

ISSN 2710-3056

# Grail of Science

Periodical scientific journal

No 32

October  
2023

## The issue of journal contains

Proceedings of the VI Correspondence  
International Scientific and Practical Conference

### **GLOBALIZATION OF SCIENTIFIC KNOWLEDGE: INTERNATIONAL COOPERATION AND INTEGRATION OF SCIENCES**

held on October 13<sup>th</sup>, 2023 by

NGO European Scientific Platform (Vinnytsia, Ukraine)  
LLC International Centre Corporative Management (Vienna, Austria)

 **OU CI**  
Open Ukrainian Citation Index



Euro Science Certificate № 22479 dated 10.09.2023  
UKRISTEI (Ukraine) Certificate № 300 dated 16.06.2023

INDEX  COPERNICUS  
INTERNATIONAL

ТЕХНОЛОГІЧНІ ОСНОВИ АВТОМАТИЗОВАНОГО КЕРУВАННЯ  
ВЛАСТИВОСТЯМИ ЯДЕРНОГО ПАЛИВА  
Науково-дослідна група:  
Панін В.О., Зубак В.В., Белоусов В.С., Кіпріанов І.А. .... 158

### **СЕКЦІЯ XIII. ЕНЕРГЕТИКА ТА ЕНЕРГЕТИЧНЕ МАШИНОБУДУВАННЯ**

#### **СТАТТІ**

APERÇU DES MODÈLES DE DISTRIBUTION DE FLUX D'ÉNERGIE DANS LES  
RÉSEAUX ÉLECTRIQUES  
Okhrimenko V., Gnatovski M. .... 166

### **СЕКЦІЯ XIV. СИСТЕМНИЙ АНАЛІЗ, МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ**

#### **ТЕЗИ ДОПОВІДЕЙ**

МЕТОД ПАРАМЕТРИЗАЦІЇ ДЕТЕРМІНОВАНИХ КОГНІТИВНИХ МОДЕЛЕЙ  
Адаменко А.А. .... 170

### **СЕКЦІЯ XV. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ**

#### **СТАТТІ**

ПОРІВНЯЛЬНИЙ АНАЛІЗ КЛАСИФІКАТОРІВ НА БАЗІ НЕЙРОННИХ  
МЕРЕЖ У ЗАДАЧАХ РОЗПІЗНАВАННЯ ОБРАЗІВ  
Науково-дослідна група:  
Давидов В.О., Курінько Д.Д., Перфільєв Д.С., Філіппов Є.Г. .... 173

ПРОБЛЕМИ РОЗВИТКУ ПАРАЛЕЛЬНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ  
ТА ЇХ МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РІШЕННЯ ЗАДАЧ  
ДИСКРЕТНОЇ ОПТИМІЗАЦІЇ  
Науково-дослідна група:  
Коломійцев О.В., Рибальченко А.О., Голубничий Д.Ю., Третяк В.Ф.,  
Воронін В.В., Комаров В.О., Пустоваров В.В., Філіппенков О.В.,  
Любченко О.В., Рудаков І.С. .... 182

РІШЕННЯ ЗАВДАННЯ РОЗПІЗНАННЯ ЗОБРАЖЕНЬ НА ПРИКЛАДІ  
АЛГОРИТМУ ЕЛМАНА  
Кесов Д.О., Шпінарьова І.М., Рудніченко М.Д., Шibaєва Н.О. .... 198


DOI 10.36074/grail-of-science.13.10.2023.032

## ПРОБЛЕМИ РОЗВИТКУ ПАРАЛЕЛЬНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ ТА ЇХ МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ РІШЕННЯ ЗАДАЧ ДИСКРЕТНОЇ ОПТИМІЗАЦІЇ

НАУКОВО-ДОСЛІДНА ГРУПА:

Коломійцев Олексій Володимирович 

Заслужений винахідник України, доктор технічних наук, професор, професор кафедри комп'ютерної інженерії та програмування  
*Національний технічний університет "Харківський політехнічний інститут", Україна*

Рибальченко Аліна Олександрівна 

аспірантка кафедри комп'ютерної інженерії та програмування  
*Національний технічний університет "Харківський політехнічний інститут", Україна*

Голубничий Дмитро Юрійович 

кандидат технічних наук, доцент, доцент кафедри інформаційних систем  
*Харківський національний економічний університет імені Семена Кузнеця, Україна*

Третяк Вячеслав Федорович 

кандидат технічних наук, доцент, старший науковий співробітник, науковий співробітник наукового центру Повітряних Сил  
*Харківський національний університет Повітряних Сил імені Івана Кожедуба, Україна*

Воронін Віктор Валерійович 

кандидат технічних наук, доцент, старший науковий співробітник наукового центру Повітряних Сил  
*Харківський національний університет Повітряних Сил імені Івана Кожедуба, Україна*

Комаров Володимир Олександрович 

Заслужений винахідник України, кандидат технічних наук, провідний науковий співробітник  
*Військовий інститут телекомунікацій і інформатизації імені Героїв Крут, Україна*

**Пустоваров Володимир Володимирович** 

кандидат технічних наук, начальник науково-дослідного відділу  
*Державний науково-дослідний інститут випробувань і  
сертифікації озброєння та військової техніки, Україна*

**Філіппенков Олексій Володимирович** 

кандидат технічних наук, начальник науково-дослідного  
відділу наукового центру Повітряних Сил  
*Харківський національний університет Повітряних Сил  
імені Івана Кожедуба, Україна*

**Любченко Олексій Вікторович** 

аспірант кафедри комп'ютерної інженерії та програмування  
*Національний технічний університет "Харківський політехнічний  
університет", Україна*

**Рудаков Ігор Сергійович** 

аспірант кафедри комп'ютерної інженерії та програмування  
*Національний технічний університет "Харківський політехнічний  
університет", Україна*

**Анотація.** *Висвітлено проблеми, які пов'язані із розвитком паралельних обчислювальних систем та їх математичного забезпечення для рішення задач дискретної оптимізації. Проведено дослідження ключових аспектів таких, як швидкість обчислень, масштабованість та ефективність паралельних систем у вирішенні складних задач оптимізації. Також, досліджено математичне забезпечення, яке використовується для створення алгоритмів оптимізації та показано важливість вибору правильних методів і підходів у даному підході. Отримані результати досліджень допоможуть вдосконалити розробку та використання паралельних обчислювальних систем для рішення складних завдань щодо оптимізації у різних галузях науки.*

**Ключові слова:** *паралельна обчислювальна структура, дискретна оптимізація, математичне забезпечення, програма, ранговий підхід.*

Розвиток паралельних обчислювальних структур та їх математичного забезпечення є актуальною науковою проблемою в сучасному обчислювальному середовищі [1-5]. До основних проблем, пов'язаних з даними аспектами можливо віднести наступні:

**Паралелізація програм:** розробка програм, які можуть ефективно використовувати паралельні структури такі, як багатоядерні процесори або обчислювальні кластери. Розробка програм, які можуть розділити завдання на незалежні потоки обчислень та ефективно координувати їх виконання, що є складною задачею.

**Масштабованість:** здатність масштабувати паралельні обчислення при збільшенні кількості доступних обчислювальних ресурсів. Це означає, що системи повинні бути здатні працювати як на невеликих кластерах, так і на суперкомп'ютерах, а також збільшувати продуктивність відповідно до росту

ресурсів.

*Взаємодія між задачами:* за умови, коли багато завдань виконуються паралельно, важливо керувати їх взаємодією та конфліктами. Це може включати в себе управління доступом до спільних ресурсів, уникнення гонок за даними та ефективне планування задач.

