k\min}, q_{k\max}$ виразимо в такий спосіб:

$$v_{jh} = a_{kh}v_{jk} + b_{kh}. \quad (3.3)$$

Коефіцієнт перерахування a_{kh} і зміщення b_{kh} рівні:

$$a_{kh} = (q_{h\max} - q_{h\min}) / (q_{k\max} - q_{k\min}); \quad (3.4)$$

$$b_{kh} = (q_{k\max}q_{h\min} - q_{h\max}q_{k\min}) / (q_{k\max} - q_{k\min}). \quad (3.5)$$

Після перетворення оцінок об'єктів до єдиної шкали з використанням (3.3)-(3.5) будується спектри ваги кожного об'єкта. У спектр S_j j -го об'єкта включаються оцінки $v_{j1}, v_{j2}, \dots, v_{jm}$, дані йому різними експертами. У загальному випадку різні об'єкти мають різні кількості m_j оцінок, причому розрізняються і множини експертів, що оцінили різні об'єкти. Після цього відповідно до (3.1) для кожного j -го спектра обчислюються коефіцієнт узгодженості $k(S_{gj})$, а також поріг T_{oj} виявлення і поріг T_{uj} застосування. При цьому для кожного об'єкта потрібно попередньо нормувати до одиниці коефіцієнти компетентності експертів, що дали оцінки цього об'єкта, поклавши рівними нулю коефіцієнти компетентності експертів, що утрималися від оцінювання даного об'єкта. Далі множина \bar{S} спектрів розділяється на три непересічних підмножини. У підмножину \bar{S}_1 включаються спектри, для яких

$$k(S_{gj}) < T_{oj}, \quad (3.6)$$

у підмножину \bar{S}_2 входять ті спектри, для яких

$$T_{oj} \leq k(S_{dj}) < T_{uj}. \quad (3.7)$$

Елементами підмножини \bar{S}_3 є спектри, що задовольняють умові

$$k(S_{lj}) \geq T_{uj}. \quad (3.8)$$

Виконання умови (3.6) означає, що спектри $S_j \in \bar{S}_1$ не містять інформації. При цьому всім експертам пропонується переглянути свої оцінки тих об'єктів, спектри ваг яких задовольняють цю умову, для чого за рішенням керівника роботи можуть бути організовані консультації експертів.

Щодо спектрів підмножини \bar{S}_2 можна стверджувати, що кожний з них містить інформацію, однак ступінь узгодженості недостатній для визначення узагальненої оцінки з достатньою точністю. Тому для поліпшення узгодженості таких спектрів організується діалог з експертами. Процедура організації діалогу щодо кожного спектра аналогічна і полягає в наступному. Обчислюється середня оцінка ваги об'єкта. Оскільки при використанні методу безпосереднього оцінювання оцінки ваги об'єкта представлені в абсолютній шкалі, то за умови рівної компетентності експертів середня оцінка v_d дорівнює середньому арифметичному компонент спектра, даних різними експертами. Якщо експерти мають різну компетентність, то:

$$v_d = \sum_{h=1}^m v_{dh} c_h. \quad (3.9)$$

Далі визначається експерт e_f , для якого $c_f > 0$ і в розглянутому спектрі S_d величина $\Delta = |v_{df} - v_d|/c_f$ – максимальна. Цьому експерту пропонується переглянути свою оцінку v_{df} об'єкта o_d у напрямку зменшення Δ . (Приклад звертання: "Ваша оцінка величини ваги об'єкта o_d істотно відрізняється від оцінок інших експертів. Чи не згодні Ви збільшити (зменшити) Вашу оцінку? Колишня оцінка дорівнює v_{df} ". Якщо експерт згоден, то його оцінка змінюється так, щоб у спектрі S_d компонента v_{df} перемістилася вбік середнього на одну поділку шкали, у якій представлена експертні оцінки. Після цього знову розраховується коефіцієнт узгодженості перетвореного спектра S_d і виконується описаний вище аналіз з наступним звертанням до експерта

при необхідності. Якщо експерт e_f відмовився коригувати оцінку v_{df} , то він виключається з множини E_d експертів, до яких можна звертатися з пропозицією змінити оцінку ваги об'єкта o_d на наступних кроках, і описаним вище методом визначається експерт e_p , якому відповідає наступне по величині відхилення Δ , після чого організується аналогічний діалог з цим експертом. Якщо на деякому кроці виявилося, що $[E_d = \emptyset] \wedge [T_{od} \leq k(S_d) < T_{ud}]$, то це означає, що можливості поліпшення узгодженості вичерпані, незважаючи на те, що вона недостатня. У цьому випадку робиться висновок про необхідність зміни складу бригади експертів.

Виконання описаних процедур поліпшення узгодженості або приводить до того, що на деякому кроці спектри ваг всіх об'єктів належать підмножині \bar{S}_3 , або робиться висновок про неможливість з цією бригадою експертів одержати достатньо узгоджених оцінок. Приналежність спектра підмножині \bar{S}_3 означає, що він досить узгоджений, і тому по ньому можна знайти агреговану оцінку ваги об'єкта, що дорівнює середньому значенню, що обчислюється виразом (3.9). На завершальному етапі нормують значення v_j і одержують коефіцієнти w_j відносної значимості об'єктів:

$$w_j = v_j / \sum_{h=1}^t v_h. \quad (3.10)$$

З викладеного алгоритму випливає необхідна і достатня умова конструктивності щодо методу безпосередньої оцінки множини $D = \{D_h\}, 1 \leq h \leq m$, первинних оцінок множини $O = \{o_j\}, 1 \leq j \leq t$ об'єктів:

$$\forall j : 1 \leq j \leq t [\exists h : 1 \leq h \leq m [v_{jh} \neq *]],$$

де $v_{jh} = *$, якщо h -й експерт не оцінював j -й об'єкт.

3.3. Метод “лінія”

Сутність методу полягає в наступному. Виберемо довільно деякий об'єкт o_e і запропонуємо кожному експерту зробити його попарне порівняння з кожним з $(t-1)$ об'єктів, що залишилися щодо заданого критерію K . У загальному випадку експерти в силу різних причин будуть утримуватися від порівняння деяких об'єктів з об'єктом o_e . У

результаті кожен експерт e_h сформує множину $D_{he} = \{d_{hej}\}$, $j \neq e$ ступені переваги об'єкта o_e над іншими об'єктами у відношенні критерію K , причому $|D_{he}| \leq (t - 1)$. Сформуємо рядок об'єднаної матриці парних порівнянь, елементи якого визначаються виразом:

$$d_{ej} = \bigcup_h (d_{hej}; c_{hj})$$

де c_{hj} – коефіцієнт компетентності h -го експерта, що виконав порівняння об'єктів o_e і o_j .

Примітка: якщо експерт e_h відмовився порівняти об'єкти o_e і o_j , то $d_{hej} = *$.

Присвоїмо об'єкту o_e вагу v_e . Задамо функцію

$$v_{jh} = f(v_e, d_{hej}) \quad (3.11)$$

і обчислимо оцінки ваги v_{jh} кожного j -го об'єкта, обумовлені компонентами, записаними в клітку d_{ej} . Очевидною вимогою до функції (3.11) є її монотонність. Вигляд цієї функції залежить від типу питань, що ставляться перед експертами в ході парних порівнянь. Якщо експерту ставиться питання: "У скільки разів об'єкт o_e перевершує об'єкт o_j у відношенні щодо критерію K ?" (мультиплікативні порівняння), то

$$v_{jh} = v_e \cdot \varphi(d_{hej}) \quad (3.12)$$

де $\varphi(d_{hej})$ – довільна монотонна функція, що задовольняє умові

$$\varphi(1) = 1.$$

При використанні адитивних парних порівнянь експерту пропонують визначити: "На скільки об'єкт o_e перевершує об'єкт o_j у відношенні щодо критерію K ?" . При цьому

$$v_{jh} = v_e + \lambda(d_{hej}) \quad (3.13)$$

де $\lambda(d_{hej})$ – довільна монотонна функція, що задовольняє умові

$$\lambda(0) = 0.$$

Таким чином, кожний з об'єктів o_j одержує деяку множину експертних оцінок v_{jh} ваги, кількість яких у найкращому випадку дорівнює t . Побудуємо для кожного об'єкта o_j зважений спектр значень ваги, включивши в нього компоненти

$v_{j1}, v_{j2}, \dots, v_{jm}$, попередньо пронормувавши коефіцієнти компетентності експертів, що порівняли ті ж самі об'єкти з o_e . Неважко бачити, що при цьому задача досягнення необхідного рівня узгодженості експертних оцінок зводиться до розглянутої вище задачі узгодження оцінок, отриманих методом безпосереднього оцінювання. Єдина відмінність полягає в тому, що для зміни величини оцінки v_{jf} експерту e_f ставляться запитання типу: "Чи не погодились би Ви зменшити (збільшити) ступінь переваги об'єкта o_e над об'єктом o_j ?" При цьому напрямок бажаної зміни визначається видом використовуваної функції (3.11) і положенням компонент спектра v_{jf} відносно середнього значення. Після досягнення достатнього ступеня узгодженості спектра ваги кожного об'єкта (тобто виконання умови (3.8) для кожного спектра) середні значення ваг приймаються в якості агрегованих узгоджених оцінок, що, використовуючи (3.10), нормують, у результаті чого утворюються коефіцієнти w_j відносної значимості об'єктів щодо критерію K . Трудомісткість алгоритму дорівнює $(t - 1)$, де t – число об'єктів.

З викладеного алгоритму визначення відносних ваг об'єктів випливає необхідна і достатня умова конструктивності щодо методу "лінія" множини первинних оцінок множини об'єктів:

$$\forall j, e : j \neq e [\exists h : d_{hej} \neq *]. \quad (3.14)$$

Розглянемо метод організації парних порівнянь, використовуючи цю методику за умови, що експерти можуть при необхідності виконувати порівняння додаткових пар об'єктів. При цьому можливі кілька критеріїв якості такого процесу. Перший очевидний критерій полягає в мінімізації часу отримання оцінок відносних ваг об'єктів. Як випливає з (3.14), мінімально необхідне число порівнянь різних об'єктів з об'єктом o_e , що повинні спільно виконати t експертів, дорівнює $t - 1$. Таким чином, оптимальна за часом (за умови, що час порівняння пари об'єктів не залежить від того, які це об'єкти і який експерт виконує порівняння) процедура полягає в наступному: множина об'єктів розділяється на $](t - 1)/m[$ різних підмножин і кожному експерту пропонується виконати порівняння об'єктів з відповідної підмножини з об'єктом o_e , після чого результати обробляються описаним вище способом.

Другий очевидний критерій якості процесу визначення відносних ваг об'єктів полягає в оптимізації вірогідності одержуваної інформації. Як випливає з описаного алгоритму, у загальному випадку ненормовані ваги об'єктів визначаються по різних множинах оцінок, даних експертами різної компетентності. Оскільки результуюче нормоване значення відносної ваги деякого об'єкта залежить від ненормованих значень ваг всіх об'єктів, то як показник вірогідності результуючої інформації природно використовувати величину:

$$Q_j = \inf_h \left(\sum c_{hj} \right). \quad (3.15)$$

Звідси випливає оптимальна процедура організації парних порівнянь, що забезпечує необхідну вірогідність результатів: визначити об'єкт o_j , якому відповідає мінімальна сума коефіцієнтів компетентності експертів, що порівняли його з об'єктом o_e , і запропонувати найбільш компетентному експерту з підмножини експертів, що ухилились від порівняння цих об'єктів, виконати це порівняння. Цю процедуру потрібно продовжувати доти, поки показник вірогідності, обумовлений виразом (3.15), менший припустимої величини.

3.4. Метод “трикутник”

Не втрачаючи спільності, будемо вважати для спрощення викладу, що множину $O = \{o_j\}, j = (1, t)$ об'єктів упорядковано по номерах. Виберемо довільно деякий об'єкт $o_1 = o_{e(t)}$ (це може зробити хто-небудь з експертів) і запропонуємо кожному експерту e_h зробити ті його попарні порівняння з $(t - 1)$ об'єктами, що залишилися щодо заданого критерію K , для яких експерт має достатню інформацію, і не існує причин іншого роду, що перешкоджають цим його діям. У результаті кожний з експертів сформує власну, у загальному випадку неповну, множину ступенів переваги об'єкта $o_1 = o_{e(t)}$ над іншими. Позначимо таку множину, сформовану h -им експертом, через $D_{h1} = \{d_{h1z}\}, z = (2, t)$. При цьому якщо h -й експерт відмовився виконати порівняння z -го об'єкта з o_1 , то вважаємо, що $d_{h1z} = *$. Об'єкт $o_{e(t)}$ виключається з множини O і для підмножини $(t - 1)$ об'єктів, що залишилися, виконується та ж процедура, тобто вибирається об'єкт $o_2 = o_{e(t-1)}$ і кожним експертом виконуються припустимі для

нього попарні порівняння з іншими ($t - 2$) об'єктами, у результаті чого утворюються множини $D_{h2} = \{d_{h2x}\}, x = (3, t), h = (1, m)$ ступенів переваги об'єкта $o_{e(t-1)}$ над іншими. Після цього цей об'єкт $o_{e(t-1)}$ виключається з множини об'єктів і описані процедури повторюються. Цей процес продовжується доти, поки в скороченні в такий спосіб множині не залишиться єдиний об'єкт. У результаті кожний з m експертів сформує трикутну, неповну в загальному випадку, матрицю порівнянь.

Для кожного рядка кожної такої матриці застосуємо описаний вище алгоритм “лінія” при обчисленні абсолютнох ваг об'єктів, причому при виконанні алгоритму для u -го рядка кожної матриці $u = (2, t)$ у якості еталонного будемо вибирати об'єкт $o_u = o_{e(t-u+1)}$. Особливість застосування алгоритму “лінія” у цьому випадку полягає в тому, що абсолютна вага v_{uh} об'єкта $o_u = o_{e(t-u+1)}$ при виконанні алгоритму “лінія” для u -го рядка матриці, сформованої h -им експертом, вибирається не довільно, а рівною

$$v_{uh} = f(v_{(u-1)}^*, d_{h(u-1), u}), \quad (3.16)$$

де $v_{(u-1)}^*$ – узгоджена агрегована оцінка ваги об'єкта $o_{(u-1)}$;

$d_{h(u-1), u}$ – відмінний від $*$ елемент $d_{(u-1), u}$ матриці порівнянь, сформованої по оцінках h -го експерта.

Через v_j^i будемо надалі позначати абсолютну вагу об'єкта o_j , обчислену відповідно до (3.11) за умови, що для підмножини, яка містить k -й об'єкт, у якості “еталона” обрано об'єкт $o_{e(k)} = o_i$.

Для визначення агрегованої узгодженої оцінки ваги об'єкта $o_{(u-1)}$, що є еталонним для u -их рядків $u = (2, t)$ матриць порівнянь, даних різними експертами, побудуємо спектр $S_{(u-1)}$ значень $v_{(u-1)h}$, обчислених з (3.16) по оцінках, даних всіма експертами відносно ваг об'єктів, що були еталонними для попередніх рядків.

Оскільки вага об'єкта $o_{e(t)}$ не залежить від дій експерта, варто покласти її рівною деякій константі $v_{e(t)}$, однаковій для всіх експертів. У зв'язку з цим спектр $S_{e(t)}$ оцінок ваги об'єкта $o_{e(t)}$ складається з m одинакових компонент, рівних $v_{e(t)}$. З (3.1), (3.2) випливає, що при цьому $k(S_{e(t)}) = 1 > T_u$, тому для спектра $S_{e(t)}$ можна обчислити агреговану оцінку ваги, що, як випливає з (3.9), дорівнює $v_{e(t)}$.

Як випливає з (3.16), величина ваги $v_{e(t-1)}^{e(t)}$ залежить не тільки від однакової для всіх експертів величини $v_{e(t)}$, але і від обумовленої експертами величини $d_{he(t)e(t-1)}$ ступеня переваги об'єкта $o_{e(t)}$ над об'єктом $o_{e(t-1)}$. Спектр $S_{e(t-1)}$ ваги об'єкта $o_{e(t-1)}$ містить $m_1 \leq m$ у загальному випадку різних компонент $v_{e(t-1)1}^{e(t)}, v_{e(t-1)2}^{e(t)}, \dots, v_{e(t-1)m_1}^{e(t)}$ обчислених по оцінках $d_{he(t)e(t-1)}$, даним m_1 експертами, за умови, що узгоджене агреговане значення ваги об'єкта $o_{e(t)}$, котрий є еталонним для множини, яка містить t об'єктів, дорівнює $v_{e(t)}$. Відмінності цих компонент є наслідком неузгодженості оцінок експертів. Виконаємо описану в попередньому розділі процедуру поліпшення узгодженості і визначення агрегованої оцінки ваги $v_{e(t-1)}$ об'єкта $o_{e(t-1)}$. Визначивши агреговані оцінки $v_{e(t)}^*$, $v_{e(t-1)}^*$, можна, використовуючи (3.16), обчислити значення компонент $v_{e(t-2)h}^{e(t)}, v_{e(t-2)h}^{e(t-1)}$ спектра $S_{e(t-2)}$ ваги об'єкта $o_{e(t-2)}$, по оцінках m_2 експертами ступенів переваги об'єкта $o_{e(t)}, o_{e(t-1)}$ над об'єктом $o_{e(t-2)}$. У такий спосіб спектр $S_{e(t-2)}$ буде мати $m_1 + m_2$ компонент. Відмінності їхніх значень викликаються як внутрішньою неузгодженістю оцінок кожного експерта, так і неузгодженістю оцінок різних експертів. Внутрішні протиріччя можуть бути викликані як природною суперечливістю проявів властивостей об'єктів, так і впливом на експерта різноманітних психофізіологічних факторів (утомою, неуважністю, впливом зовнішніх факторів і т.д.). Хоча природа внутрішньої неузгодженості відмінна від природи зовнішньої неузгодженості, прояв цих двох видів неузгодженості однаковий і полягає в розходженні компонент спектра ваги. Тому для підвищення ступеня узгодженості спектра можна скористатися однією і тією ж процедурою, описаною в попередньому підрозділі. У результаті її виконання або буде досягнуте достатнє значення коефіцієнта узгодженості спектра ваги кожного об'єкта, або буде зроблено висновок про неможливість одержання узгоджених агрегованих оцінок за допомогою наявної бригади експертів. У першому випадку визначається агрегована узгоджена оцінка $v_{e(t-2)}^*$, що дає можливість обчислити по (3.16) значення $v_{e(t-3)h}^3$. Описаний процес повторюється t разів. При цьому кількість компонент спектра $S_{e(t-u)}$ ваги об'єкта $o_{e(t-u)}$ дорівнює $\sum_1^{(t-1)} m_u$. На завершальному етапі пронормуємо агреговані узгоджені значення v_j , використовуючи (3.10), і одержимо коефіцієнти w_j значимості об'єктів щодо критерію.

рію K .

Визначимо необхідні і достатні умови конструктивності щодо методу “трикутник” множини неповних індивідуальних матриць парних порівнянь, скориставшись для цього графовою інтерпретацією. Побудуємо мультиграф $G(O, D)$, де $O = \{o_j\}, j = (1, t)$ – множина вершин, кожна з яких позначена відповідним об’єктом, що підлягає оцінюванню; D – множина дуг. Дуга $d_{ijh} \in D$ виходить з вершини, позначену об’єктом o_i і входить у вершину, позначену об’єктом o_j , якщо h -й експерт дав оцінку d_{ij} ступеня переваги об’єкта o_i над об’єктом o_j . Необхідні і достатні умови конструктивності щодо методу “трикутник” множини індивідуальних експертних оцінок визначає наступна теорема.

Теорема 3.4.1. *Для того, щоб множина D первинних експертних оцінок множини O об’єктів була конструктивною щодо методу “трикутник”, необхідно і достатньо, щоб у мультиграфі $G(O, D)$ існуває хоча б один простий шлях, що включає усі вершини $o_j \in O$.*

Доведення. *Достатність.* Якщо умови теореми виконуються, то $\forall j \exists h, i : (1 \leq h \leq m, 1 \leq i \leq t)[(d_{ij} \neq *) \vee (d_{ji} \neq *) \vee [(d_{ij} \neq *) \wedge (d_{ji} \neq *)]]$.

Якщо $\exists h, i : (1 \leq h \leq m, 1 \leq i \leq t)[(d_{ij} \neq *)]$, то o_i на деякому кроці алгоритму буде еталонним об’єктом, тому можна обчислити ненормоване значення ваги об’єкта o_j , рівне

$$v_{jh} = f(v_i^*, d_{hij}).$$

Якщо виконано тільки умова $\exists h, i : (1 \leq h \leq m, 1 \leq i \leq t)[(d_{ij} \neq *)]$, то відповідно до алгоритму “трикутник” об’єкт o_j повинен бути обраним у якості еталонного вже на першому кроці і відносно його ваги v_{jh} повинні виражатися ваги всіх інших об’єктів. Аналогічно, при $\exists h, i : (1 \leq h \leq m, 1 \leq i \leq t)[(d_{ij} \neq *) \wedge (d_{ji} \neq *)]$ виконуються умови обчислення компонент спектра ваги v_j як функції від узгодженого на попередньому кроці значення ваги еталонного об’єкта і використання узгодженого значення v_j як ваги еталонного об’єкта на наступному кроці алгоритму.

Необхідність. Допустимо зворотне, що в графі не існує такого шляху, але множина D конструктивна. Тоді граф $G(O, D)$ включає принаймні два ізольованих під-

графи G_1 та G_2 . (В загальному випадку кількість таких підграфів $k \geq 2$). Якщо кожний з ізольованих підграфів G_1, G_2, \dots, G_k є сильно зв'язний, то згідно з доказаною вище достатністю цієї умови для конструктивності підмножин оцінок, які відповідають G_1, G_2, \dots, G_k , для кожної з них можна визначити відносні ваги об'єктів в межах цих підмножин. Але через те, що відсутні дуги, які з'єднують вершини різних підграфів, неможливо визначити відношення між вагами об'єктів, що належать до різних підграфів, і через це неможливо визначити відносні ваги в межах повної множини об'єктів, що доводить неконструктивність множини D . Отримане протиріччя доводить необхідність умови.

Таким чином, конструктивність множини D доведена.

3.5. Метод “квадрат”

Сутність цього методу визначення агрегованих узгоджених значень відносних ваг об'єктів o_1, o_2, \dots, o_t полягає в наступному. Запропонуємо кожному експерту виконати t разів алгоритм “лінія” стосовно до повної множини об'єктів, причому, при кожнім виконанні алгоритму в якості еталонного експерт повинен вибирати новий об'єкт. Змінимо і спосіб обчислення абсолютнох ваг об'єктів при кожнім виконанні алгоритму “лінія”. При виконанні його в k -й раз $k = (1, t)$ експертом $e_h, h = (1, m)$ визначимо найменший елемент d_{hkp} сформованого k -го рядка матриці парних порівнянь (МПП). Якщо без втрати спільноті припустити, що $\partial f(v_k; d_{kp}) / \partial (d_{kp}) < 0$, то об'єкт o_{ph} , яким позначений p -й стовпець h -ої МПП ($1 \leq k \leq t$) має найбільшу вагу v_{ph}^k . Назовемо цей об'єкт домінуючим по k -ому рядку h -ої МПП і позначимо його через o_{ph} , а його вагу – через v_{ph}^k . Якщо при парному порівнянні експерту задається питання: ”На скільки об'єкт o_k перевершує об'єкт o_r ?“ (адитивне порівняння), то запропонуємо йому визначити вагу v_{ph}^k об'єкта o_k методом безпосереднього оцінювання. Якщо ж використовується мультиплікативне порівняння, тобто експерту задається питання ”У скільки ... ?“, то вважають $v_{ph}^k = 1$. Використовуючи вираз:

$$v_{kh}^k = f(v_{ph}^k, d_{hpk}), \quad (3.17)$$

визначимо по елементах k -го рядка h -ої МПП ненормалізоване значення ваги k -го об'єкта. Будемо називати v_{kh}^k власною вагою об'єкта o_k .

З виразу:

$$v_{jh}^k = f(v_{kh}^k, d_{hkk}), \quad (3.18)$$

визначимо ваги v_{jh}^k всіх інших об'єктів. Ці значення надалі будемо називати транзитивними.

Повторюючи цей процес для всіх рядків h -ої МПП, для кожного об'єкта o_j побудуємо об'єднаний зважений спектр ваги $S_j = \cup S_{jh}$, де S_{jh} – зважений спектр оцінок абсолютних значень ваги, даних h -им експертом, що містить усі компоненти ваги об'єкта o_j , отримані по оцінках всіх експертів щодо всіх об'єктів. У загальному випадку кожен експерт не оцінює деякі пари об'єктів. Тому кількості експертів, що оцінюють ту чи іншу компоненту спектра ваги об'єкта, різні. Це змушує для кожної компоненти об'єднаного спектра розраховувати показники відносної компетентності експертів, по оцінках яких визначалася ця компонента. Присутність у спектрі оцінок ваги об'єкта декількох компонент, що знаходяться на різних поділках шкали, свідчить про наявність протиріч у результатах парних порівнянь, обумовлених як внутрішньою неузгодженістю результатів, даних одним експертом, так і зовнішньою неузгодженістю оцінок різних експертів.

Представлення результатів парних порівнянь, що виконані декількома експертами методом “квадрат”, множиною об'єднаних спектрів ваг об'єктів дає можливість поширити метод кількісної оцінки узгодженості множини результатів безпосередньої оцінки, розглянутий в розділі 3.2, а також метод оцінки узгодженості результатів парних порівнянь одним експертом [80], на випадок, коли неповні парні порівняння t об'єктів виконують m експертів. Для цього, використовуючи (3.1), (3.2), по об'єднаних спектрах S_j ваг обчислимо коефіцієнти узгодженості $k(S_j)$, а також граничні значення коефіцієнта узгодженості, з огляду на дійсну кількість експертів, що дали оцінки, по яких обчислені компоненти ваги відповідного об'єкта.

Коефіцієнтом узгодженості $k(O)_m$ результатів парних порівнянь множини $O =$

$\{o_j\}, j = (1, t)$ об'єктів, виконаних t експертами, будемо вважати величину

$$k(O)_m = \inf k(S_j). \quad (3.19)$$

Якщо

$$\exists j : [k(S_j) < T_{oj}], \quad (3.20)$$

то спектр S_j не несе інформації. Тому не можна обчислити середнє значення v_j і на підставі цього видати рекомендації про напрямок змін переваг об'єкта, що приводить до поліпшення узгодженості. У цьому випадку варто провести тур консультацій експертів, використовуючи метод Делфі [81]. Тому при викладі алгоритму досягнення достатньої узгодженості будемо припускати, що

$$\forall j : 1 \leq j \leq t [k_c(S_j) \geq T_{oj}] \wedge \exists j : [k(S_j) < T_{uj}]. \quad (3.21)$$

У цьому випадку всі спектри S_j містять корисну інформацію, але точність визначення wag об'єктів, обчислених по спектрах, що задовольняють (3.21) – недостатня. Використовуючи припустимі для розглянутої шкали операції [19], обчислимо середнє значення v_j wag об'єкта o_j для спектра S_j і визначимо множину V_j значень wag v_{jh}^k , для яких у розглянутому спектрі S_j величина $\Delta = |v_{df} - v_d| / c_f$ – максимальна. Цю процедуру необхідно виконати для всіх S_j , що задовольняють (3.21).

Розділимо множину $V = \bigcup_{j=1}^t V_j$ на три підмножини V^p, V^o, V^t . Значення v_{ph} wag домінуючих об'єктів включаються в підмножину V^p ; V^o – це підмножина власних wag об'єктів. Компоненти спектра, що є транзитивними wagами, включаються в підмножину V^t .

Упорядкуємо елементи підмножин V^p, V^o, V^t відповідно до зменшення величин Δ . Збільшити коефіцієнт узгодженості спектра можна за рахунок зменшення відмінностей його компонент від середнього значення шляхом відповідної зміни елементів матриць парного порівняння.

Процес підвищення коефіцієнта узгодженості починається з елементів підмножини V^p , тому що зміна $v_{ph}^k \in V^p$ приводить до змін власних значень v_{kh}^k і транзитивних значень v_{jh}^k wag об'єктів. Визначимо спектр S_j , якому відповідає перший елемент v_{ph}^k підмножини V^p . Тому що в цьому випадку підвищення значення $k_c(S_j)$ може бути

досягнуто зменшенням різниці між v_k і v_{ph}^k , то запропонуємо h -ому експерту змінити оцінку ваги домінуючого об'єкта o_{kh} . Оскільки завжди $v_{ph}^k > v_k$, то збільшення $k_c(S_k)$ можна досягти, зменшуючи v_{ph}^k , для чого потрібно запропонувати h -ому експерту збільшити d_{hkp} . Якщо експерт погодився на це, то для нового значення d_{hkp} обчислюється v_{ph}^k і для всіх спектрів обчислюються нові значення компонент, що мають верхній індекс ' k '. Для цього використовуються описані вище процедури. Якщо h -й експерт відмовився змінити значення d_{hkp} , то значення v_{ph}^k виключається з підмножини V^p і, якщо $V^p \neq \emptyset$, то ця операція виконується для другого елемента підмножини V^p і т.д.; у протилежному випадку переходять до перетворення підмножини V^o .

Нехай $v_{yh}^y \in V^o$ – перший елемент цієї підмножини. У цьому випадку збільшення значення $k_c(S_y)$ може бути досягнуто зменшенням різниці між v_y і v_{yh}^y . Для цього потрібно запропонувати h -ому експерту змінити ступінь переваги d_{hyp} домінуючого об'єкта o_{ph} над об'єктом o_y . Напрямок цієї зміни (збільшення або зменшення) визначається типом функції (3.11) і знаком різниці між v_y і v_{yh}^y . Після цього відповідно до нового значення v_{yh}^y для всіх спектрів обчислюються нові значення компонент, що мають верхній індекс ' y ', і описана вище процедура повторюється. Якщо h -й експерт відмовився змінити значення d_{hyp} , то компонента v_{yh}^y виключається з підмножини V^o . Якщо $V^o \neq \emptyset$, то описана операція виконується для другого елемента підмножини V^o і т.д., інакше починається перетворення підмножини V^t . Відзначимо, що якщо компоненти v_{ph}^k , v_{yh}^y аналізувалися на q -ому кроці алгоритму, то на $(q+u)$ -ому ($u \geq 1$) кроці вони не будуть аналізуватися, тому що до початку цього кроку вони будуть виключені з множини V .

Операція перетворення підмножини V^t аналогічна описаній операції перетворення підмножини V^o за тим тільки виключенням, що в цьому випадку h -ому експерту пропонується змінити ступінь переваги d_{hkz} об'єкта o_k над об'єктом o_z , якщо v_{zh}^k – перший елемент підмножини V^t . Якщо експерт відмовився змінити d_{hkz} , то значення v_{zh}^k виключається з підмножини V^t . Якщо $V^t \neq \emptyset$, то ця операція виконується для другого елемента підмножини V^t і т.д., інакше діалог з експертами припиняється. Якщо при цьому хоча б для одного спектра виконується умова (3.21), варто зробити

висновок про те, що результати парних порівнянь, виконаних цією бригадою експертів, не можуть бути використані, інакше величини $v_j (1 \leq j \leq t)$ приймаються в якості узгоджених агрегованих оцінок ненормалізованих значень ваг об'єктів. Нормалізовані значення цих ваг визначаються зі співвідношення (3.10). Після кожної корекції значень v_{ph}^k або v_{yh}^y або v_{zh}^k необхідно обчислити нові значення ваг всіх об'єктів і відкоригувати всі спектри S_k для всіх $1 \leq k \leq t$. Якщо умова (3.18) виконується хоча б для одного спектра, необхідно повторно виконати описаний вище алгоритм.

Відзначимо, що при використанні цього методу “тиск” на експертів не спричиняється, тому що в них зберігається свобода прийняття рішень на кожнім кроці діалогу.

Для доведення стабільності описаного вище алгоритму досить переконатися в тому, що він за кінцеве число кроків дозволяє з прийнятною точністю визначити множину ваг об'єктів або зробити висновок про неможливість цього. Оскільки в описаному процесі підвищення ступеня узгодженості оцінок неповних парних порівнянь ніяких додаткових оцінок не пропонується використовувати, а мова йде тільки про корекцію вже даних експертами оцінок, то умови стабільності алгоритму при неповних групових парних порівняннях збігаються з такими ж умовами для повних групових порівнянь, виконуваних одним експертом, які визначено у [44].

Для визначення необхідних і достатніх умов конструктивності множини індивідуальних МПП побудуємо об'єднану МПП, визначаючи її елементи в такий спосіб:

$$d_{ij} = \bigcup_h (d_{ijh}; c_{hij}),$$

де c_{hij} – коефіцієнт компетентності h -го експерта, що виконав порівняння об'єктів o_i і o_j ;

$d_{ijh} = *$, якщо h – експерт відмовився порівнювати об'єкти o_i і o_j .

Необхідні і достатні умови конструктивності щодо методу “квадрат” множини індивідуальних МПП визначає наступна теорема.

Теорема 3.5.1. Для того, щоб множина D первинних експертних оцінок множини O об'єктів була конструктивною щодо методу “квадрат”, необхідно і достатньо, щоб об'єднана МПП задовільняла умові:

$$\forall j : 1 \leq j \leq t [\exists i, h : (1 \leq i \leq t, 1 \leq h \leq m, i \neq j) : [d_{ijh} \neq *]]. \quad (3.22)$$

Доведення. *Необхідність.* Якщо умова (3.22) не виконується, тобто існує хоча б один стовпець з номером s об'єднаної МПП, що містить тільки (відмінні від d_{ss}) елементи зі значеннями $*$, то це означає, що жоден експерт не порівняв жоден об'єкт з об'єктом o_s . Як випливає з алгоритму, при цьому не може бути визначена жодна з компонент ваги об'єкта o_s , тобто множина первинних оцінок – неконструктивна, що і було потрібно довести.

Достатність. Якщо виконується умова (3.22), то кожен рядок $i, 1 \leq i \leq t$ об'єднаної МПП містить хоча б один елемент $d_{ijh} \neq *, (1 \leq h \leq m; i \neq j)$. Тому відповідно до алгоритму по кожнім рядку можуть бути визначені ненормовані компоненти ваг об'єктів o_i і o_j по оцінках, даних принаймні одним h -им експертом, що і доводить конструктивність множини D первинних оцінок.

Якщо кожний з експертів може виконувати порівняння будь-яких пар об'єктів, то описаний алгоритм можна використовувати для оптимізації процесу за критерієм часу отримання мінімально необхідної множини первинних оцінок шляхом розподілу об'єктів між експертами, або за критерієм вірогідності оцінок. Для мінімізації часових витрат множина O об'єктів, $|O| = t$, розбивається на m у загальному випадку пересічних підмножин O_1, O_2, \dots, O_m , $|O_j| = \lceil t/m \rceil$, причому оцінювання об'єктів підмножини O_x доручається x -ому експерту. Особливість алгоритму “квадрат” для неповних парних порівнянь полягає в тому, що для утворення об'єднаної МПП на основі окремих МПП, необхідно, щоб кожна така МПП містила t рядків і $\lceil t/m \rceil$ стовпців. Іншими словами, об'єкти кожної з підмножин O_x повинні порівнюватися з кожним об'єктом з множини O . Для мінімізації загального числа порівнянь, необхідного для формування МПП, що задовольняє умові (3.22), кожний з об'єктів множини O досить порівняти з одним об'єктом підмножини O_x . Таким чином, кожен рядок об'єднаної МПП буде містити t оцінок, а кількість операцій порівняння одночасно виконуваних t експертами дорівнює мінімально можливому числу t , тобто за t тактів виходить об'єднана МПП, що задовольняє умові конструктивності (3.22). Сказане ілюструється прикладом організації спільної роботи експертів при $t = 8, m = 3$, приведеним у табл. 3.1.

