УДК 519.854.2

О. Павлов, О. Халус, О. Мельников, В. Дрозд, В. Кобельський, М. Медведєв

СТАТИЧНІ АЛГОРИТМИ УПРАВЛІННЯ ПАРАЛЕЛЬНИМИ АСИНХРОННИМИ ОБЧИСЛЮВАЛЬНИМИ ПРОЦЕСАМИ

Анотація: Розглядається проблема управління паралельними асинхронними обчислювальними процесами. Приведено критичний порівняльний аналіз статичних та динамічних методів управління паралельними асинхронними обчислювальними процесами. Розглядаються дві постановки задачі управління паралельними асинхронних обчислювальних процесів як одноетапних задач теорії розкладів. Пропонується для її розв’язання використовувати алгоритми, зокрема, ПДС-алгоритми, розроблені проф. Павловим О.А. та його учнями, що зазвичай використовуються для реалізації календарного та оперативного планування дискретними виробничими системами. Пропонуються модифікації ПДС-алгоритмів відповідно до сформульованих задач комбінаторної оптимізації. Наведено приклади конкретних наукових та практичних проблем, для яких ефективність програмного забезпечення їх розв’язання досягається за допомогою розглянутих статичних моделей управління паралельними асинхронними обчислювальними процесами та запропонованими методами їх розв’язання.

Ключові слова: паралельні асинхронні обчислювальні процеси; комбінаторна оптимізація; статичне управління; ПДС-алгоритми; регресійні моделі; прийняття рішень; дискретні виробничі системи.

ВСТУП

Ефективність реалізації алгоритмів розв’язання складних прикладних задач залежить як від ефективності запропонованих алгоритмів, так і архітектури програмного забезпечення, що їх реалізує. Суттєво підвищити ефективність програмного забезпечення можна в першу чергу за рахунок реалізації паралельних обчислень та відповідних алгоритмів їх оптимальної організації. Управління паралельними обчисленнями розглядається в двох можливих аспектах. Перший – управління синхронними паралельними обчислювальними процесами, та другий – управління асинхронними паралельними обчислювальними процесами. В першому випадку на реалізацію паралельних обчислень накладаються динамічні обмеження, а саме, залежність між проміжними станами паралельних обчислень. Це – найбільш складна задача ефективної організації паралельних обчислень. Можна відмітити роботи проф. Стеценко І.В. в області створення Петрі-об’єктних моделей симуляції паралельних обчислень [1, 2]. Останнім часом з’явились роботи, зокрема, [3, 4], в яких на основі результатів Петрі-об’єктного моделювання пропонуються алгоритми оптимізації параметрів паралельних обчислень.

Більш простою є задача оптимального управління асинхронними паралельними обчисленнями. Існує достатньо широка бібліографія [5–12], де наводиться та аналізується ефективність різних статистичних та динамічних методів управління. Статичні методи [5] оптимізують такі критерії, як мінімальний час використання обчислювальних ресурсів (мінімізація сумарного часу виконання мета-задачі), мінімальний час завершення обчислень та інші. Загальновизнані методи розв’язання – це методи теорії розкладів, модифікації генетичного алгоритму, метод імітації відпалу, генетична імітація відпалу, двофазний метод планування [8], в якому кластеризація задач виконується перед процесом планування, що дозволяє знаходити субоптимальну кількість необхідних процесорів. Динамічні методи управління [10] враховують непередбачувані зміни в процесі роботи обчислювальної системи. До них відносять, зокрема, метод Fast Critical Path [11] (один з алгоритмів класу «планування за списком», що реалізує достатньо прості евристики, зокрема, в першу чергу розглядається процесор, що простоює більше часу).

В даній роботі розглядаються методи статичного планування асинхронних обчислювальних процесів для двох запропонованих постановок задач, наводяться конкретні наукові та практичні задачі, методи їх розв’язання, ефективна реалізація яких в програмному забезпеченні потребує використання запропонованих моделей і методів.

Актуальність проведених досліджень випливає з того, що запропоновані моделі планування асинхронних паралельних обчислювальних процесів є задачами комбінаторної оптимізації, зокрема, з класу NP-повних та NP-трудних, і вперше для їх розв’язання використовуються як відомі ПДС-алгоритми (алгоритми, що містять достатні ознаки оптимальності їх розв’язку та поліноміальні підалгоритми побудови розв’язків, що перевіряють достатні ознаки оптимальності), створені проф. Павловим О.А. та його учнями, так і їх модифікації.