*Відмовостійкість:* збільшення кількості обчислювальних елементів збільшує ймовірність відмов пристроїв та програмного забезпечення (ПЗ). Тому, важливо розробляти методи та відповідну інфраструктуру для виявлення і відновлення відмов, щоб забезпечити надійність паралельних систем.

*Математичне моделювання:* розробка математичних моделей для аналізу та оптимізації паралельних обчислень є складною задачею. Моделі повинні враховувати різні аспекти такі, як розподіл даних, синхронізація потоків, навантаження на мережу та інші фактори.

*Програмування та інструменти:* інструменти і мови програмування для паралельних обчислень також потребують подальшого розвитку та удосконалення, щоб спростити розробку і налагодження паралельних програм.

Загалом, розвиток паралельних обчислювальних структур та математичного забезпечення вимагає багато зусиль та досліджень, оскільки це важливий аспект для подальшого підвищення продуктивності обчислювальних систем і вирішення складних завдань у багатьох областях, включаючи науку, інженерію та бізнес.

**Використання паралельної обробки** для рішення задач дискретної оптимізації може стикатися з деякими викликами і проблемами. До основних викликів і проблем можливо віднести наступні.

*Залежності між задачами:* у задачах дискретної оптимізації, таких як задачі комбінаторної оптимізації або задачі розкладання, може бути багато обмежень та залежності між різними рішеннями. Це ускладнює паралельну обробку, оскільки деякі задачі можуть бути залежними від результатів інших та не завжди можливо ефективно розділити їх між обчислювальними ресурсами.

*Конфлікти при доступі до ресурсів:* у паралельних системах можуть виникати конфлікти при доступі до спільних ресурсів таких, як спільна пам'ять або об'єкти, які використовуються кількома потоками. Розробка механізмів синхронізації та управління конфліктами може бути набагато складною.

*Вартість комунікації:* у паралельних системах вартість комунікації між обчислювальними одиницями може бути великою. У задачах дискретної оптимізації, де необхідно обмінюватися інформацією між різними рішеннями або підзадачами. Це може призвести до збільшення часу виконання.

*Різноманітність розмірів задач:* розмір задачі може значно варіюватися у залежності від конкретної ситуації. Підбір оптимального розподілу завдань між обчислювальними ресурсами може бути непростою задачею, оскільки деякі задачі можуть бути надто великими для одного обчислювального вузла, а інші – дуже малими.

*Специфічність алгоритмів:* деякі алгоритми оптимізації можуть бути важко адаптувати для паралельного виконання. Наприклад, алгоритми, які використовують ітеративні методи зі змінними, можуть вимагати складних

механізмів управління синхронізацією та взаємодією потоків.

*Складність аналізу та налагодження:* паралельні програми (ПП) можуть бути складними для аналізу та налагодження через можливу непередбачуваність взаємодії між потоками. Виявлення та усунення помилок може бути також важливим завданням.

*Підтримка апаратної архітектури:* використання паралельних обчислювальних ресурсів може вимагати адаптації алгоритмів та ПЗ до конкретної апаратної архітектури. Це може вимагати додаткової роботи та відповідних зусиль.

Незважаючи на перелічені виклики, паралельна обробка може значно покращити швидкість та продуктивність при розв'язанні задач дискретної оптимізації, особливо на великих обчислювальних кластерах або суперкомп'ютерах. Вирішення даних проблем вимагає дбайливого проектування та оптимізації паралельних алгоритмів та програм.

Як показано в [5] алгоритм представлено у вигляді графу. При цьому, алгоритм та його правило, яке його описує, дозволяють визначити наступне:

- множину змінних, у перетворенні яких полягає реалізація алгоритму;
- множину операцій, які виконуються у процесі реалізації алгоритму;
- відповідність, що показує які результати виконання попередніх операцій є аргументами для кожної операції.

Нехай, заданий алгоритм  $A$  (рис. 1). Його можна представити у вигляді підмножини алгоритмів  $\{A_j\}$   $j \in \overline{1, n}$ , яку необхідно виконати у певній послідовності, яка задана графом  $G(V, E)$ . В останньому, кожна вершина відповідає алгоритму  $A_j$ , а кожне ребро між вершинами  $i-j$ , спрямоване від вершини  $i$  до  $j$ , існує тільки у тому випадку, якщо алгоритм  $A_j$  можна виконати тільки після реалізації алгоритму  $A_i$ .

На рисунку 1а наведена структурна схема алгоритму  $A$ , який заданий у вигляді графа  $G(V, E)$ , із якого бачимо, що для його реалізації спочатку потрібно виконати алгоритм  $A_1$ .

Результати роботи приведеного алгоритму є вхідними даними для виконання алгоритмів  $A_2, A_3, A_4$ . Після їхнього виконання реалізується алгоритм  $A_5$ , а алгоритм  $A_6$  виконується після алгоритмів  $A_3, A_4$ , а  $A_7$  – слідом за  $A_5$  і  $A_6$ . Отже, число різних подань графів алгоритму  $A$  визначається множиною способів, які використовуються для рішення задачі.

Граф  $G$  (рис. 1а) розглядаємо як структуру програмного виробу (ПВ), що реалізує алгоритм  $A$ , де  $\{A_j\}$  – є множиною програмних модулів, з яких складається програма реалізації алгоритму  $A$ . За умови, якщо у кожному вершину графа  $G$  помістити процесор, орієнтований на виконання алгоритму  $A_j$ , то можливо одержати обчислювальну структуру, яка дозволяє реалізувати алгоритм  $A$ . Припустимо, що кожний процесор  $P_j$  виконує алгоритм  $A_j$  відразу, як тільки на його вхід надійдуть усі необхідні дані. Позначимо  $t_j$  – час, необхідний на реалізацію кожного з алгоритмів  $A_j$  на відповідному процесорі  $P_j$ . Таким чином, за умови, якщо привласнити вершинам графа  $G(V, E)$  усі ваги  $t_j$ , мінімальний час реалізації алгоритму  $A$  не може перевищити довжину критичного (самого довгого) шляху  $M_{1-7}$  у графі  $G$ .