Таблиця 3.1.

	a	b	c	d	e	f	g	h
a		1		2		3		
b			1		2		3	
c	1					2		3
d		1		2		3		
e			1		2		3	
f	1					2		3
g		1		2			3	
h			1		2			3

У цьому прикладі $O_1 = \{a, b, c\}$; $O_2 = \{d, e, f\}$; $O_3 = \{f, g, h\}$. У клітинах таблиці записані номери експертів, що виконують порівняння відповідних пар об'єктів. Як випливає з описаного алгоритму, так і розглянутих вище алгоритмів, у загальному випадку ненормовані ваги об'єктів визначаються по різних множинах оцінок, даних експертами різної компетентності. Тому оптимізація вірогідності оцінок здійснюється так само, як і при використанні раніше розглянутих алгоритмів.

3.6. Висновки

В розділі запропоновано підхід до вирішення задачі визначення оцінок відносної важливості об'єктів при наступних припущеннях, а саме: коли ці оцінки визначаються на основі первинних оцінок, даних декількома експертами, кожний з яких виконує неповні парні порівняння, а при визначенні агрегованих групових оцінок об'єктів враховується відносна компетентність експертів в рамках групи. Цей підхід розповсюджено на метод безпосередньої оцінки, а також на три методи парних порівнянь, що відрізняються трудомісткістю та потенційною точністю.

Характерним для цих методів є принципова можливість обчислення оцінок по частині матриці парних порівнянь, а також спосіб формального визначення діалогів з експертами у ході покращення узгодженості їхніх оцінок. Методи розроблені з урахуванням двох видів неповноти первісної інформації. Неповнота першого виду породжена неможливістю для експерта виконати порівняння пар об'єктів по причині відсутності для цього необхідної інформації, другий вид неповноти породжується

прагненням отримати оцінки за мінімальну кількість кроків. Розглянута організація роботи експертів з метою отримання потрібної вірогідності оцінок за мінімальний час.

Розділ 4.

Експериментальне дослідження методів отримання кардинальних експертних оцінок

Наявність достатньо великого різноманіття методів отримання кардинальних експертних оцінок ставить перед ОПР, або перед адміністратором, що розробляє разом з експертами базу знань для СППР, питання про вибір методу визначення кардинальних оцінок. Неважко бачити, що при цьому варто враховувати різні характеристики методів, такі як похибки оцінювання, узгодженість результатів, час, необхідний для одержання оцінок, психологічну привабливість методів. При цьому ОПР може оцінити тільки відносну важливість для нього того чи іншого критерію, у той час як оцінки методів за названими критеріями можуть бути визначені тільки шляхом експерименту, який забезпечує статистичну спроможність оцінок.

Даний розділ присвячено експериментальному дослідженю методів отримання кардинальних експертних оцінок альтернатив. На початку обґруntовується вибір об'єктів оцінювання різними методами, далі викладається методика експериментального дослідження, описуються досліджувані індивідуальні методи без зворотного зв'язку з експертом та зі зворотним зв'язком з експертом, результати обробки експериментальних даних.

4.1. Вибір оцінюваних об'єктів

Для визначення перерахованих вище характеристик методів експертного оцінювання необхідно знати еталонні значення параметрів оцінюваних об'єктів. Стосовно до параметрів об'єктів, використовуваних при розв'язанні задач підтримки прийняття рішень (коєфіцієнти відносної значимості критеріїв, цілей, альтернатив), можна стверджувати, що такі еталонні значення в принципі невідомі. Тому виникає необхідність використовувати інформаційні моделі цих параметрів для визначення оцінок методів експертної оцінки.

Вимоги, що висуваються до вибору цих моделей, досить очевидні:

- в інформаційному відношенні ступінь невизначеності параметрів моделі повинна бути аналогічною ступеню невизначеності параметрів об'єктів, оцінюваних при підтримці прийняття рішень;
- необхідно мати можливість точно вимірювати параметри моделі;
- простота і загальне розуміння моделі, що дозволило б проводити експеримент з експертами різного ступеня компетентності.

Можливими типами об'єктів для застосування їх в якості моделей в методах отримання кардинальних експертних оцінок є, наприклад, різні типи геометричних фігур на площині з їх параметрами: відрізок (пропонується визначити його довжину), пряма (визначити кут її нахилу до горизонталі), плоскі фігури, однотипні чи різно-типні (визначити їхню площину, або приміром, ступінь заповнення якимось кольором). Можливі так само і менш строгі, але більш практичні приклади – визначити вік незнайомої людини по фотографії.

Виходячи з цих вимог та з простоти реалізації на ПК, у якості моделі оцінюваного об'єкта (цілі, критерію, альтернативи) був обраний прямокутник, заповнений випадково розміщеними зафарбованими точками, а моделлю оцінюваного параметра (відносна значимість цілі, критерію, альтернативи) послужив відносний ступінь заповнення випадково розміщеними пофарбованими точками прямокутника (рис. 4.1). Цей показник задається для кожного прямокутника програмно і тому відомий до-

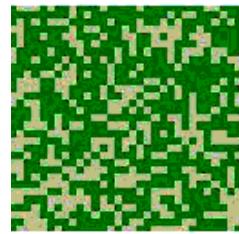


Рис. 4.1.

стовірно. Отже, ці дані можна використовувати в якості еталонних і, як наслідок, визначати оцінки досліджуваних методів експертного оцінювання за названими критеріями.

4.2. Досліджувані методи оцінювання без зворотного зв'язку з експертом

4.2.1. Метод безпосереднього оцінювання (#1)

Сутність методу безпосереднього оцінювання абсолютних параметрів (тривалість виконання проекту, його вартість, затримка поширення впливу досягнення цілі [29, 30, 54], ймовірність успішного завершення проекту [53], полягає в тому, що експерт називає свою оцінку абсолютноного значення цього параметра. При визначенні відносних коефіцієнтів значимості цілей, критеріїв, альтернатив експерт називає абсолютноні оцінки v_i , $i = (1, k)$ значимості, після чого обчислюються їхні нормовані значення:

$$w_i = v_i / \sum_{j=1}^k v_j$$

4.2.2. Метод “трикутник” мультиплікативних парних порівнянь з виразом дискретних значень переваг у фундаментальній шкалі, словесним введенням даних, попереднім упорядкуванням і послідовним методом обробки матриці порівнянь (#2)

Експерту пропонується спочатку виконати порівняння $k(k - 1)/2$ пар об'єктів у тернарній шкалі зі значеннями “більше”, “рівнозначно”, “менше”. Унаслідок цього формується ранжування об'єктів, де об'єкт a_h , що переважає над об'єктом a_r , знаходиться на i -ому місці, тобто має ранг i , а об'єкт a_r має ранг $(i + g), g \leq 1$; якщо при порівнянні об'єктів була дана оцінка “рівнозначно”, то їм присвоюються однакові ранги.

Опісля цьому експертові пропонується виконати порівняння кожного об'єкта з об'єктами, що мають більші ранги, і висловити результати порівняння d_{ih} кожної пари в так званій фундаментальній шкалі зі значеннями “слабка перевага”(3), “проста перевага”(4), “сильна перевага”(5), “домінуюча перевага”(7), “абсолютна перевага”(9). (У дужках проставлені числові еквіваленти ступенів переваг). Експерт вводить назву своєї оцінки ступеня переваги, вибираючи відповідне значення в меню.

4.2.2.1. Послідовний метод обробки

Домінуючому об'єкту a_d , тобто об'єкту, що має ранг $r_d = 1$, присвоюється ненормоване значення ваги $v_d = 1$. Ненормовані ваги інших об'єктів визначаються відповідно до такого алгоритму

- 1) $r := r_d; i := d;$
- 2) Для об'єктів a_h , що мають ранги $r + 1; r + 2, \dots, r_{max}$, обчислити компоненту v_h^i ненормованого значення ваги об'єкта a_h , обумовлену перевагою об'єкта a_i , згідно з виразом [45]:

$$v_h^i = f(v_i, d_{ih}) = v_i \frac{d_m - \varepsilon - d_{ih} (1 - \varepsilon)}{d_m - 1}; \quad (4.1)$$

де v_i – ненормоване значення ваги об'єкта a_i , $v_d = 1$;

d_{ih} – числовий еквівалент ступеня переваги об'єкта a_i над об'єктом a_h ;

d_m – числовий еквівалент найбільшого значення ступеня переваги (абсолютна перевага), $d_m = 9$;

ε – відносна похибка обчислення значень ваг об'єктів.

- 3) Обчислити агреговане ненормоване значення ваги v_h об'єкта a_h , що має ранг $r_h = (r + 1)$:

$$v_h = \frac{1}{n_h} \sum_{a_g \in A^h} v_h^g, \quad (4.2)$$

де A^h – множина об'єктів, що мають ранг менше, ніж r_h ;

$n_h = |A^h|$; v_h^g – компонента ненормованого значення ваги об'єкта a_h , що обумовлена перевагою об'єкта $a_g \in A^h$.

- 4) $r := r + 1$; якщо $r \leq r_{max} - 1$, то $i = i_r$; де i_r – індекс об'єкта, що має ранг r , пункт 2, інакше – кінець.

Після цього обчислюються відносні ваги об'єктів, тобто нормовані значення w_j ваг:

$$w_j = v_j / \sum_{h=1}^k v_h$$

де k – кількість об'єктів.

4.2.3. Метод “трикутник” мультиплікативних парних порівнянь з вираженням квазібезперервних значень переваг у фундаментальній шкалі, графічним введенням даних, попереднім упорядкуванням і послідовним методом обробки (#3)

Цей метод відрізняється від методу, розглянутого в підрозділі 4.2.2, лише тим, що для введення ступеня переваги використовується графічний метод, відповідно до якого числовий еквівалент ступеня переваги d_{ih} об'єкта a_i над об'єктом a_h дорівнює кількості пікселів у зафарбованому стовпчику від початку до курсору (рис. 4.2).

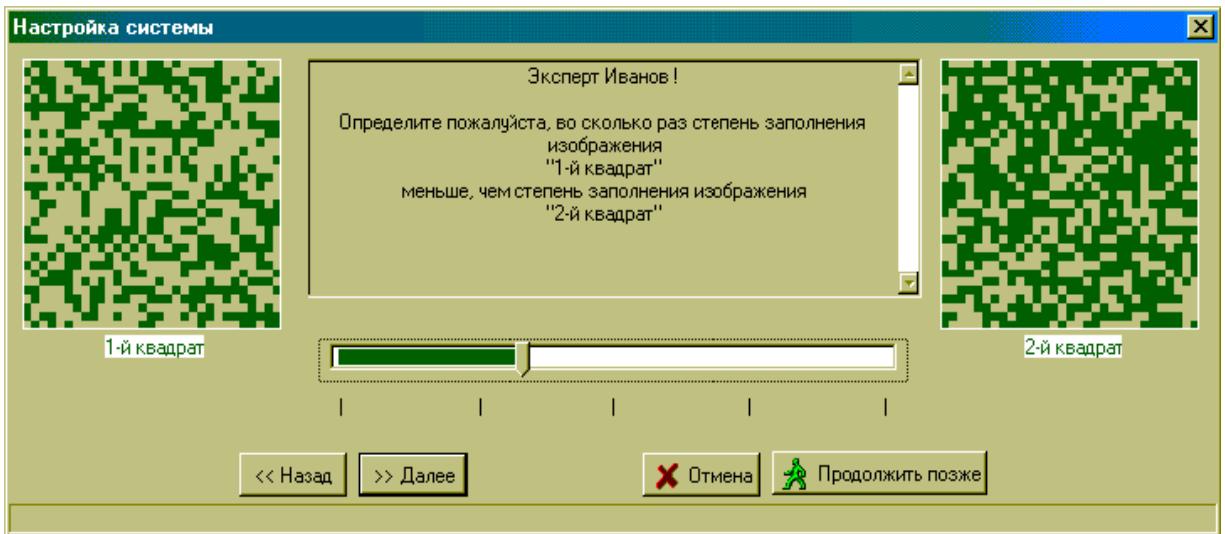


Рис. 4.2.

4.2.4. Метод “трикутник” мультиплікативних парних порівнянь з вираженням дискретних значень переваг у фундаментальній шкалі, словесним введенням даних і паралельним методом обробки (#4)

Цей метод парних порівнянь є базовим для запропонованих Сааті методів аналітичних ієрархічних процесів [26] і аналітичних мережніх процесів підтримки прийняття рішень [27]. Сутність його полягає в наступному. Експерту послідовно пред’являються пари об’єктів $\langle a_i, a_j \rangle$ і пропонується визначити ступінь d_{ij} переваги об’єкта a_i над об’єктом a_j щодо деякого якісного критерію K . При цьому, якщо експерту була представлена пара $\langle a_i, a_j \rangle$ і він визначив ступінь переваги d_{ij} , то пара $\langle a_j, a_i \rangle$ вже не пред’являється (саме тому цей метод називається “трикутник”), а ступінь переваги d_{ji} визначається, виходячи з метризованого мультиплікативного співвідношення:

$$d_{ji} = \begin{cases} d_{ij}, & \text{якщо } \langle a_i, a_j \rangle \in P \wedge \langle a_j, a_i \rangle \notin P \\ 1, & \text{якщо } \langle a_i, a_j \rangle \in P \wedge \langle a_j, a_i \rangle \in P \\ 1/d_{ij}, & \text{якщо } \langle a_i, a_j \rangle \notin P \wedge \langle a_j, a_i \rangle \in P \end{cases} \quad (4.3)$$

Таким чином, при наявності k об’єктів експерт має виконати $k(k - 1)/2$ порівнянь. Відмітимо, що співвідношення $d_{ji} = 1/d_{ij}$ є фундаментальним для методу Сааті

обчислення відносних ваг об'єктів. Ступені переваги виражаються у так званій фундаментальній шкалі, що має поділки, названі вище (п.4.2.2). Щі поділки відповідають числам, що показують, у скільки разів один об'єкт перевершує інший щодо загальної властивості або критерію. Елементи $d_{ij}, i, j = (1, k)$ утворюють зверхтранзитивну квадратну матрицю парних порівнянь D . При цьому елемент d_{ij} можна трактувати як відношення ваг об'єктів a_i і a_j , тобто w_i/w_j :

$$D = \begin{vmatrix} w_1/w_1 & w_1/w_2 & \cdots & w_1/w_k \\ w_2/w_1 & w_2/w_2 & \cdots & w_2/w_k \\ \vdots & & \ddots & \vdots \\ w_k/w_1 & w_k/w_2 & \cdots & w_k/w_k \end{vmatrix}$$

Запропоновано декілька методів обчислення відносних ваг об'єктів, виходячи з матриці D . Найбільш математично обґрунтованим є метод власного вектора, запропонований в [26].

4.2.5. Метод “трикутник” мультиплікативних парних порівнянь з вираженням квазібезперервних значень переваг у фундаментальній шкалі, графічним введенням даних і паралельним методом обробки (#5)

Цей метод відрізняється від методу, розглянутого в підрозділі 4.2.4, лише тим, що для введення ступеня переваги використовується графічний метод, описаний у підрозділі 4.2.3.

4.2.6. Метод “квадрат” адитивних парних порівнянь з вираженням квазібезперервних значень переваг у довільній шкалі, числовим введенням даних і послідовним методом обробки по рядках матриці порівнянь (#6)

При використанні цього методу експерт відповідає на запитання ‘На скільки об’єкт a_i перевершує об’єкт a_j щодо критерію K ?’ Відповідь на нього визначає значення d_{ij} матриці порівнянь розмірністю $k \times k$, де k – кількість об’єктів, d_{ij} визначається з виразу:

$$d_{ij} = v_i - v_j, \quad (4.4)$$

де v_i, v_j – ненормовані значення ваг об’єктів a_i, a_j відповідно.

Величина d_{ij} знаходиться в межах $(-1, 1)$. Якщо експерт визначив деякі значення $|d_{ij}| > 1$, то елементи матриці порівнянь приводяться до меж $(-1, 1)$ шляхом ділення всіх d_{ij} на $|d_{ij}|_{max}$.

Послідовний метод обробки матриці порівнянь по рядках полягає в наступному. Для рядка, позначеного об’єктом a_i , визначається мінімальний елемент d_{ib} . З (4.4) випливає, що об’єкт a_b , яким позначений b -й стовпець, має найбільшу вагу. Тому об’єкт a_b називають домінуючим по i -ому рядку; йому присвоюється вага $v_b^i = 1$ (верхній індекс означає, що ця компонента ваги об’єкта a_b обчислена по i -ому рядку). Далі визначається власна вага v_i^i об’єкта a_i :

$$v_i^i = v_b^i + d_{ib}, \quad (4.5)$$

і транзитивні компоненти v_h^i ваги інших об’єктів a_h , $h \neq i; h \neq b$:

$$v_h^i = v_i^i - d_{ih}. \quad (4.6)$$

Експерту пропонується методом безпосередньої оцінки визначити ненормоване абсолютне значення V_b домінуючого по i -му рядку об’єкта. На підставі цього визначаються ненормовані абсолютноні значення ваг об’єктів:

$$V_i^i = v_i^i V_b;$$

$$V_h^i = v_h^i V_b.$$

Зазначені операції виконуються для кожного рядка, унаслідок чого об'єкт a_i одержує k ненормованих абсолютних значень ваги. Далі визначають середні арифметичні абсолютні значення ваг об'єктів:

$$V_h^* = \frac{1}{k} \sum_{i=1}^k V_h$$

На останньому етапі визначають показники w_i відносної значимості об'єктів щодо обраного критерію:

$$\forall i \left[w_i = V_h^* \Bigg/ \sum_{i=1}^k V_h \right]$$

4.2.7. Метод “квадрат” адитивних парних порівнянь з вираженням квазібезперервних значень переваг у довільній шкалі, числовим введенням даних і послідовним методом обробки по стовпцях рядках матриці порівнянь (#7)

Цей метод обробки матриці порівнянь легко виводиться з описаного в попередньому розділі методу #6 з урахуванням таких особливостей:

- 1) Домінуючому по i -ому стовпцю об'єктові a_d відповідає найбільший у i -ому стовпці елемент d_{hi} , $h = (1, k)$. Цьому об'єкту присвоюється вага $v_d^i = 1$;
- 2) Власна компонента ваги v_i^i визначається виразом

$$v_i^i = v_d^i - d_{di};$$

- 3) Транзитивні компоненти ваг об'єктів обчислюються в такий спосіб:

$$v_h^i = v_i^i + d_{hi}.$$

4.2.8. Метод “квадрат” мультиплікативних парних порівнянь з вираженням квазібезперервних значень переваг у довільній шкалі, числовим введенням даних і послідовним методом обробки по рядках матриці порівнянь (#8)

Ідея методу аналогічна розглянутому в розділі 4.2.6 методу #6. Відмінності полягають у наступному:

- 1) Елементи матриці порівнянь визначаються виразом

$$d_{ij} = v_i/v_j;$$

- 2) З $d_{ih} = \inf_j d_{ij}$ випливає, що об'єкт a_h є домінуючим по i -му рядку і йому повинно бути привласнено значення ваги $v_d^i = v_h^i = 1$;

- 3) Власні і транзитивні компоненти ваг об'єктів визначаються відповідно виразами :

$$v_i^i = v_h^i d_{ih},$$

$$v_r^i = v_i^i / d_{ir};$$

- 4) Визначати абсолютні значення ваг об'єктів немає необхідності.

4.2.9. Метод “квадрат” мультиплікативних парних порівнянь з вираженням квазібезперервних значень переваг у довільній шкалі, числовим введенням даних і послідовним методом обробки по стовпцях матриці порівнянь (#9)

Ідея методу аналогічна розглянутому в розділі 4.2.7 методу #7. Відмінності полягають у наступному:

- 1) Елементи матриці порівнянь визначаються виразом

$$d_{ij} = v_i/v_j;$$

- 2) З $d_{ih} = \sup_i d_{ij}$ випливає, що об'єкт a_i є домінуючим по h -ому стовпцю і йому варто присвоїти значення ваги $v_d^h = v_i^h = 1$;
- 3) Власні і транзитивні компоненти ваг об'єктів визначаються відповідно виразами:

$$v_h^h = v_i^h / d_{ih},$$

$$v_r^h = v_h^h d_{rh};$$

- 4) Визначати абсолютні значення ваг об'єктів немає необхідності.

4.2.10. Метод “квадрат” мультиплікативних парних порівнянь з вираженням дискретних значень переваг у фундаментальній шкалі, словесним введенням даних і послідовним методом обробки по рядках матриці порівнянь (#10)

Цей метод аналогічний методу #8, розглянутому в розділі 4.2.8. Відмінності полягають у наступному:

- 1) Експерту пред'являють пару $\langle a_i, a_j \rangle$ об'єктів і пропонують визначити, який з них превалює над іншим. Ступені d_{ij} переваг об'єктів можуть приймати одне з можливих значень, названих вище (п.4.2.2). У випадку, якщо a_i має перевагу над a_j , у матрицю заноситься значення d_{ij} , рівне числовому еквіваленту визначеного експертом ступеня переваги. Якщо a_i поступається a_j , d_{ij} дорівнює зворотній величині.
- 2) Власні і транзитивні компоненти ваг об'єктів визначаються відповідно виразами [45]:

$$v_i^i = v_{pj}^i \frac{d_m - 1}{d_m - \varepsilon - d_{ib}(1 - \varepsilon)},$$

$$v_h^i = v_i^i \frac{d_m - \varepsilon - d_{ih}(1 - \varepsilon)}{d_m - 1}.$$

де d_{ib} – числовий еквівалент ступеня переваги об'єкта a_i над домінуючим по i -ому рядку об'єктом a_b ;

d_m – числовий еквівалент найбільшого значення ступеня переваги (абсолютна перевага), $d_m = 9$;

ε – відносна похибка обчислення значень ваг об'єктів.

4.2.11. Метод “квадрат” мультиплікативних парних порівнянь з вираженням дискретних значень переваг у фундаментальній шкалі, словесним введенням даних і послідовним методом обробки по стовпцях матриці порівнянь (#11)

Метод аналогічний розглянутому в розділі 4.2.9, за тим тільки виключенням, що замість безперервних значень ступенів переваг, що вводяться числовим способом, використовуються дискретні значення (1,3,4,5,7,9) числових аналогів ступенів переваги у фундаментальній шкалі (рівнозначно, слабка перевага, проста перевага, сильна перевага, домінуюча перевага, абсолютна перевага), що вводяться словесно.

4.3. Досліджувані методи індивідуального експертного оцінювання зі зворотним зв’язком з експертом

У ході проведених експериментів були досліджені наступні методи зі зворотним зв’язком з експертом:

- 1) Метод “трикутник” мультиплікативних парних порівнянь із представленням дискретних значень переваг у фундаментальній шкалі, словесним введенням даних, попереднім упорядкуванням і послідовним методом обробки (#12);

- 2) Метод “трикутник” мультиплікативних парних порівнянь із представленням квазібезперервних значень переваг у фундаментальній шкалі, графічним введенням даних, попереднім упорядкуванням і послідовним методом обробки (#13);
- 3) Метод “квадрат” адитивних парних порівнянь із представленням квазібезперервних значень переваг у довільній шкалі, числовим введенням даних і послідовним методом обробки по рядках матриці порівнянь (#14);
- 4) Метод “квадрат” адитивних парних порівнянь із представленням квазібезперервних значень переваг у довільній шкалі, числовим введенням даних і послідовним методом обробки по стовпцях матриці порівнянь (#15);
- 5) Метод “квадрат” мультиплікативних парних порівнянь із представленням квазібезперервних значень переваг у довільній шкалі, числовим введенням даних і послідовним методом обробки по рядках матриці порівнянь (#16);
- 6) Метод “квадрат” мультиплікативних парних порівнянь з вираженням квазібезперервних значень переваг у довільній шкалі, числовим введенням даних і послідовним методом обробки по стовпцях матриці порівнянь (#17);
- 7) Метод “квадрат” мультиплікативних парних порівнянь із представленням дискретних значень переваг у фундаментальній шкалі, словесним введенням даних і послідовним методом обробки по рядках матриці порівнянь (#18);
- 8) Метод “квадрат” мультиплікативних парних порівнянь із представленням дискретних значень переваг у фундаментальній шкалі, словесним введенням даних і послідовним методом обробки по стовпцях матриці порівнянь (#19);
- 9) Метод “трикутник” адитивних парних порівнянь із представленням квазібезперервних значень переваг у довільній шкалі, числовим введенням даних і комбінаторним методом обробки матриці порівнянь (#20);
- 10) Метод “трикутник” мультиплікативних парних порівнянь із представленням квазібезперервних значень переваг у довільній шкалі, числовим введенням даних і комбінаторним методом обробки матриці порівнянь (#21);

- 11) Метод “трикутник” мультиплікативних парних порівнянь із представленням квазібезперервних значень переваг у фундаментальній шкалі, графічним введенням даних і комбінаторним методом обробки матриці порівнянь (#22)

Зважаючи на те, що перераховані методи, що використовують зворотний зв’язок з експертом, характеризуються:

- a)** алгоритмами отримання експертної інформації і змістом спілкування з експертом (використання зворотного зв’язку);
- б)** типом парних порівнянь (мультиплікативні, адитивні);
- в)** типом шкали, використовуваної для представлення експертних оцінок (фундаментальна – шкала відношень, довільна – шкала відношень, шкала інтервалів);
- г)** способом введення експертної інформації (словесний, графічний, числовий);
- д)** способом обробки експертної інформації (послідовний, комбінаторний);

а також те, що сутності ознак б), в) і г), а також послідовного методу обробки матриці порівнянь – описані в розділі 4.2, а комбінаторний метод обробки матриці парних порівнянь описаний в підрозділі 2.1, обмежимося викладом сутності алгоритмів отримання інформації при реалізації методів “трикутник” і “квадрат” зі зворотним зв’язком з експертом.

4.3.1. Метод парних порівнянь “трикутник” зі зворотним зв’язком з експертом

Метод для довільної функції залежності ваг об’єктів від ступенів переваг запропонований у [44]. У [45] метод конкретизований для практично важливих випадків.

Сутність методу “трикутник” полягає в наступному. Виберемо деякий об’єкт a_e і виконаємо алгоритм “лінія” [44] щодо множини об’єктів A . Відповідно до цього алгоритму запропонуємо експерту зробити попарні порівняння a_e з кожним із тих, що залишилися $(k - 1)$ об’єктів щодо заданого критерію C . Нагадаємо, що в експерименті як такий критерій використовується ступінь заповнення квадрата випадково

розташованими точками. У результаті виконання порівнянь буде отримана множина $D_e = \{d_{ei}\}$, $i \neq e$ ступенів переваги об'єкта a_e над іншими об'єктами у відношенні щодо критерію C . У загальному випадку об'єкт a_e перевершує щодо цього критерію деякі об'єкти й поступається іншим. Уведемо поняття абсолютної ваги v_i об'єкта a_i , під яким будемо розуміти кількісну міру ступеня вираженості у об'єкта a_i властивості, обумовленої критерієм C . Присвоїмо об'єкту a_e вагу v_e .

Задамо функцію

$$v_i = f(v_e, d_{ei}) \quad (4.7)$$

і обчислимо ваги всіх інших об'єктів.

Очевидною вимогою до функції (4.7) є її монотонність. Вид цієї функції залежить від типу питань, що ставляться перед експертом у ході парних порівнянь. Якщо експерту ставиться питання: ‘У скільки разів об'єкт a_e перевершує об'єкт a_i у відношенні щодо критерію C ?’ (мультиплікативні порівняння), то

$$v_i = v_e \cdot \varphi(d_{ei}) \quad (4.8)$$

де $\varphi(d_{ei})$ – довільна монотонна функція, що задовольняє умові

$$\varphi(1) = 1.$$

З (4.7), (4.8) випливає, що при розглянутому мультиплікативному алгоритмі парних порівнянь коефіцієнти відносної значимості об'єктів не залежать від величини абсолютної ваги ‘еталонного’ об'єкта a_e , з яким порівнюються всі інші.

При використанні адитивних парних порівнянь експерту пропонують визначити ‘На скільки об'єкт a_e перевершує об'єкт a_i у відношенні щодо критерію C ?’. При цьому

$$v_i = v_e + \lambda(d_{ei})$$

де $\lambda(d_{ei})$ – довільна монотонна функція, що задовольняє умові

$$\lambda(0) = 0.$$

Очевидно з використанням об'єкта a_e одержати додаткову інформацію вже неможливо, тому виключимо його з множини і для підмножини з $(k - 1)$ об'єктів, що

залишилися, виконаємо ту ж процедуру, тобто виберемо деякий $a_h = a_{e(k-1)}$ і виконаємо алгоритм “лінія” щодо інших ($k-2$) об’єктів, внаслідок чого одержимо множину $D_h = s \{d_{hi}\}$ ступенів переваг об’єкта $a_{e(k-1)}$ над іншими. Є надія, що залучення додаткової інформації, отриманої таким чином, підвищить точність визначення відносних ваг об’єктів.

Після цього виключимо об’єкт $a_{e(k-1)}$ з множини об’єктів, що залишилися, і повторимо описані процедури. Ці дії варто виконувати доти, поки в скороченій в такий спосіб множині не лишиться один єдиний об’єкт. Особливість застосування алгоритму “лінія” у складі алгоритму “трикутник” полягає в тому, що абсолютна вага $v_{e(k-1)}$ об’єкта $a_{e(k-1)}$ вибирається не довільно, а відповідно до виразу

$$v_{e(k-1)} = f(v_e, d_{e,e(k-1)}) = v_{e(k-1)}^e. \quad (4.9)$$

Через $v_h^{e(t)}$ будемо надалі позначати абсолютну вагу об’єкта a_h , обчислену згідно (4.9) за умови, що для підмножини $A(t)$, що містить t об’єктів, у якості ‘еталона’ обраний об’єкт $a_{e(t)}$.

Наприклад:

$$\begin{aligned} a_{e(4)} &= a_3; a_{e(3)} = a_1; a_{(2)} = a_2; \\ v_1^{e(4)} &= v_1^3 = f(v_3; d_{31}); v_2^{e(4)} = v_2^3 = f(v_3; d_{23}); v_4^{e(4)} = v_4^3 = f(v_3; d_{34}); \\ v_2^{e(3)} &= v_2^1 = f(v_1; d_{12}); v_4^{e(3)} = v_4^1 = f(v_1; d_{14}); v_4^{e(2)} = v_4^2 = f(v_2; d_{24}). \end{aligned} \quad (4.10)$$

З (4.10) випливає, що абсолютна вага $v_{e(t)}$ об’єкта a_3 , обраного як еталон для підмножини $A(k) = A$, визначається однозначно. Те ж саме можна сказати і про абсолютну вагу $v_{(k-1)}$ об’єкта $a_1 = a_{e(k-1)}$, обраного в якості еталона на другому кроці, тобто для підмножини $A(k-1)$. Однак вага об’єкта a_2 , що є еталонним на третьому кроці, приймає два значення: $v_2^{e(4)} = v_2^3$ і $v_2^{e(3)} = v_2^1$. У загальному випадку вони не збігаються, що свідчить про наявність внутрішніх протиріч оцінок ступенів переваг об’єктів, даних експертом.

Відзначимо, що протиріччя оцінок експерта можуть бути викликані як природними протиріччями властивостей об’єкта, так і впливом на експерта різних психофізіологічних факторів. Другий тип протиріч викликається, наприклад, утомою, неуважністю

експерта, впливом на нього зовнішніх подразників і т.д. Зрозуміло, що для підвищення вірогідності результатів парних порівнянь необхідно вказати експерту джерело виниклих протиріч і надати йому можливість відкоригувати результати, якщо на його думку протиріччя викликані факторами, що заважають. Для цього необхідно використовувати додаткову інформацію, яку можна одержати тільки від експерта.

Наслідуючи підхід, запропонований в [80], для об'єкта $a_{e(k-2)}$ побудуємо спектр $R_{e(k-2)}$ значень її абсолютної ваги і використаємо запропонований у [52] метод за-безпечення достатньої зовнішньої узгодженості оцінок експертів, розглядаючи різні значення абсолютної ваги об'єкта $a_{e(k-2)}$ як оцінки, дані різними експертами з рівною компетентністю. Для цього спочатку обчислимо коефіцієнт узгодженості $k_c(R_{e(k-2)})$ спектра $R_{e(k-2)}$, прийнявши кількість експертів рівним 2.

Якщо $k_c(R_{e(k-2)}) < T_o$, де T_o – поріг виявлення, то спектр $R_{e(k-2)}$ не несе інформації, а являє собою ‘інформаційний шум’. По такому спектру не можна обчислити достовірної узгодженої оцінки ваги $v_{e(k-2)}$. З (4.7) випливає, що, оскілки значення $v_{e(k)}, v_{e(k-1)}$ компонент спектра $R_{e(k-2)}$ визначаються однозначно, то зміни компонент $v_{e(k-2)}^{e(k)}, v_{e(k-2)}^{e(k-1)}$ можна домогтися тільки шляхом зміни ступенів переваги $d_{e(k)e(k-2)}$ і/або $d_{e(k-1)e(k-2)}$ таким чином, щоб відхилення $v_{e(k-2)}^{e(k)}, v_{e(k-2)}^{e(k-1)}$ зменшилося. Напрямок зміни (збільшення/зменшення) $d_{e(k)e(k-2)}$ і/або $d_{e(k-1)e(k-2)}$ визначається видом функції (4.7). Тому задамо експерту питання: ‘Чи згодні Ви збільшити (зменшити) ступінь переваги об'єкта $a_{e(k)}$ над об'єктом $a_{e(k-2)}$?’ Якщо експерт згоден, то вибираємо найближче більше(менше) можливе значення ступеня переваги $d_{e(k)e(k-2)}$, обчислюємо нове значення $v_{e(k-2)}^{e(k)}$, вносимо відповідну корективу в спектр $R_{e(k-2)}$, обчислюємо нове значення його коефіцієнта узгодженості і повторюємо описаний вище аналіз. Якщо експерт не згоден змінити ступінь переваги об'єкта $a_{e(k)}$ над об'єктом $a_{e(k-2)}$, йому пропонується розглянути питання про можливість зміни ступеня переваги $a_{e(k-1)}$ перед об'єктом $a_{e(k-2)}$. Якщо ж експерт відмовляється переглянути ступінь переваги і цих об'єктів, йому повідомляється про неможливість уточнення коефіцієнтів відносної значимості об'єкта в порівнянні зі значеннями, отриманими методом “лінії”, і останні приймаються в якості шуканих.

Якщо $T_o \leq k_c(R_{e(k-2)}) < T_u$, то спектр $R_{e(k-2)}$ містить інформацію, однак достовірність значення узагальненої оцінки, визначеного по ньому, нижче припустимої. Тому необхідно аналогічно тому, як описано вище, виконати процедуру зміни ступенів переваги $d_{e(k)e(k-2)}$ і/або $d_{e(k-1)e(k-2)}$. Якщо в результаті виконується умова $k_c(R_{e(k-2)}) \geq T_u$, то, використовуючи припустиму для використованого типу шкали операцію [19], знаходять середнє значення компонент спектра $R_{e(k-2)}$, що і приймається в якості узагальненої узгодженої оцінки ваги $v_{e(k-2)}$ об'єкта $a_{e(k-2)}$. Підставляючи його у вираз (4.7), одержуємо:

$$v_i^{e(k-2)} = f(v_{e(k-2)}, d_{e(k-2),i}).$$

У загальному випадку

$$v_i^{e(k-t)} = f(v_{e(k-t)}, d_{e(k-t),i}).$$

При $t > 2$ процес визначення узагальнених узгоджених оцінок абсолютних ваг об'єктів має особливість, яка полягає в тому, що експерту пропонують змінити компоненту спектра, модуль відстані якої від середнього значення – найбільший.

4.3.2. Метод парних порівнянь “квадрат” зі зворотним зв’язком з експертом

Метод, так само як і попередній, використовується для визначення значень відносних ваг об'єктів a_1, a_2, \dots, a_k . Запропонуємо експерту виконати k разів алгоритм “лінія”, причому при кожному виконанні його в якості еталонного будемо вибирати новий об'єкт. Змінimo і спосіб обчислення абсолютнох ваг об'єктів при кожному виконанні алгоритму “лінія”. При виконанні його в i -й раз ($i = 1, k$) визначимо найменший елемент d_{ih} сформованого i -го рядка матриці парних порівнянь (МПП). Якщо без втрати загальності припустити, що $\partial f(v_i; d_{ih}) / \partial(d_{ih}) < 0$, то об'єкт a_h , якому відповідає h -й стовпець МПП ($1 \leq h \leq k$), має найбільшу вагу v_h^i . Назовемо цей об'єкт домінуючим по i -ому рядку і позначимо його через a_{ph} , а його вагу – через v_{ph}^i . Далі, якщо експерту задається при парному порівнянні питання: ”На скільки об'єкт

a_i перевершує об'єкт a_h ?” (адитивне порівняння), запропонуємо йому визначити вагу v_{ph}^i об'єкта a_h методом безпосереднього оцінювання. Якщо ж використовується мультиплікативне порівняння, тобто експерту задається питання ”У скільки ...?”, то вважають $v_{ph}^i = 1$.