1. Моделі та алгоритми статичного управління паралельними асинхронними обчислювальними процесами

1.1. Перша модель

В складі однієї мета-програми є n підпрограм, які можуть починати незалежно виконуватись одночасно чи в довільні фіксовані моменти часу $s_j > 0,\ j = \overline{1,m}$ багатопотоковою паралельною програмою.

Примітка 1. В розглянутих комбінаторних моделях кількість процесорів m ($m < n$), на яких проводяться незалежні паралельні обчислення, є відомою і може за необхідності бути заздалегідь оптимізована з використанням, наприклад, алгоритму кластеризації [8].

В першій моделі комп’ютерні процесори вважаються ідентичними. Відома середньостатистична тривалість виконання l_i i-ї підпрограми на одному процесорі, $i = \overline{1,n}$.

Необхідно побудувати розклад σ виконання n підпрограм на m комп’ютерних процесорах такий, щоб досягався мінімум функціоналу

$$\min_{\sigma} \left\{ F(\sigma) = C_{\max}(\sigma) - C_{\min}(\sigma) \right\}, \quad (1)$$

де $C_{\max}(\sigma)$ – момент завершення виконання всіх підпрограм у розкладі σ, тобто, $C_{\max}(\sigma) = \max_{i=1,n} C_i(\sigma)$, де $C_i(\sigma)$ – момент завершення виконання підпрограм на *i*-му комп’ютерному процесорі; аналогічно, $C_{\min}(\sigma) = \min_{i=1,n} C_i(\sigma)$.

Примітка 2. Обґрунтування функціоналу (1). Здавалось би, що логічно було використати в якості функціонала вираз $\min_{\sigma} \max_{i} C_i(\sigma)$, але, по-перше, зменшення $C_{\max}(\sigma)$ логічно реалізовувати за рахунок збільшення $C_{\min}(\sigma)$, а по-друге, розклад $\sigma_{\text{опт}}$, що мінімізує (1), є максимально можливо вирівняним, що може бути важливим при використанні комп’ютерних процесорів для виконання наступних програм.

В [13, 14] було запропоновано ефективний ПДС-алгоритм для розв’язання цієї задачі, що містить дві достатні ознаки оптимальності довільного розкладу σ та складається з наступних двох етапів: перший – побудові початкового розкладу, що реалізується за 6 кроків:

1. Пронумерувати підпрограми $i = \overline{1, n}$ за незростанням значень l_i.

2. Пронумерувати процесори $j = \overline{1, m}$ за неспаданням значень s_j.

3. Встановити часи звільнення процесорів $C_j = s_j$, $j = \overline{1, m}$.

4. Обрати підпрограму *i* з максимальним l_i з непризначених підпрограм та призначити на процесор *j* з мінімальним часом звільнення C_j.

5. Визначити новий час звільнення процесору *j*: $C_j = C_j + l_j$.

6. Якщо розподілені на виконання всі підпрограми, кінець алгоритму. Інакше перехід на крок 4.

Другий етап полягає в послідовному виконанні перестановок типів А і Б [15, 16], що покращують розклад, мінімізуючи значення функціоналу.

Проведені статистичні дослідження ПДС-алгоритму показали його ефективність, він рекомендується авторами як алгоритм статичного управління за першою моделлю.

1.2. Друга модель

Друга модель формулюється як перша з наступними змінами. Потужність паралельних комп’ютерних процесорів різна, тому час виконання підпрограм задається як

l_i^j – середній час виконання *i*-ї підпрограми ($i = \overline{1, n}$) *j*-м процесором ($j = \overline{1, m}$). Треба знайти розклад σ виконання підпрограм на незалежних паралельних процесорах такий, у якого досягається мінімум функціоналу

$$\min_{\sigma} \left\{ \max_{j=\overline{1,m}} C_j(\sigma) - \min_{j=\overline{1,m}} s_j \right\}, \tag{2}$$

де s_j – початковий момент запуска *j*-го процесора (для виконання багатопотокової паралельної програми).