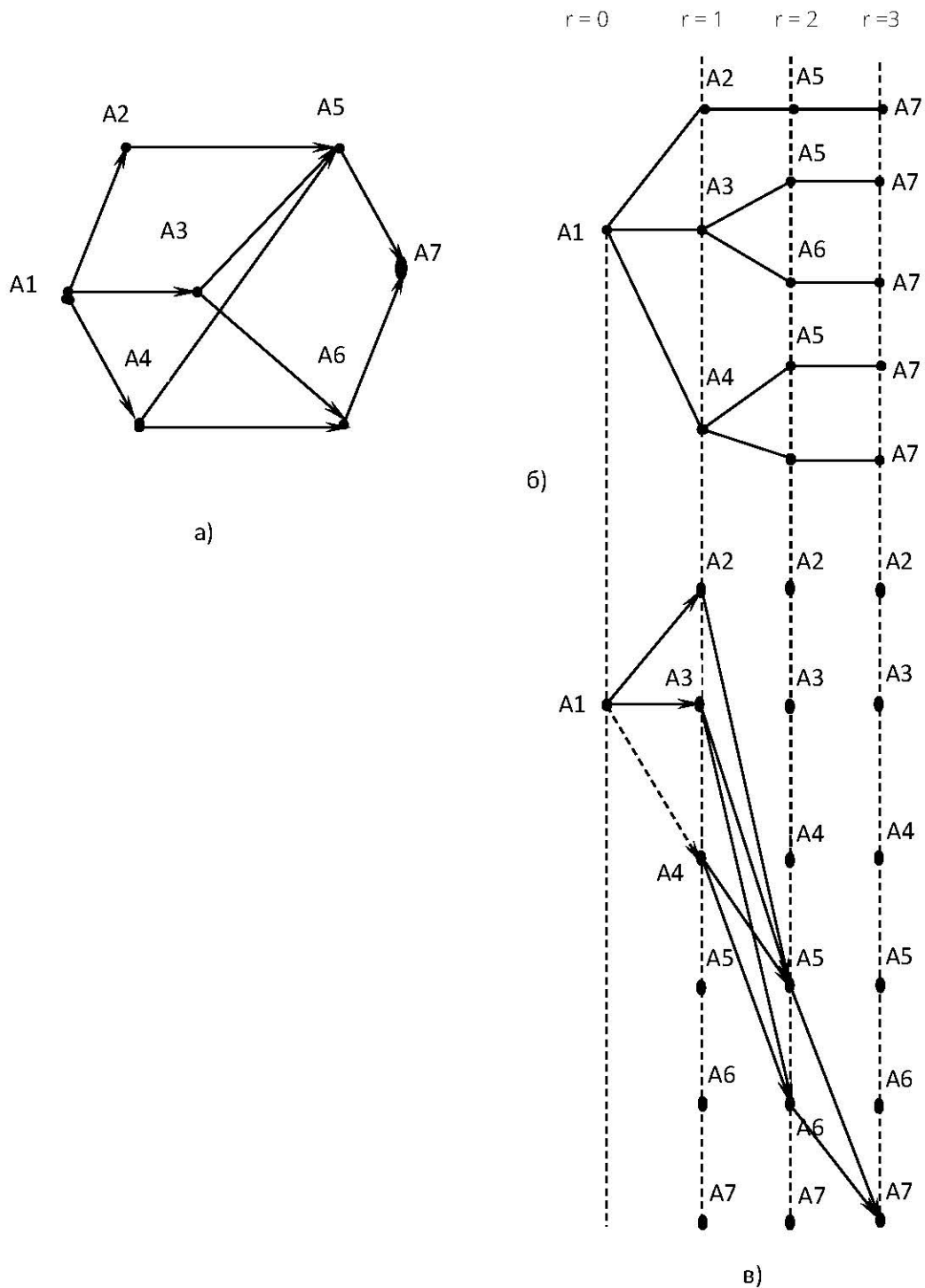


Рис.1. Структура графа G

Таким чином, обрана обчислювальна структура ідеальна для даного подання графа алгоритму A, оскільки вона забезпечує рішення завдання за мінімальний час розпаралелювання процесів обчислень.

Ранг критичного шляху (число ребер у шляху) у графі  $G(V,E)$ , який відповідає деякому алгоритму A, називається глибиною алгоритму A, а максимальне число алгоритмів  $A_j \in A$ , яких можна одночасно виконати на довільному  $g$ -м кроці реалізації алгоритму A – шириною алгоритму. Надалі,

глибина й ширина довільного алгоритму позначається літерами  $a$  й  $h$ . Припустимо, що кожний з алгоритмів  $\{A_j\} \in A$  реалізується на будь-якому процесорі  $P_j$ . Нехай, задана структурна схема алгоритму  $A$  графом  $G(V,E)$ . Виникає завдання визначення оптимальної стратегії розподілу алгоритмів  $\{A_j\}$  за процесорами  $\{P_j\}$ , при якій час на вирішення задачі відрізняється від довжини критичного шляху у графі  $G(V,E)$  на мінімально можливу величину. Обрана стратегія їхнього використання та повинна визначити обчислювальну структуру. За умови, якщо ширина алгоритму дорівнює  $h$ , то необхідно обмежитися  $h$ -процесорами для реалізації алгоритму  $A$ .

Для цього, граф  $G(V,E)$  алгоритму  $A$  можливо представити у вигляді стягнутого дерева усіх шляхів. Дерево усіх шляхів  $D$  графа  $G(V,E)$  (рис. 1а), побудовано від першої вершини наведено на рисунку 1б. Воно будується по матриці суміжності  $B = \parallel b_{ij} \parallel$  від вершини  $S$  наступним чином:

– позначимо вершину  $S$  та берем  $S$ -й рядок матриці  $B$ .

– за даним рядком вибирається номер вершин  $j_1$  для яких  $b_{sj_1} \neq 0$  та отримуємо множину вершин, що мають від вершини  $S$  шлях рангу  $r=1$  (вершини першого ярусу);

– вершини  $r$ -го ярусу зі шляхами рангу  $r$  від вершини  $S$  перебувають почерговим переглядом рядків вершин  $j_{r-1}$ , в  $(r-1)$  ярусі, які не зустрічалися у шляху від  $S$  до  $j_{r-1}$ ;

– побудова дерева триває до одержання шляхів максимального можливого рангу або до тих пір, поки для вершин  $j_{r-1}$  не виявиться, що усі вершини вже увійшли до шляху  $\mu_{sj_{r-1}}$ .

У загальному випадку, кількість розгалужень у такому дереві  $(N-1)$ , тобто для даного утворення буде потрібний  $(N-1)$  регістр пам'яті. Щоб уникнути подібного роду труднощів, необхідно перейти від дерева  $D$  усіх шляхів до стягнутого дерева усіх шляхів (рис. 1в). Воно може бути отримано стягуванням однойменних вершин дерева  $D$  на кожному ярусі. При цьому, вершини з однаковими номерами на кожному ярусі впорядковуються у горизонтальні лінійки. За даними рисунку 1в, множина шляхів у стягнутому дереві шляхів така ж, як і у дереві  $D$  (рис. 1б), але для прочитання шляхів на кожній горизонтальній лінійці дозволяється бути один раз.

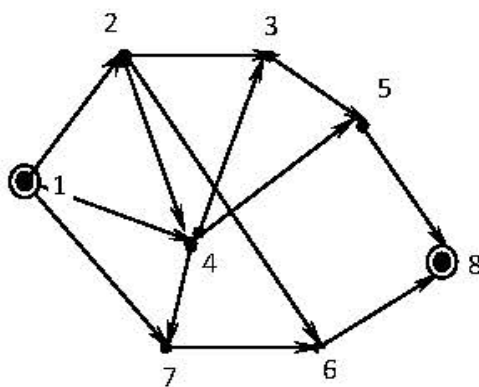
Розглянемо стратегію розподілу процесорів за алгоритмами  $A_i$  у припущенні, що є  $N$ -процесори та кількість вершин у графі алгоритму так само дорівнює  $N$ . Стратегія полягає у тому, щоб задіяти процесори у відповідності із стягнутим деревом шляхів графа алгоритму  $A$ . Тобто, на першому кроці працює процесор  $P_1$ , реалізуючи алгоритм  $A_1$ , розташований на першому ярусі ( $r=1$ ). Далі по матриці суміжності  $B$  встановлюється зв'язок між процесорами  $P_1, P_2, P_3$  та  $P_4$ . Після закінчення роботи усіх процесорів  $P_2, P_3, P_4$  відповідно до матриці суміжності  $B$  встановлюється зв'язок процесорів  $P_2, P_3, P_4$  із процесорами  $P_5$  та  $P_6$ . Результати роботи процесорів  $P_2, P_3, P_4$  передаються на процесори  $P_5, P_6$ , а потім процесори  $P_5, P_6$  реалізують алгоритми  $A_5, A_6$ . По закінченню даного виконання, по матриці суміжності встановлюється зв'язок між процесорами  $P_5, P_6$  і процесором  $P_7$  та йому передається необхідна інформація для роботи. У такий спосіб рішення завдання здійснюється у процесі побудови стягнутого дерева шляхів. На кожному етапі обчислень взаємодіють процесори, що відповідають вершинам у стягнутому дереві шляхів, розташованим на  $n$ -м і  $n+1$  ярусах.

