УДК 51-77: 65.011.56

РОДІЧЕВА О.С.,
ЧЕМЕРИС А.М.,
ТЮТЮННИК С.В.

МАТЕМАТИЧНІ МОДЕЛІ ЗАДАЧІ ВИБОРУ ОПТИМАЛЬНОГО НАБОРУ ЗАПИТІВ МАТЕРІАЛІЗАЦІЇ

Запропоновано моделі роботи бази даних, що використовують формалізовані знання аналітиків у вигляді правил алгебри показників, які використовуються для вирішення задачі вибору оптимального набору запитів матеріалізації. Моделі є розширенням базової моделі параметричної бази даних Гриші С.М. Запропоновано метод визначення обернених ребер орграфу предметної області з використанням правил алгебри показників для отримання графу моделі з альтернативним джерелом даних. Для проведення експериментальних тестів запропонованих моделей було формалізовано нелінійну модель бульового програмування. Результати експериментів підтверджують ефективність використання запропонованих моделей.

Two formal models of database work are offered for using in solving task of choosing of optimal set of materialized views and that use formalized analyst knowledge in a form of indices algebra rules. The proposed models are extensions of the base parameterized database model of Dr. Grysha S.M. A method of adding back calculated edges to an oriented graph of the subject area is proposed through the use of indices algebra for obtaining a directed graph of a model with an alternative data source. A nonlinear model of Boolean programming was formalized for experimentally testing. The results of experiments prove the effectiveness of implementing the proposed models.

ВСТУП

Проблема здешевлення сукупної вартості володіння інформаційною системою управління бізнесом (ІСУБ) на всіх стадіях життєвого циклу є актуальним напрямком дослідження, адже досі вартість інформаційних систем залишається надзвичайно високою. Однією з найважливіших задач в рамках вирішення вказаної проблеми є автоматизація вирішення задачі вибору оптимального набору запитів матеріалізації (ВОНЗМ), що в свою чергу дозволить отримати оптимальну або хоча б ефективну структуру бази даних(далі БД) ІСУБ. Математичні моделі та застосовані в роботі підходи є частиною технології автоматизованої розробки ІСУБ «КІТ-XXI» та призначені для вирішення вказаної задачі. Для опису предметної області використовується алгебра показників [1, 2, 12].

В системах чи підсистемах типу OLTP матеріалізовані дані включають первинні показники та деяку невелику підмножину вторинних показників. Для складних запитів над великим об'ємами даних широко розповсюдженими є системи типу OLAP, які дозволяють швидко обробляти запити, але і в них аналогічний перелік даних для матеріалізації визначає користувач. Наприклад в Oracle BI аналітик має визначити три рівні: фізичний рівень, бізнес-модель та рівень представлення [13]. Ціль дослідження автору є пошук ефективних шляхів автоматизації, а в подальшому і переведення в автоматичний режим методів розподілу всього простору показників, що потребують користувачі, на множини віртуалізованих та матеріалізованих показників для забезпечення своєчасного отримання інформації. Зазначимо, що дане питання потребує автоматизації на об'ємах даних в десятки, сотні мільйонів записів та більше.

1. ОГЛЯД ЛІТЕРАТУРИ

Задача пошуку оптимальної структури БД розглядається в багатьох роботах: [3]-[7]. В зарубіжній літературі проблема пошуку оптимального набору представлень для матеріалізації відома під назвою "*view-selection problem*", семантична оптимізація в БД відома під назвою "*semantic query optimization*" [8]. В роботі [9] показано, що задача вибору представлень є *np*-повною задачею для потоку запитів без самоз'єднань таблиць (self-joins). В роботі [10] формалізовано задачу пошуку набору представлень (views) БД для матеріалізації, які б задовольнили обмеження дискового простору. В роботі [11] описано підхід з використанням оптимізації запитів, що є самонастроюваним в процесі роботи системи. В опрацьованих роботах не було знайдено такої формалізації, яка б дозволяла вирішувати питання матеріалізації вже під час розробки ІСУБ з використанням

формалізованих знань аналітиків. Також не було знайдено бульової формалізації цільової функції задачі (далі ЦФ) на базі алгебри показників чи реляційної алгебри(далі – РА), що відображає предметну область. В більшості робіт моделі спрощені і опущені важливі обмеження на час обробки запитів та час матеріалізації при зміні даних, в той час як вони зазвичай є вирішальними при визначенні питання матеріалізації відношення. В описаних методах не використовуються семантичні знання про структуру запитів та їх зв'язки між собою, які дають додаткову інформацію для оптимізації структури БД.

2. МОДЕЛЬ СТРУКТУРИ БД З БЕЗАЛЬТЕРНАТИВНИМ ДЖЕРЕЛОМ ДАНИХ

Запропонована модель базуються на моделі параметричної БД Гриші С.М [12], яка доопрацьована наступним чином: 1) використовується невідкладний режим актуалізації БД, за якого актуалізація показників виконується примусово ще до появи запиту до похідних показників; 2) використовується такий режим підтримки БД, за якого для визначеної підмножини показників P^{μ} актуалізація може виконуватись відкладено з певним зсувом в часі з визначеними обмеженнями $B^{\mu}: b^{\mu} \in B^{\mu}$; 3) визначено множину елементів відношення обчислювальності.

Показник – відношення БД чи певним чином обрана підсхема, яка може бути матеріалізована чи віртуалізована (обчислюватись кожен раз з попередніх показників при кожному новому зверненні). *Актуалізація* – процес встановлення відповідності значень зв'язаних показників. *Деактуалізація* – поява невідповідності значень матеріалізованих показників, що викликане змінами первинних показників. Будемо використовувати для позначення схему показника p позначення: $sh(p)$.

