https://doi.org/10.31891/2219-9365-2022-72-4-17
УДК 519.7:007:004

Сергій БАБИЧ
ВСП Рівненський фаховий коледж НУБіП України
https://orcid.org/0000-0002-8669-7288

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ГЕНЕРАЦІЇ МАТРИЦЬ РОЗКЛАДІВ ЗГІДНО ПЕРМАНЕНТНОЇ ДЕКОМПОЗИЦІЇ

В роботі проведено аналіз відомих підходів, методів та інформаційних технологій для розв'язання задач складання розкладів занять. Актуальність таких задач зумовлюється не лише значною обчислювальною складністю відомих методів, але й наявністю низки додаткових вимог в кожному конкретному випадку, що суттєво впливають на метод розв'язання. Вимоги, що виникають при складанні розкладів, часто є антагоністичними і вимагають певних компромісних рішень. А тому розробка нових підходів та методів розв'язання задач календарного планування є актуальною задачею.

Результатом роботи є створення інформаційної технології складання розкладу занять, в основу якої покладено метод перманентної декомпозиції. Цей метод дозволяє генерувати комбінаторні об'єкти, зокрема, системи різних представників стовпців матриць розкладів, та відрізняється наявністю процедури розкладання модифікованого перманента за рядком із запам'ятовуванням ідентифікаторів елементів матриці інцидентності. При цьому мова йде не про класичний перманент, а про певну його модифікацію, що пропонується в роботі. Також запропоновано поняття модифікованої матриці інцидентності, що дозволяє коректно врахувати інформацію про потоки, та відрізняється введенням додаткових стовпчиків для кожного викладача, що має поточні пари з коректним відображенням інформації про структуру потоку. Процедура декомпозиції перманента застосовується саме до такої матриці інцидентності.

Алгоритм декомпозиції модифікованого перманента включає у себе процедуру формування систем різних представників стовпчиків матриці розкладів, запису індексів стовпчиків в комірки пам'яті та забезпечує можливість одночасної присутності викладача у кількох групах, якщо допускаються потоки. Відповідний алгоритм декомпозиції із "запам'ятовуванням" може бути використаний для розв'язання широкого класу задач генерації комбінаторних об'єктів, зокрема, перестановок, комбінацій, підмножин різних елементів деякої системи множин.

Проведені експерименти та розрахунки обчислювальної складності підтверджують ефективність методу перманентної декомпозиції.

Ключові слова: інформаційні системи, перманент, декомпозиція, розклад, матриці розкладу, інцидентність, ефективність.

Sergii BABYCH
Rivne Vocational College of NUBiP

INFORMATION TECHNOLOGY OF THE TIMETABLE MATRIX GENERATION ACCORDING TO PERMANENT DECOMPOSITION

The paper analyzes known approaches, methods, and information technologies for solving the problems of drawing up class schedules. The relevance of such problems is determined not only by the significant computational complexity of known methods but also by the presence of a number of additional requirements in each specific case, which significantly affect the solution method. The requirements that arise when drawing up schedules are often antagonistic and require certain compromise solutions. Therefore, the development of new approaches and methods for solving calendar planning problems is an urgent task.

The result of the work is the creation of information technology for drawing up a schedule of classes based on the method of permanent decomposition. This method allows you to generate combinatorial objects, in particular, systems of different representatives of the columns of the distribution matrices and is distinguished by the presence of a procedure for decomposing the modified permanent by row with memorization of the identifiers of the elements of the incidence matrix. At the same time, we are not talking about a classic permanent, but about a certain modification of it, which is proposed in the work. The concept of a modified incidence matrix is also proposed, which allows to correctly take into account the information about flows, and is distinguished by the introduction of additional columns for each teacher who has current pairs with the correct display of information about the structure of the flow.

The modified permanent decomposition algorithm includes a constructive procedure of forming systems of various representatives of the columns of the schedule matrix and recording column indices in memory cells and provides the possibility of the simultaneous presence of the teacher in several groups if flows are allowed. The appropriate decomposition algorithm with "memorization" can be used to solve a wide class of problems of generating combinatorial objects, particularly permutations, combinations, and subsets of various elements of some system of sets.

Conducted experiments and calculations of computational complexity confirm the effectiveness of the permanent decomposition method.

Keywords: information systems, permanent, decomposition, schedule, schedule matrices, incidence, efficiency.

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Задачі календарного планування, зокрема, складання розкладів, сьогодні не втрачають своєї актуальності, незважаючи на наявність великої кількості теоретичних результатів і практичних підходів до їх вирішення, напрацьованих щонайменше протягом останніх 100 років. І це не дивно, адже проблема складання розкладу – одна з найскладніших задач прикладної математики. Вона характеризується наявністю багатьох критеріїв, вага яких на практиці може бути різною навіть для схожих на перший погляд задач, необхідністю розв'язувати проблему впорядкування значних за розміром дискретних множин, що призводить до виникнення алгоритмів експоненційної складності, відповідні задачі часто є NP-повними.