Розглянутий приклад (рис. 1), коли структурна схема алгоритму складена



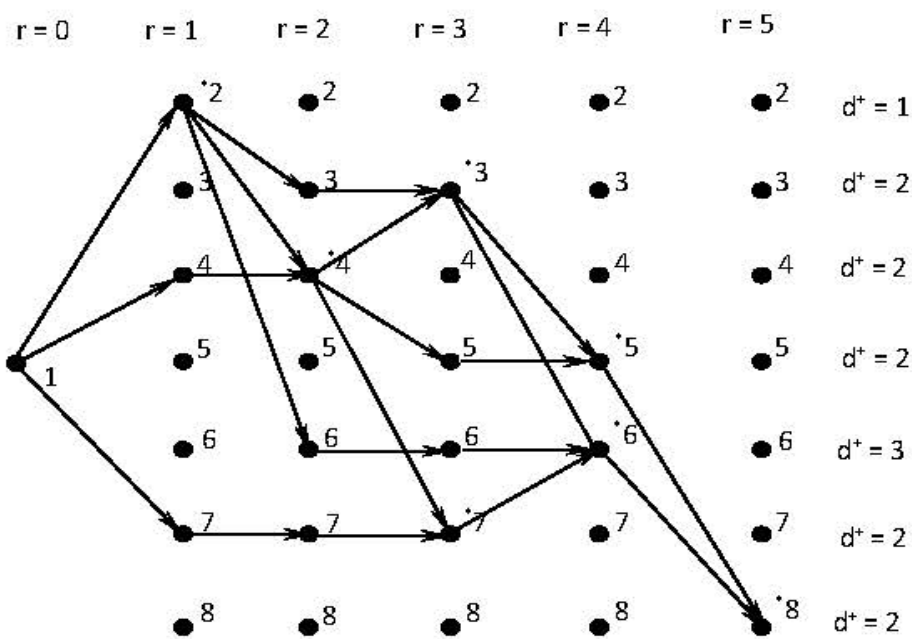
таким чином, що алгоритми  $A_j$  на кожному ярусі одержують необхідні вихідні дані для їхньої реалізації. У загальному випадку, при довільному завданні структурної схеми алгоритму дана умова може не дотримуватись (рис. 2).

	1	2	3	4	5	6	7	8	$d^+$
1	0	1	0	1	0	0	1	0	3
2	0	0	1	1	0	1	0	0	3
3	0	1	0	0	0	0	0	0	1
4	0	0	1	0	1	0	1	0	3
5	0	0	0	0	0	0	0	1	1
6	0	0	0	0	0	0	0	1	1
7	0	0	0	0	0	1	0	0	1
8	0	0	0	0	0	0	0	0	0
$d^-$	0	1	2	2	2	2	2	2	



a)

б)



в)

Рис. 2. Структура графа  $G'$

Наприклад, структурна схема алгоритму  $A$  задана графом  $G'(V,E)$ , який зображений на рисунку 2а. На рисунку 2б наведена матриця суміжності графа  $G$  та значень ступеня  $d^-$  – результату кожної вершини й  $d^+$  – ступеня заходу. Зрозуміло, що ступінь вершини графа  $G'$  дорівнює  $d=d^-+d^+$ .

У даному випадку побудова стягнутого дерева шляхів (рис. 2б) відрізняється тільки лише тим, що від вершини та зв'язок з вершинами наступного ярусу встановлюється лише тоді, коли ступінь заходу вершини досягла величини, що обумовлена матрицею суміжності  $V$ . Фактично, це означає, що процесор з номером  $j$  зайнятий з першої подачі на нього інформації, яка необхідна для реалізації алгоритму  $A$  та він залишається зайнятим до тих пір, поки кількість звертань до нього не досягне ступеня заходу  $d_j^+$ . Ступінь вершини графа  $G'$  дорівнює  $d=d^-+d^+$ . Це означає, що уся вихідна інформація для виконання алгоритму  $A_j$  отримана та процесор  $j$  який здатний його реалізувати. Відповідно до стягнутого дерева шляхів (рис. 2в) виконання алгоритму  $A$ , наданого графом  $G$  (рис. 2а), здійснюється наступним чином.

Процесор  $P_1$  виконує алгоритм  $A_1$ , далі, відповідно до  $V$ , встановлюються зв'язки із процесорами  $P_2, P_4, P_7$  і на них пересилаються результати виконання  $A_1$ . При цьому, процесор  $P_2$  отримує необхідну інформацію та відпрацьовує алгоритм  $A_2$ . Процесори  $P_4, P_7$  зайняті та перебувають у режимі очікування. Далі, по  $V$  встановлюються зв'язки із процесорами  $P_3, P_4, P_6$  та передається інформація, яка необхідна для їх роботи. Ступінь заходу вершини 4 буде дорівнювати  $d^+=2$ , яка обумовлена матрицею  $V$ , і таким чином процесор  $P_4$  може виконати алгоритм  $A_4$ .

На рисунку 2 номери процесорів, що звільнилися на кожному ярусі, позначені зірочками (\*). Для процесора  $P_4$  визначаємо по  $V$  зв'язки із процесорами  $P_3, P_5, P_7$ . При цьому, процесори  $P_3, P_7$  виконують свої алгоритми та передають інформацію на процесори  $P_5, P_6$ , які реалізують алгоритми  $A_5, A_6$  та передають вихідні дані на процесор  $P_8$ . Останній і завершить виконання алгоритму  $A$ .

Для здійснення такої організації обчислень передбачається використати обчислювальну структуру (рис. 3), яка має назву – циклічна паралельна обчислювальна структура. Вона містить  $n$ -процесорів (що складаються з пам'яті  $M_i$  та арифметичного логічного пристрою  $A_i$ ), входи й виходи яких приєднані до комутатора розмірності  $(n \times n)$ .

Комутатор управляється пристроєм управління введення та виведення (ПУВВ) інформації відповідно до матриці  $V$ .

Фактично, така обчислювальна структура й реалізує побудову стягнутого дерева шляхів графа, забезпечуючи зв'язок між процесорами при переході від ярусу до ярусу за допомогою ПУВВ і комутатора  $K$ , на основі матриці  $V$ . Обчислювальний процес у такій структурі можна зобразити у вигляді паралельно обертових кілець, рух яких синхронний або асинхронний, залежно від прийнятої організації синхронізації.

Слід зазначити, що якщо число процесорів дорівнює  $N$  – числу вершин у структурній схемі алгоритму, заданого графом  $G$ , – то на кожному кроці роботи процесорів частина з них простоює. За умови, якщо максимальна ширина алгоритму  $A=h$ , то одночасно на довільному ярусі буде працювати не більше  $h$ -процесорів. Такою кількістю процесорів можна обмежитися у кільцевій структурі, за рахунок чого зменшити число процесорів, що простоюють. Таке можливо досягти, якщо при переході від ярусу до ярусу у дереві стягнутих

шляхів будуть автоматично змінюватися номери процесорів, що комутують, відповідно до матриці суміжності  $B$ .