Визначення 1. БД зі збереженням частини показників (відношень) можна представити моделлю наступного вигляду:

$$\langle P, A, M, R, \lambda, \mu \rangle \qquad (2.1)$$

де P – множина показників БД така, що

$$P = P^x \cup P^v, P^x \cap P^v = \varnothing, P^{\mu} \subseteq P$$

P^x – множина показників, що зберігаються у зовнішніх запам'ятовуючих пристроях (матеріалізовані показники) і тому при зверненні до них процедури актуалізації виконуються тільки в разі, коли в початкових показниках зареєстровані зміни; P^v – множина віртуальних показників, тобто таких при зверненні до яких завжди вимагається їх актуалізація;

P^{μ} – множина показників, яка визначається аналітиком при проектування, що може бути матеріалізована відкладено, для якої аналітик задає вектор максимального часу матеріалізації B^{μ}. $P^{\mu} \cap P^x$ – множина матеріалізованих показників, при зверненні користувача до яких система не буде очікувати завершення актуалізації даних, та буде надавати останні актуальні дані на попередній період часу. $P^{\mu} \cap P^v$ - множина віртуалізованих показників, обчислення яких відбувається при зверненні користувача, тобто після отримання рішення задачі ВОНЗМ, тобто $P^{\mu} \cap P^v \subseteq P^v$. Також для результуючої множини, коли відомо P^x можна записати наступне:

$$\left(\forall p_i \in P^x \cap P^{\mu}\right) \wedge \left(\exists p_j \in P^{\mu} : \left(p_i \in p_j{}^{\circ\circ}\right)\right) \Rightarrow$$

$$\Rightarrow P^{\mu} := P^{\mu} \setminus p_i$$

$A \subseteq P \times P$ - відношення обчислювальності (джерела даних), тобто $\langle p_1, p_2 \rangle \in A$ означає що p_2 обчислюється з p_1 або, що теж саме, p_1 необхідне для обчислення p_2 (p_1 є джерелом даних для обчислення p_2).

$^{\leq}P$ – множина всіх первинних показників БД, тобто таких, які не можуть виводитись з інших; $P^{\geq}$ – множина всіх кінцевих показників БД чи її підсистеми, тобто таких, які не приймають участі при виведенні інших показників; оскільки для деяких показників є альтернативні джерела даних, то одне джерело отримання даних для кожного такого показника остаточно буде визначена тільки після рішення задачі ВОНЗМ;

$M = \{m_p\}\ p \in P$ - вектор об'ємів показників для збереження на деякий заданий часовий інтервал роботи системи, де

m_p – об'єм пам'яті, що необхідний для збереження показника p, що є, в загальному випадку, деякою випадковою величиною;

R – вектор трудомісткості операцій вибірки та зміни даних над показниками;

λ – вектор інтенсивності запитів користувачів до показників БД;

μ – вектор інтенсивності потоків зміни первинних показників. Вказані потоки з інтенсивностями λ та μ є стандартними пуасонівськи-

ми потоками подій.

Визначення 2. *Множиною елементів відношення обчислювальності A* може бути будь-який оператор алгебри показників.

За базис операцій алгебри показників обрано наступні операції: сортування $\tau_B(p)$, проекція $\pi_B(p)$, селекція $\sigma_C(p)$, агрегування $\gamma_B(p)$, з'єднання $\triangleright\triangleleft(p_1,p_2)$ та об'єднання $\cup(p_1,p_2)$. Коротко нагадаємо відмінність від однойменних в РА. Оператори *сортування* $\tau_B(p)$ та *селекції* $\sigma_C(p)$ та *агрегування* $\gamma_{B^{group},B^{agg}}(p)$ такі ж як і однойменні в реляційній алгебрі. Оператор *проекції* $\pi_B(p)$ виконує розширену проекцію над мірами показника p, при цьому виміри залишаються незмінними. Оператори *натурального з'єднання* $\triangleright\triangleleft(p_1,p_2)$ та *об'єднання* $\cup(p_1,p_2)$ відрізняються від однойменних в РА тим що з'єднує/об'єднує відповідно множини показників за рівністю доменів відповідних атрибутів а не за назвою імен. Також До схеми результуючого відношення при застосуванні оператора об'єднання автоматично додається ще одна міра – u_p, значення якої визначає до якого показника належить кортеж: до p_1, до p_2 чи одночасно до p_1 та p_2: $sh(\cup(p_1,p_2)) = sh(p_1) \cup u_p$.

Визначення 2 дозволяє отримати практичні рекомендації щодо правил побудови вхідного орграфу предметної області: для кожного запиту на базі алгебри показників будь-який оператор виразу алгебри показників будь-якого показника вважається показником. Це, окрім іншого, забезпечує можливість редукції орграфу при побудові тим, що якщо серед показників, що визначені аналітиком, присутній вузол p_j зі структурою як у проміжного вузла k, то в якості такого внутрішнього вузла береться такий знайдений існуючий показник p_j, інакше створюється новий показник $p_k \in P^3$, де P^3 – множина всіх довизначених автоматично показників;

$^{\circ}p$ – множина показників, з яких безпосередньо виводиться (обчислюється) показник p за допомогою відповідної процедури. Помітимо, що $(p \in^{\leq} P \Leftrightarrow {}^{\circ}p = \varnothing)$.

$^{\circ\circ}p$ – множина всіх показників, що приймають участь в актуалізації показника p. Помітимо, що ${}^{\circ}p = \varnothing \Rightarrow {}^{\circ\circ}p = \varnothing$.