Використовуючи вираз:

$$v_i^i = f(v_{ph}^i, d_{ih}), \quad (4.11)$$

визначимо по елементах i -го рядка ненормалізоване значення ваги i -го об'єкта. Будемо називати v_i^i власною вагою об'єкта a_i .

Із виразу:

$$v_h^i = f(v_i^i, d_{ih}), \quad (4.12)$$

визначимо ваги v_h^i всіх інших об'єктів. Ці значення надалі будемо називати транзитивними.

Повторюючи цей процес для всіх рядків МПП, для кожного об'єкта a_h одержимо множину $V_h = \{v_h^i\}$, $i = (1, k)$ абсолютнох значень ваги. Аналогічно описаному в попередньому розділі побудуємо для кожного об'єкта a_h спектр оцінок ваги v_h . Особливість цих спектрів полягає в тому, що кожний з них містить k компонент. Наявність у спектрі оцінок ваги об'єкта декількох компонент, що знаходяться на різних поділках шкали, свідчить про наявність протиріч у результатах парних порівнянь. Тому в загальному випадку далі виконується процес забезпечення достатнього рівня внутрішньої узгодженості. Відмінність його від описаного вище обумовлюється відмінністю використовуваних способів обчислення ваг. В алгоритмі “трикутник” вага об'єкта з множини $A(k-t)$ обчислюється після того, як знайдені узгоджені узагальнені значення ваг об'єктів $a_{e(k)}, a_{e(k-1)}, \dots, a_{e(k-t-1)}$. У той самий час в алгоритмі “квадрат” ці ваги обчислюються щодо неузгоджених власних ваг об'єктів, з якими порівнюються всі інші (іншими словами, з об'єктами, що позначають рядки МПП). Процес забезпечення необхідного рівня внутрішньої узгодженості для цього алгоритму полягає в наступному.

Якщо

$$\exists h : 1 \leq h \leq k [k_c(R_h) < T_o], \quad (4.13)$$

то спектр R_h не несе корисної інформації, тобто являє собою ‘інформаційний шум’. У цьому випадку експерту пропонується повторити парні порівняння всіх об’єктів з об’єктом a_h , тобто визначити нові значення всіх d_{ih} , $i = (1, k)$, $i \neq h$, що утворять стовпець матриці порівнянь, позначений об’єктом a_h . Потім необхідно повторювати процес, описаний вище, доти, доки буде виконуватися умова (4.13).

Допустимо, що $\forall h : 1 \leq h \leq k [k_c(R_h) \geq T_o]$, але снує підмножина $R^* \subset R$, $R^* \neq \emptyset$ спектрів, для якої з $R_h \in R^*$ випливає, що

$$k_c(R_h) < T_u. \quad (4.14)$$

У цьому випадку всі спектри R_h містять корисну інформацію, але точність визначення ваг об’єктів, обчислених по спектрах з підмножини R^* – недостатня. Використовуючи припустимі для розглянутої шкали операції [19], обчислимо середнє значення v_h ваг об’єкта a_h для спектра $R_h \in R^*$ і визначимо множину D_h значень ваг v_h^i , найбільш віддалених від v_h . Цю процедуру необхідно виконати для всіх $R_h \in R^*$.

Розділимо множину $D = \bigcup_{h=1}^k D_h$ на три підмножини D^p , D^o , D^t . Значення v_{ph}^i ваг домінуючих об’єктів включаються в підмножину D^p ; D^o – це підмножина власних ваг об’єктів. Компоненти спектра, що є транзитивними вагами, включаються в підмножину D^t . Упорядкуємо підмножини D^p , D^o , D^t відповідно до зменшення їхніх відмінностей від середніх значень ваг.

У [52] показано, що збільшити коефіцієнт узгодженості будь-якого спектра можна за рахунок зменшення відмінностей його компонент від середнього значення. Для зміни значень v_h^i або v_i^i експерт повинен змінити величини d_{ih} або d_{ip} відповідно.

Процес підвищення коефіцієнта узгодженості починається з елементів підмножини D^p , тому що зміна v_{ph}^i приводить до змін значень v_i^i і v_h^i . Визначимо спектр R_h , якому відповідає перший елемент v_{ph}^i підмножини D^p . Тому що в цьому випадку підвищення значення $k(R_h)$ може бути досягнуто зменшенням різниці між v_h і v_{ph}^i , запропонуємо експерту змінити оцінку ваги домінуючого об’єкта a_h . Напрямок пропонованої зміни визначається знаком різниці між v_h і v_{ph}^i . Після цього відповідно до нового значення v_{ph}^i для всіх спектрів обчислюються нові значення компонент, що ма-

ють верхній індекс ‘ i ’. Для цього використовуються описані вище процедури. Якщо експерт відмовився змінити значення v_{ph}^i , то значення v_{ph}^i виключається з підмножини D^p і, якщо $D^p \neq \emptyset$, то ця операція виконується для другого елемента підмножини D^p і т.д.; у протилежному випадку переходят до перетворення підмножини D^o .

Нехай $v_y^y \in D^o$ – перший елемент цієї підмножини. У цьому випадку збільшення значення $k(R_y)$ може бути досягнуто зменшенням різниці між v_y і v_y^y . Для цього варто запропонувати експерту змінити ступінь переваги d_{yp} домінуючого об’єкта a_p над об’єктом a_y . Напрямок цієї зміни (збільшення або зменшення) визначається типом функції (4.7) і знаком різниці між v_y і v_y^y . Після цього відповідно до нового значення v_y^y для всіх спектрів обчислюються нові значення компонент, що мають верхній індекс y , і описана вище процедура повторюється. Якщо експерт відмовився змінити значення d_{yp} , то компонента v_y^y виключається з підмножини D^o . Якщо $D^o \neq \emptyset$, то описана операція виконується для другого елемента підмножини D^o і т.д., інакше починається перетворення підмножини D^t . Відзначимо, що якщо компоненти v_{ph}^i , v_y^y аналізувалися на q -ому кроці алгоритму, то на $(q + u)$ -ому ($u \geq 1$) кроці вони не будуть аналізуватися, тому що до початку цього кроку вони будуть виключені з множини D .

Операція перетворення підмножини D^t аналогічна описаній операції перетворення підмножини D^o за тим тільки виключенням, що в цьому випадку експерту пропонується змінити ступінь переваги d_{iz} об’єкта a_i над об’єктом a_z , якщо v_z^i – перший елемент підмножини D^t . Якщо експерт відмовився змінити d_{iz} , то значення v_z^i виключається з підмножини D^t . Якщо $D^t \neq \emptyset$, то ця операція виконується для другого елемента підмножини D^t і т.д., інакше діалог з експертом припиняється. Якщо при цьому хоча б для одного спектра виконується умова (4.13), потрібно зробити висновок про те, що результати парних порівнянь, виконаних цим експертом, не можуть бути використані, інакше величини v_h ($1 \leq h \leq k$) приймаються як ненормалізовані значення ваг об’єктів. Нормалізовані значення цих ваг визначаються зі співвідношення:

$$w_h = v_h \Big/ \sum_{q=1}^k v_q .$$

Для організації збору статистичних даних, що характеризують методи описані в п.4.2, 4.3 було розроблено програмну систему, в якій моделюється процес експертного оцінювання об'єктів за допомогою вищезгаданих методів. Програмну реалізацію основних модулів системи на мові програмування Pascal 7 представлена в додатку В.

4.4. Алгоритми обробки даних експериментального дослідження методів експертного оцінювання

Головною метою експериментального дослідження є одержання даних, що дозволяють вибрати метод відповідно до оцінок ОПР вагомості для ней критеріїв оцінки методів і експериментально визначених їхніх кардинальних оцінок за цими критеріями. Виходячи з цього, в якості критеріїв оцінки методів були обрані:

- коефіцієнт узгодженості множини експертних оцінок відносних ваг об'єктів;
- математичне очікування відносної похибки визначення відносних ваг об'єктів;
- математичне очікування тривалості процесу отримання експертних оцінок відносних ваг об'єктів.

Нижче викладаються алгоритми обробки експериментальних даних для отримання кардинальних оцінок методів за цими критеріями.

4.4.1. Визначення спектрального коефіцієнта узгодженості експертних оцінок відносних ваг об'єктів, що забезпечується досліджуваним методом експертного оцінювання

4.4.1.1. Визначення спектрального коефіцієнта зовнішньої узгодженості множини експертних оцінок відносної ваги об'єкта

В ході експерименту одну й ту саму множину об'єктів оцінюють m експертів. Внаслідок цього для кожного з об'єктів формується множина, що містить m оцінок від-

носної ваги. В ідеальному випадку всі ці оцінки збігаються. Реально вони різні, що є проявом зовнішньої неузгодженості (далі будемо говорити про узгодженість оцінок) результатів експертного оцінювання. Кількісною мірою ступеня зовнішньої узгодженості є коефіцієнт узгодженості [52,80]. Алгоритм обробки множин експертних оцінок відносних ваг об'єктів не залежить від методу, за допомогою якого була отримана ця множина, тому будемо викладати алгоритм стосовно до одного методу. Сутність алгоритму полягає в наступному:

- 1) Для кожного i -го об'єкта, $i = (1, 7)$ формується множина $V_i = \{v_{ij}\}, j = (1, m)$, m – кількість експертів, ненормованих оцінок ваги (ступеня заповнення прямокутника точками). Оцінки v_{ij} нормуються:

$$v_{ij} := v_{ij}/v_{ij \max},$$

після чого кожна з них округляється до найближчої поділки рівномірної шкали. Шкала має n поділок, тобто $(n - 1)$ інтервалів, перше зліва має оцінку 0.1, а останнє – оцінку 1.0.

- 2) Множина V_i задається спектром, що являє собою n -компонентний вектор $R_i = \{r_{ih}\}, h = (1, n)$, де r_{ih} – кількість експертів, що вказали, як оцінку h -у позначку шкали. Нехай для прикладу маємо: кількість експертів $m = 6$; кількість позначок шкали $n = 11$; $V_i = \{0.2; 0.2; 0.3; 0.4; 0.4; 0.7\}$. Тоді $R_i = \{0; 2; 1; 2; 0; 0; 2; 0; 0; 0; 0\}$.
- 3) Обчислюються коефіцієнти узгодженості кожної з множин експертних оцінок відносних ваг об'єктів. Цей коефіцієнт для множини ваг i -го об'єкта, описаного спектром R_i , визначається виразом [52,80]:

$$k(R) = \left(1 - \frac{\frac{1}{m} \sum_{i=1}^n r_i |i - a| - \sum_{i=1}^n r_i/m \ln(r_i/m)}{G \sum_{i=1}^n \left| i - \frac{n+1}{2} \right| + \ln(n)} \right) \cdot z, \quad (4.15)$$

де

$$z = \begin{cases} 1, & \text{якщо } z^* = \text{"истина"} \\ 0, & \text{якщо } z^* = \text{"хиба"} \end{cases}$$

$$z^* = \overline{(r_1 = r_n = m/s)} \bigvee_{d=1}^{s-1} \overline{(r_d = r_{d+1})} \bigvee_{d=1}^{s-1} \overline{(i_{d+1} - i_d)} = \text{const}$$

$$G = \frac{m}{\ln(m)n \ln(n)} - \text{масштабний коефіцієнт};$$

m – кількість експертів; s – кількість груп експертів, що дали однакові оцінки; i_d – номер позначки шкали, що обрана як експертна оцінка, $d = (1, s)$; r_d – кількість експертів, що дали в якості оцінки позначку шкали з номером i_d ; z^* – булева функція, що задає необхідні і достатні умови рівності нулю спектрального коефіцієнта узгодженості; a – середнє значення елементів множини ЕО. Для шкал інтервалів, відношень, різностей або абсолютної в якості середнього використовується середнє арифметичне [19].

4.4.1.2. Обчислення узгодженого агрегованого значення коефіцієнта узгодженості, що забезпечується q -им методом експертного оцінювання

Підставою для визначення цієї величини є множина $K_y^{(q)} = \{k_{yi}^{(q)}\}$, $i = (1, 7)$, де $k_{yi}^{(q)}$ – коефіцієнт узгодженості множини ненормованих значень ваг i -го об'єкта, що були отримані q -им методом експертного оцінювання. Спосіб обчислення цих величин викладений у попередньому підрозділі.

Означення 4.4.1. Узгодженим агрегованим значенням коефіцієнта узгодженості $K^{(q)}$, що забезпечується q -им методом експертного оцінювання, називається коректно обчислене середнє значення значущої підмножини множини $K_y(q) = \{k_{yi}^{(q)}\}$, $i = (1, 7)$ коефіцієнтів узгодженості множин ненормованих значень відносних ваг об'єктів, що визначені q -им методом експертного оцінювання.

Коректність обчислення середнього визначається відповідністю операції обчислення середнього типу шкали, у якій представлені значення коефіцієнтів узгодженості. Неважко бачити, що вони представлені в абсолютній шкалі, для якої припустимою є операція обчислення середнього арифметичного [19].

Для пояснення терміну “значуча підмножина” потрібно привести означення порога застосування, що введено в [52, 80]:

Означення 4.4.2. Порогом застосування T_u називається коефіцієнт узгодженості множини експертних оцінок, що забезпечує обчислення агрегованої експертної оцінки з припустимою точністю.

Означення 4.4.3. Значущою називається підмножина $V_h \subseteq V$, для якої $k(V_h) \geq T_u$.

Неважко бачити, що будь-яка множина ЕО містить деяку множину значущих підмножин. Це випливає з того, що будь-яка підмножина ЕО, що містить одну оцінку, має, як це випливає з (4.15), коефіцієнт узгодженості рівний 1, і тому, відповідно до означення 4.4.3, є значущою підмножиною. Однак використовувати таку підмножину для визначення агрегованої ЕО немає сенсу, тому що при цьому ігноруються оцінки коефіцієнтів узгодженості множин оцінок інших об'єктів. Приведене твердження означає лише те, що виключаючи із вихідної множини ЕО деякі значення коефіцієнтів узгодженості множин оцінок об'єктів, можна підвищити коефіцієнт узгодженості множини $K_y(q)$. Разом з тим, щоб процес підвищення узгодженості цієї множини не суперечив “здоровому глузду”, слід, по-перше, застосовувати процедуру вилучення частини оцінок тільки щодо множини ЕО, що несе інформацію, і, по-друге, прийняти обмеження на гранично прийнятний мінімум кількості оцінок коефіцієнтів узгодженості в значущій підмножині. Виходячи з того ж “ здорового глузду ”, що граничну кількість слід прийняти рівною 3.

Пояснимо також термін “множина ЕО, що несе інформацію”. У [52, 80] було обґрунтовано, що такою є множина оцінок, коефіцієнт узгодженості якої – не менше порога виявлення T_o .

Таким чином, визначення узгодженого агрегованого значення коефіцієнта узгодженості $K^{(q)}$, що забезпечується q -им методом експертного оцінювання, зводиться до виконання наступного:

- 1) Обчислюється значення порога виявлення.

Поріг виявлення визначається виразом [52, 80]: $T_o = k(S_\tau)$, де S_τ являє собою спектр, у якому $m = n$; $r_1 = 0$; $r_g = 2/n$; $r_{y \neq g \neq 0} = 1/n$; $g = [n/2 + 1]$.

2) Обчислюється значення порога застосування.

Поріг застосування T_u дорівнює коефіцієнту узгодженості $k(S_u)$ спектра S_u , у якому $m = 2$, $r_i = r_{i+1} = 1/2$; $r_{h \neq i} = 0$; $r_{h \neq (i+1)} = 0$, [52, 80].

- 3) Обчислюється значення коефіцієнта узгодженості множини $K_y^{(q)} = \{k_{yi}^{(q)}\}$, $i = (1, 7)$ коефіцієнтів узгодженості множин ненормованих значень відносних ваг об'єктів, що визначені q -им методом експертного оцінювання, згідно (4.15).
- 4) Визначається значуща підмножина множини $K_y^{(q)} = \{k_{yi}^{(q)}\}$.

У залежності від значення коефіцієнта узгодженості множини $K_y^{(q)}$ варто використовувати різні способи його підвищення. Якщо

$$k(K_y^{(q)}) < T_o,$$

то $K_y^{(q)}$ не несе інформації. У цьому випадку робиться висновок про неможливість визначення достовірних значень параметра $K^{(q)}$ на підставі отриманої множини $K_y^{(q)}$ і необхідності продовжувати експериментальне дослідження q -го методу експертного оцінювання.

Якщо

$$T_o \leq k(K_y^{(q)}) < T_u, \quad (4.16)$$

то множина $K_y^{(q)}$ містить інформацію, і тому можна шукати значущу підмножину шляхом виключення на кожнім кроці з множини $K_y^{(q)}$ коефіцієнта узгодженості k_j множини оцінок j -го об'єкта, для якого максимальним є значення $\delta_j = |a - k_j|$, де – середнє значення $K_y^{(q)}$, серед всіх об'єктів множини.

Після виключення оцінки перевіряється виконання умови (4.16) і умови

$$m_g > 3, \quad (4.17)$$

де m_g – кількість оцінок, що залишилися в множині $K_y^{(q)}$ після виконання g -го кроку алгоритму. Процес триває, поки виконуються обидві умови. Якщо на деякому кроці алгоритму виконується умова

$$k(K_y^{(q)}) \geq T_u,$$

то поточна підмножина $K_y^{(q)}$ вважається значущою, а середня оцінка, що визначена для неї, приймається в якості узгоджененої агрегованої оцінки. Якщо ж на деякому кроці процес переривається через невиконання умови (4.17), а умова (4.16) виконується, як і раніше, то обчислюється середня оцінка і користувачу повідомляється поточне значення коефіцієнта узгодженості і порога застосування. Якщо його задовільняє досягнутий рівень узгодженості, процес закінчується. Інакше процес експериментального дослідження q -го методу повинен продовжитися.

4.4.2. Обчислення математичного очікування відносної похибки визначення відносних ваг об'єктів

Вихідними даними для обчислення цього параметра є множина $W^{(q)} = \{W_j^{(q)}\}$, $j = (1, m)$ множин нормованих значень оцінок відносної ваги об'єкта, даних j -им експертом при використанні q -го методу експертного оцінювання. Розділимо кожну множину $W_j^{(q)}$ на t підмножин $W_{ij}^{(q)}$, $i = (1, t)$, де $W_{ij}^{(q)}$ – підмножина нормованих значень оцінок відносної вагомості i -го об'єкта, даних j -им експертом при використанні q -го методу; t – кількість об'єктів. Визначимо також множини

$$W_i^{(q)} = \bigcup_j W_{ij}^{(q)}.$$

Множина $W_i^{(q)}$ відносних ваг i -го об'єкта має потужність t (кількість експертів). Його елементами є позитивні числа, які не перевищують 1. Таким чином, структура цієї множини подібна структурі множини $K_y^{(q)}$ коефіцієнтів узгодженості. Тому для визначення узгодженої агрегованої оцінки коефіцієнта відносної ваги i -го об'єкта можна скористатися алгоритмом, описаним у розділі 4.4.1.2. Застосування цього алгоритму до кожного з t підмножин дозволяє обчислити узгоджену агреговану оцінку v_i коефіцієнта відносної вагомості кожного об'єкта, а також нормовані значення w_i цих оцінок.

Еталонні значення w_i^* цих величин відомі для кожного об'єкта (це відносний ступінь заповнювання прямокутників точками), що дозволяє обчислити величини похи-

бок

$$\Delta_i = w_i^* - w_i,$$

відносні похибки

$$\delta_i = \Delta_i / w_i^*$$

і математичне очікування відносної похибки:

$$M_\delta = (1/t) \sum_{i=1}^t \delta_i.$$

4.4.3. Обчислення математичного очікування тривалості визначення коефіцієнтів відносної ваги об'єктів

Дослідницька програма визначає для кожного q -го методу експертної оцінки множину $T^{(q)} = \{t_j^{(q)}\}$, де $t_j^{(q)}$ – тривалість визначення оцінок відносних ваг об'єктів j -им експертом, що використовує q -й метод. Величини $t_j^{(q)}$ нормуються:

$$t_j^{(q)} := t_j^{(q)} / t_j^{(q)\max}.$$

Внаслідок цього задача обчислення математичного очікування тривалості визначення оцінок відносних ваг об'єктів зводиться до задачі визначення агрегованої узгодженої оцінки $t^{(q)}$ цієї величини, алгоритм рішення якої описаний у розділі 4.4.1.2. Оскільки $t^{(q)}$ є нормованою величиною, на останньому етапі здійснюється її денормалізація:

$$t^{(q)} := t^{(q)} \cdot t_{j\max}^{(q)}.$$

4.4.4. Визначення відносних показників методів експертного оцінювання

Визначені відповідно до описаних вище алгоритмів оцінки методів за критеріями:

- коефіцієнт узгодженості множини експертних оцінок відносних ваг об'єктів $K^{(q)}$;

- математичне очікування відносної похибки визначення коефіцієнта відносної ваги об'єкта $M_{\delta}^{(q)}$;
- математичне очікування тривалості процесу отримання експертних оцінок відносних ваг об'єктів $t^{(q)}$;

є ненормованими величинами. Для того, щоб їх можна було використовувати для вибору методу з деякої множини згідно з перевагами ОПР, потрібно їх нормалізувати в межах розглянутої множини методів. Далі, розглядаючи методи експертної оцінки, як альтернативи, можна застосовувати багатокритеріальні методи підтримки прийняття рішень (наприклад, метод аналітичних ієрархічних процесів [26]) для вибору методу експертної оцінки з урахуванням експериментально визначених параметрів і переваг ОПР.

4.4.5. Результати обробки експериментальних даних та їх аналіз

Для наочності характеристики методів, що розглядаються, наведені в таблиці 4.1.

Таблиця 4.1.

Ознака методу	Номери методів																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Зворотний зв'язок з експертом											+	+	+	+	+	+	+	+	+	+	+	+	+
Безпосередня оцінка	+																						
Трикутник	+	+									+	+									+	+	+
Квадрат				+	+	+	+	+	+	+			+	+	+	+	+	+	+				
Попереднє упорядкування	+	+									+	+											
Фундаментальнашкала	+	+	+	+						+	+	+						+	+				
Довільнашкала	+				+	+	+	+					+	+	+	+				+	+	+	+
Адитивний						+	+						+	+					+		+		+
Мультиплікативний	+	+	+	+				+	+	+	+	+				+	+	+	+		+		+
Паралельна обробка				+	+																		
Послідовна обробка по рядках	+	+			+		+	+	+	+	+	+	+	+	+	+							
Послідовна обробка по стовпцях						+		+	+	+					+	+	+	+					
Комбінаторна обробка(КО)																				+	+	+	+
Перевірка узгодженості для КО																					+	+	
Словесне введення		+	+							+	+	+							+	+			
Числове введення	+				+	+	+	+							+	+	+	+		+	+	+	+
Графічне введення			+	+										+	+								
Дискретні значення		+	+	+						+	+	+						+	+				
Квазібезперервні значення	+	+	+	+	+	+	+						+	+	+	+	+			+	+	+	+
Номер спареного метода		12	13	5	4	14	15	16	17	18	19	2	3	6	7	8	9	10	11	22	23	20	21

Таблиця 4.2.

#	K_s	$E(K_s)$	$R(K_s)$	M_δ	$E(M_\delta)$	$R(M_\delta)$	M_τ	$E(M_\tau)$	$R(M_\tau)$	N_e
1	0.93807	0.04563	3	0.08748	0.02235	14	133.81	0.16677	1	157
2	0.90513	0.04403	10	0.23541	0.00831	21	295.81	0.07544	2	145
3	0.90894	0.04421	8	0.18579	0.01052	20	617.41	0.03615	12	148
4	0.96171	0.04678	1	0.50873	0.00384	23	326.29	0.06839	3	177
5	0.94984	0.04620	2	0.45695	0.00428	22	668.78	0.03337	14	142
6	0.87017	0.04232	19	0.11498	0.01701	15	733.18	0.03044	16	61
7	0.89217	0.04339	14	0.07739	0.02527	11	825.57	0.02703	19	54
8	0.90786	0.04416	9	0.0665	0.02940	8	728.52	0.03063	15	118
9	0.91057	0.04429	7	0.08473	0.02308	13	741.39	0.03010	18	116
10	0.89331	0.04345	13	0.02644	0.07395	4	474.97	0.04698	5	119
11	0.89487	0.04353	11	0.02849	0.06863	5	479.42	0.04655	6	118
12	0.91350	0.04443	5	0.14993	0.01304	18	342.61	0.06514	4	154
13	0.91326	0.04442	6	0.12245	0.01597	16	653.99	0.03412	13	154
14	0.88639	0.04311	15	0.08386	0.02332	12	1379.90	0.01617	23	72
15	0.87412	0.04252	16	0.06753	0.02895	9	1359.19	0.01642	22	68
16	0.86867	0.04225	20	0.06753	0.02895	10	1036.55	0.02153	20	54
17	0.87408	0.04251	17	0.12763	0.01532	17	1183.89	0.01885	21	89
18	0.83731	0.04073	22	0.01533	0.12755	2	599.04	0.03725	11	70
19	0.83523	0.04063	23	0.18563	0.01053	19	737.93	0.03024	17	74
20	0.87113	0.04237	18	0.06485	0.03015	7	540.09	0.04132	8	60
21	0.84530	0.04111	21	0.01685	0.11604	3	480.90	0.04641	7	40
22	0.91400	0.04446	4	0.00794	0.24626	1	555.63	0.04016	10	106
23	0.89386	0.04348	12	0.03414	0.05727	6	550.69	0.04052	9	81

В підзаголовках граф цієї таблиці вказані номери методів, знаком “+” відмічено наявність властивостей у перелічених методів. Крім того приведені номери аналогічних методів, що відрізняються наявністю зворотного зв’язку, або деякої іншої ознаки (спарені методи).

Результати обробки експериментальних даних наведені в таблиці 4.2, виконано ранжування методів по кожному з параметрів, і вказані кількості експертів, що прийняли участь у тестуванні методів. Колонки таблиці позначені наступним чином: K_s – коефіцієнт узгодженості; $E(K_s)$ – відносна оцінка по коефіцієнту узгодженості; $R(K_s)$ – ранжування по коефіцієнту узгодженості; M_δ – математичне очікування (МО) відносної похибки визначення КЕО; $E(M_\delta)$ – відносна оцінка по МО відносної похибки; $R(M_\delta)$ – ранжування по МО відносної похибки; M_τ – МО тривалості визначення КЕО; $E(M_\tau)$ – відносна оцінка по МО тривалості визначення КЕО; $R(M_\tau)$ – ранжування по МО тривалості визначення КЕО; N_e – кількість експертів, що приймали участь у тестуванні методів.

На підставі аналізу отриманих експериментальних даних можна зробити наступні

висновки:

- 1) Всі досліджені методи забезпечують достатній коефіцієнт узгодженості (що перевищує поріг застосування) результатів оцінювання об'єктів різними експертами, що свідчить про малу залежність результатів експертного оцінювання від особистості експерта.
- 2) Серед досліджених методів без зворотного зв'язку є такі, котрі забезпечують досить високу точність результатів: відносна похибка визначення відносних ваг знаходиться в межах 2,64-7,7% (методи 10,11, 8,7).
- 3) Використання зворотного зв'язку з експертом для більшості методів приводить до зниження похибки. Так для методу 22 відносна похибка визначення відносних ваг не перевищує 1%, для методу 18 дорівнює 1.5%.
- 4) Залучення додаткової інформації підвищує точність експертних оцінок. Це випливає з того, що майже всі методи “квадрат”, що вимагають $k(k - 1)$ порівнянь, перевершують по точності результатів методи “трикутник”, що вимагають $k(k - 1)/2$ порівнянь (за винятком комбінаторних методів).
- 5) Немає підстав стверджувати про превалювання методів “трикутник” над методами “квадрат” у відношенні часу оцінювання.
- 6) Використання квазібезперервних значень ступенів переваг об'єктів як правило приводить до підвищення точності оцінок у порівнянні з використанням дискретних значень.
- 7) Послідовний метод обробки матриці порівнянь по рядках або стовпцях, а також комбінаторний метод забезпечує більшу точність результатів у порівнянні з паралельним методом обробки матриці цілком.
- 8) Існують методи, що поступаються іншим по всіх параметрах, однак, у більшості випадків поліпшення одних параметрів спричиняє погіршення інших. Це

приводить до необхідності вирішення задачі виділення Парето-оптимальної підмножини методів з наступним вирішенням задачі вибору методу на основі мультикритеріального оцінювання.

Розділ 5.

Розробка методу підтримки прийняття рішення при виборі методу отримання кардинальних експертних оцінок

Характерною особливістю процесу отримання КЕО є участь у ньому експерта, що унеможлилює отримання значень показників застосування методів аналітичним шляхом. Тому були ініційовані експериментальні дослідження розроблених методів отримання КЕО з метою визначення їх показників застосування. Ці дослідження описані в розділі 4. Результати досліджень наведені у таблиці 4.2. Наявність цих даних є необхідною, але недостатньою умовою обґрунтованого вибору методу отримання КЕО у конкретній ситуації. Цей вибір має розглядатися як задача підтримки прийняття рішень при мультикритеріальному оцінюванні альтернатив. В даному розділі розглядається спосіб вирішення цієї задачі. Вибір методу отримання КЕО базується на використанні інформації двох типів, а саме:

- 1) експериментально визначених значень вказаних показників застосування досліджених методів отримання КЕО;
- 2) характеристики поточної ситуації у вигляді вибраної ОПР підмножини вагомих

з її точки зору показників застосування методів отримання КЕО і визначених нею значень відносної вагомості цих показників.

Задачу вибору методу отримання КЕО можна сформулювати наступним чином:

- Дано:**
- множина $M = \{m_i\}, i \in I, I = \{1, 2, \dots, n\}$ методів отримання КЕО;
 - множина $H = \{h_l\}, l \in L, L = \{1, 2, \dots, c\}$ критеріїв оцінки $m_i \in M$;
 - матриця K розмірністю $n \times c$, де елемент матриці k_{il} – це КЕО методу $m_i \in M$ по j -критерію $h_l \in H$;
 - c -вимірний вектор V , де елемент вектора v_l – це КЕО показника відносної важливості критерію $h_l \in H$ для даного конкретного випадку.

Знайти: Вектор $W = \{w_i\}$ узагальнених КЕО методів $m_i \in M$.

На основі цих оцінок буде прийматись рішення про вибір конкретного методу у даній ситуації.

Спосіб вибору методу отримання КЕО визначається наступним алгоритмом. Викладення алгоритму будемо ілюструвати конкретним прикладом, в якому використовуються експериментальні дані, наведені у таблиці 4.2.

1) Запропонувати ОПР вибрati пiдмножину $H_b \subseteq H$ критерiїв оцiнки МОКЕО.

Наприклад, припустимо, що ОПР вибрала наступнi критерiї оцiнки методiв:

- коефiцiєнт узгодженостi отриманих КЕО,
- математичне очiкування вiдносної похибки та
- математичне очiкування часу вiзначення КЕО.

2) Оцiнки МОКЕО по кожному критерiю привести до єдиної формi.

Оскiльки алгоритм передбачає вибiр методу, якому вiдповiдає максимальна iнтегрована КЕО, що враховує оцiнки по всiх критерiях $h_l \in H_b$ КЕО, то експериментальнi оцiнки за критерiями $h_l \in H_b$ мають бути перетворенi таким чином, щоб зростання перетвореної оцiнки свiдчило про пiдвищення ефективностi методу.

Для прикладу, що розглядається, в колонках 2, 5 і 8 табл. 4.2 розміщені значення критеріїв, що отримані в результаті експериментального дослідження, а в колонках 3, 6 і 9, відповідно, нормовані значення перетворених та відносних оцінок цих критеріїв. Так з 2-ї колонки отримано 3-ю – шляхом нормування:

$$k_{i4} = k_{i3} \left/ \sum_j k_{j3} \right.$$

а з 5-ї отримано 6-у і з 8-ї – 9-у, взявши зворотну величину від кожного із значень і нормуючи отримані значення:

$$k_{i7} = \frac{1}{k_{i6}} \left/ \sum_j \frac{1}{k_{j6}} \right., k_{i10} = \frac{1}{k_{i9}} \left/ \sum_j \frac{1}{k_{j9}} \right..$$

- 3) Визначити коефіцієнти вагомості критеріїв із підмножини H_b , тобто значення елементів вектора V , запропонувавши ОПР використати будь-який із методів парних порівнянь, що перелічені в табл. 4.2. Результатом роботи будуть нормовані значення ступенів важливості критеріїв $h_r \in H_b$ для даної ОПР і даної проблеми.

Нехай для прикладу, що розглядається, ОПР визначила такі коефіцієнти вагомості критеріїв: $v_1 = 0.5$; $v_2 = 0.3$; $v_3 = 0.2$.

- 4) Визначити Парето-оптимальну підмножину методів отримання КЕО. Необхідність виконання цієї процедури викликана тим, що в загальному випадку деякі методи отримання КЕО поступаються перед іншими відносно всіх, без виключення, критеріїв $h_l \in H_b$. Очевидно, що ці методи не зможуть мати максимальну узагальнену КЕО і, тим самим, бути обраними для використання. Тому, для зменшення трудомісткості подальшого процесу аналізу множини H_b методів отримання КЕО потрібно визначити в цій множині таку підмножину методів, для якої виконується властивість Парето-оптимальності, тобто підмножину методів $M_p \subseteq M, M_p = \{m_i\}, i \in P, P = \{1, 2, \dots, p\}$, що для $\forall i : i \in P \negexists j : (j \in P, j \neq i) [\forall l : (l \in L, L = \{1, 2, \dots, c\}), k_{il} \leq k_{jl}]$, де k_{ab} – значення КЕО a -го методу по b -му критерію, при умові прагнення вибору методу отримання КЕО, який має максимальну інтегровану оцінку.

Отже, на цьому кроці алгоритму шляхом виключення із множини методів тих, які по всіх критеріях не кращі за решту методів, знаходимо Парето-оптимальну підмножину M_p методів.

Визначення цієї підмножини полягає у перетворенні матриці K розмірністю $n \times c$ в матрицю K_p розмірністю $p \times c$ наступним чином:

- 4а) $i := 1;$
- 4б) $j := 1;$
- 4в) якщо $\forall l : l \in L, [(k_{il} \leq k_{jl}) \wedge (i \neq j)]$, то i -й рядок видаляється з матриці, інакше – п. 4д;
- 4г) $n := n - 1$; якщо $i \leq n$, то п. 4б, інакше – п. 4ж;
- 4д) $j := j + 1$; якщо $j \leq n$, то п. 4в;
- 4е) $i := i + 1$; якщо $i \leq n$, то п. 4б;
- 4ж) $p := n$; перетворення матриці K закінчено.

Для прикладу, що розглядається, внаслідок виконання п. 4а-4ж над матрицею вихідних даних із табл. 4.2 до Парето-оптимальної підмножини включені методи: #1, #4, #5, #10, #11, #21, #22. У табл. 5.1 наведені їх відносні оцінки по

Таблиця 5.1.

N методу	$E(K_s)$	$E(M_\delta)$	$E(M_\tau)$
1	0.04563	0.02235	0.16677
4	0.04678	0.00384	0.06839
5	0.04620	0.00428	0.03337
10	0.04345	0.07395	0.04698
11	0.04353	0.06863	0.04655
21	0.04111	0.11604	0.04641
22	0.04446	0.24626	0.04016

вибраній підмножині критеріїв.

- 5) Перенормувати значення отриманої матриці K_p по кожному з критеріїв $m_i \in M_p$:

$$k_{il} = k_{il} \left/ \sum_{j=1}^p k_{jl} \right..$$

Результати перенормування оцінок критеріїв з Парето-оптимальної підмножини для прикладу, що розглядається, наведені у табл. 5.2.