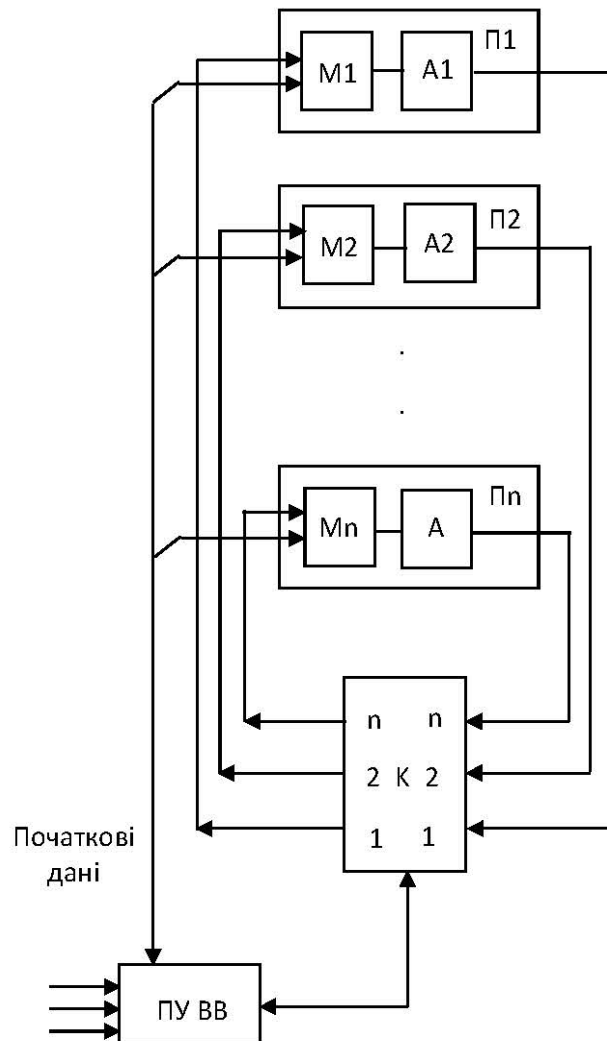


Рис. 3. Обчислювальна структура

Архітектура кільцевої структури здобуває вигляду, який наведено на рисунку 4, де ПКП (пристрій керування перенумерацією комутатора  $K$ ) для залучення процесорів, що відповідають наступному ярусу стягнутого дерева шляхів. При реалізації алгоритму  $A$  на кільцевій структурі час виконання алгоритму відрізняється від довгі критичного шляху тільки на величину  $at_k$ , де  $t_k$  – час спрацьовування комутатора. При побудові комутатора на основі програмувальних логічних матриць  $t_k$  визначається часою перемикання логічного елемента "1". За умови, якщо  $t_k < \min \{t_j\}$ , то час виконання алгоритму  $A$  на циклічній обчислювальній структурі буде у точності дорівнювати довжині критичного шляху.

Розглянемо варіанти побудови паралельної обчислювальної структури циклічного типу для задач оптимізації.

Процес реалізації алгоритму  $A$  розбивається на множину алгоритмів  $\{A_j\}$ , кожний з яких у свою чергу знову розбивається на підмножину алгоритмів  $\{A_j^i\}$  і т.д., до тих пір, поки алгоритм буде зображувати просто одну з базових операцій  $u_i \in Y$ . Множина  $Y$  утворює функціонально повну систему. Тому, процесори  $P_i$

доцільно створювати з використанням транспютерної технології побудови високопродуктивних мікропроцесорів, утворених за принципом RISC – архітектури, яка має локальну пам'ять, що становить два регістри та пристрої запису і зчитування інформації.

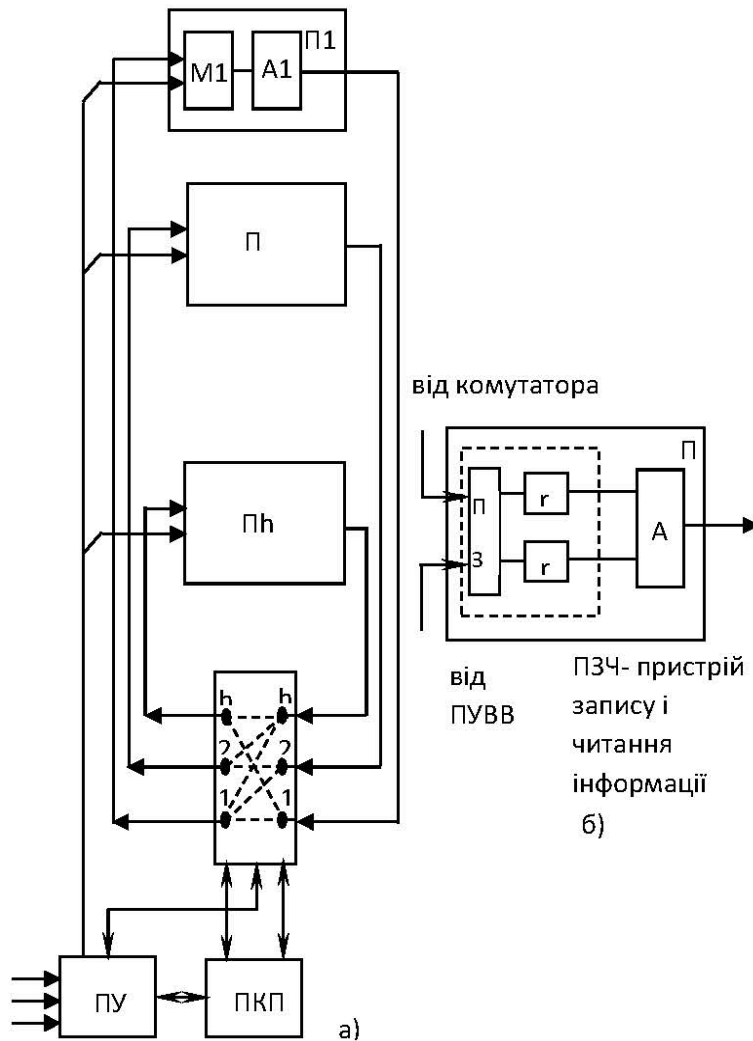


Рис. 4. Архітектура кільцевої циклічної ПОС

RISC процесор містить арифметичний пристрій  $A_i$  (рис. 5), який забезпечує виконання базових арифметичних операцій. Процесор, у цьому випадку, реалізує зазначені операції над двома числами. Таким чином, структурна схема алгоритму  $A$  повинна подаватися у вигляді орієнтованого графа, у якому ступінь заходу вершини не більше двох, а ступеня результату – довільна. У процесорах можна звільнитися від ПЗЧ. Тобто, розпаралелити процес запису вихідної інформації у регістри пам'яті, що вимагає додаткового комутатора  $K$ , який працює у паралель із основними. При цьому, архітектура обчислювальної системи буде мати вигляд, представлений на рисунку 5.

Для оцінки роботи алгоритмів на таких системах можливо використати критерій – коефіцієнт прискорення:

$$K_y = \frac{T_1}{T_n} \tag{1}$$

де  $T_1$  – час реалізації алгоритму  $A$  на гіпотетичній однопроцесорній електронно-обчислювальній системі з такою ж швидкодією, як і  $\Pi_j$  та оперативною пам'яттю, що дорівнює сумарній пам'яті  $\Pi_j$ , при наявності необхідної кількості зовнішніх пристроїв зі швидкістю обміну інформацією як і у циклічній обчислювальній системі;  $T_n$  – час реалізації алгоритму  $A$  на циклічній обчислювальній системі, що містить  $n$ -процесорів.

За умови, якщо алгоритм  $A$  зображений структурною схемою, заданою графом  $G$ , що містить  $n$ -вершин, тоді можливо отримати:

$$T_1 = \sum_{i=1}^n t_i + nt_0 \tag{2}$$

де  $t_i$  – час виконання  $i$ -тої базової операції;  $t_0$  – час обміну.