$^{<}p$ – множина первинних показників, що впливає на актуальність показника p, тобто ${}^{<}p = {}^{\circ\circ}p \cap^{\leq} P$.

p° – множина показників, при обчисленні яких безпосередньо використовується показник p у відповідних процедурах. Помітимо, що $(p \in P^{\geq} \Leftrightarrow p^{\circ} = \varnothing)$.

$p^{\circ\circ}$ – множина показників, при обчисленні яких опосередковано використовується показник p.

$p^{>}$ – множина кінцевих показників, в актуалізації яких приймає участь показник p;

$P_j^{\square}$ – множина показників, що запитані в результаті запиту до довільного показника j;

$P^{\downarrow}$ – множина показників, що знаходяться в неактуальному стані;

$P^{\uparrow}$ – множина актуальних показників;

$p^{\square}$ – факт безпосереднього запиту до показника p;

$p^{\square\square}$ – факт опосередкованого запиту до показника p через інші показники;

$p^{\downarrow}$ – факт неактуальності показника p;

$p^{\uparrow}$ – факт актуальності показника p (всі віртуальні показники вважаються **не**актуальними).

Функціонування БД є чергування процесів актуалізації та деактуалізації. В класі БД, що досліджується, актуалізація ініціюється завершенням актуалізації первинних показників. Якщо актуалізовано показник $p, p \in^{\leq} P$, то далі поетапно актуалізуються показники з множини $p^{\circ\circ} \cap P^x$. Таким чином, при зверненні до будь-якого показника користувачем всі показники, окрім показників відкладеної матеріалізації P^{μ}, мають бути актуалізовані: $P_j^{\square} \cap (P^{\downarrow} \setminus P^{\mu}) = \varnothing$ або знаходитись в стані актуалізації.

Деактуалізація в класі БД, що розглядається, виконується подібним чином. Якщо внесено зміни до показнику $p \in^{\leq} P$, то показники $P^x \cap p^{\circ\circ}$ оголошуються неактуальними, тобто

$$P^{\downarrow} = P^{\downarrow} \cup (P^x \cap p^{\circ\circ})$$

Для строгих оцінок ефективності БД потрібно аксіоматизувати описаний вище порядок обробки запитів (запам'ятовування) та деактуалізації (забуття).

Аксіоми запитів:

A.1. $(\forall p \in P)\left(p^{\square} \Rightarrow \neg p^{\square\square}\right)$

A.2. $\left(\forall p_i, p_j \in P\right)\left(p_j^{\square} \wedge p_i^{\square} \Rightarrow j = i\right)$

A.3. $\left(\forall p_j \in {}^{\circ}p_i, p_i \notin P^x\right)\left(p_i^{\square} \vee p_i^{\square\square} \Rightarrow p_j^{\square\square}\right)$

A.4. $p_j^{\square\square} \Rightarrow \left(\exists p_i \in p_j^{\circ\circ}\right)\left(p_i \notin P^x \wedge p_i^{\square}\right)$

Аксіома А.1. забороняє можливість замкнених циклів в БД при отриманні даних. Аксіома А.2. забороняє одночасні безпосередні запити до двох чи більше різних показників. Аксіома А.3. визначає появу опосередкованого запиту для показника. Аксіома А.4. визначає обов'язкову наявність показника з множини $p_j^{\circ\circ}$ до якого відбувся прямий запит, якщо є опосередкований запит $p_j^{\square\square}$. Аксіоми А.1.–А.4. породжують множину ситуацій запиту (ситуацій, що мають місце в момент початку запиту).

Стан після зміни первинного показника визначається аксіомою актуалізації:

A.5. $\left(\forall p \in^{\leq} P\right)\left(p^{\uparrow} \Rightarrow p^{\circ\uparrow} \wedge p^{\circ\circ\uparrow}\right)$

Аксіоми деактуалізації:

A.6. $\left(\forall p_j, p_i \in^{\leq} P\right)\left(p_j^{\downarrow} \wedge p_i^{\downarrow} \Rightarrow j = i\right)$

A.7. $\left(\forall p \in^{\leq} P\right)\left(p^{\downarrow} \Rightarrow p^{\circ\downarrow} \wedge p^{\circ\circ\downarrow} \Rightarrow p^{\uparrow}\right)$

Аксіома А.6. забороняє одночасну деактуалізацію двох або більше показників. Зазначимо, що деактуалізація первинних показників для великих об'ємів даних може займати багато часу, тому що в БД для відповідних таблиць є індекси, можливо секції, тощо, які в свою чергу зменшують час отримання інформації. Аксіома А.7. визначає перехід показника в неактуальний стан коли він безпосередньо змінився чи хоча б один показник, який приймав участь в його виведенні та визначає наступний запуск актуалізації після завершення процесу деактуалізації.

3. МОДЕЛЬ СТРУКТУРИ БД З АЛЬТЕРНАТИВНИМ ДЖЕРЕЛОМ ДАНИХ

Модель структури БД з альтернативним джерелом даних базується на попередньо описаній моделі та має суттєве розширення – в модель включено можливість альтернативної обчислювальності для підмножини показників, яку визначено далі. Структура даної моделі описана визначенням 1. В даній моделі використовуються такі ж показники, як і в попередній моделі P, характеристики показників (M, R, λ, μ) такі самі, як і в моделі з безальтернативним джерелом даних. Аналогічно використовуються позначення показників, що обчислюються відносно показника p ($^{\circ}p, ^{\circ\circ}p, p^{\circ}, p^{\circ\circ}$), множини первинних та кінцевих показників ($^{\leq}P$ та $P^{\geq}$ відповідно), множини показників, що запитані $P^{\square}$, деактуалізовані $P^{\downarrow}$, актуальні $P^{\uparrow}$. Також використовуються позначення факту безпосереднього та опосередкованого запиту до показника $p^{\square}$ та $p^{\square\square}$ відповідно. В даній моделі актуальні всі формули попередньої моделі та додано ті, що описано нижче.