Таблиця 5.2.

N методу	$E(K_s)$	$E(M_\delta)$	$E(M_\tau)$
1	0.146645	0.041748	0.371732
4	0.150341	0.007173	0.152442
5	0.148477	0.007995	0.074382
10	0.139639	0.138134	0.104719
11	0.139896	0.128197	0.103760
21	0.132119	0.216755	0.103448
22	0.142885	0.459998	0.089517

- 6) Визначити узагальнені інтегровані оцінки по кожному методу з Парето-оптимальної підмножини за наступною формулою:

$$w_i = \sum_{l=1}^c k_{il} \cdot v_l.$$

- 7) Ранжувати методи у відповідності до значень узагальнених оцінок в порядку зменшення. Результати знаходяться в табл. 5.3.

Таблиця 5.3.

N методу	узагальнені КЕО	ранжування
1	0.160193	2
4	0.107811	6
5	0.091513	7
10	0.132203	4
11	0.129159	5
21	0.151776	3
22	0.227345	1

По отриманих результатах можна зробити висновок, що при даних вибраних критеріях оцінки МОКЕО (коєфіцієнт узгодженості отриманих КЕО, математичне очікування відносної похибки та математичне очікування часу визначення КЕО) та конкретному ступеню важливості цих критеріїв (0.5; 0.3 і 0.2 відповідно) рекомендовано для отримання КЕО вибрati метод #22 – “трикутник” адитивних парних порівнянь із вираженням безперервних значень переваг у довільній шкалі, вербальним вводом даних і комбінаторним методом обробки матриці парних порівнянь із перевіркою

узгодженості. Якщо б ОПР на 3-му кроці алгоритму були б отримані інші коефіцієнти вагомості критеріїв, наприклад: $v_1 = 0.35; v_2 = 0.25; v_3 = 0.4$, то в результаті роботи алгоритму було б рекомендовано вибрати метод #1 – безпосередньої оцінки.

Висновки

В дисертаційній роботі досліджено методи отримання та обробки експертної інформації, а саме методи отримання та обробки кардинальних ЕО, які можуть застосовуватись в експертних системах та СППР різноманітного напрямлення. В процесі дослідження розроблено обґрунтовану методику вибору методу для використання в конкретній ситуації.

Незважаючи на те, що існує достатня кількість методів експертної оцінки, і такі методи часто використовують парні порівняння для підвищення достовірності отримуваних оцінок, в основному вони призначені для отримання рангів оцінюваних об'єктів. Існуючі ж методи отримання кардинальних ЕО не дозволяють виявити джерело можливих протиріч, що виникають під час порівнянь об'єктів і порекомендувати експертові спосіб узгодити ці порівняння і тим самим підвищити достовірність результатів. Наявність цього недоліку спонукала до створення нових методів обробки матриць парних порівнянь зі зворотним зв'язком з експертом.

Крім того, оскільки всі методи ЕО досить різноманітні і характеризуються цілим рядом параметрів, то актуальною науковою проблемою є вибір методу ЕО, за допомогою якого оцінки можливо знайти з найбільшою ефективністю в конкретній ситуації.

Основні наукові та практичні результати дисертаційної роботи полягають у наступному:

- 1) Розроблено метод визначення серед множини об'єктів підмножин сумісних об'єктів, який характеризується значно меншою трудоемністю в порівнянні з методами, що застосовувались раніше для вирішення цієї проблеми. Знаходжен-

ня цих підмножин необхідне для визначення області можливого застосування методів ЕО, тобто знаходження серед множини об'єктів таких підмножин, в рамках яких є коректним застосування методів ЕО.

Цей метод разом з його різновидами широко застосовується в СППР цільового типу для знаходження серед множини підцілей деякої цілі підмножин взаємно сумісних цілей.

- 2) Розроблено індивідуальні методи парних порівнянь з застосуванням зворотного зв'язку з експертом для підвищення достовірності результатів ЕО.

Ці методи використовують так звану послідовну обробку сформованої шляхом попарних порівнянь об'єктів матрицю парних порівнянь для знаходження відносних ваг об'єктів. Ця обробка на відміну від паралельної обробки дає можливість виявити джерела протиріч, які виникли при формуванні матриці і за допомогою зворотного зв'язку з експертом усунути або, принаймні, зменшити цю неузгодженість і, тим самим, поліпшити достовірність отримуваних результатів.

Рівень узгодженості в МПП вимірюється при допомозі спектральних коефіцієнтів узгодженості, які дозволяють визначити достатність рівня узгодженості, при якому застосування результатів парних порівнянь для розрахунків відносних ваг об'єктів є коректним.

Слід відзначити, що серед різновидів методів парних порівнянь крім методів, які використовують мультиплікативні порівняння (коли експертові пропонується визначити, *у скільки разів* один об'єкт переважає інший), розроблено ще й методи, що використовують адитивні порівняння (коли експертові пропонується визначити, *на скільки умовних одиниць* один об'єкт переважає інший). Адитивні порівняння зручніше використовувати у випадках, коли переваги порівнюваних об'єктів є незначними.

- 3) Розроблено індивідуальні комбінаторні методи парних порівнянь, які дають змогу звести кількість звернень за думкою експерта до мінімуму.

Ці методи мають сенс тільки з використанням зворотного зв'язку з експертом при обробці МПП. Як і розроблені методи з послідовною обробкою МПП, вони застосовуються як для мультиплікативних, так і для адитивних типів порівнянь.

Одним із достоїнств методів є їх нечутливість до наявних в МПП багатократних неузгодженостей та до протиріч в ранжуванні об'єктів.

- 4) Розроблено групові методи отримання кардинальних ЕО, які дозволяють прийти до єдиної думки групі експертів з різною компетентністю в даному питанні, при умові відмови в оцінюванні деяких об'єктів деякими експертами.

На відміну від існуючих методів, які обробляють неповні МПП [79] для методу парних порівнянь з паралельною обробкою МПП, цей підхід розповсюджено на метод безпосередньої оцінки, а також на три методи парних порівнянь, що відрізняються трудомісткістю та потенційною точністю: ”лінія”, ”трикутник” та ”квадрат”.

Характерним для цих методів є принципова можливість обчислення оцінок по частині матриці парних порівнянь, а також спосіб формального визначення діалогів з експертами у ході покращення узгодженості їхніх оцінок. Методи розроблені з урахуванням двох видів неповноти матриці парних порівнянь. Неповнота першого виду породжена неможливістю для експерта виконати порівняння пар об'єктів по деякій причині (наприклад, недостатності для цього необхідної інформації), другий вид неповноти породжується прагненням отримати оцінки за мінімальну кількість кроків.

Розглянуто організацію роботи експертів з метою отримання потрібної вірогідності оцінок за мінімальний час.

- 5) Розроблено методику експериментального дослідження та проведено експеримент з метою визначення основних показників, що характеризують кожний з методів ЕО.

В результаті проведеного дослідження, яке включало статистичний збір та оброб-

ку інформації, проаналізовано 23 методи експертних оцінок по декількох параметрах, а саме: по коефіцієнту узгодженості отриманих оцінок, по відносній похибці та по тривалості визначення цих оцінок.

Отримані в результаті дослідження дані використовуються при виборі методу ЕО для подальшого застосування.

- 6) Розроблено методику та алгоритм вибору конкретного методу ЕО на основі отриманих експериментальним шляхом характеристик методів та вимог ОПР до методу, який має застосовуватись для оцінювання в конкретній ситуації.

Список використаних джерел

- [1] Панкова Л.А., Петровский А.М., Шнейдерман М.В. Организация экспертиз и анализ экспертной информации. - М.:Наука, 1984. - 120 с.
- [2] Орлов А.И. Современный этап развития теории экспертных оценок // <http://antorlov.chat.ru/expertoc.htm>
- [3] Бешелев С.Д., Гурвич Ф.Г. Экспертные оценки. - М.: Наука, 1973. - 79 с.
- [4] Райхман Э.П., Азгальдов Г.Г. Экспертные методы в оценке качества товаров. - М.: Экономика, 1974. - 151 с.
- [5] Бурков В.Н. Большие системы: моделирование организационных механизмов. - М.: Наука, 1989. - 354 с.
- [6] Китаев Н.Н. Групповые экспертные оценки. - М.: Знание, 1975. - 64 с.
- [7] Ларичев О.И. Объективные модели и субъективные решения. - М.: Наука, 1987. - 143 с.
- [8] Моисеев Н.Н. Неформальные процедуры и автоматизация проектирования. - М.: Знание, 1979. - 64 с.
- [9] Моисеев Н.Н. Математические задачи системного анализа. - М.: Наука, 1981. - 487 с.
- [10] Экспертные оценки в системных исследованиях // Сборник трудов. - Вып.4. - М.: ВНИИСИ, 1979. - 120 с.

- [11] Орлов А.И. Задачи оптимизации и нечеткие переменные. - М.: Знание, 1980. - 64 с.
- [12] Тюрин Ю.Н., Литвак Б.Г., Орлов А.И., Сатаров Г.А., Шмерлинг Д.С. Анализ нечисловой информации // Препринт.- М.: Научный совет АН СССР по комплексной проблеме "Кибернетика", 1981. - 80 с.
- [13] Экспертные оценки в задачах управления // Сборник трудов. - М.: Институт проблем управления, 1982. - 106 с.
- [14] Анализ нечисловых данных в системных исследованиях // Сборник трудов. - Вып.10. - М.: ВНИИСИ, 1982. - 155 с.
- [15] Методы анализа данных, оценивания и выбора // Сборник трудов. - Вып.11. - М.: ВНИИСИ, 1984. - 92 с.
- [16] Методы анализа данных, оценивания и выбора в системных исследованиях // Сборник трудов. - Вып.14. - М.: ВНИИСИ, 1986. - 124 с.
- [17] Сидельников Ю.В. Теория и организация экспертного прогнозирования. - М.: ИМЭМО АН СССР, 1990. - 196 с.
- [18] Кендэл М. Ранговые корреляции. - М.: Статистика, 1975.
- [19] Литвак Б.Г. Экспертная информация. Методы получения и анализа. - М.: Радио и связь, 1982. - 183 с.
- [20] Дэвид Г. Метод парных сравнений: Пер. с англ. / Пер. Н. Космарской и Д. Шмерлинга. - М.: Статистика, 1978. - 144 с.
- [21] Миркин Б.Г. Проблемы группового выбора. - М.: Наука, 1974. - 257 с.
- [22] Николаев В.И., Темнов В.Н. Об одном методе формирования экспертной оценки // Изв. АН СССР, Техническая кибернетика. - 1973. - N5. - C.32-36.

- [23] Gulyniysky L.F., Sergienko I.V. Refinement of the rules of choice in multiobjective decision-making problems using expert judgment // Systems Analysis, Modeling and Simulation. - 1994. - N15. - P.39-46.
- [24] Михалевич В.С., Волкович В.Л. Вычислительные методы исследования и проектирования сложных систем. - М.: Наука, Главная редакция физико-математической литературы. - 1982. - 286 с.
- [25] Штоиер Р. Многокритериальная оптимизация. Теория, вычисления и приложения: Пер. с англ. - М.: Радио и связь, 1992.
- [26] Saaty T.L. Multicriteria Decision Making: The Analytical Hierarchy Process. - McGraw Hill, NY, 1990. - 502 p.
- [27] Saaty T.L. The Analytic Network Process. - RWS Publications, Pittsburgh, 1996. - 370 p.
- [28] Тоценко В.Г., Цыганок В.В. Метод парных сравнений с обратной связью с экспертом // Проблемы управления и информатики. - 1999. - N3. - C.111-125. (Totsenko V.G., Tsyganok V.V Method of Paired Comparisons Using Feedback with expert // Journal of Automation and Information Sciences. - 1999. - Vol.31, No9. - P.86-97.)
- [29] Тоценко В.Г. Оценка сравнительной эффективности проектов комплексных целевых программ методом моделирования иерархии целей // Электронное моделирование. - 1998, - Т.20, N3. - C.76-90.
- [30] Totsenko V.G. Estimation of Comparative Efficiency of Projects of Complex Target – oriented Programs Using the Simulation Method of Goal Hierarchy // Engineering Simulation. - 1999. - Vol.16. - P.361-375.
- [31] Charnes A. and Cooper W.W. Goal Programming and multiple Objective Optimization – Part1 // European Journal of Operational Research. - 2000. - v.1, No. 1. - P.39-54.

- [32] Van den Honert R.C. The effect of uncertain pairwise comparative judgements in the multiplicative AHP // Proc. Fourth Int. Symp. on the Analytic Hierarchy Process, Simon Fraser University, Vancouver, Canada, July 12-15, 1996. - P. 273-279.
- [33] Saaty T.L., Vargas L.G. Uncertainty and rank order in analytic hierarchy process// Eur. J. Oper. Res. - 1987. - 32. - P. 107-117.
- [34] Zahir M.S. Incorporating the uncertainty of decision Judgements in the analytic hierarchy process // Eur. J. Oper. Res. - 1991. - 53. - P. 206-216.
- [35] Basak I. Incorporating within-pair order effects in the analytic hierarchy process // Math. and Comp. Modelling. - 1993. - 17(4/5). - P. 83-92.
- [36] Salo A. Inconsistency analysis by approximately specified priorities // Math. and Comp. Modelling. - 1993. - 17(4/5). - P. 123-133.
- [37] Moreno-Jimenez J.M., Vargas L., A probabilistic study of preference structures in the analytic hierarchy process as with interval judgements // Math. and Comp. Modelling. - 1993. - 17(4/5). - P. 73-81.
- [38] Cenest C., Rivest L.P. A statistical look at Saaty's method of estimating pairwise preferences expressed on a ratio scale// J. of Math. Psychol. - 1994. - 38. - P. 477-496.
- [39] Paulson D., Zahir M. Consequences for uncertainty in the analytic hierarchy process: A simulation approach // Eur. J. Oper. Res. - 1995. - 87. - P. 45-56.
- [40] M'Kay D., Bowen W., Zinnes J.L. A thrustonian view of the analytic hierarchy process. // Eur. J. Oper. Res. - 1996. - 89. - P. 427-444.
- [41] Lipovetsky S., Tishler A. Interval estimation of priorities in the AHP: Methodology and application // Israel Institute of Business Research, working Paper. - 1997. - N 16/97
- [42] Lipovetsky S., Tishler A. Interval estimation of priorities in the AHP // Eur. J. Oper. Res. - 1999. - 114. - P. 153-164.

- [43] Lipovetsky S., Conclin W.M. Robust estimation of priorities in the AHP // Eur. J. Oper. Res. - 2002. - 137. - P. 110-122.
- [44] Тоценко В.Г. Генерация алгоритмов парных сравнений для моделирования предпочтений эксперта при поддержке принятия решений, часть 1 // Электронное моделирование. - 2000. - N3. - C.11-24. (Totsenko V.G. Generation of Pair Comparisons Algorithms for Simulation of Expert Preferences during Decision Making Support,1 // Engineering Simulation. - 2001. - Vol.18. - P.303-316.)
- [45] Тоценко В.Г. Генерация алгоритмов парных сравнений для моделирования предпочтений эксперта при поддержке принятия решений, часть 2 // Электронное моделирование. - 2000. - N4. - C.16-24 (Totsenko V.G. Generation of Pair Comparisons Algorithms for Simulation of Expert Preferences during Decision Making Support,2 // Engineering Simulation. - 2001. - Vol.18. - P.447-456.)
- [46] Jang W.Ra Chainwise Paired Comparisons // Decision Sciences Journal. - 1999. - V.30, N 2. - P.581-599.
- [47] Комп'ютерна програма "Система підтримки прийняття рішень СОЛОН-2"(СППР СОЛОН-2) Україна, Міністерство освіти і науки України державний департамент інтелектуальної власності, Свідоцтво про державну реєстрацію прав автора на твір ПА №4137 / В.Г.Тоценко, В.В.Циганок, О.С.Олійник, П.Т.Качанов; зареєстровано 17.04.2001; видано 16.05.2001.(Додаток А)
- [48] Тоценко В.Г., Цыганок В.В., Олийнык А.С. Системы поддержки решений по профессиональному отбору // Труды 9-й международной школы-семинара (спец. выпуск ж-ла Информационно-управляющие системы на железнодорожном транспорте. - 1996. - N3-4. - 46с.)
- [49] Тоценко В.Г., Цыцарев В.Н., Перова Т.С., Цыганок В.В., Литвинчук О.В., Настенко И.М. Экспертная система диагностирования по произвольным диагностическим признакам с интегрированной базой знаний // Труды 9-й ме-

- ждународной школы-семинара (спец. вып. ж-ла Информационно-управляющие системы на железнодорожном транспорте. -1996. - N3-4. - 57с.)
- [50] Тоценко В.Г. О декомпозиции задач мультикритериального оценивания альтернатив при поддержке принятия решений // Системные исследования и информационные технологии. - 2002. - N1. - C.109-119.
- [51] Zanakis S.H., Solomon A., Wishart N., Dubliss. Multi-attribute decision making: A simulation of select methods // European Journal of Operational Research. - 1998. - N107. - P.507-529.
- [52] Totsenko V.G. The agreement degree of estimations set with regard of experts competency // Proceedings of the Fourth International Symposium on the Analytic Hierarchy Process. - Simon Fraser University. - Vancouver(Canada). - 1996. - P.229-241.
- [53] Тоценко В.Г. Метод підтримки прийняття рішень на основі цільового динамічного оцінювання альтернатив з урахуванням імовірностей їх реалізації // Ресурсація, зберігання і обробка даних. - 2001. - Т.3, №.4. - С.102-109.
- [54] Тоценко В.Г. Об одном подходе к поддержке принятия решений при планировании исследований и развития. Часть 2. Метод целевого динамического оценивания альтернатив // Проблемы управления и информатики. - 2001. - №.2. - С.127-139.
- [55] Kemeny J. Mathematics without numbers. - Daedalus. - 1959. - 88p.
- [56] Кемени Дж., Снелл Дж. Кибернетическое моделирование, некоторые приложения: Пер. с англ. М.: - Сов. радио, 1972.
- [57] Arrow K.J. Social choice and individual values. - New York: Wiley, 1963.
- [58] Dalkey N.C. The Delphi method: an experimental study of group opinion. Memorandum.-RM-5888-PR. - Santa Monica, Calif.: Rand corporation, 1969.
- [59] Fishburn P.C. The theory of Social Choice. - Princeton University Press, 1973.

- [60] Гюйбо Д.Т. Теории общего интереса и логическая проблема агрегирования: Пер. с англ. // Математические методы в социальных науках. - М.: Прогресс, 1973.
- [61] Young H.P., Levenglick A. A consistent extension of Condorset's election principle // SIAM J. Appl. Math. - 1978. - Vol. 35, N 2.
- [62] Aszel J., Saaty T.L. Procedures for synthesizing ratio scale judgements // J. Math. Psychol. -1983. - 27. - P. 93-102.
- [63] Fishburn P.C. Interprofile Conditions and Impossibility. - Harwood Acad. Publishers, 1987.
- [64] Dyer R.F. Forman E.H. Group Decision Support with the Analytic Hierarchy Process // Decision Support Syst. - 1992. - N 8. - P. 99-124.
- [65] Davis J.H., Stasson M.F., Parks C.D. et al. Quantitative Decisions by Groups and Individuals: Voting Procedures and Monetary Awards by Mock Civil Juries // J. Exp. Social Psychol. -1993. - 29. - P. 326-346 (doi:10.1006/jesp.1993.1015).
- [66] Iz P.H and Gardiner L.R. Analysis of Multiple Criteria Decision Support Systems for Cooperative Groups // Group Decision and Negotiation. - 1993. - N 2. - P. 61-79.
- [67] Гуляницкий В.Ф., Малышко С.А., Сергиенко И.В. О программных средствах поддержки принятия решений // Управляющие машины и системы. - 1993. - ж 5. - С. 90-97.
- [68] Гуляницкий Л.Ф., Волкович О.В., Малышко С.А. Один подход к формализации и исследованию задач группового выбора // Кибернетика и системный анализ. - 1994. - ж 3. - С. 120-127.
- [69] Forman E., Peniwati K. Aggregating individual judgments and priorities with the analytic hierarchy process // Proc. Fourth Int. Symp. Analytic Hierarchy Process, Simon Fraser University, Vancouver, Canada, July 12-15, 1996. - P. 383-391.

- [70] Peniwati K. The Analytic Hierarchy Process: The Possibility Theorem for Group Decision Making // Proc. Fourth Int. Symp. Analytic Hierarchy Process, Simon Fraser University, Vancouver, Canada, July 12-15, 1996. - P. 202-214.
- [71] Saaty T.L., Shang J.S. The Analytic Hierarchy Process and the Voting System // Proc. Fourth Int. Symp. Analytic Hierarchy Process, Simon Fraser University, Vancouver, Canada, 1996. - P. 505-517.
- [72] Timmermans D., Vlek C. Effects on Decision Quality of Supporting Multi-attribute Evaluation in Groups // Organizational Behav. and Human Decision Processes. - 1996. - 68. - P. 158-170 (doi:10.1006/obhd.1996.0096).
- [73] Tone K. Two Technical Notes on the AHP Based on Geometric Mean Method // Proc. Fourth Int. Symp. Analytic Hierarchy Process, Simon Fraser University, Vancouver, Canada, July 12-15, 1996. - P. 375-381.
- [74] Hinsz V.B. Group Decision Making with Responses of a Quantitative Native: The Theory of Social Decision Schemes for Quantities // Organizational Behav. and Human Decision Processes. 1999. - 80. - P. 28-49 (doi:10.1006/obhd.1999.2853).
- [75] Marchant T. Group decision support systems: A brief review // INFOR. - 2000. - 38, N 1. - P. 1-13.
- [76] Шаров О.І. Удосконалення графічної моделі для використання експертної інформації в структурах методу аналізу ієрархій // Вчені записки ІЕУГП. - Вип. 2.- К.: ІЕУГП, 1998. - С. 131-136.
- [77] Морозов А.О. , Косолапов В.Л. Інформаційно-аналітичні технології підтримки прийняття рішень на основі регіонального соціально-економічного моніторингу. - К.: Наукова думка, 2002.- 231c.
- [78] Коршунов Ю.М. Математические основы кибернетики: Учебное пособие. - М.: Энергия, 1980. - 424 с.

- [79] Harker P.T. Incomplete Pairwise Comparisons in the Analytic Hierarchy Process // Mathematical Modelling. - 1987. - N.9. - P.837-848.
- [80] Тоценко В.Г. Спектральный метод определения согласованности множества экспертических оценок // Электрон. моделирование. - 1999. - Т.21, N 5. - C.82-92. (Totsenko V.G. Spectral Method for Determination of Consistency of Expert Estimate Sets // Engineering Simulation. - 2000. - N17. - P.715-727.)
- [81] Brockhoff K. The performance of forecasting groups in computer dialogue and face-to-face discussion // The Delphi method. Techniques and applications. Reading. - Addison Wesley Publ. Co., Mass., 1975.
- [82] Циганок В.В. Комбінаторний алгоритм парних порівнянь зі зворотним зв'язком з експертом // Реєстрація, зберігання і обробка даних. - 2000. - Т.2, №2. - C.92-102.
- [83] Циганок В.В. Про один метод виділення в множині об'єктів підмножин взаємно сумісних об'єктів // Реєстрація, зберігання і обробка даних. - 1999. - Т.1, №2. - C.78-81.
- [84] Миллер Г. Магическое число семь плюс минус два. Инженерная психология. - М.: Прогресс, 1964.
- [85] Totsenko V.G. A Generalized Concept of Diagnosis Expert Systems // Engineering Simulation. - 1996. - Vol.13. - P.761-772.
- [86] Косолапов В.Л. Автоматизированная система обработки информации и экспертических оценок при анализе общественно-политических процессов // Управляющие системы и машины. - 1998. - №1. - C. 18-24.
- [87] Rosenberger R.S., Peterson G.L., Loomis J.B. Applying a Method of Paired Comparisons to Measure Economic Values for Multiple Goods Sets // Journal of Agricultural & Applied Economics. - 2002. - No34(1). - P.215-229.

- [88] Тульчинський В.Г., Шаров О.І. Використання комп'ютерних технологій прогнозування у фаховій підготовці студентів-менеджерів // Зб. матеріалів Міжнародної науково-практичної конф. "Економетричні методи і моделі в економіці".- Львів, ЛДУ ім. І. Франка, 1998.- С. 56-62.
- [89] Litvin I.N. Paired Comparisons in Science, Social Science, Economics and Health // Inaugural and Emeritus Addresses. - University of Port Elizabeth. - 1999. - P.1-36.
- [90] Litvin I.N., Hilliard-Lomas J.L. Poisson Model for Paired Comparisons Applied to Soccer Tournaments // South African Statistical Journal. - 1998. - P.185-199.
- [91] Тоценко В.Г., Цыганок В.В., Качанов П.Т., Деев А.А., Качанова Е.В., Торба Л.Т. Экспериментальное исследование методов получения кардинальных экспертных оценок альтернатив. Ч.І. Методы без обратной связи с экспертом // Проблемы управления и информатики. - 2003. - N1. - C.34-48.
- [92] Тоценко В.Г., Цыганок В.В., Качанов П.Т., Деев А.А., Качанова Е.В., Торба Л.Т. Экспериментальное исследование методов получения кардинальных экспертных оценок альтернатив. Ч.ІІ. Методы с обратной связью с экспертом // Проблемы управления и информатики. - 2003. - N2. - C.112-125.
- [93] Тоценко В.Г., Циганок В.В., Качанов П.Т. Підтримка прийняття рішення щодо вибору методу експертного оцінювання // Системные исследования и информационные технологии. - 2002. - N4. - C.52-60.
- [94] Розробка елементів теорії ієархій як основи побудови прогнозуючих систем підтримки рішень: Звіт про НДР / Інститут проблем реєстрації інформації НАН України. - N ДР 0197U016092. - 1999р.
- [95] Система підтримки рішень при комплексному цільовому плануванні НДДКР та закупівлі озброєння і військової техніки для Збройних Сил України: Звіт про НДР / Інститут проблем реєстрації інформації НАН України. - N ДР 0199U003984. - 1999р.

- [96] Розробка системи інформатизації діяльності науково-технічної ради Міністерства Оборони України: Звіт про НДР / Інститут проблем реєстрації інформації НАН України. - N ДР 0199U003985. - 1999р.
- [97] Розробка елементів теорії прийняття колективних рішень: Звіт про НДР "Рубікон" / Інститут проблем реєстрації інформації НАН України. - N ДР 0100U002077. - 2002. - 128с.

Додаток А



УКРАЇНА

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ДЕПАРТАМЕНТ ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ

**ПРО ДЕРЖАВНУ РЕЄСТРАЦІЮ ПРАВ
АВТОРА НА ТВІР
ПА № 4137**

Тоценко Віталій Георгійович,
Циганок Віталій Володимирович,
Олійник Олександр Степанович, Качанов Петро Тимофійович
(Прізвище, ім'я, по батькові автора)

Комп'ютерна програма "Система підтримки прийняття рішень
СОЛОН-2" (СППР СОЛОН-2)

(вид і жанр, повна та скорочена назва твору)

Зареєстровано в Державному департаменті інтелектуальної власності
 Міністерства освіти і науки України.

Дата реєстрації 17.04.2001
(число, місяць, рік)

М.П.

Голова Державного департаменту
 інтелектуальної власності
 Міністерства освіти і науки України

Дата видачі 16.05.2001
(число, місяць, рік)

М.В. Паладій
(підпис)

М. В. Паладій

Додаток Б

Акт про впровадження результатів дисертаційного дослідження

Додаток В

Програмна реалізація основних модулів системи збору даних експериментального дослідження

```

program Expert;
{$V-, X+}
uses
  Objects, Drivers, Views, Menus, Dialogs, App, MsgBox,
  AsciiTab, {Gadgets, ViewText,} ShowAlgo,
  ConstAll, Simple, Goal_Var, Ternar, GoalPref, UGoalPre, ReadGoal, TextWin,
  EnterNam, CaseAlg, WriteRez, Na_Input, NaN_Calc, In_Input, InN_Calc,
  Saat_Toc, Toc_koef, UToc_koe, St_Input, StN_Calc, {Fun_Calc,} CombCalc,
  Crt, Dos, Graph, GVision;
type
  TGV = object(TApplication)
    FindTern:boolean;
    constructor Init;
    procedure Idle; virtual;
    procedure HandleEvent(var Event: TEvent); virtual;
    procedure InitMenuBar; virtual;
    procedure InitStatusLine; virtual;
    procedure About;
  end;
{ TGV }
procedure TGV.Idle;
begin
  inherited Idle;
  ShowAlg^.Update;
end;

procedure TGV.InitMenuBar;
var
  R: TRect;
begin
  GetExtent(R);
  R.B.Y := R.A.Y + 1;
  MenuBar := New(PMenuBar, Init(R, NewMenu(
    NewSubMenu('М'єгоды', hcNoContext, NewMenu(
      NewSubMenu('Б'ез обратной связи', hcWithoutFeedBack, NewMenu(
        NewItem(' #~1~', '', kbNoKey, cmMethods+NMethod+NMatrix, hcMethods,
        NewItem(' #~2~', '', kbNoKey, cmMethods+1+NMethod+NMatrix, hcMethods+1,
        NewItem(' #~3~', '', kbNoKey, cmMethods+2+NMethod+NMatrix, hcMethods+2,
        NewItem(' #~4~', '', kbNoKey, cmMethods+3+NMethod+NMatrix, hcMethods+3,
        NewItem(' #~5~', '', kbNoKey, cmMethods+4+NMethod+NMatrix, hcMethods+4,
        NewItem(' #~6~', '', kbNoKey, cmMethods+5+NMethod+NMatrix, hcMethods+5,
        NewItem(' #~7~', '', kbNoKey, cmMethods+6+NMethod+NMatrix, hcMethods+6,
        NewItem(' #~8~', '', kbNoKey, cmMethods+7+NMethod+NMatrix, hcMethods+7,
        NewItem(' #~9~', '', kbNoKey, cmMethods+8+NMethod+NMatrix, hcMethods+8,
        NewItem(' #~10~', '', kbNoKey, cmMethods+9+NMethod+NMatrix, hcMethods+9,
        NewItem(' #~11~', '', kbNoKey, cmMethods+10+NMethod+NMatrix, hcMethods+10,
        NewItem(' #~12~', '', kbNoKey, cmMethods+11+NMethod+NMatrix, hcMethods+11,
        NewItem(' #~13~', '', kbNoKey, cmMethods+12+NMethod+NMatrix, hcMethods+12,
        NewItem(' #~14~', '', kbNoKey, cmMethods+13+NMethod+NMatrix, hcMethods+13,
        nil))))))),),
    NewSubMenu('С' обратной связью', hcWithFeedBack, NewMenu(
      NewItem(' #~2~', '', kbNoKey, cmMethods+1, hcMethods+1,
      NewItem(' #~3~', '', kbNoKey, cmMethods+2, hcMethods+2,
      NewItem(' #~6~', '', kbNoKey, cmMethods+5, hcMethods+5,
      NewItem(' #~7~', '', kbNoKey, cmMethods+6, hcMethods+6,
      NewItem(' #~8~', '', kbNoKey, cmMethods+7, hcMethods+7,
      NewItem(' #~9~', '', kbNoKey, cmMethods+8, hcMethods+8,
      NewItem(' #~10~', '', kbNoKey, cmMethods+9, hcMethods+9,
      NewItem(' #~11~', '', kbNoKey, cmMethods+10, hcMethods+10,
      NewItem(' #~12~', '', kbNoKey, cmMethods+11, hcMethods+11,
      NewItem(' #~13~', '', kbNoKey, cmMethods+12, hcMethods+12,
      NewItem(' #~14~', '', kbNoKey, cmMethods+13, hcMethods+13,
      nil))))))),),
    NewLine(

```

```

NewItem('`~А~втоматический выбор', '', kbNoKey, cmAutoRun{RunAll}, hcAllAlg,
.NewLine(
NewSubMenu(`~С~равнения', hcCompare, NewMenu(
  newItem('для #2,#4 (#~15~)', '', kbNoKey, cmMethods+NMethod, hcMethods+NMethod,
  newItem('для #3,#5 (#~16~)', '', kbNoKey, cmMethods+NMethod+1,
  hcMethods+NMethod+1,
  newItem('для #6,#7,#12 (#~17~)', '', kbNoKey, cmMethods+NMethod+2,
  hcMethods+NMethod+2,
  newItem('для #8,#9,#13 (#~18~)', '', kbNoKey, cmMethods+NMethod+3,
  hcMethods+NMethod+3,
  newItem('для #10,#11,#14 (#~19~)', '', kbNoKey, cmMethods+NMethod+4,
  hcMethods+NMethod+4,
  nil)))))),,
.NewLine(
{ newItem('Тест (отладка)', '', kbNoKey, cmRabNew, hcNoContext,
  NewLine( }
  newItem('Выход', 'Alt-X', kbAltX, cmQuit, hcNoContext,
  nil))))))),,
 newItem(`~Р~езультаы', '', kbF3, cmViewRez, hcResults,
 newItem(`~И~нформация', '', kbAltSpace, cmAbout, hcNoContext,
  nil)))););
end;

procedure TGV.InitStatusLine;
var
  R: TRect;
  PSD: PStatusDef;
begin
  PSD:=
    NewStatusDef(hcMethods, hcMethods,
      NewStatusKey(NmTst[1], kbNoKey, cmNo, nil),
      NewStatusDef(hcMethods+1, hcDef+1,
        NewStatusKey(NmTst[2], kbNoKey, cmNo, nil),
        NewStatusDef(hcMethods+2, hcMethods+2,
          NewStatusKey(NmTst[3], kbNoKey, cmNo, nil),
          NewStatusDef(hcMethods+3, hcMethods+3,
            NewStatusKey(NmTst[4], kbNoKey, cmNo, nil),
            NewStatusDef(hcMethods+4, hcMethods+4,
              NewStatusKey(NmTst[5], kbNoKey, cmNo, nil),
              NewStatusDef(hcMethods+5, hcMethods+5,
                NewStatusKey(NmTst[6], kbNoKey, cmNo, nil),
                NewStatusDef(hcMethods+6, hcMethods+6,
                  NewStatusKey(NmTst[7], kbNoKey, cmNo, nil),
                  NewStatusDef(hcMethods+7, hcMethods+7,
                    NewStatusKey(NmTst[8], kbNoKey, cmNo, nil),
                    NewStatusDef(hcMethods+8, hcMethods+8,
                      NewStatusKey(NmTst[9], kbNoKey, cmNo, nil),
                      NewStatusDef(hcMethods+9, hcMethods+9,
                        NewStatusKey(NmTst[10], kbNoKey, cmNo, nil),
                        NewStatusDef(hcMethods+10, hcMethods+10,
                          NewStatusKey(NmTst[11], kbNoKey, cmNo, nil),
                          NewStatusDef(hcMethods+11, hcMethods+11,
                            NewStatusKey(NmTst[12], kbNoKey, cmNo, nil),
                            NewStatusDef(hcMethods+12, hcMethods+12,
                              NewStatusKey(NmTst[13], kbNoKey, cmNo, nil),
                              NewStatusDef(hcMethods+13, hcMethods+13,
                                NewStatusKey(NmTst[14], kbNoKey, cmNo, nil),
                                NewStatusDef(hcMethods+NMethod, hcMethods+NMethod,
                                  NewStatusKey(NmTst[1+NMethod], kbNoKey, cmNo, nil),
                                  NewStatusDef(hcMethods+NMethod+1, hcMethods+NMethod+1,
                                    NewStatusKey(NmTst[2+NMethod], kbNoKey, cmNo, nil),
                                    NewStatusDef(hcMethods+NMethod+2, hcMethods+NMethod+2,
                                      NewStatusKey(NmTst[3+NMethod], kbNoKey, cmNo, nil),
                                      NewStatusDef(hcMethods+NMethod+3, hcMethods+NMethod+3,
                                        NewStatusKey(NmTst[4+NMethod], kbNoKey, cmNo, nil),
                                        NewStatusDef(hcMethods+NMethod+4, hcMethods+NMethod+4,
                                          NewStatusKey(NmTst[5+NMethod], kbNoKey, cmNo, nil),
                                          nil))))))))))))));
  GetExtent(R);
  R.A.Y := R.B.Y - 1;
  StatusLine := New(PStatusLine, Init(R,
    NewStatusDef(0, $FF,
      NewStatusKey(`~Alt-X~выХод', kbAltX, cmQuit,