Час роботи алгоритму  $A$  на структурах можливо визначити наступним чином (рис. 6):

$$T_n = \sum_{j=1}^a t_j^{max} + at_0 \tag{3}$$

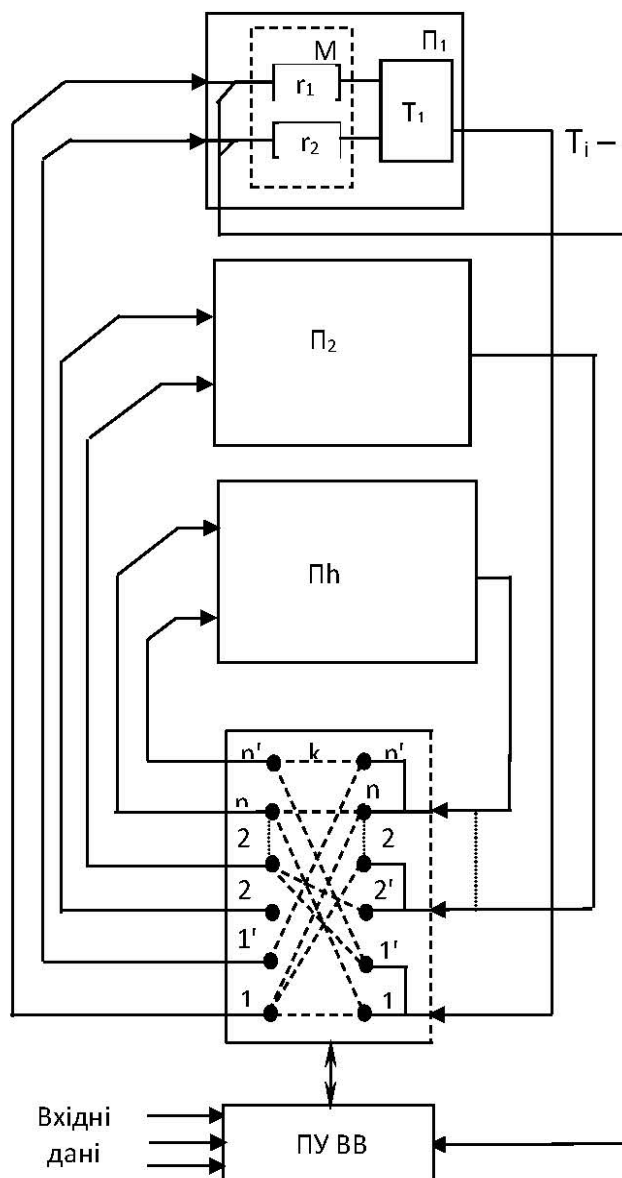


Рис. 5. Обчислювальна структура

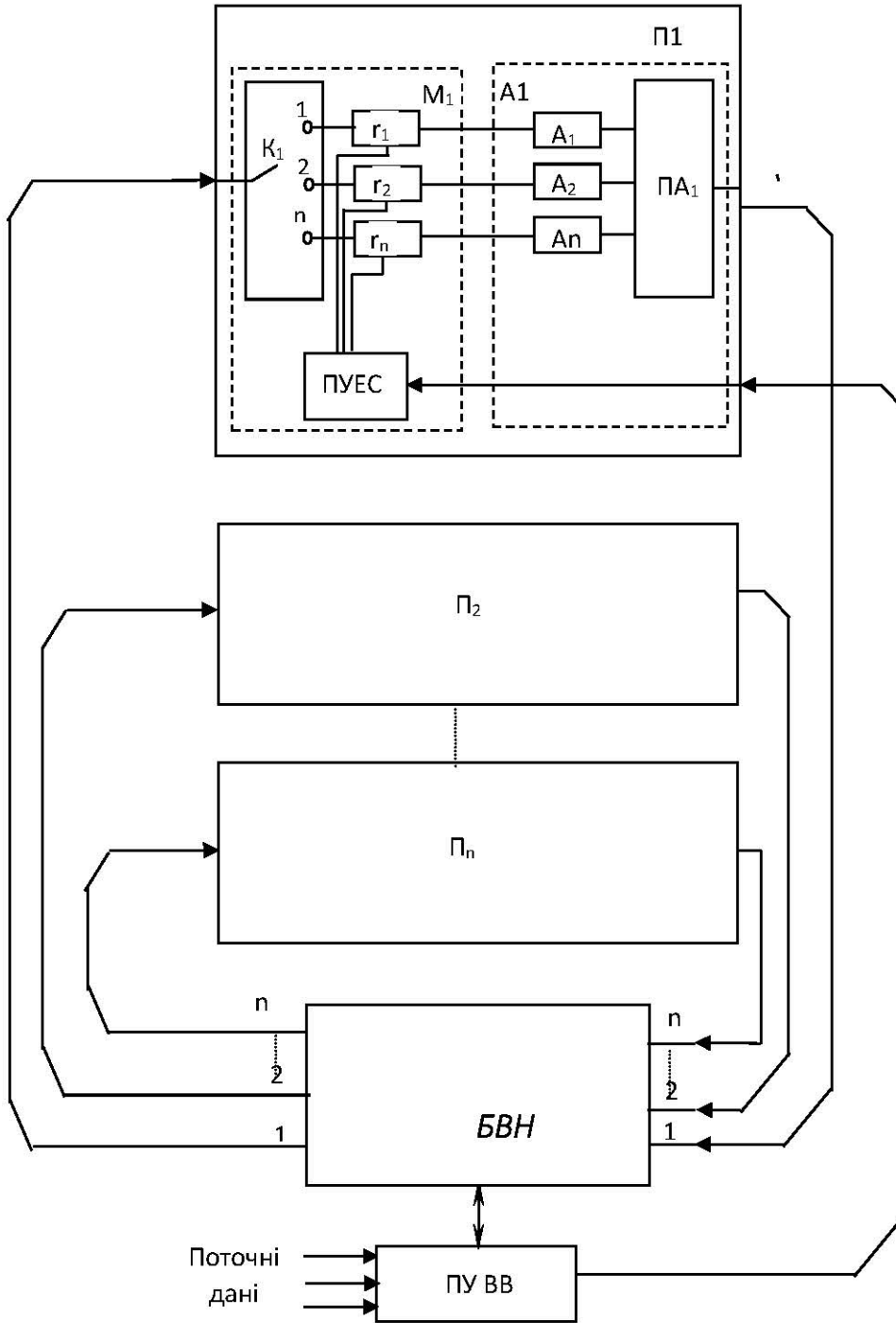


Рис. 6. Обчислювальна структура

де  $t_j^{\max}$  – найбільше із часу основних операцій, які виконуються одночасно декількома процесорами на  $j$ -му кроці;  $a$  – глибина алгоритму  $A$  який заданий графом  $G$ .

Тоді, коефіцієнт прискорення можливо представити наступним чином:

$$K_y = \frac{\sum_{i=1}^n t_i + nt_0}{\sum_{j=1}^a t_j^{\max} + at_0} \quad (4)$$

Для спрощення аналізу допустимо, що:  $t = \bar{t}_k$  - середнє значення по  $i=(1,n)$ ,

а  $t_j^{\max} = \bar{t}_a$  – середнє значення по  $j=(1,a)$ , при цьому:

$$K_y = \frac{n}{a} \alpha, \quad (5)$$

$$\alpha = \frac{\overline{t_k \pm t_0}}{\overline{t_a \pm t_0}}.$$

Із (5) можливо зробити висновок, що зі зменшенням глибини алгоритму  $A$  – зростає коефіцієнт прискорення.

У випадку, коли кожний процесор спеціалізується на виконанні алгоритмів із підмножини алгоритмів  $\{A_j\}$ , можлива організація обчислень циклічного типу на обчислювальній структурі (рис. 6).

У структурі, що зображена на рисунку 6 пам'ять процесора  $P_i$  включає  $(n-1)$  реєстр, закріплений за кожним  $P_i$  процесором ( $i=\overline{1,n}$ ); комутатор  $K_i$ ; пристрій управління записом та зчитуванням; арифметико-логічні блоки (за числом реєстрів); пристрій аналізу  $PA_i$ , що за певним критерієм вибирає з результатів роботи  $A_j$  найкращий.