До структури показників $P(P^x, P^v, P^{\mu})$ та відношення обчислювальності A додамо:

$P^{\square}$ – множина показників альтернативної обчислювальності, які можуть бути джерелом даних для деяких вершин з $^{\circ}P$; $P^{\square} \cap P^x$ – множина показників, які можуть бути джерелом даних для вершин, з яких вони обчислюються при вибірці (не при актуалізації).

До множини аксіом А.1.–А.4. додамо аксіому, які визначає наявність альтернативного джерела даних для деякої підмножини вершин:

A.8. $\forall p : p^{\circ} \in \left(P^{\square} \cap \left(P^x \setminus P^{\mu}\right)\right)$ показник p при наявності факту прямого $p^{\square}$ чи опосередкованого запиту $p^{\square\square}$ може обчислюватись як з попередніх вершин $^{\circ}p$, так і з вершин p°, що визначені аксіомою.

Введення множин показників альтернативної обчислювальності, що є можливим тільки для деякого обмеженого типу вершин, дає можливість обрати джерело даних за критерієм мінімальних витрат при обчисленні даних.

4. ПРАВИЛА ФОРМУВАННЯ РЕБЕР ОБЕРНЕНОЇ ОБЧИСЛЮВАЛЬНОСТІ В ОРГРАФІ ПОКАЗНИКІВ

Кожна вершина орграфу показників отримується з попередніх застосуванням відношення обчислювальності A. Кожний показник з імовірністю, відмінною від нуля, в результаті рішення задачі ВОНЗМ може бути матеріалізований і тому розглянемо можливість обчислення віртуалізованих показників з похідних матеріалізованих. Далі наведено твердження алгебри показників, на базі яких отримується перелік показників, для яких потрібно добудувати обернені ребра в орграфі.

Твердження 4.1. Якщо для деяких показників виконується ліва частина однієї з формул (4.1) – (4.6), то в орграф показників потрібно додати ребро (будемо називати його ребром альтернативного джерела отримання даних),

формула обчислення для якого описується в правій частині відповідної формули, яке можна задіяти для отримання таких даних за умови матеріалізованості та актуальності показників-джерел даних.

$$\left(p_2=\pi_B\left(p_1\right)\right)\wedge\left(sh\left(p_1\right)\subseteq sh\left(p_2\right)\right)\Rightarrow \Rightarrow\left(p_1=\pi_{B(sh(p_1))}\left(p_2\right)\right) \quad (4.1)$$

$$\left(p_2=\tau\left(p_1\right)\right)\wedge\left(p_1=\tau_B\left(p\right)\right)\wedge \wedge\left(B\subseteq sh\left(p_1\right)\right)\Rightarrow\left(p_1=\tau_B\left(p_2\right)\right) \quad (4.2)$$

$$\left(p_2=\tau\left(p_1\right)\right)\wedge \left(\neg\exists p_i: p_1=\tau_B\left(p_i\right)\right)\Rightarrow\left(p_1=p_2\right) \quad (4.3)$$

$$\left(p_2=\sigma_{C_1}\left(p_1\right)\right)\wedge\left(p_1=\sigma_{C_2}\left(p\right)\right)\wedge \wedge\left(C_1\subseteq C_2\right)\Rightarrow\left(p_1=p_2\right) \quad (4.4)$$

$$\left(p_3=p_1\triangleright\triangleleft p_2\right)\wedge\left(shr(p_1)=shr(p_2)\right)\Rightarrow \Rightarrow\left(p_i=\pi_{sh(p_1)}\left(p_3\right), i\in\overline{1,2}\right) \quad (4.5)$$

$$\left(p_3=p_1\cup p_2\right)\wedge\left(sh(p_1)=sh(p_2)\right)\Rightarrow \Rightarrow\left(p_i=\sigma_{p_3^{\arg}=p_i}\left(p_3\right), i\in\overline{1,2}\right)\wedge \wedge\left(sh\left(p_3\right)=sh\left(p_1\right)\cup u_p\right) \quad (4.6)$$

де $sh\left(p\right)$ – схема показника, shr – ключ показника p, u_p - додатковий атрибут.

Доведення. Наведемо доведення твердження для формули (4.2), для всіх інших формул твердження доводиться аналогічно. Згідно твердження показник p_1 є результатом застосування оператора сортування показника p та показник p_2 є результатом застосування оператора сортування над p_1, то p_1 можна отримати з p_2 застосувавши той самий оператор сортування з переліком атрибутів, що відповідає показнику p_1. Доведемо від зворотнього. Нехай виконується ліва частина формули (4.2) та не виконується права, тобто p_1 не можна отримати з p_2, застосувавши оператор сортування з переліком атрибутів, що відповідає показнику p_1. Але за визначенням оператор сортування не змінює кортежів множини [14], а порядок сортування буде збережено при застосуванні того самого оператора сортування τ_B до p_2. Тобто застосувавши оператор τ_B до p_2 отримаємо показник p', еквівалентний p_1 (не враховуючи порядок сортування за атрибутами, що не входять в B), тобто права частина (4.2) виконується■

5. ПОСТАНОВКА ЗАДАЧІ БУЛЬОВОГО ПРОГРАМУВАННЯ

Нехай P^1 – набір всіх первинних показників, тобто $P^1 =^{\leq} P$; P^2 – набір всіх похідних показників, що визначені користувачем; P^3 – набір всіх добудованих показників (тобто таких, які користувач визначив у вигляді формул та не виділив в окремі показники), кожен з яких відповідає одній з вершин показників (очевидно, внутрішній).

$$\left|P^1\right|=n, \left|P^2\right|=m, \left|P^3\right|=l$$

x_i – бульова ознака матеріалізації i-го показника: $\forall i=\overline{1,n}: x_i=1$.