```

```

    NewStatusKey('`F10`Меню', kbF10, cmMenu,
    nil)),
    NewStatusDef(hcWithFeedBack, hcWithFeedBack,
    NewStatusKey(' Выбор методов с обратной связью с экспертом ', kbNoKey, cmNo,
    nil),
    NewStatusDef(hcWithoutFeedBack, hcWithoutFeedBack,
    NewStatusKey(' Выбор методов без обратной связи с экспертом ', kbNoKey, cmNo,
    nil),
    NewStatusDef(hcResults, hcResults,
    NewStatusKey(' Просмотр результатов работы методов ', kbNoKey, cmNo,
    nil),
    NewStatusDef(hc3But, hc3But,
    NewStatusKey('Используйте стрелки, пробел, <ENTER> для выбора', kbNoKey, cmNo,
    nil),
    NewStatusDef(hc5But, hc5But,
    NewStatusKey(' Превосходства нет', kbNoKey, cmNo,
    nil),
    NewStatusDef(hc5But+1, hc5But+1,
    NewStatusKey(' Превосходство площади левого изображения над площадью правого
    незначительно', kbNoKey, cmNo, nil),
    NewStatusDef(hc5But+2, hc5But+2,
    NewStatusKey(' Превосходство площади левого изображения над площадью правого
    несомненно', kbNoKey, cmNo, nil),
    NewStatusDef(hc5But+3, hc5But+3,
    NewStatusKey(' Превосходство площади левого изображения над площадью правого
    существенно', kbNoKey, cmNo, nil),
    NewStatusDef(hc5But+4, hc5But+4,
    NewStatusKey(' Превосходство площади левого изображения над площадью правого
    подавляющее', kbNoKey, cmNo, nil),
    NewStatusDef(hc5But+5, hc5But+5,
    NewStatusKey(' Высшая степень превосходства площади левого изображения над
    площадью правого', kbNoKey, cmNo, nil),
    NewStatusDef(hcGLine, hcGLine,
    NewStatusKey(' Выбор с помощью манипулятора "мышь", kbNoKey, cmNo,
    nil),
    NewStatusDef(hcRealInp, hcRealInp,
    NewStatusKey(' Наберите значение в диапазоне от 0.01 до 1.00 и нажмите
    <ENTER>', kbNoKey, cmNo, nil),
    NewStatusDef(hcNaInp, hcNaInp,
    NewStatusKey(' Наберите значение в диапазоне от -1.00 до 1.00 и нажмите
    <ENTER>', kbNoKey, cmNo, nil),
    NewStatusDef(hcInInp, hcInInp,
    NewStatusKey(' Наберите значение в диапазоне от 0.01 до 99.99 и нажмите
    <ENTER>', kbNoKey, cmNo, nil),
    NewStatusDef(hcCompare, hcCompare,
    NewStatusKey(' Предварительный повторный ввод матриц парных сравнений', kbNoKey,
    cmNo, nil),
    NewStatusDef(hcPlusMinus, hcPlusMinus,
    NewStatusKey(`~+`Увеличить предпочтение', kbGrayPlus, cmPlus,
    NewStatusKey(`~-`Уменьшить предпочтение', kbGrayMinus, cmMinus,
    nil)),
    NewStatusDef(hcAllAlg, hcAllAlg,
    NewStatusKey(' Последовательный выбор всех методов, по которым еще не получены
    результаты', kbNoKey, cmNo, nil),
    NewStatusDef(hcTxtWin, hcTxtWin,
    NewStatusKey(`~Alt+X`выход', kbAltX, cmQuit,
    NewStatusKey(`~F10`Меню', kbF10, cmMenu,
    NewStatusKey(`~Alt+F3`Закрыть окно', kbAltF3, cmClose,
    NewStatusKey(`~F6`Следующее', kbF6, cmNext,
    NewStatusKey(`Shift+F6`Предыдущее', kbShiftF6, cmPrev,
    nil)))),
    PSD
  ))))))))))))))));
end;

procedure TGV.About;
begin
  MessageBox(`Система предназначена для накопления '
  + 'информации о "методах экспертной оценки" для последующего их анализа',
  nil, mfInformation + mfOkButton);
end;

```

```

procedure TGV.HandleEvent(var Event: TEvent);
var IS, OS : real;
    EigenVecs : glNpArray;
    SaveMatrix:GLNpbyNP;
    FM:FNameStr;
    R: TRect;
    alg:byte;
procedure InputDiscr;
var i,j:integer;
begin
  if not FindTern then begin
    ShowAlg^.Ftext:='Предварит. упорядочение';
    FindTernar(false);
    if ReturnCode <> 0 then exit;
    FindTern:=true;
  end;
{ goal_var.First:=1;}
NGoal := Group[1];
{-----Определение степеней предпочтения для подгруппы целей-----}
ShowAlg^.Ftext:='Опр. Ст.предп.в Дискр.Шк.';
DefineStepen(1);
if ReturnCode = 0 then begin
  SaveMatrix:=SaatiMatrix;
  for i:=1 to NGoal do
    for j:=1 to NGoal do
      if i<>j then
        SaatiMatrix[IndexAr[i],IndexAr[j]]:=SaveMatrix[i,j];
  WriteInform(NMethod+1,false);
end;
end;
Procedure InputUnLim;
var i,j:integer;
begin
  if not FindTern then begin
    ShowAlg^.Ftext:='Предварит. упорядочение';
    FindTernar(false);
    if ReturnCode <> 0 then exit;
    FindTern:=true;
  end;
{ Goal_var.First:=1;}
NGoal := Group[1];
{-----Определение степеней предпочтения для подгруппы целей-----}
ShowAlg^.Ftext:='Опр. Степ.Предп.в Непр.шк.';
UDefineStepen(1);
if ReturnCode = 0 then begin
  SaveMatrix:=SaatiMatrix;
  for i:=1 to NGoal do
    for j:=1 to NGoal do
      if i<>j then
        SaatiMatrix[IndexAr[i],IndexAr[j]]:=SaveMatrix[i,j];
  WriteInform(NMethod+2,false);
end;
end;
procedure CaseProc(alg:byte);
var i,n:integer;
    ReInp: boolean;
    MaxWeight:real;
    st:String[4];
begin
  ReInp:=alg > (NMatrix+NMethod);
  if ReInp then alg:=alg-(NMatrix+NMethod);
  case alg of
  0: MessageBox(^С'Вы прошли (или попытались пройти) все тесты. ,
               + 'На этом - сеанс работы закончен.' ,
               + ^М^С'СПАСИБО ЗА СОТРУДНИЧЕСТВО',
               nil, mfInformation + mfOkButton);

  1: begin
    ShowAlg^.Ftext:='Метод:Непосред.Оцен. ';
    WriteInform(alg,true);
    Simple_proc;
  end;
  2: begin

```

```

if (ReInp and (CaseAlgor(NMethod+1)<2)) or not FindTern or (CaseAlgor(NMethod+1)=0)
then InputDiscr;
if ReturnCode <> 0 then exit;
{Расчет по ТКФ}
str(alg:2,st);
ShowAlg^.Ftext:='Метод : #' + st;
WriteInform(alg,true);
SaveMatrix:=SaatiMatrix;
for i:=1 to NGoal do
  for n:=1 to NGoal do
    if i<>n then
      SaatiMatrix[i,n]:=SaveMatrix[IndexAr[i],IndexAr[n]];
Koeff;
end;
3:begin
if (ReInp and (CaseAlgor(NMethod+2)<2)) or not FindTern or (CaseAlgor(NMethod+2)=0)
then InputUnLim;
if ReturnCode <> 0 then exit;
{#### Расчет по ТКФ #####}
str(alg:2,st);
ShowAlg^.Ftext:='Метод : #' + st;
WriteInform(alg,true);
SaveMatrix:=SaatiMatrix;
for i:=1 to NGoal do
  for n:=1 to NGoal do
    if i<>n then
      SaatiMatrix[i,n]:=SaveMatrix[IndexAr[i],IndexAr[n]];
UKoeff;
end;
4: begin
if (ReInp and (CaseAlgor(NMethod+1)<2)) or (CaseAlgor(NMethod+1)=0) then InputDiscr;
if ReturnCode <> 0 then exit;
{Расчет по Saati}
str(alg:2,st);
ShowAlg^.Ftext:='Метод : #' + st;
WriteInform(alg,true);
CalcWeights(NGoal, SaatiMatrix, IS, OS, EigenVecs); {модуль Saat}
MaxWeight:=0;
for i:=1 to NGoal do
  if EigenVecs[i]>MaxWeight then
    MaxWeight:=EigenVecs[i];
  for i:=1 to NGoal do Koef[i]:=EigenVecs[i]/MaxWeight;{с разнораспределением}
end;
5: begin
if (ReInp and (CaseAlgor(NMethod+2)<2)) or (CaseAlgor(NMethod+2)=0) then InputUnLim;
if ReturnCode <> 0 then exit;
{Расчет по Saati}
str(alg:2,st);
ShowAlg^.Ftext:='Метод : #' + st;
WriteInform(alg,true);
CalcWeights(NGoal, SaatiMatrix, IS, OS, EigenVecs); {модуль Saat}
MaxWeight:=0;
for i:=1 to NGoal do
  if EigenVecs[i]>MaxWeight then
    MaxWeight:=EigenVecs[i];
  for i:=1 to NGoal do Koef[i]:=EigenVecs[i]/MaxWeight;{с разнораспределением}
end;
6: begin
if (ReInp and (CaseAlgor(NMethod+3)<2)) or (CaseAlgor(NMethod+3)=0) then begin
  InputMatrNa;
  if ReturnCode = 0 then
    WriteInform(NMethod+3,false)
  else exit;
end;
str(alg:2,st);
ShowAlg^.Ftext:='Метод : #' + st;
{ ShowAlg^.Ftext:='Тест:0бр"НА"Vдом-оц.'+' Стр'; }
WriteInform(alg,true);
CalcNaN(false,true);
end;
7: begin
if (ReInp and (CaseAlgor(NMethod+3)<2)) or (CaseAlgor(NMethod+3)=0) then begin
  InputMatrNa;

```

```

        if ReturnCode = 0 then
            WriteInform(NMethod+3,false)
        else exit;
    end;
    str(alg:2,st);
    ShowAlg^.Ftext:='Метод : #' + st;
{ ShowAlg^.Ftext:='Тест:0бр"НА"Вдом-оп.'+' Стлб';}
    WriteInform(alg,true);
    CalcNaN(false,false);
end;
8: begin
    if (ReInp and (CaseAlgor(NMethod+4)<2)) or (CaseAlgor(NMethod+4)=0) then begin
        InputMatrIn;
        if ReturnCode = 0 then
            WriteInform(NMethod+4,false)
        else exit;
    end;
    str(alg:2,st);
    ShowAlg^.Ftext:='Метод : #' + st;
{ ShowAlg^.Ftext:='Тест:0обработка "В'+' Стр';}
    WriteInform(alg,true);
    CalcInN(true);
end;
9: begin
    if (ReInp and (CaseAlgor(NMethod+4)<2)) or (CaseAlgor(NMethod+4)=0) then begin
        InputMatrIn;
        if ReturnCode = 0 then
            WriteInform(NMethod+4,false)
        else exit;
    end;
    str(alg:2,st);
    ShowAlg^.Ftext:='Метод : #' + st;
{ ShowAlg^.Ftext:='Тест:0обработка "В'+' Стлб';}
    WriteInform(alg,true);
    CalcInN(false);
end;
10: begin
    if (ReInp and (CaseAlgor(NMethod+5)<2)) or (CaseAlgor(NMethod+5)=0) then begin
        InputMatrSt;
        if ReturnCode = 0 then
            WriteInform(NMethod+5,false)
        else exit;
    end;
    str(alg:2,st);
    ShowAlg^.Ftext:='Метод : #' + st;
{ ShowAlg^.Ftext:='Тест:0раб.Фунд.шк.'+' Стр';}
    WriteInform(alg,true);
    CalcStN(true);
end;
11: begin
    if (ReInp and (CaseAlgor(NMethod+5)<2)) or (CaseAlgor(NMethod+5)=0) then begin
        InputMatrSt;
        if ReturnCode = 0 then
            WriteInform(NMethod+5,false)
        else exit;
    end;
    str(alg:2,st);
    ShowAlg^.Ftext:='Метод : #' + st;
{ ShowAlg^.Ftext:='Тест:0раб.Фунд.шк.'+' Стлб';}
    WriteInform(alg,true);
    CalcStN(false);
end;
12: begin
    if (ReInp and (CaseAlgor(NMethod+3)<2)) or (CaseAlgor(NMethod+3)=0) then begin
        InputMatrNa;
        if ReturnCode = 0 then
            WriteInform(NMethod+3,false)
        else exit;
    end;
    str(alg:2,st);
    ShowAlg^.Ftext:='Метод : #' + st;
{ ShowAlg^.Ftext:='Тест:0раб.НА комб.'+' ';}
    WriteInform(alg,true);

```

```

    CalComb(1);
end;
13: begin
  if (ReInp and (CaseAlgor(NMethod+4)<2)) or (CaseAlgor(NMethod+4)=0) then begin
    InputMatrIn;
    if ReturnCode = 0 then
      WriteInform(NMethod+4,false)
    else exit;
  end;
  str(alg:2,st);
  ShowAlg^.Ftext:='Метод : #' + st;
{ ShowAlg^.Ftext:='Тест:Обраб."В"комбинаторная';}
  WriteInform(alg,true);
  for i:=1 to NGoal do begin
    for n:=i to NGoal do
      if SaatiMatrix[i,n]<1 then
        Matrix[i,n]:=-Round(10/SaatiMatrix[i,n])
      else
        Matrix[i,n]:=Round(SaatiMatrix[i,n]/10);{преобразуем в ShortInt}
    end;
    CalComb(2);
  end;
14: begin
  if (ReInp and (CaseAlgor(NMethod+5)<2)) or (CaseAlgor(NMethod+5)=0) then begin
    InputMatrSt;
    if ReturnCode = 0 then
      WriteInform(NMethod+5,false)
    else exit;
  end;
  str(alg:2,st);
  ShowAlg^.Ftext:='Метод : #' + st;
{ ShowAlg^.Ftext:='Тест:Обраб.фунд.комбинаторная';}
  WriteInform(alg,true);
  CalComb(3);
end;
end;{case}
if (alg in [1..NMethod]) and (ReturnCode=0) or (ReturnCode=2) then
  WriteInform(alg,false);
end;
begin
  inherited HandleEvent(Event);
  if Event.What = evCommand then
  begin
    ReturnCode := 0;
    NofQ:=0;
    TimeQ:=0; {обнулили счетчик секунд}
    case Event.Command of
      cmAbout : About;
      cmAutoRun : begin
        alg:=0;
        repeat
          if alg>(NMethod+NMatrix) then alg:=alg-(NMethod+NMatrix);
          alg:=CaseAlgor(0{alg});
        NofQ:=0;
        TimeQ:=0; {обнулили счетчик секунд}
        CaseProc(alg);
        if ReturnCode=1 then break;
      until alg=0;
    end;
    cmMethods..cmMethods+NMethod-1:
    begin
      FeedBack:=true;
      CaseProc(Event.Command-cmMethods+1);
    end;
    cmMethods+NMethod+NMatrix..cmMethods+NMethod+NMatrix+NMatrix-1:
    begin
      FeedBack:=false;
      CaseProc(Event.Command-cmMethods-NMethod-NMatrix+1);
    end;
    cmMethods+NMethod: begin
      FindTern:=false;
      InputDiscr
    end;
  end;
end;

```

```

cmMethods+NMethod+1:begin
  FindTern:=false;
  InputUnLim
end;
cmMethods+NMethod+2: begin
  InputMatrNa;
  if ReturnCode = 0 then
    WriteInform(NMethod+3,false)
end;
cmMethods+NMethod+3: begin
  InputMatrIn;
  if ReturnCode = 0 then
    WriteInform(NMethod+4,false)
end;
cmMethods+NMethod+4: begin
  InputMatrSt;
  if ReturnCode = 0 then
    WriteInform(NMethod+5,false)
end;
(* cmRabNew: begin
(* {престановка матрицы}
for i:=1 to NGoal do for j:=1 to NGoal do
  DebMatrix[i,j]:=SaatiMatrix[IndexAr[i],IndexAr[j]];
  SaatiMatrix:=DebMatrix;
  NofQ:=0;
  TimeQ:=0; {обнулили счетчик секунд}
  (* ShowAlg^.Ftext:='Тест: Ввод НА';
  { NGoal:=7; {для отладки *****}
  { SaatiMatrix:=DebMatrix; {для отладки *****}
  InputMatrNa;
  if ReturnCode = 0 then begin
    ShowAlg^.Ftext:='Тест:Обраб.НА комб.'+' Стр';
    for i:=1 to NGoal do begin
      IndexAr[i]:=i;
      for n:=i to NGoal do
        Matrix[i,n]:=round(SaatiMatrix[i,n]*100);{преобразуем в ShortInt}
    end;
    CalCombN;
    if ReturnCode = 0 then begin
      WriteInform(9);
      DisableCommands([cmRabNew])
    end;
    end;
    end; *)
  (* NGoal:=7; {для отладки *****}
  SaatiMatrix:=DebMatrix; {для отладки *****}
  {if CaseAlgor(NMethod+5)=0 then begin
    ShowAlg^.Ftext:='Тест: Ввод в фунд.шкале';
    InputMatrSt;
    if ReturnCode = 0 then
      WriteInform(NMethod+5)
    else begin ClearEvent(Event); exit end;
  end;}
  ShowAlg^.Ftext:='Тест:отладка';
  CalComb(2);
  if ReturnCode = 0 then
    WriteInform(14);
  end; *)
  cmViewRez : begin
    R.Assign(1, 1, 78, 23);
    FN:=FilNam;
    InsertWindow(New(PTextWindow, Init(R, FN , 1, hcTxtWin)));
  end;
  end;
else
  Exit;
end;
ShowAlg^.Ftext:='      ';
ClearEvent(Event);
end;
end;

type PStT = ^TStT;
TStT = object(TStaticText)

```

```

        function GetPalette: PPalette; virtual;
      end;

      function TStT.GetPalette: PPalette;
      const
        CStT = #01;
        P: String[Length(CStT)] = CStT;
      begin
        GetPalette := P;
      end;

      constructor TGV.Init;
      var R:TRect;
          PST:PStaticText;
          i:integer;
      begin
        inherited Init;
        CaseName;
        ReadGoalFile;
        GetExtent(R);
        R.A.X := R.B.X - 35; R.B.Y := R.A.Y + 1;
        ShowAlg := New(PShowAlg, Init(R));
        Insert(ShowAlg);
        GetExtent(R);
        R.A.X:=R.B.X-length(ExpertName)-12; R.B.Y:=R.B.Y-2; R.A.Y:=R.B.Y-1;
        PST:=New(PStaticText,Init(R,'Имя эксперта:' +Copy(ExpertName,5,length(ExpertName)-4)));
        DeskTop^.Insert(PST);
        GetExtent(R);
        R.B.X:=R.B.X-1; R.A.X:=R.B.X-51; R.B.Y:=R.B.Y-16; R.A.Y:=R.B.Y-1;
        DeskTop^.Insert(New(PStT,Init(R,^С'окращения:')));
        for i:=1 to 12 do begin
          R.Move(0,1);
          DeskTop^.Insert(New(PStT,Init(R,Abbr[i])));
        end;
        FindTern:=false;
      end;

      var GVApp: TGV;
          AMode: word;
          AFontSize1: byte;
          FontPtr1: pointer;

      begin
        AMode := ModeAvail;
        case AMode of
          smVGA: AMode := smVGA400;
          smEGA:;
          smVGA400:;
        else begin
          WriteLn('Может работать на EGA, VGA, ....');
          Halt
        end
      end;
      ReadFont('GVFONT14.FNT', FontPtr1, AFontSize1);
      if FontPtr1 = nil then
        AFontSize1 := 14;
      InitGVision(AMode, AFontSize1);
      if GError <> grOk then begin
        WriteLn(GraphErrorMsg(GError));
        Halt(1);
      end;
      SetFont(FontPtr1);
      GVApp.Init;
      GVApp.Run;
      GVApp.Done;
    end.
  
```

```

unit ShowAlgo;
interface
uses
  Objects, Views, MsgBox, Dialogs, Drivers;
type

```

```

PShowAlg=^TShowAlg;
TShowAlg= object(TView)
  OldText,{Mem : LongInt;}
  FText: String;
  constructor Init(var Bounds: TRect);
  procedure Draw; virtual;
  procedure Update;
end;

implementation

constructor TShowAlg.Init(var Bounds: TRect);
begin
  inherited Init(Bounds);
  FText:='      ';
  OldText:='Начало';
end;

procedure TShowAlg.Draw;
var
  B: TDrawBuffer;
  C: Byte;
begin
  OldText := FText;
  C := GetColor(2);
  MoveChar(B, ' ', C, Size.X);
  MoveStr(B, Ftext, C);
  WriteLine(0, 0, Size.X, 1, B);
end;

procedure TShowAlg.Update;
begin
  if (OldText <> FText) then DrawView;
end;
end.

```

```

unit ConstAll;

interface
uses Goal_Var, ShowAlgo;

Const
  maxRect=GlobMaxNGoal;

  ExpertName: string = '';

  cmAbout = 200;
  cmMethods = 301; {- 314+5+14}
  cmPlus = 208;
  cmMinus = 209;
  cmViewRez = 210;
{ cmObrab = 211; }
  cmStepBack= 215; {ReturnCode для возврата на шаг}
  cmRabNew = 217;
  cmAutoRun = 219;
  cmWithoutFeedback= 220;
  cmCombKsog= 221; {+2}

  hc3But = $F000; {-$F002}
  hc5But = $F001; {-$F006}
  hcGline = $F007;
  hcGVertText = $F008;
  hcRealInp = $F009;
  hcNaInp = $F00A;
  hcCompare = $F00B;
  hcPlusMinus = $F010;
  hcInInp = $F011;
  hcAllAlg = $F018;
  hcTxtWin = $F019;
  hcWithFeedBack = $F01A;

```

```

hcWithoutFeedBack = $F01B;
hcResults = $F01C;
hcMethods = $F101; {-$F10D}
NMethod = 14; {количество методов}
NMatrix = 5; {количество матриц}
Abbr: array[1..12] of string[52] = (
{1}' МпПС - мультиплекативных парных сравнений',
{2}' АдПС - аддитивных парных сравнений',
{3}' ДискрЗП - дискретных значений предпочтений',
{4}' КвНепрЗП - квазинепрерывных значений предпочтений',
{5}' НепрЗП - непрерывных значений предпочтений',
{6}' ФШ - фундаментальной шкале',
{7}' ПШ - произвольной шкале',
{8}' ВербВвД - вербальным вводом данных',
{9}' ГрВвД - графическим вводом данных',
{10}', ПрУпор - предварительным упорядочением',
{11}', посл. - последовательным',
{12}', МОМС - методом обработки матрицы сравнений');

Nmstt: array[1..NMethod+NMatrix] of string[80] = (
{1}'Непосредственной оценки',
{2}"трехугольник"МпПС с выражением ДискрЗП в ФШ, ВербВвД, ПрУпор и посл. МОМС',
{3}"трехугольник"МпПС с выражением КвНепрЗП в ФШ, ГрВвД, ПрУпор и посл. МОМС',
{4}'квадрат"МпПС с выражением ДискрЗП в ФШ, ВербВвД и паралельным МОМС (Саати)',
{5}'квадрат"МпПС с выражением КвНепрЗП в ФШ, ГрВвД и паралельным МОМС (Саати)',
{6}'квадрат"АдПС с выражением НепрЗП в ПШ, ВербВвД и посл. МОМС по строкам',
{7}'квадрат"АдПС с выражением НепрЗП в ПШ, ВербВвД и посл. МОМС по столбцам',
{8}'квадрат"МпПС с выражением НепрЗП в ПШ, ВербВвД и посл. МОМС по строкам',
{9}'квадрат"МпПС с выражением НепрЗП в ПШ, ВербВвД и посл. МОМС по столбцам',
{10}'квадрат"МпПС с выражением ДискрЗП в ФШ, ВербВвД и МОМС по строкам',
{11}'квадрат"МпПС с выражением ДискрЗП в ФШ, ВербВвД и МОМС по столбцам',
{12}'трехугольник"АдПС с выражением НепрЗП в ПШ, ВербВвД и комбинаторным МОМС',
{13}'трехугольник"МпПС с выражением НепрЗП в ПШ, ВербВвД и комбинаторным МОМС',
{14}'трехугольник"МпПС с выражением НепрЗП в ФШ, ВербВвД и комбинаторным МОМС',
{1}'Матрица предпочтений в дискретной шкале',
{2}'Матрица предпочтений в квазинепрерывной шкале',
{3}'Матрица предпочтений аддитивных парных сравнений', {НА}
{4}'Матрица предпочтений мультиплекативных парных сравнений', {В}
{5}'Матрица предпочтений с оценкой в фундаментальной шкале';

dm=9; {dm-размер шкалы предпочтений}
NmPr: array[1..dm] of string[20] = (
{1}'равнозначность',
{2}'слабое',
{3}'умеренное',
{4}'заметное',
{5}'очень заметное',
{6}'сильное',
{7}'очень сильное',
{8}'доминирующее',
{9}'абсолютное');

Uvel='увеличить';
Umen='уменьшить';
FailMess='*** НЕ успешное окончание ***';
eps=0.1; {точность ( epsилон)}
MaxSec=180; {максимальное время обдумывания вопроса}
FileNazva:string=''; {for automatic calculating}

var
  Safe: array[1..maxRect] of real(* = (0.67, 1.0, 0.5, 0.87, 0.68, 0.09)*);
  NRect, {количество квадратов}
  TimeQ, {количество времени в секундах затраченного на тест}
  NofQ {количество задаваемых вопросов в тесте} :integer;
  ShowAlg: PShowAlg;
  FeedBack: boolean; {обратная связь}
const _FB:boolean=false; {without feedback autocalc}
  _Ks:boolean=false; {break comb if Ks>=Tp}
implementation
end.

(* Модуль Simple . Метод непосредственной оценки *)
unit Simple;
INTERFACE
uses Objects, Views, App, Dialogs, MsgBox,

```

```

WinRect1, ConstAll, DialSimp,
Goal_Var;

procedure Simple_Proc;
IMPLEMENTATION
procedure Simple_Proc;
VAR
  R:TRect;
  Win: PWinRect;
  Dial: PDialSimp;
  Control,i: word;
  Kreal: real;

begin
  for i:=1 to NRect do Koef[i]:=0.0;
  i:=1;
  Dial:=New(PDialSimp, Init(^M^C'степень заполнения изображения',true));
  Dial^.DisableCommands([cmStepBack]);
  repeat
    DeskTop^.GetExtent(R);
    R.B.X := R.B.X div 2;
    R.A.Y := R.B.Y div 2;
    R.Move(8,0);
    Win := New(PWinRect, Init(R, i, Safe[i]));
    DeskTop^.Insert(Win);{образец}
    Dial^.SetData(Koef[i]);
    Control := DeskTop^.ExecView(Dial);
    DeskTop^.Delete(Win);
    DisPose(Win, Done);
    if Control=cmOk then begin
      Dial^.GetData(Kreal);
      Koef[i]:=kreal;
      Inc(i);
      Dial^.EnableCommands([cmStepBack]);
      if i>NRect then
        if MessageBox(^C'Подтвердите правильность последнего ответа'+^M^C
          '(Но-шаг назад)',nil,mfConfirmation+mfYesButton+mfNoButton) <> cmYes then
          Control:=cmStepBack;
    end;
    if Control=cmStepBack then begin
      Dec(i);
      if i=1 then
        Dial^.DisableCommands([cmStepBack]);
      Dial^.SelectNext(True)
    end;
    if Control=cmCancel then
      if MessageBox(^C'Тест еще не закончен!'+
        ^M^C'Вы действительно хотите прервать тест?',nil,
        mfConfirmation+mfYesButton+mfNoButton) <> cmYes then
        Control:=cmOk;
    until (Control=cmCancel) or (i>NRect);
    ReturnCode:=0;
    if Control=cmCancel then ReturnCode:=1;
  end;
end.



---


unit WinRect1;

interface

uses
  Objects, Views, Dialogs, Dos,
  GVision,
  ConstAll,RectRnd;

type
  PWinRect = ^TWinRect;
  TWinRect = object(TDialog)

```

```

    Seconds:LongInt;
constructor Init(var Bounds: TRect; aNumber: word; aPart: real);
destructor Done; virtual;
end;

implementation

constructor TWinRect.Init(var Bounds: TRect; aNumber: word; aPart: real);
var
  Hour1, Minute1, Second1, Hand1: Word;
  FileView: PGView;
  tit: TTITLEStr;
begin
  Str(aNumber, Tit);
  TDialog.Init(Bounds, Tit);
  {Подсчет количества Вопросов - кол-ва квадратов слева на Экране}
  if Bounds.A.X<40 then begin
    Inc(NofQ);
    {инициализация счетчика времени}
    GetTime(Hour1, Minute1, Second1, Hand1);
    Seconds:=LongInt(Hour1)*3600+Minute1*60+Second1;
  end;
  Flags := 0;{wfClose}
  GetExtent(Bounds);
  Bounds.Grow(-1, -1);
  Insert(New(PRectRnd, Init( Bounds, aNumber, aPart)));
end;

destructor TWinRect.Done;
var Hour2, Minute2, Second2, Hand2: Word;
  Sec: LongInt;
begin
  if Origin.X<40 then begin
    GetTime(Hour2, Minute2, Second2, Hand2);
    Sec:=LongInt(Hour2)*3600+Minute2*60+Second2;
    if Abs(Sec-Seconds)>MaxSec then
      Inc(TimeQ,MaxSec)
    else
      Inc(TimeQ,Abs(Sec-Seconds));
  end;
  inherited Done;
end;
end.

```

```

unit RectRnd;

interface

uses
  Objects, Drivers, Views,
  Crt, Dos, Graph, GVision,
  ConstAll;

type
  PRectRnd = ^TRectRnd;
  TRectRnd = object(GView)
    Number: word;
    Part: real;
    constructor Init( R:TRect; aNumber: word; aPart: real);
    procedure Draw; virtual;
    function GetPalette: PPalette; virtual;
  end;

implementation

const CRectGr = #20;

constructor TRectRnd.Init( R:TRect; aNumber: word; aPart: real);
begin
  inherited Init(R, FullBuf);

```

```

    Number:=aNumber;
    Part:=aPart;
end;

function TRectRnd.GetPalette: PPalette;
const
  P: String[Length(CRectGr)] = CRectGr;
begin
  GetPalette := P;
end;

procedure TRectRnd.Draw;
const Discr:word = 30;
var
  x,y, LenSide, BegX, Color_: word;
  S: string;
begin
  Str(Number, S);
  InitDraw;
  SetViewPort(0, 0, GetSizeX, GetSizeY, ClipOn);
  SetFillBkColor( LightGray );
  ClearViewPort;
  Color_:=GetColor(1);
  SetFillStyle(0, Color_ and $f);
  SetColor(Color_);
  SetFillBkColor(Color_ shr 4);
  LenSide:=GetSizeY div Discr;
  BegX:=GetSizeX div 2 - GetSizeY div 2;
  for x:=0 to Discr-1 do
    for y:=0 to Discr-1 do
      if Random <= Part then
        Bar(BegX + LenSide * x, LenSide * y,
             BegX + LenSide * (x+1)-1, LenSide * (y+1)-1);
  Rectangle(BegX, 0, BegX + LenSide * Discr, LenSide * Discr);
  DoneDraw
end;
end.