Кільця циркуляції інформації замикаються через блок вибору напрямку (БВН) передачі інформації, він послідовно здійснює опитування процесорів  $P_i$ . Інформація процесора  $I=1$  за допомогою комутатора  $K_i$  записується у реєстри пам'яті усіх процесорів  $P_i$ , закріплених за процесором з номером  $I$ .

Після закінчення опитування процесорів вони виконують операції за отриманими даними, а результати їх роботи за допомогою БВН та комутаторів  $K$  заносяться у реєстри пам'яті, закріпленої за процесорами.

Описані операції виконуються до повної реалізації алгоритму  $A$ . Для цього варіанту коефіцієнт прискорення визначається:

$$K_y = \frac{\sum_{i=1}^n t_i + nt_0}{\sum_{j=1}^a \overline{t_j^{\max}} + ant_0} \quad (6)$$

Для забезпечення прискорення, у цьому випадку, повинна виконуватись нерівність:

$$\sum_{i=1}^n t_i \quad \sum_{j=1}^a \overline{t_j^{\max}} > nt_0(a-1), \quad (7)$$

$$npu \quad t_i = t_k \quad i \quad \overline{t_j^{\max}} = t_0:$$

$$\frac{n}{a} > \frac{t_k \pm nt_0}{t_s \pm t_0}. \quad (8)$$

У реальних завданнях час виконання алгоритму  $A_j \in A$  відмінний. Тому, за умови, якщо  $\Delta t = (t_1^{\max} - t_2^{\min})$  – час між закінченням виконання самого трудомісткого алгоритму  $A_1 \in \{A_i\}$  та найпростішого  $A_2 \in \{A_j\}$  – використати для опитування процесорів  $P_i$ , то прискорення, що обирається із (6) може зрости до величини обумовленої (5). Останнє еквівалентно пропозиції, що  $n=1$  у знаменнику дроби, обумовленої (6). Таку структуру кільцевої мережі ефективно прийняти для визначення найкоротших шляхів рішення завдань динамічного програмування та варіаційного обчислення.

Реалізація кільцевих структур (рис. 4, 5) можлива й на базі багатоканальних систем зв'язку, що використовують частотне або тимчасове ущільнення. Кількість каналів зв'язку буде визначатися числом регістрів пам'яті, які є у наявності у процесорах  $P_i$ , тобто кожний процесор забезпечується багатоканальним приймачем та передавачем.

Розглянуті структури циклічного типу зможуть успішно конкурувати з конвеєрними обчислювачами, побудованими на базі однорідних обчислювальних систем.

Для задач цілочисельного лінійного програмування (ЦЛП) із булевими змінними (БЗ) паралельна форма узагальненої процедури  $A_0$  збігається із графом  $DD$ . У даному випадку паралельна обчислювальна структура циклічного типу має вигляд, який зображений на рисунку 5.

З рисунку 6 зрозуміло, що роль комутатора виконує процесор  $P_x$ . Процесори  $P_i$   $i=(1,n)$  служать для зберігання й вибору шляху, яке відповідає обраному правилу відсікання  $\{L_w\}$ .

ПУВВ відповідає за початкове завантаження вихідних даних (ваг вершин, каліброваних векторів і т.д.) та вивід результату після одержання припустимого або екстремального рішення. Усім процесом управляє пристрій управління (ПУ). Кожний  $P_i$  процесор відповідає  $i$ -тій вершині графа  $DD$  та зберігає постійно усі властиві її ваги.

На даній структурі завдання вирішується наступним чином. Пристрій керування формує  $n$  раз керуючу послідовність тактів. У кожному такті формується множина шляхів у процесорі  $P_x$ , що відповідають номеру  $k$  даного такту. Для цього, із процесорів  $P_i$  та  $J$   $k$  вибираються, згідно з правилами відсікання  $\{L_w\}$ , шляхи, які узагальнюються у  $P_x$ . Після цього, результат сформованої множини записується у  $k$ -і процесор. Таким чином доти, поки усі множини поточного рангу не будуть сформовані.

По закінченні формування шляхів рангу  $r$  виробляється побудова шляхів рангу  $r=r+1$  і так доти, поки або усі множини не будуть порожніми (що відповідає нульовому лічильнику векторів поточного рангу), або ранг шляху буде дорівнювати  $n$ . Слід зазначити, що для організації зберігання множин як попереднього, так і наступного рангів, необхідно передбачити режим поділу адрес оперативної пам'яті усередині процесорів, щоб при запису множин сформованого рангу не затерти множини збережених векторів поточного рангу.

Систолічні масиви – це спеціалізовані обчислювальні структури, які використовуються для вирішення певних класів обчислювальних задач. Вони мають деякі характеристики та властивості, які роблять їх ефективними для відповідних завдань. До складу деяких з них належать наступні.

*Паралельність:* систолічні масиви призначені для паралельних обчислень. Кожен елемент масиву має власний обчислювальний блок і усі ці блоки працюють одночасно, виконуючи операції над вхідними даними.

*Фіксована структура:* систолічні масиви мають фіксовану структуру, яка означає, що кількість обчислювальних елементів та спосіб їх підключення фіксовані на етапі проектування. Ця фіксована структура дозволяє оптимізувати масив для конкретного завдання.



*Лінійний перебіг даних:* вхідні дані у систолічному масиві пересуваються через обчислювальні елементи послідовно, як у систолі. Це дозволяє досягти великої швидкості обчислень для завдань, де послідовний доступ до даних важливий.

*Локальне зберігання даних:* систолічні масиви зазвичай мають локальне зберігання даних у кожному обчислювальному елементі. Це дозволяє уникнути затрат на доступ до загальної пам'яті та підвищує швидкодію.

*Використання у спеціалізованих задачах:* систолічні масиви найчастіше використовуються для спеціалізованих обчислювальних задач, таких як обробка сигналів, обчислення матричних операцій, обробка зображень тощо. Для загальних задач вони можуть бути менш ефективними.

*Швидкодія:* завдяки паралельній обробці та фіксованій структурі, систолічні масиви можуть досягати високої швидкодії для відповідних задач. Особливо це корисно для реального часу та задач з обмеженим часом виконання.

*Складність проектування:* розробка систолічного масиву може бути складною та вимагати глибоких знань у області архітектури обчислювальних систем і математики. Досягнення оптимальної архітектури для конкретної задачі може бути викликом.

Таким чином, систолічні масиви – є потужним інструментом для спеціалізованих обчислювальних задач, де вони можуть забезпечити високу швидкодію та ефективність, але вони вимагають ретельного проектування та налаштування для кожного конкретного застосування. Приклад реалізації на ПОС систолічного типу розглянуті у [1-5, 15-16].