ω_i – бульова ознака відкладеної матеріалізації i-го показника

Визначимо ЦФ задачі, переписавши формулу критерію ефективності, запропоновану в [12] так, щоб враховувати тільки змінні витрати експлуатації системи:

$$c=R^{Sb}+R^{Cb}+R^{\mu b}+M \quad (5.1)$$

де R^{Sb} – вартість простоїв, що викликані порушенням часового обмеження отримання даних, R^{Cb} – вартість простоїв, що викликані порушенням часу актуалізації даних, $R^{\mu b}$ – вартість простоїв, що викликані порушенням часу отримання даних показників відкладеної матеріалізації; M – вартість додаткового сховища для збереження даних. Для $R^{Sb}, R^{Cb}, R^{\mu b}$ та M аналітик визначає деякий проміжок часу (на кожний такий проміжок часу потрібно буде здійснити витрату c). Постійні витрати зменшуються іншими методами, зокрема, використанням самої технології KIT-XXI, що описано в [2].

Деталізуємо ЦФ:

$$c=\sum_{i=1}^{n+m}\max\left(\left(R_i^S-b_i^S\right),0\right)\alpha\lambda_i k_i^S + + \sum_{C\in\{I,U,D\}}\left(\begin{array}{l}\sum_{i=1}^{n}\max\left(\left(R_i^C-b_i^C\right),0\right)\alpha\mu_i k_i^C + \\ + \sum_{p_j\in P^\mu}\max\left(\left(R_j^C-b_j^{C\mu}\right),0\right)* \\ *\alpha\mu_i^\mu k_i^\mu\end{array}\right)+ \sum_{i=1}^{n+m+l} m_i x_i\beta \quad (5.2)$$

де $R_i^S, R_i^I R_i^U, R_i^D$ – час виконання операції над показником відповідно вибірки, вставки, онов-

лення, видалення; k_i – ваговий коефіцієнт важливості забезпечення швидкодії роботи відповідного показника для операцій вибірки, вставки, оновлення, видалення; α – усереднена вартість простоїв для підприємства кожної часової одиниці.

Математична постановка задачі ВОНЗМ на орграфі алгебри показників наступна: для заданого набору запитів БД (P^1, P^2), частот роботи з ними λ_i , μ_i , (вибірки, зміни даних), при заданих обмеженнях на час b_i^S, b_i^C (вибірки, зміни даних), заданих зважених значень порушень оперативності реагування показників k_i^S, k_i^C (вибірки, зміни даних), вартості часу користувачів α, простій яких відбувається в разі порушення часового регламенту, при заданих періоді генерування T^{gen}, значені вартості зовнішньої пам'яті β, векторі можливості відкладеної матеріалізації M та часовому обмеженні на таку відкладену матеріалізацію B^{μ}, знайти такий вектор матеріалізації X, за якого значення функції сумарної зваженої вартості змінних витрат експлуатації системи (5.2) буде мінімальним.

6. ВАРТІСНІ ОЦІНКИ ПАРАМЕТРІВ МОДЕЛІ ДЛЯ ЗАДАЧІ З БЕЗАЛЬТЕРНАТИВНИМ ДЖЕРЕЛОМ ДАНИХ

Вартість вибірки даних для i-ї вершини графу залежить від значення бульової змінної матеріалізації:

$$R_i^S = R_i^{SM} x_i + (1 - x_i) R_i^{SN} \tag{6.1}$$

Де R_i^{SM} – вартість вибірки даних зі сховища, за умови, що показник матеріалізовано; R_i^{SN} – вартість виконання запиту алгебри показників вибірки даних за умови, що показник не матеріалізовано і виконується вибірка даних з показників; загальна формула для отримання значення R_i^{SN} для показника p_i, аргументами якого є показники p_j, p_k (або тільки p_j):

$$R_i^{SN} = a_i + b_{ji} R_j^S + b_{ki} R_k^S + c_i R_j^S R_k^S$$

Значення $R_i^C = f(X, \mu, V)$ обчислюється за формулою:

$$R_i^C = \sum_{\forall p_j \in p_i^{\circ\circ}} R_j^{SN} x_j \mu_j \tag{6.2}$$

7. ВАРТІСНІ ОЦІНКИ ПАРАМЕТРІВ МОДЕЛІ ДЛЯ ЗАДАЧІ З АЛЬТЕРНАТИВНИМ ДЖЕРЕЛОМ ДАНИХ

Для моделі з альтернативним джерелом даних процеси актуалізації ідентичні із процесами актуалізації в моделі з безальтернативним джерелом даних, R^C рахується також за формулою (6.2), причому R_i^S у формулі (6.2) обчислюється за формулою (6.1), а R_i^S для визначення вартості вибірки обчислюється з врахуванням альтернативної обчислювальності:

$$R_i^S = \min \begin{pmatrix} R_i^{SM} x_i + (1 - x_i) R_i^{SN}, \\ \left(R_{j_1}^{back} x_{j_1} + (1 - x_{j_1}) L\right), ..., \\ \left(R_{j_p}^{back} x_{j_p} + (1 - x_{j_p}) L\right) \end{pmatrix}, \quad \forall j_q : \exists V(j_q, i) \in V^{back} \tag{7.1}$$

де L – достатньо велика за значенням константа, щоб забезпечити умову невикористання оберненого ребра, якщо наступну вершину не матеріалізовано:

$$L > \max\left(R_i^{SM}, R_i^{SN}\right)$$

тобто вартість вибірки визначається як найменше з вартості безпосередніх обчислень та всіх можливих вартостей альтернативних обчислень.