```

```

unit Goal_Var;
INTERFACE
uses Trans;
CONST
  nsh: word = 10;{размерность дискретной шкалы предпочтений}
  MaxNGoal = 7; {макс-ное количество целей В ГРУППЕ }
  GlobMaxNGoal = 7;{30; 7-для эксперимента }
  Mmax = MaxNGoal+1{+1 для обработки хар-к};
  Nmax = MaxNGoal+1{+1 для обработки хар-к};
TYPE
  {Real = Extended;}
  glNparray = ARRAY [1..Nmax+1] of real;
  glNpByNp = ARRAY [1..Nmax+1,1..Nmax+1] of real;
  glMparray = ARRAY [1..Mmax+1] of real;
var
  SaatiMatrix : glNpByNp;
  Matrix : MatrixType;
  IndexAr : array[1..MaxNGoal] of byte;
  Ravnoznach : array[1..MaxNGoal] of char;
  GlobIndexAr : array[1..GlobMaxNGoal] of byte;
  GlobRavnoznach : array[1..GlobMaxNGoal] of char;
  NGr : byte;
  Group : array[1..5] of byte;
  GlobNGoal, {общее количество подцелей}
  NGoal, {количество подцелей в группе}
  LastNGoal, {номер последней цели по списку Goals в группе}
  ReturnCode : byte;

  koef_sog {коэф-ты соглас-сти соотв-щих весов}
  : array[1..GlobMaxNGoal] of real;
  koef {веса альтернатив}
  : array[1..GlobMaxNGoal] of real;

```

```

kpr, {коэф-нт применения}
kbn, {коэф-нт обнаружения}
ksog: real; { текущий коэф-т соглас-я}
{=====}

function BBC(r : real) : integer;

IMPLEMENTATION
{=====}
function BBC(r : real) : integer;
{Определение ближайшего большого целого}
var
  i : integer;
begin
  i:=trunc(r);
  if Frac(r) <> 0 then i:=i+1;
  BBC := i;
end;
end.



---


unit Ternar;

INTERFACE
{ Выходные данные:
  Matrix - матрица со значениями 'Л', 'Х', 'Р' с транзитивным замыканием.}
uses Objects, Views, App, Dialogs,
  Goal_Var, Trans, WinRect1, Dial3But, ConstAll;
{=====}
procedure FindTernar( Trans_: boolean );
{построение тернарного списка всех целей; Trans_ - с учетом транзитивности}

implementation

VAR R:TRect;
  WinL, WinR: PWinRect;
  Dial3B: PDial3But;
  Control: word;

procedure FormChoiceMatrix(
  var MenuTab : MatrixType; { входная матрица из string[80] }
  NMatr : byte; { размер матрицы }
  Trans_: boolean (* Включить MakeTrans *)
);
const
  PrefArr : array[1..3] of char = ( 'Л', 'Х', 'Р' );
var
  trow,tcol : byte; { текущие строка и колонка }
  c, k , i , ci, ki, PrefChoice : byte;
  ch, rch : char;
  First : boolean;
  StartItem : byte;
  NotReady : byte;
{-----}
procedure CountNotReady;
var c, k : byte;
begin
  NotReady := 0;
  for c := 1 to NMatr do
    for k := c + 1 to NMatr do
      if MenuTab[c,k] = ' ' then
        Inc(NotReady);
end;
{-----}
procedure MakeTrans;
begin
  if DefineTransitivity(MenuTab, MenuTab, NMatr) then {модуль Trans}
    CountNotReady;
end;
{-----}
procedure ProceedToPusto;
var c, k : byte;
Label Next;

```

```

begin
  for c := 1 to NMatr do
    for k := c + 1 to NMatr do
      if MenuTab[c,k] = ' ', then goto Next;
Next:
  DeskTop^.Delete(WinL);
  DisPose(WinL, Done);
  DeskTop^.Delete(WinR);
  DisPose(WinR, Done);
  DeskTop^.GetExtent(R);
  R.B.X := R.B.X div 2;
  R.B.Y := R.B.Y div 2;
  WinL := New(PWinRect, Init(R, c, Safe[c]));
  DeskTop^.GetExtent(R);
  R.A.X := R.B.X div 2;
  R.B.Y := R.B.Y div 2;
  WinR := New(PWinRect, Init(R, k, Safe[k]));
  DeskTop^.Insert(WinL);
  DeskTop^.Insert(WinR);

  trow:=c; tcol := k;
end;

{ ТЕЛО ПРОЦЕДУРЫ FormChoiceMatrix}
begin
  First := true;
  { Анализ параметров на ошибки }
  StartItem := 2;

  { Формирование элементов матрицы }
  NotReady := 0;
  for c := 1 to NMatr do
    for k := c+1 to NMatr do
      if MenuTab[c,k] = ' ', then
        NotReady := NotReady + 1;
{### Цикл управления меню ###}
{ tcol,trow - "координаты" текущего элемента;
  c,k - "координаты" следующего элемента }
  ci := BBC(StartItem / NMatr);
  ki := StartItem - (ci-1)*NMatr ;

  ch := #0; c:=ci; k:=ki; trow:=ci; tcol:=ki; i := StartItem;

  DeskTop^.GetExtent(R);
  R.B.X := R.B.X div 2;
  R.B.Y := R.B.Y div 2;
  WinL := New(PWinRect, Init(R, c, Safe[c]));
  DeskTop^.GetExtent(R);
  R.A.X := R.B.X div 2;
  R.B.Y := R.B.Y div 2;
  WinR := New(PWinRect, Init(R, k, Safe[k]));
  DeskTop^.Insert(WinL);
  DeskTop^.Insert(WinR);

  Dial3B:=New(PDial3But, Init);
repeat
  Control := DeskTop^.ExecView(Dial3B);
  case Control of
    cmOk :
    begin
      Dial3B^.GetData(PrefChoice);
      rch := PrefArr[PrefChoice+1];
      MenuTab[trow,tcol] := rch;
      if NotReady <> 0 then
        begin
          NotReady := NotReady - 1;
          if NotReady = 0 then
            Exit;
        end
      else
        Exit;
      If trans_ then
        MakeTrans;
    end
  end;
end;

```

```

        if NotReady = 0 then
            Exit;
            ProceedToPusto;
        end;
        cmCancel : begin ReturnCode := 1; Exit end;
    end; { case }
until NotReady = 0;
end; {FormChoiceMatrix}
{=====
procedure LinearOrdering;
{ Упорядочение в убывающем порядке с фиксацией индексов упорядочения
  в GlobIndexAr}
var i,p,ma, left, right : byte;
begin
    for i := 1 to GlobNGoal do
        GlobIndexAr[i] := i;
    for p:= 1 to GlobNGoal-1 do
        begin
            ma:=p;
            for i:= p+1 to GlobNGoal do
                begin
                    left := GlobIndexAr[ma];
                    right := GlobIndexAr[i];
                    if left < right then
                        begin
                            if Matrix[left,right] = 'X' then ma:=i;
                        end
                    else
                        if Matrix[right, left] = 'Л' then ma:=i;
                end;
            i := GlobIndexAr[p];
            GlobIndexAr[p] := GlobIndexAr[ma];
            GlobIndexAr[ma] := i;
        end;
    end; { LinearOrdering}
{=====}
procedure MakeRavnoznachList;
var i, l, r : byte;
    ch : char;
begin
    ch := '1';
    for i := 1 to GlobNGoal do
        GlobRavnoznach[i] := ',';
    for i := 1 to GlobNGoal-1 do
        begin
            if GlobIndexAr[i] < GlobIndexAr[i+1] then
                begin l := GlobIndexAr[i]; r := GlobIndexAr[i+1]; end
            else
                begin l := GlobIndexAr[i+1]; r := GlobIndexAr[i]; end;
            if Matrix[l,r] = 'P' then
                begin GlobRavnoznach[i] := ch; GlobRavnoznach[i+1] := ch; end
            else if ch = '1' then ch := '2' else ch := '1';
        end;
    end;
{=====}
procedure FindTernar( Trans_: boolean );
{построение тернарного списка всех целей}
var c, k : byte;
{-----}
procedure DefineGroup(n : byte);
begin
    case n of
        1..7: begin NGr := 1; Group[1] := n; end;
        8..13: begin NGr := 2;
            Group[1] := BBC(n/2);
            Group[2] := n - Group[1] + 1;
        end;
        14..19: begin NGr := 3;
            Group[1] := BBC(n/3);
            Group[2] := BBC((n - Group[1] + 2)/2);
            Group[3] := n - Group[1] - Group[2] + 2;
        end;
        20..25: begin NGr := 4;
    end;

```

```

        Group[1] := BBC(n/4);
        Group[2] := BBC((n - Group[1] + 3)/3);
        Group[3] := BBC((n - Group[1] - Group[2] + 2)/2);
        Group[4] := n - Group[1] - Group[2] - Group[3] + 3;
    end;
26..31: begin NGr := 5;
        Group[1] := BBC(n/5);
        Group[2] := BBC((n - Group[1] + 4)/4);
        Group[3] := BBC((n - Group[1] - Group[2] + 3)/3);
        Group[4] := BBC((n - Group[1] - Group[2] - Group[3] + 2)/2 + 1);
        Group[5] := n - Group[1] - Group[2] - Group[3] - Group[4] + 4;
    end;
end;

begin
    ReturnCode:=0;
{Построение матрицы Matrix со значениями 'Л','Х','Р' по диалогу
с транзитивным замыканием}
    for c := 1 to GlobNGoal do begin
        for k := 1 to GlobNGoal do
            Matrix[c,k] := ' ';
        Matrix[c,c] := 'P';
    end;
    FormChoiceMatrix(Matrix, GlobNGoal, Trans_);
    if ReturnCode=0 then begin
        {Построение упорядоченного списка номеров }
        LinearOrdering;
        {Построение индикационного списка равнозначности - требуется только
для использования последующими программами}
        MakeRavnoznachList;
        DefineGroup(GlobNGoal);
    end;
    DeskTop^.Delete(WinL);
    DisPose(WinL, Done);
    DeskTop^.Delete(WinR);
    DisPose(WinR, Done);
end;
{=====
}
end.
```

(* ##### Дискретная шкала #####
Построение матрицы Саати степеней предпочтения.
Входные данные:
Matrix - матрица со значениями 'Л','Х','Р' с транзитивным замыканием.
Выходные данные:
Saatimatrix - матрицы Саати степеней предпочтения;
IndexAr - упорядоченный тернарный список номеров сравниваемых объектов
(в порядке уменьшения предпочтения);
Ravnoznach - индикационный список равнозначности (требуется только
для использования последующими программами).
Пример:
IndexAr = (2, 3, 1, 5, 4, 6);
Ravnoznach = ('1', '1', ' ', ' ', ' ', '2', '2'),
что означает: 2 = 3 > 1 > 5 > 4 = 6. *)
unit GoalPref;
INTERFACE
uses Objects, Views, App, Dialogs, MsgBox,
 Goal_Var, WinRect1, Dial5But, ConstAll;
{=====}

procedure DefineStepen(gr : byte);
{Определение степеней предпочтения для подгруппы целей}

IMPLEMENTATION

CONST
 NPref = 6;
 PrefDegreeNumb : array[0..NPref-1] of byte = (1, 3, 4, 5, 7, 9);

VAR

```

R:TRect;
WinL, WinR: PWinRect;
Control: word;
{=====
procedure FormSaatiMatrix;
var c, k : byte;
Dial5B: PDial5But;
{-----
procedure DefineOcenki(trow, tcol : byte);
var r1, c, i : byte;
rch : char;
kk,Control: word;
begin
  ReturnCode:=0;
  r1 := IndexAr[trow]; c := IndexAr[tcol];
  rch := Matrix[r1, c];
{-----
  if rch = 'X' then begin
    DeskTop^.GetExtent(R);
    R.B.X := R.B.X div 2;
    R.B.Y := R.B.Y div 2;
    WinL := New(PWinRect, Init(R, c, Safe[c]));
    DeskTop^.GetExtent(R);
    R.A.X := R.B.X div 2;
    R.B.Y := R.B.Y div 2;
    WinR := New(PWinRect, Init(R, r1, Safe[r1]));
    DeskTop^.Insert(WinL);
    DeskTop^.Insert(WinR);
  end else begin
    DeskTop^.GetExtent(R);
    R.B.X := R.B.X div 2;
    R.B.Y := R.B.Y div 2;
    WinL := New(PWinRect, Init(R, r1, Safe[r1]));
    DeskTop^.GetExtent(R);
    R.A.X := R.B.X div 2;
    R.B.Y := R.B.Y div 2;
    WinR := New(PWinRect, Init(R, c, Safe[c]));
    DeskTop^.Insert(WinL);
    DeskTop^.Insert(WinR);
  end;
  kk:=0;
  for i:=0 to NPref-1 do begin
    if SaatiMatrix[trow, tcol] = PrefDegreeNumb[i] then begin
      kk:=i;
      break
    end;
  end;
  Dial5B^.SetData(kk);
  if (trow=1) and (tcol=2) then
    Dial5B^.DisableCommands([cmStepBack])
  else
    Dial5B^.EnableCommands([cmStepBack]);
  Control := DeskTop^.ExecView(Dial5B);
  if Control=cmCancel then
    if MessageBox(^C'Тест еще не закончен!'+  

      ^M^C'Вы действительно хотите прервать тест?',nil,  

      mfConfirmation+mfYesButton+mfNoButton) <> cmYes then
      Control:=cmOk
    else
      ReturnCode:=1;
  if Control=cmOk then
    begin
      Dial5B^.GetData(kk);
      SaatiMatrix[trow, tcol] := PrefDegreeNumb[kk];
      SaatiMatrix[tcol, trow] := 1/PrefDegreeNumb[kk];
      if (trow>=(NGoal-1)) and (tcol>=NGoal) then
        if MessageBox(^C'Подтвердите правильность последнего ответа'+^M^C  

          '(Но-шаг назад)',nil,mfConfirmation+mfYesButton+mfNoButton) <> cmYes then
          Control:=cmStepBack;
    end;
  if Control=cmStepBack then
    ReturnCode:=5;
  DeskTop^.Delete(WinL);
}

```

```

DisPose(WinL, Done);
DeskTop^.Delete(WinR);
DisPose(WinR, Done);
end; {DefineOcenki}

begin
  Dial5B:=New(PDial5But, Init);
  for c := 1 to NGoal do
    for k := 1 to NGoal do
      if c=k then SaatiMatrix[c,c] := 1 else SaatiMatrix[c,k] := 0;
  for c := 1 to NGoal-1 do
    for k := c+1 to NGoal do
      if Matrix[IndexAr[c],IndexAr[k]] = 'P' then begin
        SaatiMatrix[c,k] := 1; SaatiMatrix[k,c] := 1
      end else begin
        DefineOcenki(c, k);
        if ReturnCode=1 then Exit;
        if ReturnCode=5 then begin {шаг назад}
          { Dial5B^.SelectNext(True); }
          dec(k,2);
          if k<c then begin
            dec(c);
            k:=NGoal-1;
            if c<1 then begin
              c:=1;
              k:=c+1;
            end;
            end;
          end;
        end;
      end;
    end; { FormSaatiMatrix }
{=====
procedure DefineStepen(gr : byte);
{Определение степеней предпочтения для подгруппы целей}
procedure ProceedToNextGroup;
var n : byte;
begin
  {Пересылка в начало списка Goals названий подцелей очередной группы}
  if gr = 1 then
    begin
      for n := 1 to NGoal do
        begin
          IndexAr[n] := GlobIndexAr[n];
          Ravnoznach[n] := GlobRavnoznach[n];
        end;
      LastNGoal := NGoal;
      Exit;
    end;
  for n := 1 to NGoal do
    begin
      IndexAr[n] := GlobIndexAr[n + LastNGoal - 1];
      Ravnoznach[n] := GlobRavnoznach[n + LastNGoal - 1];
    end;
  if Ravnoznach[2] = ' ' then
    Ravnoznach[1] := ' ';
  if Ravnoznach[NGoal-1] = ' ' then
    Ravnoznach[NGoal] := ' ';
  LastNGoal := LastNGoal + NGoal - 1;
end;

begin
  ReturnCode := 0;
  ProceedToNextGroup;
  FormSaatiMatrix;
  if Control=cmCancel then
    ReturnCode := 1;
end;
end.

unit ReadGoal;
interface

```

```

Uses Goal_Var, ConstAll;
procedure ReadGoalFile;

Implementation

procedure ReadGoalFile;
var FGoals : text;
    rs : (*string*)real;
begin
  Assign(FGoals, 'Goals.dat');
  {$I-}
  Reset(FGoals);
  {$I+}
  if IOResult <> 0 then
    begin WriteLn('Отсутствует файл целей. Нажмите Enter.');?>
  readln(FGoals);
  readln(FGoals);
  readln(FGoals);
  NRect := 0;
  while not Eof(FGoals) do
    begin
      readln(FGoals, rs);
      if rs <> 0.0 then
        begin
          NRect := NRect + 1;
          {rs := Trim(rs);}
          Safe(*Goals*)[NRect] := (rs+0.5)/3.4;
(*           if Length(rs) > NameLen then
             NameLen := Length(rs);*)
        end; {rs <> ','}
      end; { Eof }
      Close(FGoals);
      NGoal:=NRect; {== для эксперимента (без разбиения на подгруппы)}
      GlobNGoal:=NRect;
    end;
end.

```

```

unit TextWin;
interface
uses Objects, Views,
      ViewText,
      GVision;
type
  PTextWindow = ^TTextWindow;
  TTextWindow = object(TWindow)
    constructor Init(R: TRect; var FileName: fNameStr; WNumber, aHelpCtx: word);
  end;

implementation

{TTextWindow}
constructor TTextWindow.Init;
begin
  TWindow.Init(R, 'после просмотра результатов ЗАКРОЙТЕ, пожалуйста, окно'{FileName},
               WNumber);
  Options := Options or ofTileable or ofBuffered2;
  HelpCtx:=aHelpCtx;
  GetExtent(R);
  R.Grow(-1, -1);
  Insert(New(PFileViewer, Init(R,
                                StandardScrollBar(sbHorizontal + sbHandleKeyboard),
                                StandardScrollBar(sbVertical + sbHandleKeyboard), filename)));
end;
end.

```

```

unit EnterNam;
interface
uses

```

```

Objects, Views, App, Dialogs, MsgBox,
ConstAll,ExpNColl;

procedure CaseName;
function FilNam:String;

implementation

procedure EnterName;
var Dial: PDialog;
  R: TRect;
begin
  R.Assign(0,0,40,7);
  Dial:=New(PDialog, Init(R, 'Имя эксперта'));
  Dial^.Options := Dial^.Options or ofCentered;
  R.Assign(15,4,23,6);
  Dial^.Insert(New(PButton, Init(R, 'ВВОД', cmOk, bfDefault)));
  R.Assign(2,2,37,3);
  Dial^.Insert(New(PInputLine, Init(R, 35)));
  Dial^.SetData(ExpertName);
  if DeskTop^.ExecView(Dial) = cmOk then
    Dial^.GetData(ExpertName);

end;

procedure CaseName;
const fl='ExpLst.dat';
var Dial: PDialog;
  S:PDosStream;
  LB:PExpNListBox;
  Rec:record Coll:PStrColl; Sel: Integer end;
  R: TRect;
  Ctrl,MaxVal,i:word;
  S4:string[4];
  F:text;

procedure Max(Item:Pointer); far;
var Code: Integer;
  Valu:word;
begin
  Val(Copy(PString(Item)^,1,4),Valu,Code);
  if (Code=0) and (Valu>MaxVal) then MaxVal:=Valu;
end;

begin
  R.Assign(0,0,39,20);
  Dial:=New(PDialog, Init(R, 'Имя эксперта'));
  Dial^.Options := Dial^.Options or ofCentered;
  R.Assign(18,17,28,19);
  Dial^.Insert(New(PButton, Init(R, 'ВЫБОР', cmOk, bfDefault)));
  R.Assign(3,17,16,19);
  Dial^.Insert(New(PButton, Init(R, 'Новое имя', cmYes, bfNormal)));
  R.Assign(1,1,37,16);
  New(LB, Init(R));
  New(Rec.Coll,Init(100,10));
  Assign(F, fl);
  {$I-}
  ReSet(F);
  {$I+}
  if IOResult <>$ 0 then begin
    Rewrite(F);
    Close(F);
    S:=New(PDosStream,Init(f1,stOpenWrite));
    Rec.Coll^.Store(S^);
    DisPose(S,Done);
  end else begin
    Close(F);
    S:=New(PDosStream,Init(f1,stOpenRead));
    Rec.Coll^.Load(S^);
    DisPose(S,Done);
  end;
  Dial^.Insert(LB);
  Dial^.Insert(LB^.VScrollBar);

```

```

Rec.Sel:=0;
Dial^.SetData(Rec);
Case DeskTop^.ExecView(Dial) of
cmOk: begin
  Dial^.GetData(Rec);
  if Rec.Coll^.Count=0 then ExpertName:=' ' else
    ExpertName:=PString(Rec.Coll^.Items^[Rec.Sel])^;
end;
cmYes: begin
  repeat
    EnterName;
    Ctrl:=MessageBox(^C'Имя эксперта:' + ^M^C+ExpertName
      + ^M^C'подтверждаете?',nil, mfConfirmation + mfYesNoCancel)
    until Ctrl<>cmNo;
  if Ctrl=cmYes then begin
    MaxVal:=0;
    Rec.Coll^.ForEach(Max);
    Str(MaxVal+1:4,S4);
    for i:=1 to 3 do if S4[i]=' ' then S4[i]:='0';
    ExpertName:=S4+ExpertName;
    Rec.Coll^.Insert(NewStr(ExpertName));
    S:=New(PDosStream,Init(f1,stOpenWrite));
    Rec.Coll^.Store(S^);
    DisPose(S,Done);
  end;
end;
end;
end;

function FilNam:String;
var FN:String[6];
begin
  if FileNazva<>' ' then
    FilNam:=FileNazva
  else begin
    if ExpertName=' ' then CaseName;
    Str(Safe[1]:6:4, FN);
    FilNam := Copy(FN,3,4) + Copy(ExpertName,1,4) + '.dat';
  end;
end;
end.

```

```

unit CaseAlg;
interface

uses ConstAll, EnterNam, Goal_Var, RezColl;

function CaseAlgor(case_:byte):byte;

implementation

function CaseAlgor(case_:byte):byte;
var F: text;
  RezCo: PRezColl;
  st:string;
  NofMatr,bt,maxbt:byte;
  code,i,j,N,nF:integer;
  nFO:boolean;
  S7:string[7];
begin
  Assign(F,FilNam);
  {$I-}
  Reset(F);
  {$I+}
  if IOResult <> 0 then begin
    if case_<NMETHOD then CaseAlgor:=1 else CaseAlgor:=0;
  end else begin
    if case_<NMETHOD then begin
      RezCo:=New(PRezColl, Init(15,5));
      RezCo^.Initialize(F);
      if (RezCo^.Count=0)or(PRez(RezCo^.At(0))^ .Rec.Num<>1) then

```

```

CaseAlgor:=1
else
  if RezCo^.Count=1 then
    CaseAlgor:=2
  else begin
    I := 0;
    while (I < RezCo^.Count-1) and
      ((PRez(RezCo^.At(I))^^.Rec.Num=PRez(RezCo^.At(I+1))^^.Rec.Num) or
       (PRez(RezCo^.At(I))^^.Rec.Num=PRez(RezCo^.At(I+1))^^.Rec.Num-1)) do
      Inc(I);
    if I < RezCo^.Count-1 then
      CaseAlgor:=PRez(RezCo^.At(I))^^.Rec.Num+1
    else
      if PRez(RezCo^.At(RezCo^.Count-1))^^.Rec.Num+1<=NMethod then
        CaseAlgor:=PRez(RezCo^.At(RezCo^.Count-1))^^.Rec.Num+1
      else begin {Second Path}
        I:=0; N:=0;
        repeat
          inc(N);
          nF:=0; nF0:=false;
          while (I < RezCo^.Count) and (PRez(RezCo^.At(I))^^.Rec.Num=N) do
            begin {по одному номеру алгоритма}
              if (PRez(RezCo^.At(I))^^.Rec.Rez='') or(PRez(RezCo^.At(I))^^.Rec.Rez=FailMess) then
                inc(nF)
              else
                nF0:=true;
              inc(I)
            end;
            if nF0 then nF:=0;
          until (I>= RezCo^.Count) or (nF=1);
        if nF=1{I<RezCo^.Count} then
          CaseAlgor:=N
        else begin {Third path}
          I:=0; N:=0;
          repeat
            inc(N);
            nF:=0; nF0:=false;
            while (I < RezCo^.Count) and (PRez(RezCo^.At(I))^^.Rec.Num=N) do
              begin {по одному номеру алгоритма}
                if (PRez(RezCo^.At(I))^^.Rec.Rez='') or(PRez(RezCo^.At(I))^^.Rec.Rez=FailMess) then
                  inc(nF)
                else
                  nF0:=true;
                inc(I)
              end;
              if nF0 then nF:=0;
            until (I >= RezCo^.Count) or (nF=2);
          if nF=2{I<RezCo^.Count} then
            CaseAlgor:=N+NMethod+NMatrix
          else
            CaseAlgor:=0
          end;
        end;
      end;
    end;
  end else begin
    NofMatr:=0;
    while not EoF(F) do begin
      bt:=0;
      While not EOF(F) and (bt<>case_) do begin
        ReadLn(F,st);
        if st[1]='#' then Val(Copy(st,2,2),bt,code);
      end;
      if bt=Case_ then begin
        ReadLn(F);
        for j := 1 to NRect do begin
          for i := 1 to NRect do begin
            read(F, S7); {(7:3) - возможно без пробелов}
            val(S7, SaatiMatrix[j,i], code);
          end;
          readln(F);
        end;
      end;
    end;
  end;
end;

```

```

        inc(NofMatr);
    end;
end;{while}
CaseAlgor:=NofMatr;
end;
Close(F);
end;
end;
end.



---


unit WriteRez;

interface

uses
  Dos, MsgBox, ConstAll, Goal_Var, EnterNam;

procedure WriteInform(test:byte;start:boolean);

implementation

procedure WriteInform(test:byte;start:boolean);
var F : text;
  i, j, n : byte;
  rs,st : string;
  ch : char;
  SumKoef,SumSafe:real;
{=====
function LeadingZero(w : Word) : String;
var
  s : String;
begin
  Str(w:0,s);
  if Length(s) = 1 then
    s := '0' + s;
  LeadingZero := s;
end;

procedure WriteTimeStamp;
var y, mon, d, dow, h, m, s, hund : Word;
begin
  GetDate(y,mon,d,dow);
  GetTime(h,m,s,hund);
  Write(F, 'Дата: ', LeadingZero(d), '/',LeadingZero(mon), '/', y:0, ' ',
        LeadingZero(h), ':',LeadingZero(m), ':',LeadingZero(s));
end;

begin
  Assign(F, FilNam);
  {$I-}
  Append(F);
  {$I+}
  if IOResult <> 0 then begin
    Rewrite(F);
    Writeln(F, 'Эксперт: ', copy(ExpertName,5,Length(ExpertName)-4));
  { for i:=1 to NRect do
    write(F, Safe[i]:8:4);
    writeln(F)}
  end;
  if start or (test in [NMethod+1..NMethod+NMatrix]) then begin
    write(F,'#',test:2,' ');
    WriteTimeStamp;
    write(F, '+NmTst[test]');
    if start then begin
      if FeedBack then Write(F, ' С обратной связью') else Write(F,'БЕЗ обратной связи');
    end;
    WriteLn(F);
  end;
  if not start or (test in [NMethod+1..NMethod+NMatrix]) then begin
    WriteLn(F, 'вопросов:',NofQ:3{c 10 поз.},'; время:',TimeQ:5{c 21 поз.}, 'сек.;');
    case test of

```

```

1..NMethod:
if ReturnCode=2 then
  writeln(F,FailMess)
else begin
  SumKoef:=0.0;
  writeln(F,'Ненормированные значения весов:');
  for i := 1 to NRect do begin
    write(F, koef[i]:8:4);
    SumKoef:=SumKoef+koef[i];
  end;
  writeln(F);
  writeln(F,'Нормированные значения весов:');
  for i := 1 to NRect do
    write(F, koef[i]/SumKoef:8:4);
  writeln(F);
  if FeedBack then begin
{stud}writeln(F,'Результирующая матрица:');
    for j := 1 to NRect do begin
      for i := 1 to NRect do
        write(F, SaatiMatrix[j,i]:7:3);
      writeln(F);
    end;
  end;
  writeln(F,'Абсолютные ошибки определения коэффициентов значимости:');
{сумма для нормировки}
  SumKoef:=0;
  SumSafe:=0;
  for i:=1 to NRect do begin
    SumKoef:=SumKoef+koef[i];
    SumSafe:=SumSafe+Safe[i];
  end;
  for i:=1 to NRect do
    write(F, Safe[i]/SumSafe-koef[i]/SumKoef:8:4);
  writeln(F);{student}
  writeln(F,'Относительные ошибки определения коэффициентов значимости:');
  for i:=1 to NRect do
    write(F, (Safe[i]/SumSafe-koef[i]/SumKoef)*SumSafe/Safe[i]*100:7:2,'%');
  writeln(F);{student}
  str(test:2,st);
  if not _FB then
    MessageBox('С'Метод '+st+' успешно завершил работу. Все данные записаны
      в файл результатов.', nil, mfInformation + mfOkButton);
  end;
NMethod+1..NMethod+NMatrix:
  for j := 1 to NRect do begin
    for i := 1 to NRect do
      write(F, SaatiMatrix[j,i]:7:3);
    writeln(F);
  end;
end;{case}
NofQ:=0;
TimeQ:=0; {обнулили счетчик секунд}
end;
Close(F);
end;
end.

```

```

unit Na_Input;
interface

uses App, Objects, Views, MsgBox,
  Goal_Var, WinRect1, Dial_Na, ConstAll;

procedure InputMatrNa;

implementation

procedure InputMatrNa;
var R:Trect;
  WinL,Win:PWinRect;
  DialNa:PDialNa;
  i,j:integer;

```

```

Control: word;
begin
  ShowAlg^.Ftext:='Ввод матр.аддит.сравнений';
  DialNa:=New(PDialNa,Init);
  for j:=1 to NRect do for i:=1 to NRect do {нач. заполнение матрицы "Саати"}
    SaatiMatrix[j,i]:=0;
  for j:=1 to NRect do begin {заполнение матрицы "Саати"по строкам}
    for i:=1 to NRect do begin {заполнение матрицы "Саати"по столбцам}
      if i<>j then begin
        if (j=1)and(i=2) then
          DialNa^.DisableCommands([cmStepBack])
        else
          DialNa^.EnableCommands([cmStepBack]);
        DeskTop^.GetExtent(R);
        R.B.X := R.B.X div 2;
        R.B.Y := R.B.Y div 2;
        WinL := New(PWinRect, Init(R, i, Safe[i]));
        DeskTop^.Insert(WinL); {левое}
        DeskTop^.GetExtent(R);
        R.B.X := R.B.X div 2;
        R.B.Y := R.B.Y div 2;
        R.Move(R.B.X,0);
        Win := New(PWinRect, Init(R, j, Safe[j]));
        DeskTop^.Insert(Win); {правое}
        DialNa^.SetData(SaatiMatrix[j,i]);
        Control := DeskTop^.ExecView(DialNa);{на сколько прав(j) > лев(i)}
        DeskTop^.Delete(WinL);
        DisPose(WinL, Done);
        DeskTop^.Delete(Win);
        DisPose(Win, Done);
      if Control=cmCancel then
        if MessageBox(^C'Тест еще не закончен!'+  

          ^M^C'Вы действительно хотите прервать тест?',nil,  

          mfConfirmation+mfYesButton+mfNoButton) <> cmYes then
          Control:=cmOk
        else
          ReturnCode:=1;
      if Control=cmOk then begin
        DialNa^.GetData(SaatiMatrix[j,i]);
        ReturnCode:=0;
        if (i>=NRect-1)and(j>=NRect) then
          if MessageBox(^C'Подтвердите правильность последнего ответа'+^M^C  

            '(Но-шаг назад)',nil,mfConfirmation+mfYesButton+mfNoButton) <> cmYes then
            dec(i);
      end;
      if Control=cmStepBack then begin
        DialNa^.SelectNext(True);
        if i-1=j then dec(i);
        dec(i,2);
        if i<0 then begin
          dec(j);
          i:=NRect-1;
          if j<1 then begin
            j:=1;
            i:=1;
          end;
        end;
        if ReturnCode=1 then Break;
      end;
    end;
    if ReturnCode=1 then Break;
  end;
end;
end;

```

```

unit Dial_Na;

interface
uses
  Objects, Views, Dialogs, Drivers, MsgBox, Validate, ConstAll;
```

```

type
  PDialNa = ^TDialNa;
  TDialNa = object(TDialog)
    constructor Init;
    procedure HandleEvent(var Event: TEvent); virtual;
  end;

implementation

type
  PMyValidator = ^TMyValidator;
  TMyValidator = object(TPXPictureValidator)
    function IsValid(const S: string): Boolean; virtual;
    procedure Error; virtual;
  end;

  PILRealType = ^TILRealType;
  TILRealType = object(TInputLine)
    Field: pointer;
    constructor Init(var Bounds: TRect; AMaxLen: Integer; Ctx: word);
    function DataSize: Word; virtual;
    procedure GetData(var Rec); virtual;
    procedure SetData(var Rec); virtual;
  end;

  function TMyValidator.IsValid(const S: string): Boolean;
  var code:integer;
    RealVar: real;
  begin
    Val(S,RealVar,code);
    if inherited IsValid(S) and (code=0) and (RealVar>=-1.0) and (RealVar<=1.0) then
      IsValid:=true
    else
      IsValid:=false;
  end;
  procedure TMyValidator.Error;
  begin
    MessageBox(^С'Некорректные данные !',
              +^М^С'Повторите ввод', nil, mfError + mfOkButton);
  end;

  constructor TILRealType.Init(var Bounds: TRect; AMaxLen: Integer; Ctx: word);
  begin
    inherited Init(Bounds,AMaxLen);
    SetValidator(New(PMyValidator, Init('[-][#][{.[#][#][#]}]', True)));
    HelpCtx := Ctx;
  end;
  function TILRealType.DataSize: Word;
  begin
    DataSize:=sizeof(real);
  end;
  procedure TILRealType.GetData(var Rec);
  var code:integer;
  begin
    val(Data^,real(Rec),code);
  end;
  procedure TILRealType.SetData(var Rec);
  begin
    str(real(Rec):5:3,Data^);
    while copy(Data^,1,1)='.' do
      Data^:=copy(Data^,2,4);
  end;

  constructor TDialNa.Init;
  var
    Bounds: TRect;
    C1: PInputLine;
  begin
    Bounds.Assign(0,0,32,10);
    Bounds.Move(46,14);
    inherited Init(Bounds, 'Определите');
    HelpCtx:=hcNoContext;
    { Options := Options or ofCentered; }
  end;

```

```

Flags := 0;
GetExtent(Bounds);
Bounds.Assign(1,1,Bounds.B.X-1,4);
Insert(New(PStaticText, Init(Bounds, ^С'На сколько степень заполнения'+  

    '{нижнего}' 'правого изображения больше'+  

    'степени заполнения левого')));
Bounds.Assign(0,5,8,6);
C1:= New(PILRealType, Init(Bounds, 6, hcNaInp));
C1^.Options := C1^.Options or ofCenterX;
Insert(C1);
Bounds.Assign(0,7,9,9);
Bounds.Move(4,0);
Insert(New(PButton, Init(Bounds, 'назад',cmStepBack,bfNormal)));
Bounds.Move(10,0);
Insert(New(PButton, Init(Bounds, 'ввод',cmOk,bfDefault)));
C1^.Select
end;

procedure TDialNa.HandleEvent(var Event: TEvent);
begin
  inherited HandleEvent(Event);
  if (Event.What = evCommand) and (Event.Command=cmStepBack) then begin
    EndModal(cmStepBack);
    ClearEvent(Event)
  end;
end;
end.