На практиці іноді необхідно враховувати і людський фактор, психолого-педагогічні особливості, що призводить до проблем виключно формалізованої постановки задачі та автоматизації процесу її оптимального вирішення. Тут виникає необхідність у застосуванні експертних оцінок, певних евристичних підходів. Складність відповідних точних оптимізаційних алгоритмів при розв'язанні задач квадратичного програмування, що часто тут виникають, досліджувалась в низці робіт [1]-[2]. Вона зумовлює широке застосування певних альтернативних підходів, де пропонувалось знаходження близького до оптимального розв'язання за допомогою певних евристичних алгоритмів [3]. Так, наприклад, останніми роками для розв'язання задач календарного планування, зокрема, складання розкладів занять ЗВО, почали широко застосовуватись генетичні алгоритми, які, як відомо, дають розв'язок, близький до оптимального, та вимагають окремого дослідження їх ефективності, збіжності [4], [5]-[6].

Розв'язання задач календарного планування, зокрема, складання розкладів занять, зустрічей вимагає застосування нестандартних методів, оригінального творчого мислення і глибокого розуміння суті й складності проблеми. Але для організацій типу закладів вищої освіти вирішення цієї проблеми вкрай необхідно, оскільки розклад або графік є основним інструментом управління часом, від нього безпосередньо залежить, наприклад, продуктивність праці викладачів і швидкість навченості студентів, а значить і ефективність ЗВО в цілому.

В зв'язку з тим, що в процесі календарного планування виникають задачі, які характеризуються багатокритеріальністю, важкоформалізованістю, значною обчислювальною складністю (вони є NP-повні) та розв'язання яких класичними методами часто вимагає залучення значних обчислювальних ресурсів та адаптації алгоритмів з врахуванням особливостей кожної конкретної задачі, виникає необхідність у розробці інформаційної технології, яка дозволила б підвищити ефективність існуючих технологій календарного планування шляхом розробки нових методів аналізу вхідних даних, пріоритизації вимог та побажань стейкхолдерів та нових швидких алгоритмів генерації комбінаторних об'єктів.

Аналіз досліджень та публікацій

Підґрунтям даного дослідження є праці як вітчизняних так і зарубіжних дослідників. Проблема календарного планування була предметом активних досліджень науковців у всьому світі, що призвело до наявності вражаючої кількості наукових публікацій в даній галузі. Щоб оцінити їх, недостатньо простого огляду, а необхідно провести глибокий бібліометричний аналіз [7]. Такий аналіз стосовно задач календарного планування був здійснений у роботах A. Verbeek [8], S.D. Dao [9], I. Rahimi [10], H.Y. Shishido and J.C. Estrella [11], J. Yu [12], де проведено, зокрема, аналіз генетичних алгоритмів, сіток і хмар, технологій Cloud (технології та генетичні алгоритми недомінованого сортування NSGA II). Систематичний огляд допомагає отримати нове розуміння та розкрити значущі знання на основі накопичених даних дослідницької галузі [7].

Проводячи огляд статей та робіт, що перекликаються із задачами календарного планування, варто зауважити, що більшість «слабких місць» та описаних, не вирішених, проблем – це наявність критеріїв, що унеможливлюють використання «єдиновірного» алгоритму. Дану концепцію можна простежити в роботі [13]. Автори вказали на проблему створення розкладу для співробітників, з метою обслуговування клієнтів, з врахуванням навичок фахівців. В цілях аргументації доцільності було наведено важливість оптимізаційного планування як проблему комбінаторної оптимізації NP-складної задачі. У деяких ситуаціях існують додаткові обмеження навіть на сумісність навичок.

Якщо провести певну класифікацію найбільш поширених відомих підходів до розв'язання задач складання розкладів, то можемо виділити, зокрема, оптимізаційні алгоритми на основі класичних методів теорії розкладів (обслуговуючий пристрій тут-аудиторія), еволюційні та генетичні алгоритми, нейромережеві підходи та машинне навчання, евристичні підходи, методи комбінаторної оптимізації.