Перепишемо формулу (5.2) з врахуванням оберненої обчислювальності так:

$$c = \sum_{i=1}^{n+m} \max\left(\left(R_i^{\min} - b_i^S\right), 0\right) \alpha \lambda_i k_i^{\lambda} + \sum_{C \in \{I,U,D\}} \sum_{i=1}^{n} \max\left(\left(\sum_{\forall p_j \in p_i^{\circ\circ}} R_j^{SN} x_j \omega_j - b_i^C\right), 0\right) \cdot \alpha \mu_i k_i^{\mu} + \sum_{i=1}^{n+m+l} m_i x_i \beta \tag{7.2}$$

$$R_i^{\min} = \min \begin{pmatrix} R_i^{SM} x_i + (1 - x_i) R_i^{SN}, \\ \left(R_{j_1}^{back} x_{j_1} + (1 - x_{j_1}) L\right), ..., \\ \left(R_{j_p}^{back} x_{j_p} + (1 - x_{j_p}) L\right) \end{pmatrix} \tag{7.3}$$

Твердження 7.2. Для циклічного орграфу G', який отримано в результаті додавання обернених ребер застосуванням правил (4.1) – (4.6), виконується:

$$\forall X_i \in X : c'(X_i) \le c(X_i) \qquad (7.4)$$

де $c'(X_i)$ – значення ЦФ, побудованої на графі G' за формулою (7.2), а $c(X_i)$ – значення ЦФ, побудованої на графі G за формулою (5.2).

Доведення. Розглянемо можливі варіанти значення матерілаізації вектору рішення X і їх вплив на значення ЦФ. Розглядаючи всі чотири варіанти сумісної матеріалізації показників p_1, p_2 легко бачити, що тільки для варіанту матеріалізації, за якого $x_1 = 0, x_2 = 1$, обернене ребро може бути використане для отримання даних. За визначенням ЦФ c та c' ((5.2) та (7.2) відповідно) відрізняються тільки в частині суми вартості вибірки показників R^S. Тому для доведення (7.4) потрібно показати, що

$$\max\left(\left(\left(R_i^{SM}x_i + \left(1-x_i\right)R_i^{SN}\right) - b_i^S\right), 0\right) \ge$$

$$\max\left(\left(\min\begin{pmatrix} R_i^{SM}x_i + \left(1-x_i\right)R_i^{SN}, \\ \left(R_{j_1}^{back}x_{j_1} + \left(1-x_{j_1}\right)L\right), \\ \ldots, \\ \left(R_{j_p}^{back}x_{j_p} + \left(1-x_{j_p}\right)L\right) \end{pmatrix} - b_i^S\right), 0\right) \qquad (7.5)$$

Якщо для вершини не знайдеться оберненого ребра, для якого

$$\max\left(R_i^{SM}x_i + \left(1-x_i\right)R_i^{SN}\right) > \max\left(\min\left(R_{j_k}^{back}x_{j_k} + \left(1-x_{j_k}\right)L\right)\right) \qquad (7.6)$$

то у нерівності (7.5) ліва і права частини будуть рівні між собою, а тому виконується нерівність (7.4). Якщо ж знайдеться ребро для якого виконується умова (7.6), а отже і (7.5) то, відповідно, (7.4) виконується також, що й потрібно було довести. ■

Твердження 7.3. Якщо для деякого ребра $< p_{j_k}, p_i >$ виконується

$$\min(R_i^{SM}, \overline{R_i^{SN}}) < R_{j_k}^{back} \qquad (7.7)$$

де $\overline{R_i^{SN}}$ – верхня оцінка значення R_i^{SN}:

$$\overline{R_i^{SN}} = R_i^{SN}\left(x_g = 0\right), \forall p_g \in {}^{\circ\circ}p_i \qquad (7.8)$$

то для жодного вектору рішень X це ребро не може бути обране до використання для обчислення значення відповідного показника.

Доведення. Доведення від зворотнього. Нехай для ребра $< p_{j_k}, p_i >$ виконується умова (7.7) і для обчислення вершини p_i було обрано це ребро $< p_{j_k}, p_i >$, якому відповідає доданок формули (7.3) $\left(R_{j_k}^{back}x_{j_k} + \left(1-x_{j_k}\right)L\right)$ ЦФ (7.2). але з умови (7.7) випливає, що для обчислення з мінімальним часом не потрібно обирати ребро $R_{j_k}^{back}$, а потрібно обирати ребро i ніколи не буде обране ребро $< p_{j_k}, p_i >$. Отже, прийшли до протиріччя, що й треба було довести. ■

Запишемо алгоритм додавання обернених ребер наступним чином.

Алгоритм 1 – Додавання обернених ребер до графу G

1 Сформувати множину правил C, для отримання ребер альтернативного джерела даних

2 **for** (для) кожної вершини $p_i \in P[G]$

3 **if**

$$\left(\exists c \in C : c\left(p_{\,i} = p_2, p_j = p_1\right) = true\right) \wedge \wedge \neg\left(\min(R_i^{SM}, \overline{R_i^{SN}}) < R_{j_k}^{back}\right)$$

4 **then** $V[G] = V[G] \cup < p_j, p_i >, a_{ji} = R_{ji}^S$

На поточний момент множину правил C складають правила (4.1) – (4.6). Зазначимо, що саме твердження (7.2) дає можливість на етапі 3 алгоритму 1 відсікти непотрібні обернені ребра, тобто ті, які ніколи не призведуть до покращення рішення.