Примітка 3. Фактично, критерій (2) не прив'язаний до конкретних значень s_j, $j = \overline{1, m}$. Він гарантує, що отриманому розкладу відповідає мінімальний сумарний час виконання на паралельних процесорах багатопотокової паралельної програми.

Для знаходження оптимального розкладу пропонується використовувати спрощену версію ПДС-алгоритму [15, 17], що був запропонований для розв'язання наступної задачі календарного планування: маємо *m* незалежних пристроїв, які без переривань виконують *n* завдань. $l_i^j \left(i = \overline{1, n},\ j = \overline{1, m}\right)$ – час виконання *i*-го завдання на *j*-му пристрої. Завдання повинні бути виконані до відповідних директивних строків d_i, $i = \overline{1, n}$. Моменти запуску пристроїв s_j, $j = \overline{1, m}$ $(m < n)$, довільні. Треба знайти розклад σ виконання завдань, на якому досягається

$$\max_{\sigma} \min_{j=\overline{1,m}} s_j. \tag{3}$$

ПДС-алгоритм розв'язання цієї задачі (алгоритм D [17]) містить дві достатні ознаки оптимальності допустимого розкладу, шість поліноміальних підалгоритмів, що реалізують можливість побудови розкладу, який задовольняє одному з достатніх ознак оптимальності, та шість поліноміальних підалгоритмів для отримання наближеного розкладу задачі. Загальна схема ПДС-алгоритму наведена на рис. 3.4 [15].

Результати п. 3.4 [15] ілюструють ефективність запропонованого ПДС-алгоритму.

Наслідок 1. Алгоритм статичного управління, що реалізує другу комбінаторну модель, є спрощеною версією розглянутого вище ПДС-алгоритму, так як для другої моделі $d_i = d > 0$, $i = \overline{1, n}$.

Наслідок 2. При розв'язанні задачі статичного управління ПДС-алгоритмом довільне значення *d* задається достатньо великим, таким, щоб на отриманому розв'язку виконувалось $\min_{j=\overline{1,m}} s_j > 0$.

Нехай

$$s_{j_1} \le s_{j_2} \le \ldots \le s_{j_m}. \tag{4}$$

Тоді момент початку роботи процесора j_1 може бути довільним r_{j_1}, а моменти запуску процесорів з номерами $j_l,\ l = \overline{2, m}$, повинні задовольняти умовам:

– якщо $r_{j_1} \geq s_{j_1}$, то

$$r_{j_{l-1}} \leq r_{j_l} \leq s_{j_l} + \left| r_{j_1} - s_{j_1} \right|,\ l = \overline{2, m}. \tag{5}$$

– якщо $r_{j_1} \leq s_{j_1}$, то

$$r_{j_{l-1}} \leq r_{j_l} \leq s_{j_l} - \left| r_{j_1} - s_{j_1} \right|,\ l = \overline{2, m}. \tag{6}$$

Виконання умови (5) чи (6) гарантує виконання умови (2).

2. Приклади можливого використання запропонованих алгоритмів статичного управління асинхронними паралельними обчислювальними процесами

2.1. Модифікований метод групового урахування аргументів

Постановка задачі побудови багатовимірної лінійної регресії (БЛР), заданої надлишковим описом, полягає в наступному. Задано

$$Y(x) = \sum_{j=0}^{m} b_j x^j + E,$$

де x – скалярна вхідна змінна, E – випадкова величина, $ME = 0$, $DE = \sigma^2 < \infty$. За результатами експериментів $\left(x_i \to y_i,\ i = \overline{1, n}\right)$ треба оцінити невідомі значення коефіцієнтів $b_j,\ j = \overline{0, m}$, враховуючи, що деякі з них можуть тотожно дорівнювати нулю. Для розв'язання цієї задачі в [18] було запропоновано модифікований метод групового урахування аргументів (ММГУА), методологія реалізації якого є наступною.

1. Дані експерименту $J = \left(x_i \to y_i,\ i = \overline{1, n}\right)$ розбиваються на дві множини $J_1 = \left\{x_{il} \to y_{il},\ l = \overline{1, n_1}\right\}$, $J_2 = \left\{x_{jl} \to y_{jl},\ l = \overline{1, n_2}\right\}$, $J_2 \cup J_2 = J$, $J_2 \cap J_2 = \varnothing$.