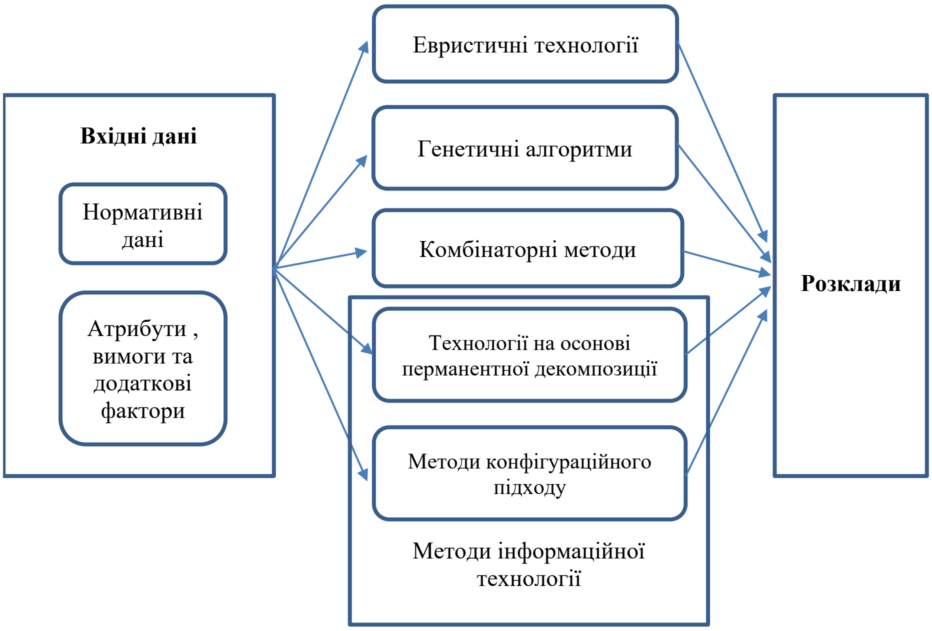

Рис. 1. Класифікація методів розв'язання задач складання розкладів

В даній роботі пропонується інформаційна технологія, яка ґрунтується на основі методу перманентної декомпозиції.

Інформаційна технологія та метод перманентної декомпозиції

В основі інформаційної технології, що пропонується, покладено поняття системи різних представників множин, утворених елементами стовпчиків матриці розкладу (СРПС), особливі способи представлення вхідної інформації на основі алгебраїчних структур - спеціальних матриць інцидентності (без використання додаткових баз даних) а також алгоритми генерації СРПС, що грунтуються на процедурах декомпозиції спеціальним чином модифікованих перманент матриць інцидентності.

Нехай маємо довільну матрицю денного розкладу, розмірністю 3xn. Розглянемо стовпчики матриці розкладу $R_1, R_2, \ldots R_n$. Модифікована матриця інцидентності будується наступним способом. По горизонталі зображуються номера викладачів, по вертикалі – групи, у яких проводяться заняття. Кожному викладачу ставиться у відповідність стовпчик матриці, у якому записуються нулі та одинички в залежності від того, чи має викладач пари у відповідних групах. Причому, якщо якийсь викладач x має поточну пару, то виділяємо йому окремий стовпчик матриці інцидентності, позначивши його x^p (можна використовувати індекс, що є кількістю елементів потоку) [14]. Таким чином, утворюється матриця виду:

$$A = \begin{array}{c} \\ R_1 \\ R_2 \\ \ldots \\ R_m \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & \ldots & x_n \end{array} \\ \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \ldots & \ldots & & \ldots \\ a_{m1} & a_{m2} & \ldots & a_{mn} \end{pmatrix} \end{array}, \qquad (1)$$

де $\alpha_{ij} = \begin{cases} 1, & \text{коли } x_i \in R_j, \\ 0 & \text{в інакшому випадку}. \end{cases}$

Наприклад, для матриці розкладів $R = \begin{pmatrix} 1 & 1 & 1 & 2 & 3 & 5 \\ 2 & 3 & 1 & 4 & 4 & 4 \\ 3 & 1 & 4 & 5 & 2 & 4 \end{pmatrix}$

отримуємо наступну матрицю інцидентності:

$$\begin{pmatrix} & 1 & 1^p & 2 & 3 & 4 & 4^p & 5 \\ R_1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ R_2 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ R_3 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ R_4 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ R_5 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ R_6 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}, \qquad (2)$$

де 1^p та 4^p - індекси відповідних викладачів, що мають "поточні" пари.

Особливістю нашої інформаційної технології є те, що матриці інцидентності вже містять у собі додаткові обмеження, наприклад, інформацію про потоки, а алгоритми знаходження їх модифікованих перманент дозволяють побудувати конструктивні способи запису в пам'ять відповідних СРПС. Таким чином, окремі проблеми тут вирішуються автоматично лише за рахунок відповідної організації структур даних, як, наприклад, проблема "потоків", що дозволяє уникнути громіздких додаткових обчислювальних процедур.