7. ПЛАН ТА РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТІВ

Метою експерименту є з'ясування області ефективного застосування запропонованих в роботі підходів за допомогою статистичної оцінки наступних параметрів: 1) відсоток ребер, для яких спрацювали правила побудови ребер оберненої обчислювальності; 2) відсоток задіяних обернених ребер; 3) відношення значень ЦФ наближених рішень задачі на графі з альтернативним джерелом даних та на графі з безальтернативним джерелом даних при одних і тих самих векторах X.

Обчислювальний експеримент проводився на випадкових, штучно згенерованих даних. Розмірності даних, на яких проводяться випробування, наведені у таблиці 1 разом з результатами експериментів. Графічне відображення результатів експерименту наведено в на рисун-

ку 1.

Табл. 1. Результати проведення експерименту для визначення ефективності використання задачі з альтернативним джерелом даних

#	Середня кількість вершин					Всього наборів	% визнач. оберн. ребер	% використ. оберн. ребер в найкр. ріш-ні	% покращення ЦФ при вик. оберн. задачі ГА
	P1	P2	P3	Кінц.	Всього				
1	3	4	3	2	10	40	42,28	11,92	70,63
2	4	7	4	2	15	40	35,48	8,44	75,38
3	4	8	6	2	18	40	40,63	8,56	80,7
4	5	11	10	2	25	40	38,67	8,76	83,7
5	10	22	18	3	50	40	43,12	9,9	89,43
6	10	25	25	4	60	40	44,76	9,94	93,25
7	20	42	38	6	100	40	44,18	9,71	90,32
8	30	89	81	9	200	40	42,52	9,75	88,05
9	100	155	145	14	400	40	44,92	10,37	89,55
10	50	153	297	18	500	40	43,57	4,59	95,3
11	50	281	269	19	600	40	43,91	0,52	93,15
12	200	418	382	32	1000	40	45,35	0,97	91,57
13	300	921	779	145	2000	40	42,14	1,67	93,41
14	200	1105	1195	109	2500	40	42,85	2,39	98,45
15	500	2407	2093	299	5000	40	42,55	1,64	99,39
16	2500	2762	2238	527	7500	40	42,72	0,53	98,49
17	200	2150	7650	495	10000	40	41,61	1,2	99,99
						min:	**35,48**	**0,52**	**70,63**
						max:	**45,35**	**11,92**	**99,99**
						avg:	**42,43**	**5,93**	**90,04**

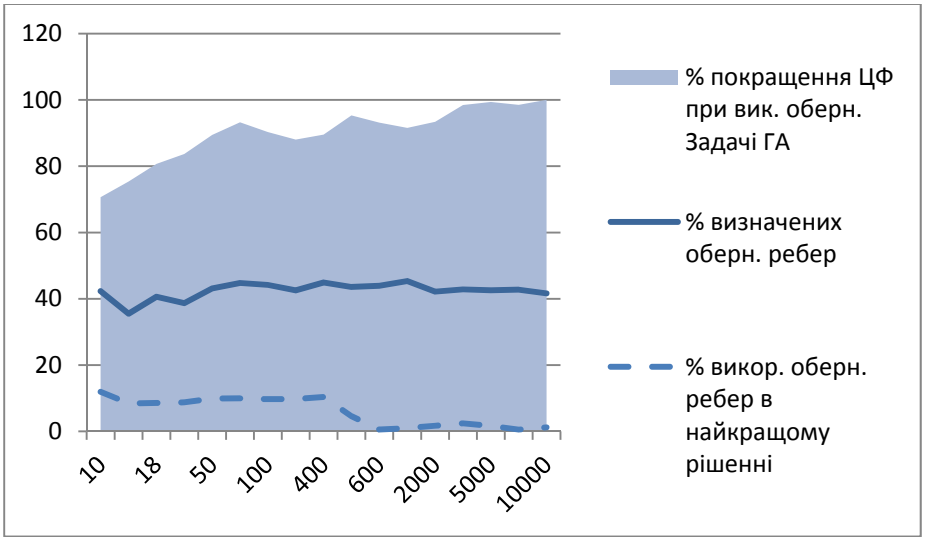

Рис. 1. Графічне відображення результатів проведення експерименту з визначення ефективності використання задачі з альтернативним джерелом даних

Для кожного графу було згенеровано відповідну кількість вершин, ребер, коефіцієнти для вершин та ребер. Оцінки потужностей вершин визначались за допомогою загальних правил, описаних в [14].

Випробування були проведені на всіх комплектах експериментальних даних. В якості наближеного алгоритму для рішення було використано генетичний алгоритм. Вибір саме цього алгоритму був зумовлений його широким

розповсюдженням при вирішенні задач оптимізації в БД[15,16]. Для визначення області ефективного застосування генетичного алгоритму для рішення даної задачі було проведено попередній експеримент для визначення ефективних параметрів алгоритму.

Результати проведеного експерименту свідчать про високу доцільність використання математичної моделі з альтернативним джерелом даних. Час на додавання обернених ребер до графу менше $O\left(n^A\right)$, де n^A – кількість всіх ребер в графі, і не впливає на результати експерименту. Тому при однакових витратах часу та однакових характеристиках генетичного алгоритму було отримано покращення значення ЦФ в середньому на 90,04, при середній кількості визначених до використання ребер – 5,93% та 42,43% усіх визначених обернених ребер згідно правил. Для вхідних даних задачі ефективність застосування суттєво збільшується при збільшенні розмірності задачі, хоча частка задіяних обернених ребер і значно зменшується при цьому, що значно розширює область ефективного застосування запропонованого підходу.