2. Методом найменших квадратів (МНК) за множиною J_1 знаходяться оцінки $\hat{b}_j,\ j = \overline{0, m}$.

3. Оригінальним алгоритмом кластерного аналізу [18] по оцінкам $\hat{b}_j,\ j = \overline{0, m}$, множина коефіцієнтів $\left\{b_j,\ j = \overline{0, m}\right\}$, розбивається на два класи M_1 та M_2, що не перетинаються.

4. МНК, з використанням даних експерименту J_1, будується множина часткових описів БЛР. В кожний частковий опис входять члени, що відповідають коефіцієнтам з

множини M_1 та члени, що відповідають всім можливим різним підмножинам з множини M_2.

5. За множиною J_2 по вибраному регулярному критерію знаходиться частковий опис, структура якого вважається такою, що відповідає шуканій БЛР.

6. За множиною J МНК знаходяться оцінки коефіцієнтів вибраного часткового опису.

Ефективність програмної реалізації запропонованого в [18] ММГУА для побудови багатовимірних регресій по обмеженому об’єму даних може бути суттєво покращена, якщо для побудови множини часткових описів багатовимірної регресії (див. п. 4 методології) використовувати асинхронні паралельні обчислення, статичне управління якими в залежності від використовуваних обчислювальних засобів, реалізується відповідно до першої чи другої комбінаторної моделі. Знаходження параметрів управління – час побудови часткового опису багатовимірної регресії на конкретному обчислювальному процесорі залежить від розмірності часткового опису та способу реалізації загальної формули методу найменших квадратів при повторних активних експериментах. Результати відповідних теоретичних та експериментальних досліджень наведені в [18, 19].

2.2. Побудова експертних ваг за емпіричною матрицею парних порівнянь з використанням моделей лінійного програмування

В [20] наведені теоретичні основи побудови експертних ваг, що кількісно оцінюють якість альтернатив, з яких треба обрати найкращу, за емпіричною матрицею парних порівнянь з використанням низки моделей лінійного програмування. Методологія знаходження експертних ваг ω_j, $j = \overline{1, n}$, наступна:

1. За емпіричною матрицею парних порівнянь $\Gamma = \left(\gamma_{ij}\right)_1^n$ (деякі коефіцієнти матриці Γ можуть бути відсутні) знаходиться множина $\left|\Gamma\right|$, що складається з усіх пар $\left(i, j\right)$, яким відповідають $\gamma_{ij} \geq 1, \forall i \neq j$.

2. Розв’язується низка задач лінійного програмування (ЛП) виду

$$\min_{\forall y_{ij}} \sum_{(ij)\in\left|\Gamma\right|} r_{ij} y_{ij}$$

$$-y_{ij} \leq \omega_i - \gamma_{ij}\omega_j \leq y_{ij}, \forall (ij) \in \left|\Gamma\right|, y_{ij} \geq 0, \sum_{i=1}^{n} \omega_i = 1 .$$

Змінними задачі ЛП є $\forall y_{ij}$, ω_i. Коефіцієнти r_{ij} в кожній задачі ЛП приймають одне із запропонованих в [20] множини значень, наприклад:

$$r_{ij} = \frac{1}{\gamma_{ij} - 1}, \ \gamma_{ij} > 1, \ r_{ij} = 1, \ \gamma_{ij} = 1,$$

$$r_{ij} = \frac{1}{\sqrt{\gamma^{*}_{ij} - 1}}, \ \gamma_{ij} > 1, \ r_{ij} = 1, \ \gamma_{ij} = 1,$$

$$r_{ij} = \frac{1}{\sqrt[3]{(\gamma^{*}_{ij} - 1)^{2}}}, \ \gamma_{ij} > 1, \ r_{ij} = 1, \ \gamma_{ij} = 1.$$

В результаті отримуємо множину $\{\overline{\omega}^{l}\}, \ l = \overline{1, P}$, векторів оцінок емпіричних ваг, де P – кількість задач ЛП.