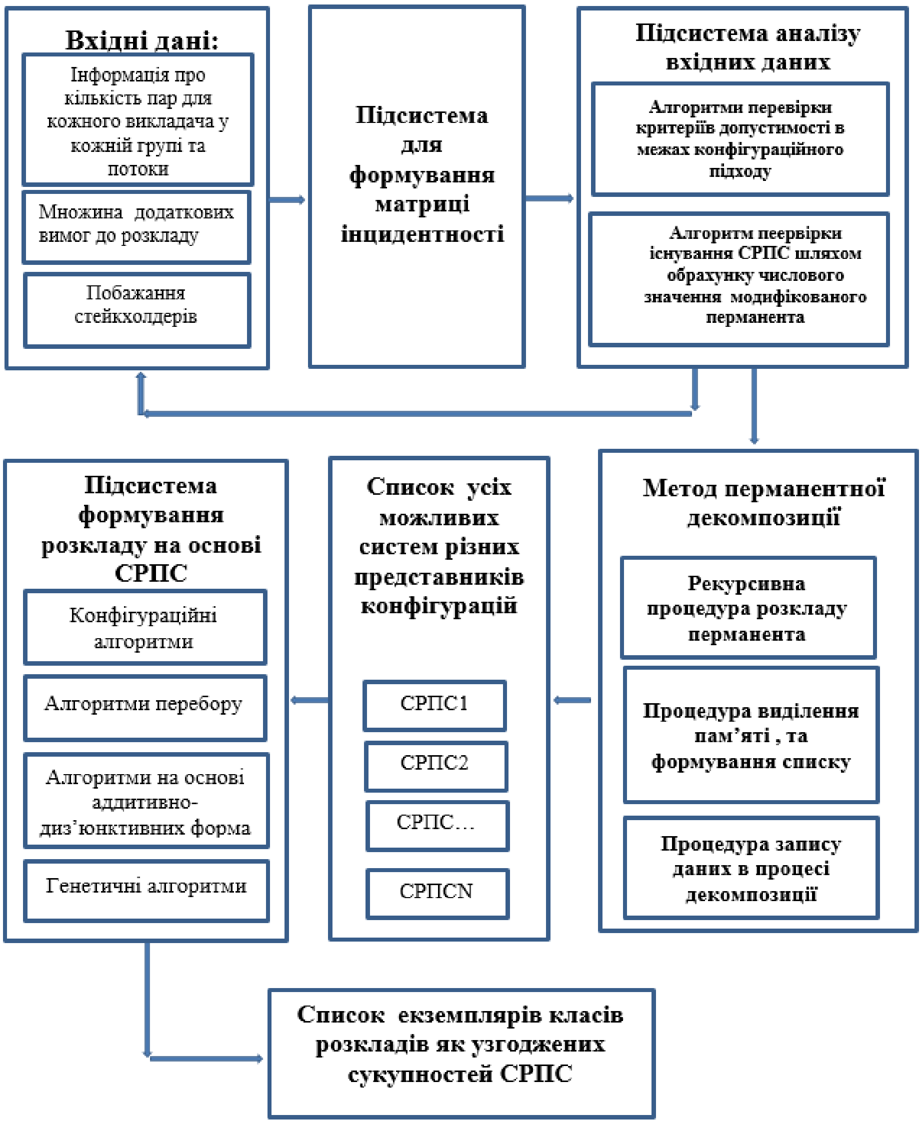

Рис. 2. Інформаційна технологія складання розкладу згідно перманентної декомпозиції

Метод перманентної декомпозиції [15] адаптований до роботи в "парі" з відповідною матрицею інцидентності. В той же час, алгоритм розкладу перманента дозволяє безпосередньо здійснювати запис в потрібну комірку пам'яті необхідного значення, що дозволяє зекономити навіть на пошуку позиції в структурах даних для запису елемента–адреса тут генерується безпосередньо в процесі рекурентної процедури декомпозиції. Таким чином, бачимо, що тут мова йде про цілісну систему, елементи якої тісно пов'язані та адаптовані для оптимальної роботи один до одного.

Ідея застосування перманента як основи алгоритмів генерації, що розглянуті в даній роботі, виникла за рахунок розгляду певних алгебраїчних підходів, зокрема, процедур розкладу перманент та визначників матриць, саме з інформаційної точки зору, з точки зору структур даних. В самому простому виразі вона полягає в тому, що до стандартної процедури розкладу перманента за рядком з метою обчислення самого значення перманента додається пам'ять, що запам'ятовує номери стовпців в процесі розкладу. Метод дозволяє отримати не саме значення перманента, яке нас тут зовсім не цікавить, а алгоритм, що дозволяє, наприклад, згенерувати усі перестановки індексів стовпчиків у лексикографічному порядку (якщо є квадратна матриця інцидентності, що складається лише з одиничок). Сам алгоритм тут досить жорстко диктує вимоги до структур даних, які можуть бути використані для програмної реалізації: має забезпечуватись можливість прямого запису біжучого індексу розкладу у необхідну комірку пам'яті. Як розглядалось вище, це реалізовано через

Означення: модифікованим перманентом матриці інцидентності будемо називати суму всіх можливих добутків елементів матриці, кожен з яких містить по одному елементу з кожного рядка та з різних стовпчиків, причому елемент потокового стовпчика (стовпчика, що відповідає потоковому елементу) не може бути в добутку разом з елементами інших рядків, що відповідають цьому ж потоку.

Зауважимо, що за відсутності потокових елементів модифікований перманент є звичайним перманентом.

Для знаходження модифікованого перманента матриці інцидентності можемо використовувати розклад за рядком. Виходячи з означення, процедура розкладу буде наступною: ненульовий елемент рядка множиться на модифікований перманент матриці, утвореної за наступними правилами – якщо елемент рядка належить потоковому стовпчику, то матриця утворюється з вихідної викреслюванням стовпчика, де цей елемент знаходиться, та всіх рядків, що відповідають всім елементам даного потоку. Якщо елемент рядка не належить потоковому стовпчику, то матриця утворюється шляхом викреслювання рядка та стовпчика, де стоїть цей елемент, а також всіх потокових стовпчиків, на перетині яких з цим рядком стоять ненульові елементи. Відмітимо й те, що вилучення всіх наявних елементів дасть у кінцевому результаті одиницю.