```

```

Unit NaN_Calc;
{#### Обработка метода "На" новая раздельно стр и столб ####}
interface

uses MsgBox,
      Goal_Var,Toc_Dial,Na_ReInput,ConstAll,SimpEst,ReEst;

procedure CalcNaN(Vd_1,Vsingl:boolean);
  {Vd=1 при Vd_1=true, Vd - оценивается при false}
  {если Vsingl=true то работаем по V (по строкам), иначе по W }

implementation

procedure CalcNaN(Vd_1,Vsingl:boolean);
const nk=11; {размерность дискретной шкалы распределения
              (количество дел. шкалы, до которого округл. знач. сил при выч. Ксог)}
  eps=0.1; {точность (эпсилон)}
type Real1Dim = array[1..MaxNGoal] of Real;
  Real2Dim = array[1..MaxNGoal] of Real1Dim;
  Byte1Dim = array[1..MaxNGoal] of byte;
  Byte2Dim = array[1..MaxNGoal] of Byte1Dim;
  Char1Dim = array[1..MaxNGoal] of char;
  Char2Dim = array[1..MaxNGoal] of Char1Dim;
  Bool1Dim = array[1..MaxNGoal] of boolean;
  Bool2Dim = array[1..MaxNGoal] of Bool1Dim;
  omass = array[1..nk] of Byte;

var Delta,
  VWsr, Domin, MinKsog, ds: real;
  ReInput: boolean;
  h,i,j,et,js,Crr: word;
  NDomin,NewDomin,NMinKsog,ForMove,WhereMove,N_ReInput: byte;
  soob: string;
  OurMatrix: glNpByNp; {OurMatrix - матрица весов альтернатив}
  VW {Точные значения распределения} :Real2Dim;
  RVW {Распределение: RVW[i,j]-позиция шкалы, где расположена
        "составляющая"Vij (Wij)} :Byte2Dim;
  Sets {'p'-predominant,'o'-own,'t'-transitive}:Char2Dim;
  IdxKoefSogl {индекс упорядочивающий Ксог по возрастанию} :Byte1Dim;
  R {Распределение в другом виде: R[i] - количество "палок"
     в i-й позиции шкалы} :omass;
  KoefSogl {Коэф. согласованности результирующие}:Real1Dim;
  MoveAble {true - перемещаемая "составляющая"; false - нет} :Bool2Dim;
  PredominantEst {оценки доминирующих элементов} :Real1Dim;

```

```

function SrednVW(Nraspr:byte;Tochn:boolean) : real;
var i:integer;
    sr:real;
begin
    sr:=0;
    for i:=1 to Ngoal do
        if Tochn then
            sr:=sr+VW[Nraspr,i]
        else
            sr:=sr+RVW[Nraspr,i];
    sr:=sr/NGoal;
    SrednVW:=sr;
end;

Procedure Ksogl(var k : real;
                  x,y:integer
                  ;
                  R : omass);
{ Процедура расчета коэффициента согласования.
  Исходные данные:
    R - массив, задающий текущее распределение;
    x - количество делений шкалы;
    у - текущее значение количества путей
    (экспертов);
  Выходные данные:
    k - коэффициент согласия;}
Var
    sp,lebzn,npznn,prr,lebr,d:real;
    i,nd:integer;
    k_is_0, ravn: boolean;

begin
    nd:=0;
    for i:=1 to x do {x-число делений шкалы}
        if (R[i]>0) then inc(nd);
    { nd - число делений шкалы, названных в качестве оценок
      (количество элементов множества значений сил) }
    d:=0.0;
    if nd>1 then
        d:=(x-1)/(nd-1);
    if (Frac(d)>0.0) or not((R[1]>0) and (R[x]>0)) then
        k_is_0 := false { необходимое условие отличия от 0 ksogl }
    else begin
        ravn:=True; { проверка равномерности распределения оценок}
        for i:=1 to (x-Trunc(d)) do
            ravn:=ravn and (R[i]=R[i+Trunc(d)]);
        if ravn then
            k_is_0 := true
        else
            k_is_0 := false
    end;
    if not k_is_0 then begin
        npznn:=ln(x);
        sp:=0.0;
        for i:=1 to x do
            sp:=sp+i*R[i]/y;
        if y = 1 then begin k:=1; exit end; {одна составляющая}
        lebzn:=0.0;
        for i:=1 to x do
            lebzn:=lebzn+Abs(i-(x+1)/2.0)*y/(x*ln(x)*ln(y));
        prr:=0.0;
        for i:=1 to x do
            if R[i]>0 then prr:=prr+R[i]/y*ln(R[i]/y);
        lebr:=0.0;
        for i:=1 to x do
            lebr:=lebr+R[i]/y*Abs(i-sp);
        k:=(lebr-prr)/(lebzn+npznn);
    end else
        k:=0;
end;{Sogl-----}

Procedure ForCalcKsog(var Rinp :Byte1Dim; var R:omass);
{ Процедура преобразования представления распределения

```

```

Входные данные: Rinp - массив, задающий текущее распределение;
Выходные данные: R - =====///===== в другом формате }

Var
    i:integer;
begin
    {### Преобразование Rinp в R ###}
    for i:=1 to nk do R[i]:=0;
    for i:=1 to NGoal do
        if Rinp[i]<>0 then inc(R[Rinp[i]]);
    {#####
    end;{ForCalcKsog-----}

begin
    {##### ИНИЦИАЛИЗАЦИЯ #####}
    for i:=1 to NGoal do
        PredominantEst[i]:=0.0;
    {Delta-Приращение, обеспечивающее движение "составляющие" при улучшении распред.-я}
    Delta:={0.51*}eps;
    {##### Находим Тобн и Тприм для шкалы #####}
    { - порога обнаружения }
    et:=round(0.5 * nk + 1);
    for js := 1 to nk do begin
        R[js]:=1;
        if js = et then R[js]:=2;
    end;
    R[1]:=0;
    Ksogl(kobn,nk,nk,R);
    { - порога применения }
    for js := 1 to nk do R[js]:=0;
    R[et-1]:=1;
    R[et]:=1;
    Ksogl(kpr,nk,2,R);
    OurMatrix:=SaatiMatrix; {OurMatrix - матрица весов альтернатив}
    MinKsog:=0; {поэтому первый раз всегда зайдем в цикл}
    {##### Инициализация (все составл. перемещаемые) Когда это делать ?###}
    for i:=1 to Ngoal do for j:=1 to Ngoal do
        begin MoveAble[i,j]:=True; RVW[i,j]:=255; VW[i,j]:=-1 end;

while (MinKsog < Kpr) and (ReturnCode<>1) do begin
    {##### Формирование распределений #####}
    for i:=1 to NGoal do begin { ЦИКЛ ПОРАСПРЕДЕЛЕНИЯ }
        N_ReInput:=0;
        repeat
            ReInput:=false;
            for j:=1 to NGoal do Sets[j,i]:=' ';{столб множества пустой}
            {### Доминирующий - min (max) элемент матрицы ###}
            if Vsingl then Domin:=1 else Domin:=-1;
            for h:=1 to NGoal do
                if Vsingl then begin
                    if OurMatrix[i,h]<Domin then begin {min}
                        Domin:=OurMatrix[i,h];
                        NDomin:=h
                    end
                end else begin
                    if OurMatrix[h,i]>Domin then begin {max}
                        Domin:=OurMatrix[h,i];
                        NDomin:=h
                    end;
                end;
            {## Доминирующему элементу присваивается значение 1 ##}
            VW[NDomin,i]:=1;
            {## Доминирующий элемент оценивается если еще не оценивался ##}
            if PredominantEst[NDomin]<>0.0 then begin
                if not Vd_1 then
                    VW[NDomin,i]:=PredominantEst[NDomin];
                Sets[NDomin,i]:='p'; { predominant (доминирующая) }
            end else begin
                if not Vd_1 then
                    Simple_Estimate(NDomin,VW[NDomin,i]);
                PredominantEst[NDomin]:=VW[NDomin,i];
            end;
            for h:=1 to NGoal do {по всех домин. элементах }
                if (Vsingl and (OurMatrix[i,h]=Domin)) or (not Vsingl
                    and (OurMatrix[h,i]=Domin)) then begin

```

```

        Sets[h,i]:='p'; { predominant (доминирующая) }
        VW[h,i]:=VW[NDomin,i];
        PredominantEst[h]:=VW[NDomin,i];
        end;
      end;
      {### Определяем Vii (Wjj) собственные значения ###}
      if Vsingl then
        VW[i,i]:=VW[NDomin,i]+OurMatrix[i,NDomin]
      else
        VW[i,i]:=VW[NDomin,i]-OurMatrix[NDomin,i];
      if Sets[i,i]<>'p' then Sets[i,i]:='o'; {own (собственные)}
      {### Остальные V (W)-транзитивные определяем относительно Vii (Wjj) ###}
      for j:=1 to NGoal do begin
        if Sets[j,i]=' ' then begin
          Sets[j,i]:='t';
          if Vsingl then
            VW[j,i] := VW[i,i] - OurMatrix[i,j]
          else
            VW[j,i] := VW[i,i] + OurMatrix[j,i];
        end;
        if VW[j,i]<0 then begin
          ReInput:=true;
          PredominantEst[NDomin]:=0.0; {чтобы спрашивать повторно}
          break {for j}
        end;
        {избавимся от потери значимости round(VW[j,i] * 100000)/100000}
        RVW[j,i]:=round(round(VW[j,i] * 100000)/10000);
        (*RVW[j,i]:=round(VW[j,i]/eps); { Округляем, до ближайш.знач. шкалы
           иначе оставляем прежнюю составл.}*)
      end;
      if ReInput then begin
        if not FeedBack then begin ReturnCode:=2; Exit end;
        MessageBox('C+'+'Обнаружена несогласованность.'
        +'M^C'+'Вам нужно повторно ввести некоторые данные.','
        nil, mfInformation + mfOkButton);
        for j:=1 to NGoal do begin
          if ReturnCode<>0 then Break;
          if i<>j then
            if Vsingl then DialReInput(i,j,OurMatrix[i,j]) {вводить i-ю строку заново}
            else DialReInput(j,i,OurMatrix[j,i]);{вводить j-й столбец заново}
        end;
        Inc(N_ReInput)
      end
      else
        break;
    until N_ReInput>2;
    if ReInput then begin
      if N_ReInput>2 then
        break {for i}
      else
        dec(i);
    end;
  end;{for i}

  if not FeedBack then minKsog:=1.0 {чтобы сразу выйти из цикла} else begin
    if N_ReInput<=2 then begin
      {#### Мы должны работать со всеми распределениями одновременно !!!!
      ##### Формируем индексный массив (упорядоч. Ksog по возрастанию )###}
      For i:=1 to NGoal do begin
        ForCalcKsog(RVW[i],R);
        Ksogl(Ksog,nk,NGoal,R);
        KoefSogl[i]:=Ksog;
        IdxKoefSogl[i]:=lo(i);
      end;
      For i:=1 to NGoal-1 do
        For j:=i+1 to NGoal do
          if KoefSogl[IdxKoefSogl[i]] > KoefSogl[IdxKoefSogl[j]] then begin
            h:=IdxKoefSogl[i];
            IdxKoefSogl[i]:=IdxKoefSogl[j];
            IdxKoefSogl[j]:=h;
          end;
      MinKsog:=KoefSogl[IdxKoefSogl[1]];
    end;
  end;

```

```

if KoefSogl[IdxKoefSogl[1]]<Kobn then begin
  MessageBox(^C+'Согласованность меньше допустимой.'
+^M^C'Вам нужно ввести некоторые данные снова.',
  nil, mfInformation + mfOkButton);
for j:=1 to NGoal do begin
  if ReturnCode<>0 then Break;
  if IdxKoefSogl[1]<>j then {чтобы исключить ввод диагональных элементов}
    if Vsingl then {вводить столбец заново}
      DialReInput(j, IdxKoefSogl[1], OurMatrix[j, IdxKoefSogl[1]])
    else { строку }
      DialReInput(IdxKoefSogl[1], j, OurMatrix[IdxKoefSogl[1], j]);
end;
Inc(N_ReInput);
end else begin

if KoefSogl[IdxKoefSogl[1]]<Kpr then begin
  repeat
    ##### Находим составл., наиболее отдаленную от ср.зн.
    - для передвижения (ForMove) #####
    ForMove:=0;
    WhereMove:=0;
    i:=1;
    while (i<=NGoal) and (KoefSogl[IdxKoefSogl[i]]<Kpr) and (Formove=0) do begin
      ##### Вычисляем среднее значение по RVW #####
      Vwsr:=SrednVW(IdxKoefSogl[i],false);
      ds:=0.5; {Max отклонение от ср.знач. в данном i-м распр-и (< 0.5 - не двигать)}
      for j:=1 to NGoal do begin
        if (Abs(Vwsr-RVW[IdxKoefSogl[i],j])>ds)and MoveAble[IdxKoefSogl[i],j] then
        begin
          ds:=Abs(Vwsr-RVW[IdxKoefSogl[i],j]);{искать по точным значениям ? - Нет}
          if (Sets[IdxKoefSogl[i],j]='p') {является домин.(predominant)} then begin
            ForMove:=j;
            WhereMove:=IdxKoefSogl[i];
            {на всякий случай найдем новое доминирующее значение }
            if Vsingl then Domin:=1 else Domin:=-1;
            for h:=1 to NGoal do
              if h<>WhereMove then {без текущего доминирующего}
                if Vsingl then begin
                  if OurMatrix[ForMove,h]<Domin then begin {min}
                    Domin:=OurMatrix[ForMove,h];
                    NewDomin:=h
                  end
                end else begin
                  if OurMatrix[h,ForMove]>Domin then begin {max}
                    Domin:=OurMatrix[h,ForMove];
                    NewDomin:=h
                  end;
                end;
            end;
          end;
        end;
        end;
        ForMove:=0
      end;
      inc(i);
    end;
    if Formove=0 then begin {predominants is absent}
      i:=1;
      while (i<=NGoal) and (KoefSogl[IdxKoefSogl[i]]<Kpr) and (Formove=0) do begin
        ##### Вычисляем среднее значение по RVW #####
        Vwsr:=SrednVW(IdxKoefSogl[i],false);
        ds:=0.5;{max откл.-я от ср.значения в данном i-м распр.(<0.5-не двигать)}
        for j:=1 to NGoal do begin
          if (Abs(Vwsr-RVW[IdxKoefSogl[i],j])>ds) and MoveAble[IdxKoefSogl[i],j]
          then begin
            ds:=Abs(Vwsr-RVW[IdxKoefSogl[i],j]);
            {может искать по точным значениям ? - Нет}
            if (Sets[IdxKoefSogl[i],j]='o')
              {является собственной (own)}then begin
                ForMove:=j;
                WhereMove:=IdxKoefSogl[i];
                {найдем домин.знач. для j-го столбца (строки)
                - пригодится при изменении матрицы}
                NDomin:=0;
                for h:=1 to NGoal do

```

```

        if (Sets[h,j]='p') then begin
            NDomin:=h;
            Break
        end;
        end else
            ForMove:=0
        end;
        inc(i);
    end;
end;
if Formove=0 then begin {owns is absent too}
    i:=1;
    while (i<=NGoal) and (KoefSogl[IdxKoefSogl[i]]<Kpr) and (Formove=0) do begin
        ##### Вычисляем среднее значение по RVW #####
        VWsr:=SrednVW(IdxKoefSogl[i],false);
        ds:=0.5; {макс отклонения от ср.значения в данном i-м распределении
                   (<0.5 -не двигать)}
        for j:=1 to NGoal do begin
            if (Abs(VWsr-RVW[IdxKoefSogl[i],j])>ds)
                and MoveAble[IdxKoefSogl[i],j] then begin
                    ds:=Abs(VWsr-RVW[IdxKoefSogl[i],j]);
                    {может искать по точным значениям ? - Нет}
                    if (Sets[IdxKoefSogl[i],j]='t')
                        {является транзитивной (transitive)}then begin
                            ForMove:=j;
                            WhereMove:=IdxKoefSogl[i];
                        end;
                    end;
                end;
            inc(i);
        end;
    end;
{#####
if ForMove>0 then begin
    Crr:=1; {1 вместо cmYes для диалога}
    {задавать вопрос}
    case Sets[WhereMove,ForMove] of
        'p': {для доминирующих} begin
            if Vsingl {and (VWsr<RVW[WhereMove,ForMove]) считаем, что домин всегда
                       больше среднего}then begin
                soob:=Uvel;{ нужно увеличить предпочтение le по срав с ri}
                Dial(Crr,WhereMove{ri},ForMove{le},soob,OurMatrix[ForMove{ri},
                           WhereMove{le}])
            end else begin
                soob:=Umen; { нужно уменьшить предпочтение le по срав с ri}
                Dial(Crr,ForMove{ri},WhereMove{le},soob,OurMatrix[WhereMove{le},
                           ForMove{ri}]);
            end;
        if Crr<>1 then begin
            if (VWsr>RVW[WhereMove,ForMove]) then
                soob:=Uvel { нужно увеличить непосредственно оцененное значение}
            else
                soob:=Umen;{ нужно уменьшить непосредств. оценку}
                Re_Estimate(Crr,soob,WhereMove,VW[WhereMove,ForMove]);
                PredominantEst[WhereMove]:=VW[WhereMove,ForMove];
                {tsv остальные дом. элементы? - не трогать}
            end;
        end;
        'o': {для собственных} begin
            if ( Vsingl and (VWsr<RVW[WhereMove,ForMove])) or
                (not Vsingl and not (VWsr<RVW[WhereMove,ForMove])) then
                soob:=Umen { нужно увеличить предпочтение le по срав с ri}
            else
                soob:=Uvel;{ нужно уменьшить предпочтение le по срав с ri}
            if Vsingl then
                Dial(Crr,NDomin{ri},ForMove{le},soob,OurMatrix[ForMove{ri},
                           NDomin{le}])
            else
                Dial(Crr,ForMove{ri},NDomin{le},soob,OurMatrix[NDomin{le},
                           ForMove{ri}]);
        end;
        't':{это для транзитивных составляющих} begin

```

```

if ( Vsingl and (VWsr<RVW[WhereMove,ForMove])) or
    (not Vsingl and not (VWsr<RVW[WhereMove,ForMove])) then
    soob:=Uvel { нужно увеличить предпочтение le по срав с ri}
else
    soob:=Umen;{ нужно уменьшить предпочтение le по срав с ri}
if Vsingl then
    Dial(Crr,WhereMove{ri},ForMove{le},soob,OurMatrix[ForMove{ri},
        WhereMove{le}])
else
    Dial(Crr,ForMove{ri},WhereMove{le},soob,OurMatrix[WhereMove{le},
        ForMove{ri}]);
end;
end;
if ReturnCode = 1 then exit;
if crr = 1 then begin { 1 вместо cmYes }
case Sets[WhereMove,ForMove] of
  'p': {для доминирующих} begin
    if Vsingl then {V}
        { увеличивать элемент OurMatrix}
        OurMatrix[ForMove,WhereMove]:=VW[NewDomin,ForMove]
        +OurMatrix[ForMove,NewDomin]-VW[NDomin,ForMove]+Delta
    else {W}
        { уменьшать элемент OurMatrix}
        OurMatrix[WhereMove,ForMove]:=-VW[NewDomin,ForMove]
        +OurMatrix[NewDomin,ForMove]+VW[NDomin,ForMove]-Delta;
  end;
  'o': {для собственных}
    if Vsingl then begin {V}
      if (VWsr<RVW[WhereMove,ForMove]) then
          {следовательно, уменьшать и элемент OurMatrix}
          OurMatrix[ForMove,NDomin]:=OurMatrix[ForMove,NDomin]-Delta
      else {двигать вправо(увеличивать V)}
          {следовательно, увеличивать и элемент OurMatrix}
          OurMatrix[ForMove,NDomin]:=OurMatrix[ForMove,NDomin]+Delta
    end else begin {W}
      if (VWsr<RVW[WhereMove,ForMove]) then
          {следовательно, увеличивать элемент OurMatrix}
          OurMatrix[NDomin,ForMove]:=OurMatrix[NDomin,ForMove]+Delta
      else {двигать вправо(увеличивать W)}
          {следовательно, уменьшать элемент OurMatrix}
          OurMatrix[NDomin,ForMove]:=OurMatrix[NDomin,ForMove]-Delta
    end;
    {здесь м.б. изменить остальные доминирующие элементы в строке, столбце}
  't':{для транзитивных}
    if Vsingl then begin {V}
      if (VWsr<RVW[WhereMove,ForMove]) then
          {следовательно, увеличивать элемент OurMatrix}
          OurMatrix[ForMove,WhereMove]:=OurMatrix[ForMove,WhereMove]+Delta
      else {двигать вправо(увеличивать V)}
          {следовательно, уменьшать элемент OurMatrix}
          OurMatrix[ForMove,WhereMove]:=OurMatrix[ForMove,WhereMove]-Delta
    end else begin {W}
      if (VWsr<RVW[WhereMove,ForMove]) then
          {следовательно, уменьшать и элемент OurMatrix}
          OurMatrix[WhereMove,ForMove]:=OurMatrix[WhereMove,ForMove]-Delta
      else {двигать вправо(увеличивать W)}
          {следовательно, увеличивать и элемент OurMatrix}
          OurMatrix[WhereMove,ForMove]:=OurMatrix[WhereMove,ForMove]+Delta
    end;
    end;
    { Break выход из цикла выбора составл. для перемещения}
  end else begin {если нет - помечаем как непередвигаемую}
    MoveAble[WhereMove,ForMove]:=false;
    if Sets[WhereMove,ForMove]='o'{для собственных} then
      MoveAble[NDomin,ForMove]:=false;
    end;
  end;
until (ForMove=0) or (Crr=1); {выход,если нечего двигать или по ДА}
end; {if Kсогл min < Тприм}
end;
end; {if N_ReInput<=2}
if (ForMove=0) or (N_ReInput>2) then begin
  { если нечего двигать - ForMove=0}

```

```

EndMsg;
for i:=1 to NGoal do KoefSogl[i]:=0.0;
Break
end
end{FeedBack}
end; {конец цикла While MinKsog<=Kpr and (ReturnCode<>1)}
if ReturnCode<>1 then begin
  for i:=1 to Ngoal do koef[i]:=SrednVW(i,true); {precisely}
end;
end;
end.

```

unit Toc_Dial;

INTERFACE

{для Graphic Vision}

Uses Objects, Views, App, MsgBox,

Goal_Var, ConstAll, WinRect1, St_Opposite;

Procedure Dial(var Crr:word; ri,le:integer;soob:string;old:real);

Procedure DialSt(var Crr:word; ri,le:integer;soob_Uvel:boolean;old:real;Oppos:boolean);

procedure EndMsg;

IMPLEMENTATION

```

{ для GraphicVision начало}
Procedure dial;
type recType = record
  ukStr,ukStr1: PString
end;
var WinL, WinR: PWinRect;
  recS: recType;
  rab:string;
  R:TRect;
begin
  DeskTop^.GetExtent(R);
  R.B.X := R.B.X div 2;
  R.B.Y := R.B.Y div 2;
  WinL := New(PWinRect, Init(R, le, Safe[le]));
  DeskTop^.GetExtent(R);
  R.A.X := R.B.X div 2;
  R.B.Y := R.B.Y div 2;
  WinR := New(PWinRect, Init(R, ri, Safe[ri]));
  DeskTop^.Insert(WinL);
  DeskTop^.Insert(WinR);

  R.Assign(0,0,32,13);
  R.Move(24,13);
  recS.UkStr:=soob;
  str(old:6:2,rab);
  recS.UkStr1:=rab;
  Crr := MessageBoxRect(R, ^C + 'ОБНАРУЖЕНЫ ПРОТИВОРЕЧИЯ',
    +' Согласны ли Вы %-10s'
    +' степень превосходства заполнения'
    +' левого изображения'
    +' по сравнению с заполнением правого ?'
    +' ^M^C+'( Прежнее значение степени'
    +' превосходства равно %-6s'),
    RecS, mfConfirmation + mfNoButton + mfYesButton);
  if Crr = cmCancel then ReturnCode := 1;
  if Crr = cmYes then Crr := 1 else Crr := 2;
  DeskTop^.Delete(WinL);
  DisPose(WinL, Done);
  DeskTop^.Delete(WinR);
  DisPose(WinR, Done);
end;

Procedure DialSt;
type recType = record
  ukStr,ukStr1: PString
end;

```

```

var WinL, WinR: PWinRect;
  recS: recType;
  soob,rab:string;
  sw:integer;
  R:TRect;
begin
  if soob_Uvel then soob:=Uvel else soob:=Umen;
  if old < 1 then begin
    if Oppos then old:=Opposite(old){для ST} else old:=1/old;{для IN}
    if not soob_Uvel then soob:=Uvel else soob:=Umen
  end else begin
    sw:=le; le:=ri; ri:=sw;
  end;

  DeskTop^.GetExtent(R);
  R.B.X := R.B.X div 2;
  R.B.Y := R.B.Y div 2;
  WinL := New(PWinRect, Init(R, le, Safe[le]));
  DeskTop^.GetExtent(R);
  R.A.X := R.B.X div 2;
  R.B.Y := R.B.Y div 2;
  WinR := New(PWinRect, Init(R, ri, Safe[ri]));
  DeskTop^.Insert(WinL);
  DeskTop^.Insert(WinR);

  R.Assign(0,0,50,13);
  R.Move(15,13);
  recS.UkStr:=soob;
  str(old:3:1,rab); rab:='['+rab+']';
  if abs(old - round(old))<0.0000001 then
    rab:=NmPr[round(old)]+rab+[9.9]
  else
    rab:='больше чем '+NmPr[trunc(old)]+', но меньше чем '+NmPr[trunc(old)+1]+rab;
  recS.UkStr1:=rab;
  Crr := MessageBoxRect(R, ^C+'ОБНАРУЖЕНЫ ПРОТИВОРЕЧИЯ'
    +^M' Согласны ли Вы %-9s'
    +' степень превосходства заполнения'
    +' левого изображения'
    +' по сравнению с заполнением правого ?'
    +' (' Прежнее значение степени'
    +' превосходства было %-0s )',
    RecS, mfConfirmation + mfNoButton + mfYesButton);
  if Crr = cmCancel then ReturnCode := 1;
  if Crr = cmYes then Crr := 1 else Crr := 2;
  DeskTop^.Delete(WinL);
  DisPose(WinL, Done);
  DeskTop^.Delete(WinR);
  DisPose(WinR, Done);
end;

procedure EndMsg;
  var R:TRect;
begin
  R.Assign(0,0,50,10);
  R.Move(15,6);
  MessageBoxRect(R, ^C+'Из-за имеющихся противоречий и невозможности,'+
    +' по Вашему мнению, изменить оценки степеней превосходства заполнения'+
    +' изображений - получить достоверную информацию Невозможно!'-
    +^M^C'Предлагаем попытаться выполнить этот метод познее еще раз',
    nil, mfinformation + mfOkButton);
  Returncode :=2
end;
end.

```

```

unit Na_ReInput;
interface

uses App, Objects, Views,
      Goal_Var, WinRect1, Dial_Na, ConstAll;

Procedure DialReInput(ri,le:integer;var rez:real);

```

```

implementation

procedure DialReInput;
var R:Trect;
    WinL,WinR:PWinRect;
    DialNa:PDialNa;
    i,j:integer;
    Control: word;
begin
    DialNa:=New(PDialNa,Init);
    DialNa^.DisableCommands([cmStepBack]);
    DeskTop^.GetExtent(R);
    R.B.X := R.B.X div 2;
    R.B.Y := R.B.Y div 2;
    R.Move(R.B.X,0);
    WinR := New(PWinRect, Init(R, ri, Safe[ri]));
    DeskTop^.Insert(WinR); {правое}
    DeskTop^.GetExtent(R);
    R.B.X := R.B.X div 2;
    R.B.Y := R.B.Y div 2;
    WinL := New(PWinRect, Init(R, le, Safe[le]));
    DeskTop^.Insert(WinL); {левое}
    {rez:=0.0;}
    DialNa^.SetData(rez);
    Control := DeskTop^.ExecView(DialNa);
    if Control=cmOk then begin
        DialNa^.GetData(rez);
        ReturnCode:=0;
    end else ReturnCode:=1;
    DeskTop^.Delete(WinL);
    DisPose(WinL, Done);
    DeskTop^.Delete(WinR);
    DisPose(WinR, Done);
end;
end.

```

```

(* Модуль SimpEst . Непосредственное оценивание *)
unit SimpEst;
INTERFACE
uses Objects, Views, App, Dialogs,
     WinRect1, ConstAll, DialSimp, Goal_Var;

procedure Simple_Estimate(i:word; var Rez: real);
IMPLEMENTATION

procedure Simple_Estimate(i:word; var Rez: real);

VAR
    R:TRect;
    Win: PWinRect;
    Dial: PDialSimp;
    Control: word;

begin
    Dial:=New(PDialSimp, Init('М^С' степень заполнения',false));
    DeskTop^.GetExtent(R);
    R.B.X := R.B.X div 2;
    R.A.Y := R.B.Y div 2;
    R.Move(8,0);
    Win := New(PWinRect, Init(R, i, Safe[i]));
    DeskTop^.Insert(Win); {образец}

    Dial^.SetData(Rez);
    Control := DeskTop^.ExecView(Dial);
    if Control=cmOk then begin
        Dial^.GetData(Rez);
    end;
    DeskTop^.Delete(Win);
    DisPose(Win, Done);
    ReturnCode:=0;
    if Control=cmCancel then ReturnCode:=1;

```

```
end;
end.
```

```
(* Модуль SimpEst . Непосредственное оценивание *)
unit ReEst;
INTERFACE
uses Objects, Views, App, Dialogs,
      WinRect1, ConstAll, DialSimp, Goal_Var;

procedure Re_Estimate(var Crr:word;msg:string;i:word; var Rez: real);

IMPLEMENTATION

procedure Re_Estimate(var Crr:word;msg:string;i:word; var Rez: real);

VAR
  R:TRect;
  Win: PWinRect;
  Dial: PDialSimp;
  Control: word;

begin
  Dial:=New(PDialSimp, Init(^M^C+msg+' оценку степени заполнения изображения',true));
  Dial^.Title:=NewStr('Согласны ли Вы');
  PButton(Dial^.Last^.Next^.Next)^.Title:=NewStr('Нет');
  PButton(Dial^.Last^.Next^.Next)^.Command:=cmNo;
  DeskTop^.GetExtent(R);
  R.B.X := R.B.X div 2;
  R.A.Y := R.B.Y div 2;
  R.Move(8,0);
  Win := New(PWinRect, Init(R, i, Safe[i]));
  DeskTop^.Insert(Win); {образец}

  Dial^.SetData(Rez);
  Control := DeskTop^.ExecView(Dial);
  if Control=cmOk then begin
    Dial^.GetData(Rez);
    Crr:=1;
  end else
    Crr:=2;
  DeskTop^.Delete(Win);
  DisPose(Win, Done);
  ReturnCode:=0;
  if Control=cmCancel then ReturnCode:=1;
end;
end.
```

```
unit In_Input;
interface

uses App, Objects, Views, MsgBox,
      Goal_Var, WinRect1, Dial_In, ConstAll;

procedure InputMatrIn;

implementation

procedure InputMatrIn;
var R:Trect;
  WinL,Win:PWinRect;
  DialIn:PDialIn;
  i,j:integer;
  Control: word;
begin
  ShowAlg^.Ftext:='Ввод матр. мультипл. сравнений';
  DialIn:=New(PDialIn,Init);
  for j:=1 to NRect do for i:=1 to NRect do {нач. заполнение матрицы "Саати"}
    SaatiMatrix[j,i]:=1.0;
  for j:=1 to NRect do begin {заполнение матрицы "Саати" по строкам}
    for i:=1 to NRect do begin {заполнение матрицы "Саати" по столбцам}
      if i<>j then begin
```

```

if (j=1)and(i=2) then
  DialIn^.DisableCommands([cmStepBack])
else
  DialIn^.EnableCommands([cmStepBack]);
DeskTop^.GetExtent(R);
R.B.X := R.B.X div 2;
R.B.Y := R.B.Y div 2;
WinL := New(PWinRect, Init(R, i, Safe[i]));
DeskTop^.Insert(WinL);
DeskTop^.GetExtent(R);
R.B.X := R.B.X div 2;
R.B.Y := R.B.Y div 2;
R.Move(R.B.X,0);
Win := New(PWinRect, Init(R, j, Safe[j]));
DeskTop^.Insert(Win); {эталон}
DialIn^.SetData(SaatiMatrix[j,i]);
Control := DeskTop^.ExecView(DialIn);
DeskTop^.Delete(WinL);
DisPose(WinL, Done);
DeskTop^.Delete(Win);
DisPose(Win, Done);
if Control=cmCancel then
  if MessageBox(^C'Тест еще не закончен!'+  

    ^M^C'Вы действительно хотите прервать тест?',nil,  

    mfConfirmation+mfYesButton+mfNoButton) <> cmYes then
    Control:=cmOk
  else
    ReturnCode:=1;
if Control=cmOk then begin
  DialIn^.GetData(SaatiMatrix[j,i]);
  ReturnCode:=0;
  if (i>=NRect-1)and(j>=NRect) then
    if MessageBox('C'Подтвердите правильность последнего ответа'+^M^C  

    '(Но шаг назад)',nil, mfConfirmation+mfYesButton+mfNoButton) <> cmYes then
      dec(i);
end;
if Control=cmStepBack then begin
  DialIn^.SelectNext(True);
  if i-1=j then dec(i);
  dec(i,2);
  if i<0 then begin
    dec(j);
    i:=NRect-1;
    if j<1 then begin
      j:=1;
      i:=1;
    end;
  end;
  if ReturnCode=1 then Break
end;
end;
if ReturnCode=1 then Break;
end;
end;
end.

```

```

unit Dial_In;

interface
uses
  Objects, Views, MsgBox, Dialogs, Drivers,
  Validate, ConstAll;

type
  PDialIn = ^TDialIn;
  TDialIn = object(TDialog)
    constructor Init;
    procedure HandleEvent(var Event: TEvent); virtual;
  end;

```

```

implementation

type
  PMyValidator = ^TMyValidator;
  TMyValidator = object(TPXPictureValidator)
    function IsValid(const S: string): Boolean; virtual;
    procedure Error; virtual;
  end;

  PILRealType = ^TILRealType;
  TILRealType = object(TInputLine)
    Field: pointer;
    constructor Init(var Bounds: TRect; AMaxLen: Integer; Ctx: word);
    function DataSize: Word; virtual;
    procedure GetData(var Rec); virtual;
    procedure SetData(var Rec); virtual;
  end;

  function TMyValidator.IsValid(const S: string): Boolean;
  var code:integer;
    RealVar: real;
  begin
    Val(S,RealVar,code);
    IsValid:= inherited IsValid(S) and (code=0) and (RealVar>0.0) and (RealVar<=99.99)
  end;
  procedure TMyValidator.Error;
  begin
    MessageBox(^С'Некорректные данные !
              +^М^С'Повторите ввод', nil, mfError + mfOkButton);
  end;

  constructor TILRealType.Init(var Bounds: TRect; AMaxLen: Integer; Ctx: word);
  begin
    inherited Init(Bounds,AMaxLen);
    SetValidator(New(PMyValidator, Init('[#][#][{.[#][#]}]', True)));
    HelpCtx := Ctx;
  end;
  function TILRealType.DataSize: Word;
  begin
    DataSize:=sizeof(real);
  end;
  procedure TILRealType.GetData(var Rec);
  var code:integer;
  begin
    val(Data^,real(Rec),code);
  end;
  procedure TILRealType.SetData(var Rec);
  begin
    str(real(Rec):4:2,Data^);
    while copy(Data^,1,1)=' do
      Data^:=copy(Data^,2,3);
  end;

  constructor TDialIn.Init;
  var
    Bounds: TRect;
    Cl: PInputLine;
  begin
    Bounds.Assign(0,0,32,10);
    Bounds.Move(46,14);
    inherited Init(Bounds, 'Определите');
    HelpCtx:=hcNoContext;
  { Options := Options or ofCentered;}
  Flags := 0;
  GetExtent(Bounds);
  Bounds.Assign(1,1,Bounds.B.X-1,5);
  Insert(New(PStaticText, Init(Bounds, ^С'Во сколько раз степень заполнения'+
    '{нижнего} ' правого изображения больше'+
    ', степени заполнения левого')));

  Bounds.Assign(0,5,6,6);
  Cl:= New(PILRealType, Init(Bounds, 4, hcInInp));
  Cl^.Options := Cl^.Options or ofCenterX;
  Insert(Cl);

```

```

    Bounds.Assign(0,7,9,9);
    Bounds.Move(4,0);
    Insert(New(PButton, Init(Bounds, 'назад',cmStepBack,bfNormal)));
    Bounds.Move(10,0);
    Insert(New(PButton, Init(Bounds, 'ввод',cmOk,bfDefault)));
    Cl^.Select
end;

procedure TDialIn.HandleEvent(var Event: TEvent);
begin
  inherited HandleEvent(Event);
  if (Event.What = evCommand) and (Event.Command=cmStepBack) then begin
    EndModal(cmStepBack);
    ClearEvent(Event)
  end;
end;
end;
end.



---


Unit InN_Calc;
{#### Обработка метода "В"новая (с разбиением на подмножества) ####}
{#### Доминирующие составляющие - перемещаемы ####}
interface

uses MsgBox,
      Goal_Var,Toc_Dial,In_ReInput,ConstAll;

procedure CalcInN(Vsingl:boolean);
  {если Vsingl=true то работаем по V (по строкам), иначе по W }

implementation

procedure CalcInN(Vsingl:boolean);
const nk=11; {размерность дискретной шкалы распределения
              (количество дел. шкалы, до которого округл. знач. сил при выч. Ксог)}
  eps=0.1; {точность (епсилон)}
type Real1Dim = array[1..MaxNGoal] of Real;
  Real2Dim = array[1..MaxNGoal] of Real1Dim;
  Byte1Dim = array[1..MaxNGoal] of byte;
  Byte2Dim = array[1..MaxNGoal] of Byte1Dim;
  Char1Dim = array[1..MaxNGoal] of char;
  Char2Dim = array[1..MaxNGoal] of Char1Dim;
  Bool1Dim = array[1..MaxNGoal] of boolean;
  Bool2Dim = array[1..MaxNGoal] of Bool1Dim;
  omass = array[1..nk] of Byte;

var
  Vwsr, Domin, MinKsog, ds: real;
  ReInput: boolean;
  h,i,j,et,js,Crr: word;
  NDomin, NewDomin, NMinKsog, ForMove, WhereMove, N_ReInput: byte;
  soob: string;
  OurMatrix: glNpByNp; {OurMatrix - матрица весов альтернатив}
  VW {Точные значения распределения} :Real2Dim;
  RVW {Распределение: RVW[i,j]-позиция шкалы, где расположена
        "палка"Vij (Wij)} :Byte2Dim;
  Sets {'p'-predominant,'o'-own,'t'-transitive}:Char2Dim;
  IdxKoefSogl {индекс упорядочивающий Ксог по возрастанию} :Byte1Dim;
  R {Распределение в другом виде: R[i] - количество "палок"
      в i-й позиции шкалы} :omass;
  KoefSogl {Коэф. согласованности результирующие}:Real1Dim;
  MoveAble {true - перемещаемая "палка"; false - нет} :Bool1Dim;

function SrednVW(Nraspr:byte;Tochn:boolean) : real;
var i:integer;
  sr:real;
begin
  sr:=0;
  for i:=1 to Ngoal do
    if Tochn then
      sr:=sr+VW[Nraspr,i]
    else
      sr:=sr+RVW[Nraspr,i];
end;