3. За критерієм

$$\min_{l = \overline{1, P}} \sum_{(ij) \in |\Gamma|} \frac{1}{\gamma_{ij}} \left| \gamma_{ij} - \frac{\omega^{l}_{i}}{\omega^{l}_{j}} \right|$$

знаходимо шуканий набір ваг $\hat{\omega}^{*}$.

4. Випадковим чином синтезуються M штучних матриць парних порівнянь (M – задане число), кожний елемент якої γ^{l}_{ij} (якщо $\gamma_{ij} \in \Gamma$ існує), $l = \overline{1, M}$, $\forall i \neq j$, знаходиться випадковим чином, при обов’язковому виконанні умови

$$\left| \gamma^{l}_{ij} - \frac{\omega^{*}_{i}}{\omega^{*}_{j}} \right| = \pm \left| \gamma_{ij} - \frac{\omega^{*}_{i}}{\omega^{*}_{j}} \right|.$$

Для кожної такої матриці знаходимо вектор $\hat{\omega}^{l}, \ l = \overline{1, M}$, оцінок емпіричних коефіцієнтів з використанням задачі ЛП, розв’язком якої був вектор ω^{*}. Відповідно [20], вектор ω^{*} є обґрунтованим рішенням, якщо $d\left(\hat{\omega}^{l} - \omega^{*}\right) \in [0, 0.02], \ l = \overline{1, M}$, де

$$d\left(\hat{\omega}^{l} - \omega^{*}\right) = \left\| \frac{\omega^{l}}{\left\|\omega^{l}\right\|} - \frac{\omega^{*}}{\left\|\omega^{*}\right\|} \right\|,$$

$\|\cdot\|$ – евклідова норма.

Ефективність програмної реалізації запропонованого в [20] методу можна суттєво підвищити, якщо при реалізації пп. 2 та 4 наведеної вище методології використовувати асинхронні паралельні обчислювальні процеси, що статично управляються на основі першої чи другої комбінаторної моделі. Параметри управління часу розв’язання конкретної задачі лінійного програмування симплекс-методом статистично ефективно задається кількістю кроків симплекс-методу ($3m$, де m – кількість лінійних нерівностей) та часом виконання одного кроку, що залежить від кількості змінних задачі лінійного програмування.

 ISSN 1560-8956

2.3. Точний метод для важкорозв’язуваної одноетапної задачі комбінаторної оптимізації «Побудова допустимого розкладу, що мінімізує сумарне зважене випередження виконання робіт відносно їх директивних строків на одному пристрої, що працює без переривань»

В [21] була сформульована наступна важкорозв’язувана одноетапна задача комбінаторної оптимізації. На одному пристрої виконуються n робіт. Для кожної роботи j задані d_j – директивний строк, l_j – час виконання на пристрої, $j = \overline{1,n}$. Треба побудувати допустимий (при якому жодний директивний строк не порушується) розклад σ, для якого виконувалось би: при умові, що значення моменту запуску пристрою є максимально великим ($s_{\max}$), досягається

$$\min_{\sigma} \sum_{j=1}^{n} \omega_j \left(d_j - C_j(\sigma) \right), \tag{7}$$

де $C_j(\sigma)$ – момент завершення виконання j-ї роботи, $\omega_j > 0$ – експертний ваговий коефіцієнт, $j = \overline{1,n}$.

Як показано в [21], задача декомпозується на дві. Спочатку знаходиться $s_{\max}$. Ефективний алгоритм його знаходження наведений в [22]. Потім при знайденому значенні $s_{\max}$ знаходиться допустимий розклад, який задовольняє (7). Для знаходження цього розкладу в [21] запропонований модифікований метод «гілок і меж». Модифікація полягає в тому, що будується «ліс», що складається з «дерев», коренем кожного з яких є робота, що в допустимому розкладі може бути останньою. Приведено два теоретично обґрунтовані правила відсікання гілок, що є наслідком властивостей сформульованої задачі. Перше випливає з того, що для двох робіт i_1, i_2, що стоять поруч в оптимальному розкладі:

а) якщо $l_{i_2} < l_{i_1}$, $\omega_{i_2} > \omega_{i_1}$, то робота i_1 повинна стояти перед роботою i_2;

б) якщо $l_{i_2} < l_{i_1}$, $\omega_{i_2} < \omega_{i_1}$, то порядок виконання робіт i_1 та i_2 в оптимальному розкладі залежить від знаку нерівності: $\omega_{i_1} \cdot l_{i_2} \gtrless \omega_{i_2} \cdot l_{i_1}$.