Розглянемо процедуру обчислення модифікованого перманента для матриці розкладів [16]. Нехай маємо матрицю інцидентності виду:

$$A=\begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}. \quad (3)$$

Тут потокові стовпчики -2-й та 4-й . При обраханку перманента будемо будувати розклад за першим рядком . Перша ітерація розкладу виглядає так:

$$permodA = 1 * permod\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} + 1 * permod\begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} +$$

$$+1 * permod\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}. \quad (4)$$

Далі процедура розкладу здійснюється аналогічно.

Для побудови процедури генерації усіх можливих СРПС будемо використовувати процедуру перманентної декомпозиції. Зауважимо, що кожній СПРС відповідає підматриця матриці інцидентності з одиничним перманентом. Крім того, саму СПРС можна отримати, якщо ввести процес "запам'ятовування" елемента стовпчика, що відповідає одиниці, яка використовується при побудові добутку в даний момент та рядка, в якому ця одиниця стоїть у вихідній матриці інцидентності. У зв'язку з цим в процесі обчислення перманента будемо кожній одиничці дописувати два індекси: верхній – елемент, що відповідає стовпчику, де стоїть ця одиничка, нижній-номер рядка, в якому ця одиниця стоїть у вихідній матриці.

Нехай, наприклад, маємо матрицю розкладу виду:

$$R = \begin{pmatrix} 2 & 2 & 2 \\ 0 & 3 & 2 \\ 3 & 2 & 4 \end{pmatrix} \quad (5)$$

Тоді матриця інцидентності має вигляд:

$$\begin{pmatrix} & 2 & 2^n & 0 & 3 & 4 \\ R_1 & 0 & 1 & 1 & 1 & 0 \\ R_2 & 1 & 1 & 0 & 1 & 0 \\ R_3 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \quad (6)$$

Побудуємо процес розкладу модифікованого перманента з "запам'ятовуванням" за першим рядком:

$$per\ mod\begin{pmatrix} 2 & 2^n & 0 & 3 & 4 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix} = 1^{2^p} * 1 + 1^0 \ * per\ mod\begin{pmatrix} 2 & 3 & 4 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} + 1^3 \ * per\ mod\begin{pmatrix} 2 & 0 & 4 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$
$$= 1^{2^n} * 1 + 1^0 \ * \left(1^2 \ per\ mod\begin{pmatrix} 3 & 4 \\ 0 & 1 \end{pmatrix} + 1^3 \ per\ mod\begin{pmatrix} 2 & 4 \\ 1 & 1 \end{pmatrix}\right) +$$
$$+1^3 \ 1^2 \ per\ mod\begin{pmatrix} 0 & 4 \\ 0 & 1 \end{pmatrix} = 1^{2^n} + 1^0 \ 1^2 \ 1^4 \ + 1^0 \ 1^3 \ 1^2 \ + 1^0 \ 1^3 \ 1^4 \ + 1^3 \ 1^1 \ 1^4 \ . \tag{7}$$

Бачимо, що величина модифікованого перманента рівна 5 . Це і це є кількість всіх можливих СРПС. Крім того, відомі й самі СРПС . Вони записуються як верхні індекси "одиничок". Причому потокові елементи повторюються відповідну кількість разів : 222, 024, 032, 034, 314.

Оскільки описаний вище підхід дозволяє побудувати всі можливі СПРС, тобто ми маємо всю інформацію про можливі рядки матриці розкладів, то, очевидно, на етапі вибору певних СРП можемо вирішувати різні задачі, що випливають з певного поняття оптимальності. Очевидно, можемо розглянути низку інших критеріїв, яким повинен задовольняти розклад. Розглянемо, наприклад, критерій K_2, який розглядався в роботі [17]. Його можна розглянути як додаткову умову при знаходженні підмножини СПРС, враховуючи наявність відповідної перестановки СПРС, що автоматично забезпечить розбиття розкладу "по днях".

Таким чином, побудовано новий метод перманентної декомпозиції, в процесі якого формуються усі СРПС в лексикографічному порядку, що дозволяє розробляти ефективні засоби програмної реалізації. Однак, вимагає детальнішого дослідження складність відповідних алгоритмів. Для порівняння розглянемо детальніше підхід до генерації матриць розкладів, що базується на основі відношень порядку.

Зауважимо, що для тижневого розкладу будемо мати матрицю розкладу, що має 15 рядків (5 робочих днів по три пари в день). Стовпчики, очевидно, можуть містити елементи, що повторюються. Тоді при побудові матриці інцидентності необхідно записувати таку кількість стовпчиків, що відповідають повторюваному елементу, скільки разів він зустрічається у стовпчику. Останню обставину необхідно враховувати і у випадку побудови матриці інцидентності денного розкладу, якщо допускається можливість проведення двох практичних у тій самій групі в день, тобто наявність двох однакових непоточних елементів у стовпчику матриці.