МАЙБУТНЯ РОБОТА

Дана робота є частиною робіт зі створення інтелектуальної технології розробки та впровадження ІСУБ на базі технології «КІТ-XXI» з функціями OLTP та OLAP. Найближчим часом планується опублікувати результати з дослідження ефективних методів рішення задачі ВОНЗМ, що використовують моделі, описані в даній роботі.

ВИСНОВКИ

1. Узагальнено дослідження вітчизняних та зарубіжних видань на тему ВОНЗМ. Аналіз підтверджує актуальність вирішення поставленої задачі та недостатність існуючих методів для ефективного рішення задачі. Окрім іншого, в описаних методах не використовуються семантичні знання про структуру ІСУБ.

2. Для вирішення поставленої задачі було сформульовано та аксіоматизовано дві математичні моделі (формально та у вигляді бульової задачі), що є розширенням моделі параметричної БД Гриші С.М [12]: модель з безальтернативним та модель з альтернативним джерелом даних. Було визначено елементи відношення обчислювальності в моделі БД; в моделі використовується невідкладний режим актуалізації БД; в моделі визначено множину показників відкладеної актуалізації з обмеженням на максимальний час матеріалізації. Друга модель з альтернативним джерелом даних базується на попередньо описаній моделі та має суттєве розширення – в модель включено можливість оберненої обчислювальності для підмножини показників.

3. Запропоновано метод визначення обернених ребер в орграфі з використанням правил алгебри показників.

4. Результати проведеного експерименту свідчать про високу доцільність використання математичної моделі з альтернативним джерелом даних. При однакових витратах часу та однакових характеристиках генетичного алгоритму, що використовувався для пошуку рішення, було отримано покращення значення ЦФ в середньому на 89,58, при середній кількості визначених до використання ребер – 46,22% з 42,18% усіх визначених обернених ребер. Для вхідних даних задачі, ефективність застосування суттєво не змінюється для різних розмірностей задачі, що значно розширює область ефективного застосування запропонованого підходу.

ЛІТЕРАТУРА

1. Формализация процедур обработки данных в АСУ на основе алгебры показателей : материалы 3-й Советско-Польской научно-технической конференции «Комплексная автоматизация промышленности» / Гриша С.Н., Фаловский А.А., Малькевич А.Б. – ПНР, Вроцлав, 1988. – т.2 с.131-134.
2. Гриша С.М. Технологічно інтелектуалізовані інформаційні системи для управління бізнесом (ІСУБ) на основі алгебри показників / Гриша С.М., Родічева О.С., Приліпко Д.І. // Вісник Національного Технічного Університету «ХПІ». – 2008. – №5. – С.123-133.
3. Agrawal S. Automated selection of materialized views and indexes in Microsoft SQL Server / Agrawal S, Chaudhuri S, Narasayya V // In: Proc VLDB. – Cairo, Egypt, 2000. – pp 496-505.
4. Gupta H. Index selection for OLAP / Gupta H, HarinarayanV, Rajaraman A, Ullman JD. // In: Proc ICDE. – 1997. – pp 208-219.
5. Harinarayan V. Implementing data cubes efficiently/ Harinarayan V, Rajaraman A, Ullman JD. // In: Proc

SIGMOD. – 1996. – pp. 205-216.
6. Karloff HJ. On the complexity of the view selection problem / Karloff HJ, Mihail M. // In: Proc PODS. – Philadelphia, Penn., USA, 1999. – pp 167-173.
7. Yang J. Algorithms for materialized view design in data warehousing environment / Yang J, Karlapalem K, Li Q. // In: Proc VLDB. – Athens, Greece, 1997. – pp 136-145.
8. U.S. Chakravarthy "Logic-based Approach to Semantic Query Optimization," / U.S. Chakravarthy, J. Grant, and J. Minker //ACM Trans. Database Systems. – 1990. –vol. 15, no. 2. – pp. 162-207.
9. Rada Chirkova. Designing Views to Answer Queries under Set, Bag,and BagSet Semantics / Rada Chirkova, Foto Afrati, Manolis Gergatsoulis, Vassia Pavlaki // Proceedings of 6th International Symposium, SARA. – Scotland, UK, 2005. – pp. 332-350.
10. Rada Chirkova. A Formal Perspective on the View Selection Problem / Rada Chirkova , Alon Halevy , Dan Suciu // Proc. of the Int. Conf. on Very Large Data Bases. –2002. – pp. 216-237.
11. V. Markl. LEO: An autonomic query optimizer for DB2 / V. Markl, G. M. Lohman, V. Raman // IBM SYSTEMS JOURNAL. – VOL 42, NO 1. – 2003. – pp. 98-106.
12. Гриша С.Н. Информационно-стоимостной анализ и синтез моделей компьютеризованного управления производственными системами : Дис. ... док. техн. наук.: 05.13.06 / Гриша Сергей Николаевич / Киевский политехнический институт. – К., 1991. – 340 с.
13. Rittman Mead – Delivered Intelligence. OBIEE 11.1.1.5 and Oracle OLAP Support http://www.rittmanmead.com/2011/05/untitled-1/
14. Гектор Гарсиа-Молина. Системы баз данных. Полный курс / Гектор Гарсиа-Молина, Джеффри Д. Ульман, Дженифер Уидом.; Пер. с англ. – М.: Издательский дом «Вильямс», 2003. – 1088 с. – ISBN 5-8459-0384-X.
15. Kristin Bennett. A Genetic Algorithm for Database Query Optimization. / Kristin Bennett, Michael C. Ferris, Yannis Ioannidis // 4th Int'l Conference on Genetic Algorithms. – San Diego, CA, July 1991. – pp. 400-407.
16. Michael Stillger. Genetic Programming in Database Query Optimization / Michael Stillger, Myra Spiliopoulou // Institut fur Informatik Humboldt-Universitat zu Berlin, 1996.