#### **Список використаних джерел:**

- [1] Listrovoy S.V., Golubnichiy D.Yu., Listrovaya E.S. Solution method on the basis of rank approach for integer linear problems with boolean variables /Engineering Simulation. – 1999. – vol.16. – P. 707-725.
- [2] Listrovoy S.V., Tretjak V.F., Listrovaya A.S. Parallel algorithms of calculation process optimization for the boolean programming problems // Engineering Simulation. – 1999. – vol.16. – P. 569-579.
- [3] Viacheslav, Tretiak, et al. "Parallel Computation Method for Fragmentation of Distributed Database Data Based on Rank-Based Approach." 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT). IEEE, 2019.
- [4] Пономаренко В.С., Голубничий Д.Ю., Третяк В.Ф. Цілочисельне програмування в економіці. – Х.: Вид. ХНЕУ, 2005. – 204 с.
- [5] Третяк, В., Коломійцев, О., Голубничий, Д., Осієвський, С., Калачова, В., & Філіппенков, О. (2023). Варіанти побудови паралельної обчислювальної структури циклічного типу для рішення задач дискретної оптимізації. Grail of Science, (31), 216-227.
- [6] Коломійцев, О., Осієвський, С., Третяк, В., Закіров, З., Романюк, А., Нікітченко, В., Логвиненко, Є., & Лисиця, А. (2021). Задачі дискретної оптимізації та їх постановка. InterConf, (75), 285-302. <https://doi.org/10.51582/interconf.19-20.09.2021.033>.
- [7] Коломійцев, О., Голубничий, Д., Третяк, В., Рибальченко, А., Любченко, О., Полтавський, Е., Кривчун, В., Крамар, О., Шутіков, О., Туленко, М., & Третяк, А. (2023). Використання методів рангового підходу до рішення задачі оптимізації

- розміщення засобів захисту інформації в хмарному середовищі. Scientific Collection «InterConf+», (29(139)), 274-292.
- [8] Коломійцев, О., Старцев, В., Третьак, В., Нікорчук, А., Шаповалов, О., Закіров, З., ... & Рибальченко, А. (2022). Метод рішення задачі оптимізації маршрутів для спеціалізованих машин логістичного забезпечення в автоматизованій інформаційній системі складського обліку на основі рангового підходу. Scientific Collection «InterConf+», (27 (133)), 417-434.
- [9] Голубничий Д.Ю. Інформаційна технологія відсікання неперспективних варіантів в алгоритмах рішення задачі цілочисельного лінійного програмування з булевими змінними на основі рангового підходу // Theoretical foundations in research in Engineering: collective monograph / Д.Ю. Голубничий, О.В. Коломійцев, В.Ф. Третьак [та ін.]; International Science Group. – Boston, 2022. – С. 96-133.
- [10] Третьак, В., Голубничий, Д., Коломійцев, О., Мегельбей, Г., Возний, О., & Філіпенков, О. (2020). Математична модель рангового підходу. Збірник наукових праць ЛОГОС, 116-122.
- [11] Третьак, В., Осієвський, С., Усачова, О., Ірха, А., Булай, А., Бабіч, О., & Шамрай, Н. (2021). Архітектури паралельних обчислювальних структур для рішення задач дискретної оптимізації. InterConf, 462-479.
- [12] Голубничий, Д., Третьак, В., Костенко, І., Поляцко, В., Апполонов, О., Крук, Б., & Закіров, З. (2021). Експериментальне дослідження алгоритмів рішення задач дискретної оптимізації на основі рангового підходу та принципу оптимізації за напрямком. InterConf, (58), 324-337.
- [13] Третьак, В., Деменко, М., Запара, Д., Новіченко, С., Доска, О., & Савельєв, А. (2021). Спосіб рішення задачі цілерозподілу сил та засобів зенітних ракетних військ. InterConf, (40). вилучено із <https://ojs.ukrlogos.in.ua/index.php/interconf/article/view/8350>.
- [14] Третьак В.Ф., Місюра О.М., Більчук В.М. Метод оптимізації структури розподіленої бази даних у вузлах інфокомунікаційної мережі хмарного середовища. // Наука і техніка Повітряних Сил Збройних Сил України. – 2017. – № 1. – С. 92-96.
- [15] Патент на корисну модель № 92925, Україна, МПК G06 F15/00. Спосіб визначення маршруту в графі / В.Ф. Третьак, Ю.Г. Бусигін, Д.Ю. Голубничий та ін. – № u201403580; заяв. 07.04.2014; опубл. 10.09.2014; Бюл. № 17. – 5 с.
- [16] Патент на корисну модель № 69487, Україна, МПК G06 F15/00. Пристрій для рішення задач на графах / В.Ф. Третьак, Д.Ю. Голубничий та ін. – № u201113667; заяв. 21.11.2011; опубл. 25.04.2012; Бюл. № 8. – 6 с.
- [17] Коломійцев, О., Голубничий, Д., Коц, Г., Третьак, В., Євстрат, Д., & Лисиця, А. (2020). Задачі дискретної оптимізації та їх постановка для розміщення засобів захисту в розподіленій системі. Збірник наукових праць ЛОГОС, 36-41. <https://doi.org/10.36074/20.11.2020.v5.12>.

*The scientific periodical*

## **GRAIL OF SCIENCE**

**№ 32 (October, 2023)**

with the proceedings of the VI Correspondence International Scientific and Practical Conference «Globalization of scientific knowledge: international cooperation and integration of sciences» held on October 13<sup>th</sup>, 2023 by NGO European Scientific Platform (Vinnytsia, Ukraine) and LLC International Centre Corporative Management (Vienna, Austria).

*Journal's frequency: monthly*

*All materials are reviewed. The editorial office did not always agree with the position of authors. Authors are responsible for the accuracy of the material.*

### **Contacts of the editorial offices:**

1. 21037, Ukraine, Vinnytsia, Zodchykh str. 18, office 81; NGO «European Scientific Platform» *[Owner of the journal]*  
Tel.: +38 098 1948380; +38 098 1526044

E-mail: info@ukrlogos.in.ua

Certificate of the subject of the publishing business: ДК № 7172 of 21.10.2020.

2. 1110, Österreich, Wien, Simmeringer Hauptstraße 24; LLC «International Centre Corporative Management»

E-mail: rachael.a@iccm.org

Signed for publication 13.10.2023.

Format 60×84/16. Offset paper.

Arial & Open Sans typefaces.

Digital printing. Circulation of 100 copies.

Conventionally printed sheets 25,34.

*Order № 46248.*

*Printed from the finished original layout.*

Publisher [printed copies]:

Sole proprietorship - Gulyaeva V.M.

08700, Ukraine, Obuhiv, Malyska str. 5.

E-mail: 5894939@gmail.com

Certificate of the subject of the publishing

business: ДК № 6205 of 30.05.2018.

*Наукове періодичне видання*

## **ГРААЛЬ НАУКИ**

**№ 32 (жовтень, 2023)**

за матеріалами VI Міжнародної науково-практичної конференції «Globalization of scientific knowledge: international cooperation and integration of sciences», що проводилася 13 жовтня 2023 року ГО «Європейська наукова платформа» (Вінниця, Україна) та ТОВ «International Centre Corporative Management» (Відень, Австрія).

*Щомісячне видання*

*Всі матеріали пройшли рецензування. Редакція не завжди поділяє позицію авторів. За точність викладеного матеріалу відповідальність несуть автори.*

### **Контактна інформація редакції:**

1. 21037, Україна, м. Вінниця, вул. Зодчих, 18/81; ГО «Європейська наукова платформа» *[власник журналу]*  
Тел.: +38 098 1948380; +38 098 1526044

E-mail: info@ukrlogos.in.ua

Свідоцтво суб'єкта видавничої справи: ДК № 7172 від 21.10.2020.

2. 1110, Österreich, Wien, Simmeringer Hauptstraße 24; LLC «International Centre Corporative Management»

E-mail: rachael.a@iccm.org

Підписано до друку 13.10.2023.

Формат 60×84/16. Папір офсетний.

Гарнітура Arial & Open Sans.

Цифровий друк. Тираж: 100 примірників.

Умовно-друк. арк. 25,34.

*Замовлення № 46248.*

*Віддруковано з готового оригінал-макету.*

Виготовлювач [друкованої продукції]:

Друкарня ФОП Гуляєва В.М.

08700, Україна, м. Обухів, вул. Малишка, 5.

E-mail: 5894939@gmail.com

Свідоцтво суб'єкта видавничої справи: ДК № 6205 of 30.05.2018.