Наприклад, для матриці розкладу

$$R = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 3 & 1 \\ 2 & 1 & 4 \end{pmatrix}$$

матриця інцидентності має вигляд:

	1	$1^п$	2	2	3	4
R1	0	1	1	1	0	0
R2	1	1	0	0	1	0
R3	1	1	0	0	0	1

Дослідження ефективності методу перманентної декомпозиції у порівнянні з класичними підходами

Для дослідження ефективності методу перманентної декомпозиції розглянемо деяку тестову задачу, наприклад задачу генерації усіх перестановок з n елементів. Розглянемо для порівняння відомий алгоритм генерації всіх перестановок на основі лексикографічного порядку. Як відомо, відношення лексикографічного порядку $<_L$ на множині рядків визначається наступним чином:

$$s = (s_1, s_2, \dots, s_m) <_L p = (p_1, p_2, \dots, p_m) \text{ якщо } \exists i: s_i < p_i, s_k = p_k \ \forall 1 \le k < i \ .$$

Можна показати, що кількість арифметичних операцій такого алгоритму не менша ніж $n!\,(4\ln 2(e-1) + \ln 2)$.

Розглянемо тепер наш метод перманентної декомпозиції [18]. Маємо формування n матриць, аналог множення-запис у список відповідного елемента, аналог додавання-розбиття біжучої послідовності на дві частини, розгалуження. Арифметичні операції можуть бути відсутні. Але тоді матимемо копіювання елементів у списки. Отже,

$$Q(n) = 2n - 1 + \mathrm{n}\mathrm{Q}(\mathrm{n}-1) = 2\mathrm{n} - 1 + \mathrm{n}\big(2(\mathrm{n}-1) - 1 + (\mathrm{n}-1)\mathrm{Q}(\mathrm{n}-2)\big)$$
$$= 2\mathrm{n} - 1 + 2\mathrm{n}(\mathrm{n}-1) - \mathrm{n} + \mathrm{n}(\mathrm{n}-1)\mathrm{Q}(\mathrm{n}-2) =$$
$$= 2\mathrm{n} - 1 + 2\mathrm{n}(\mathrm{n}-1) - \mathrm{n} + \mathrm{n}(\mathrm{n}-1)\big(2(\mathrm{n}-2) - 1 + (n-2)Q(n-3)\big) =$$

$$=$$

$$2\mathrm{n}-1+2\mathrm{n}(\mathrm{n}-1)-\mathrm{n}+2\mathrm{n}(\mathrm{n}-1)(n-2)-n(n-1)+\mathrm{n}(\mathrm{n}-1)(n-2)Q(n-3)=2\mathrm{n}-1+2\mathrm{n}(\mathrm{n}-1)-\mathrm{n}+2\mathrm{n}(\mathrm{n}-1)(n-2)-n(n-1)+\mathrm{n}(\mathrm{n}-1)(n-2)\big(2(n-3)-1+(n-3)Q(n-4)\big)=2\mathrm{n}-1+2\mathrm{n}(\mathrm{n}-1)-\mathrm{n}+2\mathrm{n}(\mathrm{n}-1)(n-2)-n(n-1)+2\mathrm{n}(\mathrm{n}-1)(n-2)(n-3)-\mathrm{n}(\mathrm{n}-1)(n-2)+\mathrm{n}(\mathrm{n}-1)(n-2)(n-3)Q(n-4)=\cdots=$$

$$=2\mathrm{n}+2\mathrm{n}(\mathrm{n}-1)+2\mathrm{n}(\mathrm{n}-1)(n-2)+\cdots+2\mathrm{n}(\mathrm{n}-1)(n-2)*..*2-(1+n+n(n-1)+n(n-1)(n-2)+\cdots+n(n-1)(n-2)*..*3)+n!=(n+n(n-1)+n(n-1)(n-2)+\cdots+n(n-1)(n-2)*..*2)+n!+n!-1=n!\left(\frac{1}{(n-1)!}+\frac{1}{(n-2)!}+\cdots+\frac{1}{2!}+1\right)+2n!-1\approx n!(e+1)-1$$

Для порівняння складності маємо:

$$\frac{(4\ln 2(e-1)+\ln 2)}{e+1}=\frac{5.457}{3.718}=1.46$$

Аналізуючи оцінки складності відповідних алгоритмів бачимо, що метод перманентної декомпозиції належить до того самого класу складності–O(n!), що і метод, який базується на відношенні порядку. Однак, незважаючи на однаковий порядок складності, маємо виграш в складності перманентного алгоритму (за рахунок мультиплікативної константи), особливо зважаючи на той факт, що при оцінці складності класичного підходу до генерації перестановок була використана оцінка знизу. Складність методу перманентної декомпозиції менша від складності порядкового методу принаймні на 46%.