Друге правило відсікання ґрунтується на необхідній і достатній умові допустимого розкладу: якщо розклад, в якому роботи, починаючи з першої позиції, розташовані в порядку за неспаданням величин їхніх директивних строків, є недопустимим, то допустимого розкладу не існує.

Ефективність програмної реалізації методу може бути суттєво підвищена, якщо при побудові «лісу» кожне «дерево» будується незалежно від інших на окремому обчислювальному процесорі. Для використання першої чи другої комбінаторної моделі статичного управління асинхронними паралельними обчислювальними процесами треба заздалегідь на основі обчислювальних експериментів вивести статистично достовірну емпіричну залежність кількості операцій побудови одного «дерева» як функції кількості робіт, тобто розмірності, сформульованої в п. 2.3. одноетапної задачі комбінаторної оптимізації.

ВИСНОВКИ

1. На основі критичного наукового аналізу методів статичного управління асинхронними паралельними обчислювальними процесами сформульовано дві важкорозв’язувані комбінаторні моделі статичного управління.

2. Для їх розв’язання запропоновано використовувати ПДС-алгоритми та їх модифікації, створені проф. Павловим О.А. та його учнями для ефективної реалізації багаторівневої моделі календарного та оперативного управління дискретними виробничими системами з мережевим представленням технологічних процесів.

3. Розглянуто теоретичні та практичні аспекти використання запропонованих моделей комбінаторної оптимізації статичного управління асинхронними паралельними обчислювальними процесами при програмній реалізації оригінальних методів розв’язання трьох прикладних задач.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Стеценко И.В.* Формальное описание систем средствами Петри-объектных моделей. *Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка*. 2011. Вип. 53. С. 74–81. URL: https://ela.kpi.ua/handle/123456789/3813 (дата звернення: 20.01.2024).

2. *Стеценко И.В.* Теоретические основы Петри-объектного моделирования систем. *Математичні машини і системи*. 2011. Том 4. С. 136–148. URL: https://dspace.nbuv.gov.ua/handle/123456789/83636 (дата звернення: 20.01.2024).

3. *Stetsenko I.V., Pavlov A.A., Dyfuchyna O.* Parallel algorithm development and testing using Petri-object simulation. *International Journal of Parallel, Emergent and Distributed Systems*. 2021. Vol. 36. Iss. 6. P. 549-564. DOI: 10.1080/17445760.2021.1955113.

4. *Дифучина О.* Метод оптимізації параметрів паралельних обчислень. *Технічні науки та технології*. 2023. Том 3. № 33. С. 130–140. URL: http://tst.stu.cn.ua/article/view/291211 (дата звернення: 20.01.2024).

5. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems / T.D. Braun et al. // *Journal of Parallel and Distributed Computing*. 2021. Vol. 61. Iss. 6. P. 810–837. DOI: 10.1006/jpdc.2000.1714.

6. *El-Rewini H., Lewis T., Ali H.* Task Scheduling in Parallel and Distributed Systems. 2010. Prentice Hall, 290 p.

7. Dynamic grid scheduling with job migration and rescheduling in the GridLab resource management system / K. Kurowski et al. // *Scientific Programming*. 2004. Vol. 12. Iss. 4. P. 263–273. DOI: 10.1155/2004/892169.

8. *Liou J., Palis M.A.* A comparison of general approaches to multiprocessor scheduling. Proceedings of the 11th International Parallel Processing Symposium (IPPS '97). *IEEE Computer Society*. 1997. P. 152–156. DOI: 10.1109/ipps.1997.580873.

9. *Минайленко Р.М.* Аналіз алгоритмів планування ресурсами в розподіленому середовищі. *National Interagency Scientific and Technical Collection of Works*. 2020. Col. 50. P. 229–235. DOI: 10.32515/2414-3820.2020.50.229-235.

10. Topics in Parallel and Distributed Computing: Enhancing the Undergraduate Curriculum: Performance, Concurrency, and Programming on Modern Platforms. S.K. Prasad et al. Springer International Publishing. 2018. 337 p. ISBN: 978-3-319-93108-1.