```

```

sr:=sr/NGoal;
SrednVW:=sr;
end;

Procedure Ksogl(var k : real;
                x,y:integer
                ;
                R : omass);
{ Процедура расчета коэффициента согласования.
  Исходные данные:
    R - массив, задающий текущее распределение;
    x - количество делений шкалы;
    у - текущее значение количества путей
      (экспертов);
  Выходные данные:
    k - коэффициент согласия; }

Var
  sp,lebzn,npzn,npr,lebr,d:real;
  i,nd:integer;
  k_is_0, ravn: boolean;

begin
  nd:=0;
  for i:=1 to x do {x-число делений шкалы}
    if (R[i]>0) then inc(nd);
    { nd - число делений шкалы, названных в качестве оценок
      (количество элементов множества значений сил) }
  d:=0.0;
  if nd>1 then
    d:=(x-1)/(nd-1);
  if (Frac(d)>0.0) or not((R[1]>0) and (R[x]>0)) then
    k_is_0 := false { необходимо условие отличия от 0 ksogl }
  else begin
    ravn:=True; { проверка равномерности распределения оценок}
    for i:=1 to (x-Trunc(d)) do
      ravn:=ravn and (R[i]=R[i+Trunc(d)]);
    if ravn then
      k_is_0 := true
    else
      k_is_0 := false
  end;
  if not k_is_0 then begin
    npzn:=ln(x);
    sp:=0.0;
    for i:=1 to x do
      sp:=sp+i*R[i]/y;
    if y = 1 then begin k:=1; exit end; {одна палка}
    lebzn:=0.0;
    for i:=1 to x do
      lebzn:=lebzn+Abs(i-(x+1)/2.0)*y/(x*ln(x)*ln(y));
    npr:=0.0;
    for i:=1 to x do
      if R[i]>0 then npr:=npr+R[i]/y*ln(R[i]/y);
    lebr:=0.0;
    for i:=1 to x do
      lebr:=lebr+R[i]/y*Abs(i-sp);
    k:=(lebr-npr)/(lebzn+npzn);
  end else
    k:=0;
end;{Sogl-----}

Procedure ForCalcKsog(var Rinp :Byte1Dim; var R:omass);
{ Процедура преобразования представления распределения
  Входные данные: Rinp - массив, задающий текущее распределение;
  Выходные данные: R - =====///===== в другом формате }
Var
  i:integer;
begin
  ##### Преобразование Rinp в R #####
  for i:=1 to nk do R[i]:=0;
  for i:=1 to NGoal do
    if Rinp[i]<>0 then inc(R[Rinp[i]]);
  #####

```

```

end;{ForCalcKsog-----}

begin
{##### Находим Тобн и Тпим для шкалы #####}
{ - порога обнаружения I
    I I I I I I I I I
    1--2--3--4--5--6--7--8--9--10 }
et:=round(0.5 * nk + 1);
for js := 1 to nk do begin
    R[js]:=1;
    if js = et then R[js]:=2;
end;
R[1]:=0;
Ksogl(kobn,nk,nk,R);
{ - порога применения
    I I
    1--2--3--4--5--6--7--8--9--10 }
for js := 1 to nk do R[js]:=0;
R[et-1]:=1;
R[et]:=1;
Ksogl(kpr,nk,2,R);
OurMatrix:=SaatiMatrix; {OurMatrix - матрица весов альтернатив}
MinKsog:=0; {поэтому первый раз всегда зайдем в цикл}
{## Инициализация (все "палки"перемещаемые) Когда это делать ???##}
for i:=1 to Ngoal do for j:=1 to Ngoal do
begin MoveAble[i,j]:=True; RVW[i,j]:=255; VW[i,j]:=-1 end;

while (MinKsog < Kpr) and (ReturnCode<>1) do begin
{#### Формирование распределений ####}
for i:=1 to NGoal do begin { Ц И К Л П О Р А С П Р Е Д Е Л Е Н И Я М}
    N_ReInput:=0;
    for j:=1 to NGoal do Sets[j,i]:=';{столб множества пустой}
{### Доминирующий - min (max) элемент матрицы ###}
    if Vsingl then Domin:=9999 else Domin:=0;
    for h:=1 to NGoal do
        if Vsingl then begin
            if OurMatrix[i,h]<Domin then begin {min}
                Domin:=OurMatrix[i,h];
                NDomin:=h
            end
            end else begin
                if OurMatrix[h,i]>Domin then begin {max}
                    Domin:=OurMatrix[h,i];
                    NDomin:=h
                end;
            end;
        end;
    {## Доминирующему элементу присваивается значение 1 ##}
    VW[NDomin,i]:=1;
    for h:=1 to NGoal do {по всех домин. элементах }
        if (Vsingl and (OurMatrix[i,h]=Domin)) or (not Vsingl
            and (OurMatrix[h,i]=Domin)) then begin
            Sets[h,i]:='p'; { predominant (доминирующая) }
            VW[h,i]:=VW[NDomin,i]; {1.0}
        end;
    {## Определяем Vii (Wjj) собственные значения ##}
    if Vsingl then
        VW[i,i]:=VW[NDomin,i]*OurMatrix[i,NDomin]
    else
        VW[i,i]:=VW[NDomin,i]/OurMatrix[NDomin,i];
    if Sets[i,i]<>'p' then Sets[i,i]:='o'; {own (собственные)}
{## Остальные V (W)-транзитивные определяем относительно Vii (Wjj) ##}
    for j:=1 to NGoal do begin
        if Sets[j,i]='' then begin
            Sets[j,i]:='t';
            if Vsingl then
                VW[j,i] := VW[i,i] / OurMatrix[i,j]
            else
                VW[j,i] := VW[i,i] * OurMatrix[j,i];
        end;
        {избавимся от потери значимости round(VW[j,i] * 100000)/100000}
        RVW[j,i]:=round(round(VW[j,i] * 100000)/10000);
    end;
end;
end;

```

```

if not FeedBack then minKsog:=1.0 {чтобы сразу выйти из цикла} else begin
{##### Мы должны работать со всеми распределениями одновременно !!!!
##### Формируем индексный массив (упорядоч. Ksog по возрастанию )####}
For i:=1 to NGoal do begin
  ForCalcKsog(RVW[i],R);
  Ksogl(Ksog,nk,NGoal,R);
  KoefSogl[i]:=Ksog;
  IdxKoefSogl[i]:=lo(i);
end;
For i:=1 to NGoal-1 do
  For j:=i+1 to NGoal do
    if KoefSogl[IdxKoefSogl[i]] > KoefSogl[IdxKoefSogl[j]] then begin
      h:=IdxKoefSogl[i];
      IdxKoefSogl[i]:=IdxKoefSogl[j];
      IdxKoefSogl[j]:=h
    end;
MinKsog:=KoefSogl[IdxKoefSogl[1]];
if KoefSogl[IdxKoefSogl[1]]<Kobn then begin
  MessageBox('C++ Согласованность меньше допустимой.'
  +'М Вам нужно ввести некоторые данные снова.',
  nil, mfInformation + mfOkButton);
for j:=1 to NGoal do begin
  if ReturnCode<>0 then Break;
  if IdxKoefSogl[1]<>j then {чтобы исключить ввод диагональных элементов}
    if Vsingl then {вводить столбец заново}
      DialReInput(j,IdxKoefSogl[1],OurMatrix[j,IdxKoefSogl[1]])
      else { строку }
    DialReInput(IdxKoefSogl[1],j,OurMatrix[IdxKoefSogl[1],j]);
end;
Inc(N_ReInput);
if N_ReInput>2 then begin
  EndMsg;
  exit
end;
end else begin

  if KoefSogl[IdxKoefSogl[1]]<Kpr then begin
repeat
  {##### Находим "палку", наиболее отдаленную от ср.зн.
  - для передвижения (ForMove) ####}
  ForMove:=0;
  WhereMove:=0;
  i:=1;
  while (i<=NGoal) and (KoefSogl[IdxKoefSogl[i]]<Kpr) and (Formmove=0) do begin
    {##### Вычисляем среднее значение по RVW ####}
    Vwsr:=SrednVW(IdxKoefSogl[i],false);
    ds:=0.5; {Max отклонение от ср.зн. в данном i-м распр-и (< 0.5 - не двигать}
    for j:=1 to NGoal do begin
      if (Abs(Vwsr-RVW[IdxKoefSogl[i],j])>ds)
        and MoveAble[IdxKoefSogl[i],j] then begin
          ds:=Abs(Vwsr-RVW[IdxKoefSogl[i],j]); {искать по точным значениям? - Нет}
          if (Sets[IdxKoefSogl[i],j]='p') {является дом.(predominant)} then begin
            ForMove:=j;
            WhereMove:=IdxKoefSogl[i];
            {на всякий случай найдем новое доминирующее значение }
            if Vsingl then Domin:=9999 else Domin:=0;
            for h:=1 to NGoal do
              if h<>WhereMove then {без текущего доминирующего}
                if Vsingl then begin
                  if OurMatrix[ForMove,h]<Domin then begin {min}
                    Domin:=OurMatrix[ForMove,h];
                    NewDomin:=h
                  end
                end else begin
                  if OurMatrix[h,ForMove]>Domin then begin {max}
                    Domin:=OurMatrix[h,ForMove];
                    NewDomin:=h
                  end;
                end;
            end else
              ForMove:=0
            end;
          end;
        end;
    end;
  end;
end;

```

```

    end;
    inc(i);
end;
if Formove=0 then begin {predominants is absent}
  i:=1;
  while (i<=NGoal) and (KoefSogl[IdxKoefSogl[i]]<Kpr) and (Formove=0) do begin
    ##### Вычисляем среднее значение по RVW #####
    VWSr:=SrednVW(IdxKoefSogl[i],false);
    ds:=0.5; {max отклон-я от ср.зн. в данном i-м распред. (<0.5 -не двигать)}
    for j:=1 to NGoal do begin
      if (Abs(VWSr-RVW[IdxKoefSogl[i],j])>ds)
        and MoveAble[IdxKoefSogl[i],j] then begin
          ds:=Abs(VWSr-RVW[IdxKoefSogl[i],j]); {искать по точным значениям? -Нет}
          if (Sets[IdxKoefSogl[i],j]='o') {является собственной (own)} then begin
            ForMove:=j;
            WhereMove:=IdxKoefSogl[i];
          {найдем домин.зн.для j-го столбца(строки) - пригодится при изм. матр.}
          NDomin:=0;
          for h:=1 to NGoal do
            if (Sets[h,j]='p') then begin
              NDomin:=h;
              Break
            end;
          end else
            ForMove:=0
          end;
        end;
      inc(i);
    end;
  end;
end;
if Formove=0 then begin {owns is absent too}
  i:=1;
  while (i<=NGoal) and (KoefSogl[IdxKoefSogl[i]]<Kpr) and (Formove=0) do begin
    ##### Вычисляем среднее значение по RVW #####
    VWSr:=SrednVW(IdxKoefSogl[i],false);
    ds:=0.5; {max отклонения от ср.зн. в данном i-м расп.(<0.5 -не двигать)}
    for j:=1 to NGoal do begin
      if (Abs(VWSr-RVW[IdxKoefSogl[i],j])>ds) and MoveAble[IdxKoefSogl[i],j]
      then begin
        ds:=Abs(VWSr-RVW[IdxKoefSogl[i],j]); {искать по точным значениям? - Нет}
        if (Sets[IdxKoefSogl[i],j]='t') {является транзитивной (transitive)}
        then begin
          ForMove:=j;
          WhereMove:=IdxKoefSogl[i];
        end;
      end;
      inc(i);
    end;
  end;
  #####
end;
if ForMove>0 then begin
  Crr:=1; {1 вместо cmYes для диалога}
  {задавать вопрос}
  case Sets[WhereMove,ForMove] of
    'p': {для доминирующих} begin
      if Vsingl then begin
        soob:=Uvel; { нужно увеличить предпочтение le по справ с ri}
        Dial(Crr,WhereMove{ri},ForMove{le},soob,
          OurMatrix[ForMove{ri},WhereMove{le}])
      end else begin
        soob:=Umen; { нужно уменьшить предпочтение le по справ с ri}
        Dial(Crr,ForMove{ri},WhereMove{le},soob,
          OurMatrix[WhereMove{le},ForMove{ri}]);
      end;
    end;
    'o': {для собственных} begin
      if (Vsingl and (VWSr<RVW[WhereMove,ForMove])) or
        (not Vsingl and not (VWSr<RVW[WhereMove,ForMove])) then
        soob:=Umen { нужно уменьшить предпочтение le по справ с ri}
      else
        soob:=Uvel; { нужно увеличить предпочтение le по справ с ri}
      if Vsingl then

```

```

    Dial(Crr,NDomin{ri},ForMove{le},soob,
          OurMatrix[ForMove{ri},NDomin{le}])
    else
      Dial(Crr,ForMove{ri},NDomin{le},soob,
            OurMatrix[NDomin{le},ForMove{ri}]);
    end;
't':{это для транзитивных составляющих} begin
  if ( Vsingl and (VWsr<RVW[WhereMove,ForMove])) or
    (not Vsingl and not (VWsr<RVW[WhereMove,ForMove])) then
    soob:=Uvel { нужно увеличить предпочтение le по срав с ri}
  else
    soob:=Umen;{ нужно уменьшить предпочтение le по срав с ri}
  if Vsingl then
    Dial(Crr,WhereMove{ri},ForMove{le},soob,
          OurMatrix[ForMove{ri},WhereMove{le}])
  else
    Dial(Crr,ForMove{ri},WhereMove{le},soob,
          OurMatrix[WhereMove{le},ForMove{ri}]);
  end;
end;
if ReturnCode = 1 then exit;
if crr = 1 then begin { 1 вместо cmYes }
  case Sets[WhereMove,ForMove] of
    'p': {для доминирующих} begin
      if Vsingl then {V}
        { увеличивать элемент OurMatrix}
        OurMatrix[ForMove,WhereMove]:= {VW[NewDomin,ForMove]* ==1.0}
        OurMatrix[ForMove,NewDomin] / 0.9
      else {W}
        { уменьшать элемент OurMatrix}
        OurMatrix[WhereMove,ForMove]:= {VW[NewDomin,ForMove]* ==1.0}
        0.9 * OurMatrix[NewDomin,ForMove]
    end;
    'o': {для собственных}
      if Vsingl then begin {V}
        if (VWsr<RVW[WhereMove,ForMove]) then
          {следовательно, уменьшать и элемент OurMatrix}
          OurMatrix[ForMove,NDomin]:=OurMatrix[ForMove,NDomin]-Eps
        else {двигать вправо(увеличивать V)}
          {следовательно, увеличивать и элемент OurMatrix}
          OurMatrix[ForMove,NDomin]:=OurMatrix[ForMove,NDomin]+Eps
      end else begin {W}
        if (VWsr<RVW[WhereMove,ForMove]) then
          {следовательно, элемент OurMatrix =}
          OurMatrix[NDomin,ForMove]:=1/(VW[ForMove,ForMove]-Eps)
        else {двигать вправо(увеличивать W)}
          {следовательно, элемент OurMatrix}
          OurMatrix[NDomin,ForMove]:=1/(VW[ForMove,ForMove]+Eps)
      end;
      {здесь м.б. изменить остальные доминирующие элементы в строке, столбце}
    ',t':{для транзитивных}
      if Vsingl then begin {V}
        if (VWsr<RVW[WhereMove,ForMove]) then
          {следовательно, увеличивать элемент OurMatrix}
          OurMatrix[ForMove,WhereMove]:=VW[ForMove,ForMove]
          /(VW[WhereMove,ForMove]-Eps)
        else {двигать вправо(увеличивать V)}
          {следовательно, уменьшать элемент OurMatrix}
          OurMatrix[ForMove,WhereMove]:=VW[ForMove,ForMove]
          /(VW[WhereMove,ForMove]+Eps)
      end else begin {W}
        if (VWsr<RVW[WhereMove,ForMove]) then
          {следовательно, уменьшать и элемент OurMatrix}
          OurMatrix[WhereMove,ForMove]:=(VW[WhereMove,ForMove]-Eps)
          /VW[ForMove,ForMove]
        else {двигать вправо(увеличивать W)}
          {следовательно, увеличивать и элемент OurMatrix}
          OurMatrix[WhereMove,ForMove]:=(VW[WhereMove,ForMove]+Eps)
          /VW[ForMove,ForMove]
      end;
    end;
  end else begin { если нет - помечаем как непередвигаемую }
    MoveAble[WhereMove,ForMove]:=false;
  end;
end;

```

```

        if Sets[WhereMove,ForMove]='о'{для собственных} then
            MoveAble[NDomin,ForMove]:=false;
        end;
    end
    until (ForMove=0) or (Crr=1); {выход,если нечего двигать или по ДА}
end; {if Ксогл min < Тприм}
end;
if (ForMove=0) or (N_ReInput>2) then begin
    { если нечего двигать - ForMove=0}
    EndMsg;
    for i:=1 to NGoal do KoefSogl[i]:=0.0;
    Break
end
end{FeedBack}
end;{конец цикла While MinKsog<=Kpr and (ReturnCode<>1)}
if ReturnCode<>1 then begin {не Cancel}
    for i:=1 to Ngoal do koef[i]:=SrednVW(i,true); {precisely}
end;
end;
end.

```

```

unit In_ReInput;
interface

uses App, Objects, Views,
      Goal_Var, WinRect1, Dial_In, ConstAll;

Procedure DialReInput(ri,le:integer;var rez:real);

implementation

procedure DialReInput;
var R:Trect;
    WinL,WinR:PWinRect;
    DialIn:PDialIn;
    i,j:integer;
    Control: word;
begin
    DialIn:=New(PDialIn,Init);
    DialIn^.DisableCommands([cmStepBack]);
    DeskTop^.GetExtent(R);
    R.B.X := R.B.X div 2;
    R.B.Y := R.B.Y div 2;
    R.Move(R.B.X,0);
    WinR := New(PWinRect, Init(R, ri, Safe[ri]));
    DeskTop^.Insert(WinR); {правое}
    DeskTop^.GetExtent(R);
    R.B.X := R.B.X div 2;
    R.B.Y := R.B.Y div 2;
    WinL := New(PWinRect, Init(R, le, Safe[le]));
    DeskTop^.Insert(WinL); {левое}
    {rez:=0.0;}
    DialIn^.SetData(rez);
    Control := DeskTop^.ExecView(DialIn);
    if Control=cmOk then begin
        DialIn^.GetData(rez);
        ReturnCode:=0;
    end else ReturnCode:=1;
    DeskTop^.Delete(WinL);
    DisPose(WinL, Done);
    DeskTop^.Delete(WinR);
    DisPose(WinR, Done);
end;
end.

```

```

Unit CombCalc;
{#### Обработка метода "На"комбинаторная ####}
interface

uses Goal_Var, CombColl, Dos,
      Toc_Dial, SimpEst{tsv},CombKsog, ConstAll;

```

```

procedure CalComb(case_:byte);{case_=1 - "HA",2-"B",3-"фунд"}
var Matrix:array[1..MaxNGoal,1..maxNGoal] of ShortInt;
implementation

procedure CalComb(case_:byte);
var i,i1,j,n,k,l: integer;
    NDomin,Num:byte;
    Yes_:boolean;
    Nabor: array[1..(MaxDim*MaxDim-MaxDim) div 2,1..2] of byte;
    Nab: array[0..MaxDim] of byte;
    SortColl:PSortColl;
    Incr,Decr: array[1..MaxDim,1..MaxDim] of boolean;
    aRec:tRec;
    realVar,realVar1:real;
    Seconds:LongInt;
    Hour, Minute, Second, Hand: Word;
    _Row,_Col:byte; _More:boolean; {для передачи параметров в far-процедуру}
    Ind:array[1..MaxChanges] of byte; {индексный массив для сортировки ел.-тов матрицы}
    soob:string; crr:word; Sum,Base:real;
    OurMatrix:g1NpByNp;{for CombKsog}

function Real2Short(r1:real):shortInt;
var r:real; sh:ShortInt;
begin
  if r1<1 then r:=1/r1 else r:=r1;
  if r<1.5 then Sh:=round((r-1)*100) else
    if r<5 then Sh:=round((r-1.5)*10)+50 else
      if r<10 then Sh:=round((r-5)*5)+85 else
        if r<20 then Sh:=round(r-10)+110 else
          if r<=50 then Sh:=round((r-20)/5)+120 else
            Sh:=127;
  if r1<1 then Real2Short:=-Sh else Real2Short:=Sh;
end;

function Short2Real(Sh1:shortInt):real;
var r:real; sh:ShortInt;
begin
  if sh1<0 then sh:=-sh1 else sh:=sh1;
  if sh<=50 then r:=1+sh/100 else
    if sh<=85 then r:=1.5+(sh-50)/10 else
      if sh<=110 then r:=5+(sh-85)/5 else
        if sh<=120 then r:=10+sh-110 else
          if sh<=126 then r:=20+(sh-120)*5 else
            r:=99;
  if sh1<0 then Short2Real:=1/r else Short2Real:=r;
end;

function ValidNab: boolean;
var se: set of 1..MaxDim;
    i,i1:byte;
begin
  se:=[Nabor[Nab[1],1]];
  se:=se+[Nabor[Nab[1],2]];
  for i1:=2 to NGoal-1 do
    for i:=2 to NGoal-1 do
      if Nabor[Nab[i],1] in se then
        se:=se+[Nabor[Nab[i],2]] else
        if Nabor[Nab[i],2] in se then
          se:=se+[Nabor[Nab[i],1]];
  ValidNab:=se=[1..NGoal]
end;

procedure GenerationIdeal(var aREC:tRec);
const Empty=127;
var n,Ro,C_:byte;
    brk:boolean;
    IntVar:integer;
    Vect:array[1..(MaxDim*MaxDim-MaxDim) div 2] of ShortInt;

procedure ToMatrCoord(vec:byte; var matrRow,matrCol:byte);
var sum:byte;

```

```

begin
  matrRow:=0; sum:=0;
  repeat
    inc(matrRow);
    sum:=sum+NGoal-matrRow;
  until sum>=vec;
  matrCol:=vec-sum+NGoal;
end;

function ToVec(matrRow,matrCol:byte) :byte;
var sum,i,n:byte;
begin
  sum:=0;
  n:=NGoal-1;
  i:=1;
  while i<MatrRow do begin
    inc(sum,n);
    dec(n);
    inc(i)
  end;
  ToVec:=sum+matrCol-matrRow;
end;

function FromVect(matrRow,matrCol:byte) :shortInt;
begin
  if matrRow>matrCol then
    FromVect:=-Vect[ToVec(matrCol,matrRow)]
  else
    FromVect:=Vect[ToVec(matrRow,matrCol)]
end;

begin
  for n:=1 to (NGoal*NGoal-NGoal) div 2 do Vect[n]:=Empty;
  aRec.Num:=0;
  for n:=1 to NGoal-1 do
    if abs(Matrix[Nabor[Nab[n],1],Nabor[Nab[n],2]])=Empty then begin
      aRec.Num:=$FF;
      break
    end else
      Vect[ToVec(Nabor[Nab[n],1],Nabor[Nab[n],2])]:=Matrix[Nabor[Nab[n],1],Nabor[Nab[n],2]];
  if aRec.Num<>$FF then
repeat
  brk:=true;
  for n:=1 to (NGoal*NGoal-NGoal) div 2 do
    if Vect[n]=Empty then begin
      ToMatrCoord(n,Ro,Co);
      C_:=1;
      while (C_<=NGoal) and ((C_=Ro)or(C_=Co)or(abs(FromVect(Ro,C_))=Empty)or(abs(FromVect(Co,C_))=Empty)) do
        inc(C_);
      if C_<=NGoal then begin
        case case_ of
          1:IntVar:=FromVect(Ro,C_)-FromVect(Co,C_);
          2:IntVar:=Real2Short(Short2Real(FromVect(Ro,C_))/Short2Real(FromVect(Co,C_)));
          3:begin
            realVar:=FromVect(Ro,C_)/10;
            if realVar<0 then realVar:=-1/realVar;
            realVar1:=FromVect(Co,C_)/10;
            if realVar1<0 then realVar1:=-1/realVar1;
            realVar:=realVar/realVar1;
            if realVar<1 then intVar:=-round(10/realVar) else intVar:=round(10*realVar);
          end;
        end;{case}
      end;
      if abs(IntVar)>=127 then begin
        aRec.Num:=$FF;{not Valid}
        brk:=true;
        Break
      end else
        Vect[n]:=IntVar
    end else
      brk:=false;
  end;

```

```

until brk;
if aRec.Num<>$FF then
  {filling aRec if not equal}
  with aRec do begin
    for n:=1 to (NGoal*NGoal-NGoal) div 2 do begin
      ToMatrCoord(n,Ro,Co);
      if Vect[n]<>Matrix[Ro,Co] then begin
        inc(Num);
        {здесь можно сделать ограничение по количеству if Num<=кол then...}
        RowCol[Num,1]:=Ro;
        RowCol[Num,2]:=Co;
        Rez[Num]:=Vect[n]
      end;
    end;
  end;
end;

procedure Save;
var counter:word;
  Matr:PMatr;
begin
  if ValidNab then begin
    {сформировать aRec по сгенерированому набору Nab}
    GenerationIdeal(aRec);
    {здесь можно сделать ограничение по количеству if aRec.Num<=кол then...}
    {if aRec.Num>0 - это уже не обязательно}
    if aRec.Num<>$ff then begin
      if memAvail<50000 then SortColl^.AtFree(SortColl^.Count-1);
      Matr:=New(PMatr,Init(aRec));
      Counter:=SortColl^.Count;
      SortColl^.Insert(Matr);
      if Counter=SortColl^.Count then
        DisPose(Matr,Done);
      {Str(SortColl^.Count:7,ShowAlg^.Ftext);}
      Str(memAvail:9,ShowAlg^.Ftext);
      ShowAlg^.Ftext:=' Подг.данных в ОП'+ShowAlg^.Ftext;
      ShowAlg^.DrawView
    end;
  end;
end;

procedure Recur(k:byte);
begin {рекурсия, сочетания}
  Nab[k]:=Nab[k-1]+1;
  if k=NGoal-1 then Save;
  while Nab[k]<>Num-NGoal+k+1 do begin
    if k<NGoal-1 then Recur(k+1);
    inc(Nab[k]);
    if Nab[1]>NGoal-1 then exit;
    if k=NGoal-1 then Save;
  end;
  if k<NGoal-1 then Save;
end;

procedure DelCase;
var i:byte;
  j:integer;
begin
  j := SortColl^.Count - 1;
  while j >= 0 do begin
    with PMatr(SortColl^.At(j))^ do begin
      i:=1;
      while (i<Num)and(RowCol[i,1]<_Row) do inc(i);
      while (i<Num)and(RowCol[i,1]=_Row)and(RowCol[i,2]<_Col) do inc(i);
      if (RowCol[i,1]=_Row)and(RowCol[i,2]=_Col) then
        if _More and (Rez[i]>Matrix[_Row,_Col])
        or not _More and(Rez[i]<Matrix[_Row,_Col]) then
          SortColl^.Free(SortColl^.At(j));
    end;
    Dec(j)
  end;
end;

```

```

begin
  {инициализация счетчика времени}
  GetTime(Hour, Minute, Second, Hand);
  Seconds:=LongInt(Hour)*3600+Minute*60+Second;
  if _Ks then InitMatrix(case_,OurMatrix);{for CombKsog}
  if _Ks and KsogGood(case_,OurMatrix) then begin
    CalcWeights(case_,OurMatrix); exit
  end;
  {Matrix prepare}
  case case_ of
    1:for i:=1 to NGoal do
      for n:=i to NGoal do
        Matrix[i,n]:=round(SaatiMatrix[i,n]*100);{преобразуем в ShortInt}
    2:for i:=1 to NGoal do
      for n:=i to NGoal do
        Matrix[i,n]:=Real2Short(SaatiMatrix[i,n]);
    3:for i:=1 to NGoal do
      for n:=i to NGoal do
        if SaatiMatrix[i,n]<1 then
          Matrix[i,n]:=-Round(10*SaatiMatrix[n,i]) {исп. симетричное знач.}
        else
          Matrix[i,n]:=Round(10*SaatiMatrix[i,n]);{преобразуем в ShortInt}

  end;{case}
  {form array }
  SortColl:=New(PSortColl,Init(1000,200));
  i1:=1;
  for i:=1 to NGoal-1 do
    for j:=i+1 to NGoal do begin
      Nabor[i1,1]:=i;
      Nabor[i1,2]:=j;
      inc(i1)
    end;
  Num:=i1-1;

  {Test for ideal }
  for i:=1 to NGoal-1 do nab[i]:=i;
  {сформировать aRec по сгенерированому набору Nabor}
  GenerationIdeal(aRec);
  if aRec.Num>0 then begin {non-ideal matrix}
    {Generation combination for NGoal-1 of Num}
    for i:=0 to NGoal-1 do nab[i]:=0;
    Recur(1);
    GetTime(Hour, Minute, Second, Hand);
    Inc(TimeQ,Abs(LongInt(Hour)*3600+Minute*60+Second-Seconds));
    case case_ of
      1>ShowAlg^.Ftext:='Обраб.Алдит комбинаторн.';
      2>ShowAlg^.Ftext:='Обраб.Мультипл.комбинаторн.';
      3>ShowAlg^.Ftext:='Обраб.Фунд.шк.комбинаторн.';
    end;
    {analysis of collection}
    for i:=1 to NGoal do for j:=1 to NGoal do
      begin Incr[i,j]:=false; Decr[i,j]:=false end;
    i:=1;
    InitMatrix(case_,OurMatrix);{for CombKsog}
    if SortColl^.Count>0 then
      repeat
        with PMatr(SortColl^.Items^[0])^.Rec do begin
          if i=1 then begin
            {сортировка ел.марицы - формир. Ind-масс.}
            for k:=1 to Num do Ind[k]:=k;
            for k:=1 to Num-1 do
              for l:=k+1 to Num do
                if abs(Rez[k]-Matrix[RowCol[k,1],RowCol[k,2]]) <
                  abs(Rez[l]-Matrix[RowCol[l,1],RowCol[l,2]]) then begin
                    n:=Ind[k]; Ind[k]:=Ind[l]; Ind[l]:=n
                  end;
            end;
            _Row:=RowCol[Ind[i],1];
            _Col:=RowCol[Ind[i],2];
            _More:=Rez[Ind[i]]>Matrix[_Row,_Col];
            i1:=Num;
          end;{with}
        end;
      end;
    end;
  end;
end;

```

```

Yes_:=More and Incr[_Row,_Col] or not _More and Decr[_Row,_Col];
if not Yes_ then begin {haven't asked yet}
  case case_ of
    1:begin
      if _More then {увеличивать}
        soob:=Uvel {INcrease}
      else {уменьшать}
        soob:=Umen; {DEcrease}
      Dial(Crr,{ri}_Col,_Row{le},soob,Matrix[_Row{ri},_Col{le}]/100);
    end;
    2:begin
      if _More then {увеличивать}
        soob:=Uvel {INcrease}
      else {уменьшать}
        soob:=Umen; {DEcrease}
      Dial(Crr,_Col{ri},_Row{le},soob,Short2Real(Matrix[_Row{ri},_Col{le}])));
    end;
    3:begin
      realVar:=Matrix[_Row{ri},_Col{le}]/10;
      if realVar<0 then realVar:=-1/realVar;
      DialSt(Crr,_Row{ri},_Col{le},_More,realVar,false);
    end;
  end;{case}
  if ReturnCode=1 then begin {выход по Escape}
    dispose(SortColl,Done);
    Exit
  end;
  Yes_:=Crr=1;
  if Yes_ then begin
    if _More then
      Incr[_Row,_Col]:=true
    else
      Decr[_Row,_Col]:=true;
    _More:=not _More;
    DelCase;
  end;
end;
if Yes_ then begin
  if _Ks then begin
    {vvvvv for CombKsog vvvvv}
    case case_ of
      1:OurMatrix[_Row,_Col]:=PMatr(SortColl^.Items^[0])^.Rec.Rez[ind[i]]/100;
      2:OurMatrix[_Row,_Col]:=Short2Real(PMatr(SortColl^.Items^[0])^.Rec.Rez[ind[i]]);
      3:begin
        OurMatrix[_Row,_Col]:=PMatr(SortColl^.Items^[0])^.Rec.Rez[ind[i]]/10;
        if OurMatrix[_Row,_Col]<0 then OurMatrix[_Row,_Col]
          :=-1/OurMatrix[_Row,_Col];
      end;
    end;
    case case_ of
      1:OurMatrix[_Col,_Row]:=-OurMatrix[_Row,_Col];
      2..3:OurMatrix[_Col,_Row]:=1/OurMatrix[_Row,_Col];
    end;
    if KsogGood(case_,OurMatrix) then begin
      CalcWeights(case_,OurMatrix);
      dispose(SortColl,Done);
      exit
    end;
    {~~~~~for CombKsog ~~~~~}
  end;
  inc(i)
end else begin {don't agree}
  DelCase;
  i:=1;
  InitMatrix(case_,OurMatrix);{for CombKsog}
end;
until (SortColl^.Count=0)or(i>i1);
if SortColl^.Count>0 then {Improvement Successful!!! }
  with PMatr(SortColl^.Items^[0])^.REC do
    for i:=1 to Num do
      Matrix[RowCol[i,1],RowCol[i,2]]:=Rez[i]
else begin

```

```

        dispose(SortColl,Done);
        EndMsg;
        Exit
    end;
end else {Матрица идеально согласована !};
dispose(SortColl,Done);
{min in 1-st row}
NDomin:=1;
for i:=2 to NGoal do
    if (Matrix[1,NDomin]>Matrix[1,i]) then
        NDomin:=i;
Koef[NDomin]:=1;
if (case_=1)and not _FB then Simple_Estimate(NDomin,Koef[NDomin]);
{Calculate weights}
case case_ of
1:for i:=1 to NGoal do
    Koef[i]:=Koef[NDomin]-(Matrix[1,i]-Matrix[1,NDomin])/100;
2:begin
    realVar1:=Short2Real(Matrix[1,NDomin]);
    for i:=1 to NGoal do
        Koef[i]:=realVar1/Short2Real(Matrix[1,i]);
    end;
3:begin
    if Matrix[1,NDomin]<0 then realVar1:=-10/Matrix[1,NDomin] else realVar1
        :=Matrix[1,NDomin]/10;
    for i:=1 to NGoal do begin
        if Matrix[1,i]<0 then realVar:=-10/Matrix[1,i] else realVar:=Matrix[1,i]/10;
        Koef[i]:=realVar1/realVar;
    end;
    end;
end;{case}
ReturnCode:=0;
end;
end.

```

```

unit CombColl;

interface

uses Objects;

const MaxDim=7; {max размерность матрицы - кол-во целей }
    MaxChanges= {макс количество несовпадений с реальной матрицей}
                (MaxDim*MaxDim-MaxDim) div 2 - (MaxDim-1);

type
PSortColl=^TSortColl;
TSortColl=object(TSortedCollection)
    function Compare(Key1, Key2: Pointer): Integer; virtual;
    procedure FreeItem(Item: Pointer); virtual;
end;

tRec =record
    Num:byte; {number of new rez in matrix}
    RowCol:array[1..MaxChanges,1..2] of byte; {row,column of matrix}
    rez:array[1..MaxChanges] of ShortInt; {new rez in matrix}
end;

PMatr=^TMatr;
TMatr=object(TObject)
    Rec:tRec;
    constructor Init(var aRec:tRec);
end;

implementation

constructor TMatr.Init(var aRec:tRec);
begin
    inherited Init;
    Rec:=aRec;

```

```

end;

{TSortColl}
procedure TSortColl.FreeItem(Item: Pointer);
begin {Just in case}
  if Item <> nil then Dispose(PMatr(Item), Done);
end;

function TSortColl.Compare(Key1, Key2: Pointer): Integer;
var i:byte;
begin
  if PMatr(Key1)^.Rec.Num >PMatr(Key2)^.Rec.Num then begin
    Compare:=1; exit
  end;
  if PMatr(Key1)^.Rec.Num < PMatr(Key2)^.Rec.Num then begin
    Compare:=-1; exit
  end; {Num1=Num2}
  for i:=1 to PMatr(Key1)^.Rec.Num do begin
    if PMatr(Key1)^.Rec.RowCol[i,1]*MaxDim+PMatr(Key1)^.Rec.RowCol[i,2]
    >PMatr(Key2)^.Rec.RowCol[i,1]*MaxDim+PMatr(Key2)^.Rec.RowCol[i,2] then begin
      Compare:=1; exit
    end;
    if PMatr(Key1)^.Rec.RowCol[i,1]*MaxDim+PMatr(Key1)^.Rec.RowCol[i,2]
    < PMatr(Key2)^.Rec.RowCol[i,1]*MaxDim+PMatr(Key2)^.Rec.RowCol[i,2] then begin
      Compare:=-1; exit
    end;
  end;
  for i:=1 to PMatr(Key1)^.Rec.Num do begin
    if PMatr(Key1)^.Rec.Rez[i] >PMatr(Key2)^.Rec.Rez[i] then begin
      Compare:=1; exit
    end;
    if PMatr(Key1)^.Rec.Rez[i] < PMatr(Key2)^.Rec.Rez[i] then begin
      Compare:=-1; exit
    end;
  end;
  Compare:=0;
end;
end.

```