Проведемо тепер обчислювальний експеримент. Будемо розглядати метод перманентної декомпозиції, який записує перестановки у відповідні поля однозв'язного списку та метод формування перестановок у вигляді лексикографічного порядку. Для методу перманентної декомпозиції всі перестановки будуть записуватись в пам'ять, оскільки результат генерації перестановок нам буде потрібен для задачі складання розкладу. В даному випадку генерація перестановок еквівалентна формуванню СРПС. В якості альтернативи використаємо Heap-алгоритм. Очевидно, виберемо однакові за потужністю множини елементів. Результати тестування наведено на Рисунку 3.

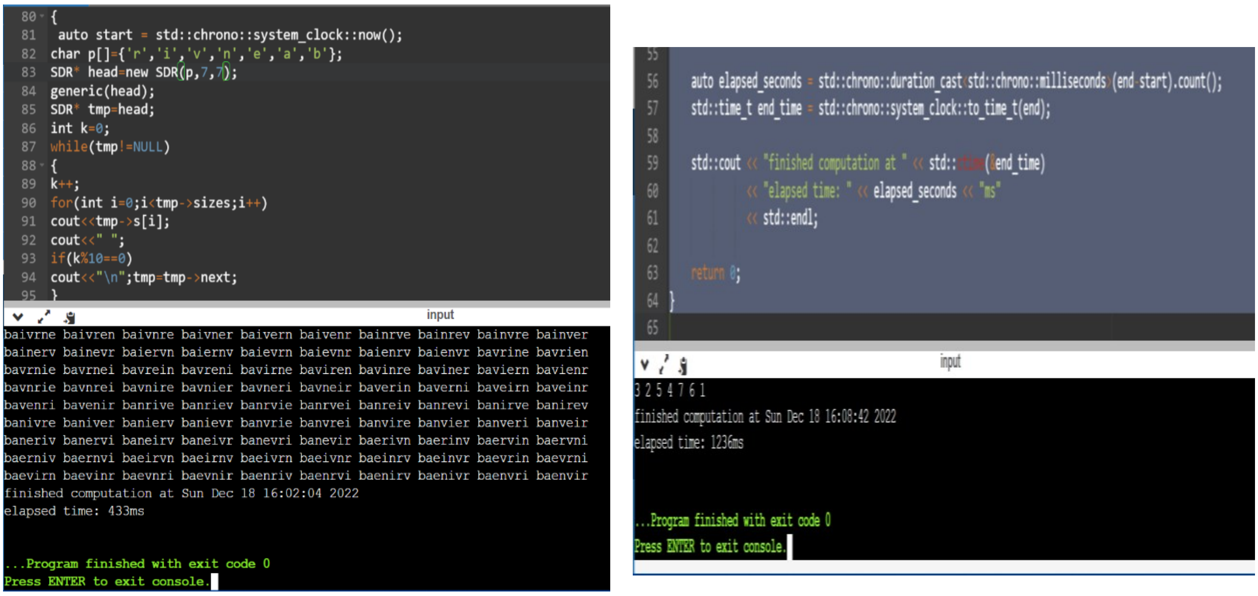

Рис. 3 Результати тестування швидкодії методу перманентної декомпозиції генерації перестановок

Бачимо, що метод перманентної декомпозиції працював 433 мс, в той час Heap-алгоритм-1236 мс. Звичайно, що це лише один приклад, який показує переваги методу перманентної декомпозиції, однак з врахуванням теоретичних оцінок складності можемо говорити про переваги методу, який пропонується.

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

Таким чином, в роботі запропоновано інформаційну технологію формування матриць розкладів, що грунтується на основі методу перманентної декомпозиції. Відповідна технологія включає в себе низку нових понять та алгоритмів, зокрема, поняття модифікованої матриці інцидентності , що дозволяє коректно врахувати інформацію про потоки, нове визначення модифікованого перманента для матриць інцидентності, яке відрізняється тим, що в сумі всіх можливих добутків елементів матриці, кожен з яких містить по одному елементу з кожного рядка та з різних стовпчиків, елемент потокового стовпчика не може бути в добутку разом з елементами інших рядків, що відповідають цьому ж потоку. Суть модифікації матриці інцидентності полягає у введенні додаткових стовпчиків для кожного викладача, що має поточні пари з коректним відображенням інформації про структуру потоку.

Побудовано алгоритм декомпозиції модифікованого перманента, що включає у себе конструктивну процедуру формування систем різних представників стовпчиків матриці розкладів та запису індексів стовпчиків в комірки пам'яті та який виключає, у підсумку, можливість одночасного проведення викладачем кількох пар, і в той же час, допускає одночасну присутність викладача у кількох групах, якщо допускаються потоки. Відповідний алгоритм декомпозиції із "запам'ятовуванням" може бути використаний для розв'язання широкого класу задач генерації комбінаторних об'єктів, зокрема, перестановок (що реалізовано в даному розділі як приклад), комбінацій, підмножин різних елементів деякої системи множин.