11. *Radulescu A., Gemund A.J.C.* On the complexity of list scheduling algorithms for distributed memory systems. Supercomputing (SC’99): Proceedings of 13th International Conference. *IEEE Computer Society*. 1999. P. 68–75. DOI: 10.1145/305138.305162.

12. *Yang T., Gerasoulis A.* DSC: Scheduling parallel tasks on an unbounded number of processors. *IEEE Transactions on Parallel and Distributed Systems*. 1994. Vol. 5. Iss. 9. P. 951–967. DOI: 10.1109/71.308533.

13. ПДС-алгоритми для двоетапної задачі календарного планування в детермінованій постановці та в умовах невизначеності / О.А. Павлов та ін. // *Адаптивні системи автоматичного управління*. 2023. Том 1. № 42. С.184–196. DOI: 10.20535/1560-8956.42.2023.279170.

14. Highly efficient scheduling algorithms for identical parallel machines with sufficient conditions for optimality of the solutions obtained / S.F. Telenyk et al. // *Bulletin of the Polish Academy of Sciences: Technical Sciences*. 2024. Vol. 72. Iss. 1. P. e148939. DOI: 10.24425/bpasts.2024.148939.

15. *Згуровский М.З., Павлов А.А. Труднорешаемые задачи комбинаторной оптимизации в планировании и принятии решений*. Наукова думка, Київ. 2016. 716 с. ISBN: 978-966-00-1543-2

16. *Павлов О.А., Жданова О.Г., Сперкач М.О.* Задача составления допустимого расписания с максимально поздним моментом запуска выполнения идентичными параллельными приборами работ с общим директивным сроком. *Вісник НТУУ "КПІ". Серія "Інформатика, управління та обчислювальна техніка".* 2014. Вип. 61. С. 93–102. URL: https://ela.kpi.ua/handle/123456789/16721 (дата звернення: 20.01.2024).

17. *Zgurovsky M.Z., Pavlov A.A.* Optimal scheduling for vector or scalar criterion on parallel machines with arbitrary due dates of tasks. In: Combinatorial Optimization Problems in Planning and Decision Making: Theory and Applications. 1st ed. *Studies in Systems, Decision and Control*. 2019. Vol. 173. P. 39–105. Springer, Cham. DOI: 10.1007/978-3-319-98977-8_3.

18. *Pavlov A.A., Holovchenko M.N.* Modified method of constructing a multivariate linear regression given by a redundant description. *Bulletin of National Technical University*

“KhPI”. Series: System analysis, control and information technologies. 2022. Vol. 2. Iss. 8. P. 3–8. DOI: 10.20998/2079-0023.2022.02.01.

19. Дослідження ефективності методу побудови багатовимірної лінійної регресії, заданої надлишковим описом / О.А. Павлов та ін. // Матеріали тез доповідей конференції “Інженерія програмного забезпечення і передові інформаційні технології” (SoftTech-2022 Осінь). 2023. С. 10–13. URL: https://drive.google.com/file/d/1CP9EaBTT_rJAXsINbanSVGnP2jkg9FJ0/view (дата звернення: 29.01.2024).

20. *Zgurovsky M.Z., Pavlov A.A.* The four-level model of planning and decision making. In: Combinatorial Optimization Problems in Planning and Decision Making: Theory and Applications. 1st ed. *Studies in Systems, Decision and Control*. Vol. 173. P. 265–290. – Springer, Cham, 2019. DOI: 10.1007/978-3-319-98977-8_8.

21. *Павлов О.А., Халус О.А., Медведєв М.Є.* Мінімізація сумарного зваженого моменту випередження виконання завдань на одному приладі. *Фізико-математичне моделювання та інформаційні технології*. 2023. Вип. 37. С. 57–61. DOI: 10.15407/fmmit2023.37.057.

22. *Згуровский М.З., Павлов А.А., Халус Е.А.* Задача построения допустимого расписания с максимально поздним моментом запуска и минимальным суммарным опережением. *Системні дослідження та інформаційні технології*. 2015. № 2. С. 7–15. URL: http://journal.iasa.kpi.ua/article/view/51990 (дата звернення: 26.01.2024).