Показано, що клас складності перманентного алгоритму генерації комбінаторних об'єктів щонайменше аналогічний до класу складності найшвидших відомих алгоритмів. Наприклад, перманентний алгоритм генерації перестановок має клас складності, аналогічний класу складності алгоритму, що базується на відношенні порядку, складність його менша принаймні на 46%. Однак, на відміну від класичних алгоритмів, перманентний алгоритм допускає узагальнення для вирішення значно складніших задач теорії розкладів, а класичні алгоритми генерації перестановок є вузькоспеціалізованими для вирішення конкретних задач.

Проведені експерименти підтверджують переваги методу перманентної декомпозиції у порівнянні з відомими підходами.

References

1. Szwarc W., Grosso A., Della Croce F. Algorithmic paradoxes of the single machine total tardiness problem. Journal of Scheduling. 2001. V. 4. P. 93–104.
2. Rykov I. Approximate solving of RCPSP. Abstract Guide of OR. 2006, Karlsruhe 6.09-8.09, Germany, P. 226.
3. M. Kumar, L. Kaur and J. Singh, "Dynamic and Static Energy Efficient Scheduling of Task Graphs on Multiprocessors: A Heuristic," in IEEE Access, vol. 8, pp. 176351-176362, 2020, doi: 10.1109/ACCESS.2020.3026839.
4. Kamaljit Kaur, Amit Chhabra and Gurvinder Singh Modified Genetic Algorithm for Task Scheduling in Homogeneous Parallel System Using Heuristics. International Journal of Soft Computing. Volume 5. Issue 2. pp. 42 – 51.
5. S.D. Dao, K. Abhary, R. Marian A bibliometric analysis of Genetic Algorithms throughout the history. Comput. Ind. Eng., 110 (2017), pp. 395-403
6. James C. Chena, Cheng Chun, Wub Chia, Wen Chenc, Kou-Huang Chend Flexible job shop scheduling with parallel machines using Genetic Algorithm and Grouping Genetic Algorithm. Expert Systems with Applications. Volume 39, Issue 11, 1 September 2012, Pages 10016-10021.
7. T.O. Omotehinwa, Temidayo Oluwatosin Examining the developments in scheduling algorithms research: A bibliometric approach. Omotehinwa, Temidayo Oluwatosin. Heliyon, Volume 8, Issue 5, e09510. Open Access. Published: May 21, 2022. doi: 10.1016/j.heliyon.2022.e09510
8. A. Verbeek, K. Debackere, M. Luwel, E. Zimmermann Measuring progress and evolution in science and technology - I: the multiple uses of bibliometric indicators. Int. J. Manag. Rev., 4 (2) (2002), pp. 179-211.
9. S.D. Dao, K. Abhary, R. Marian A bibliometric analysis of Genetic Algorithms throughout the history. Comput. Ind. Eng., 110 (2017), pp. 395-403.
10. I. Rahimi, A.H. Gandomi, K. Deb, F. Chen, M.R. Nikoo Scheduling by NSGA-II: review and bibliometric analysis. Processes, 10 (1) (2022), pp. 1-31.
11. H.Y. Shishido, J.C. Estrella Bibliometric analysis of workflow scheduling in grids and clouds. Proceedings - International Conference of the Chilean Computer Science Society, SCCC, 2017-October (2018), pp. 1-9.
12. J. Yu, Z. Yang, S. Zhu, B. Xu, S. Li, M. Zhang A bibliometric analysis of cloud computing technology research. Proceedings of 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2018 (2018), pp. 2353-2358.
13. A. Khalfay, A. Crispin and K. Crockett, "A review of technician and task scheduling problems, datasets and solution approaches," 2017 Intelligent Systems Conference (IntelliSys), 2017, pp. 288-296, doi: 10.1109/IntelliSys.2017.8324306.
14. Turbal Y. V., Babych S. V. Methods of the schedule matrix forming based on the modified permanent. *Telekomunikacja i Elektronika. Zeszyty Naukowe. Uniwersytet Technologiczno-Przyrodniczy Im. Jana i Jedrzeja Sniadeckich w Bydgoszczy.* 2018. Nr 268, 21. P. 85-92.
15. Turbal Y. V., Babych S. V., Kunanets N. E. Permanent Decomposition Algorithm for the combinatorial object's generation. *Radio Electronics, Computer Science, Control. № 2*. 2022. P. 74-79 (Фахове видання України, Q2, Scopus, WoS).
16. Бабич *С. В., Турбал Ю. В. Алгоритм побудови допустимої матриці розкладів. Вісник Національного університету водного господарства та природокористування*. Серія «Технічні науки». 2014. Випуск 4(68), С. 274-281.
17. Бабич С. В., Турбал Ю.В. Методи формування матриць розкладів на основі модифікованих перманент. Інформаційні системи та мережі. 2017. №872. С.204-209.
18. Бабич С.В., Турбал Ю.В. Застосуваня аддитивно-диз'юнктивних форм в задачах календарного планування. *Матеріали 11-ої Міжнародної науково-практичної конференції «Глушковські читання».* 2022. С.123-126.