УДК 510.52

А.О. Мельник, І.Д. Яковлєва*
Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин;
*Чернівецький національний університет імені Юрія Федьковича,
кафедра комп’ютерних систем та мереж

ПОБУДОВА УСІЧЕНИХ НА ВИХОДІ АЛГОРИТМІВ

Запропоновано метод побудови усічених на виході алгоритмів, який полягає у врахуванні факту відсутності операцій за зменшення кількості отримуваних вихідних даних та передбачає зміну правила обробки даних модифікацією структурної матриці, за допомогою якої подано структуру потокового графа алгоритму та побудову нового алгоритму меншої складності.

This paper describes the design method of truncated on output algorithm, which takes into account the lack of operations when is used the reduced number of output data and supposes changing the rule of data processing by modifying the structural matrix which presents the algorithm flow graph and further synthesis of a new algorithm of less complexity.

Вступ. Час виконання алгоритму залежить від його обчислювальної складності, яка визначається кількістю операцій, необхідних для його виконання. Існують задачі, для розв’язання яких необхідні не всі вихідні дані, які можуть бути обчислені за заданим алгоритмом, а лише якась їх частина. Очевидно, що у такому випадку потрібно виконувати не всі операції алгоритму, а лише ті, які забезпечують отримання потрібної частини вихідних даних. Постає задача побудови нових алгоритмів, які передбачають отримання заданої кількості вихідних даних. Такі алгоритми називають усіченими на виході [1, 2]. Їх обчислювальна складність є меншою порівняно з неусіченими алгоритмами.

У цій роботі запропоновано метод побудови усічених на виході алгоритмів, який полягає у врахуванні факту відсутності операцій за зменшення кількості отримуваних вихідних даних та передбачає зміну правила обробки даних модифікацією структурної матриці, за допомогою якої подано структуру потокового графа алгоритму та побудову нового алгоритму меншої складності, що уможливлює у підсумку прискорити виконання алгоритму.

Огляд літератури. Під поняттям „алгоритм” розуміють точний припис, що задає обчислювальний процес розв’язання задачі, а під задачею сформульоване намагання отримати з множини вхідних даних і початкових умов та з множини можливих вихідних даних підмножину вихідних даних, що повністю задовольняють початкові умови і вхідні дані [3]. Алгоритм характеризується множиною параметрів вхідних, проміжних і вихідних даних, правилом вводу даних, правилом початку, правилом опрацювання даних, правилом закінчення, правилом виводу даних. Правило опрацювання даних визначає структуру алгоритму. За наявності умовних передач керування структура алгоритму може залежати від значень вхідних даних і змінюватися. Надалі розглядатимемо інваріантні до зсуву алгоритми [4], структура яких не залежить від даних.

Порядок усічення алгоритму визначається правилом виведення. Побудова усічених на виході алгоритмів з аналітичного представлення є доволі складною задачею [1]. Тому, враховуючи багатоваріантність усічення, до сьогодні такі алгоритми фактично не використовувались.

Аналіз методів подання алгоритмів за допомогою графів показав перспективність використання потокового графа алгоритму (ПГА), в якому операції алгоритму відповідають вершинам

графа, що розподілені за l ярусами, так що на i-му ярусі ($i=\overline{1,l}$) розміщені тільки вершини, пов'язані дугами, що переміщають дані від вершин попередніх ярусів і не надходять від вершин того самого та наступних ярусів. Таке подання алгоритму у вигляді множини функціональних операторів та дуг, які відображають їх взаємозв'язки, дає змогу розв'язати задачу синтезу усічених на виході алгоритмів залежно від правила виведення.

Із врахуванням розподілу вершин за типами (вхідні вершини – вершини, з яких надходять дані, вихідні вершини – вершини, на які надходить результат виконання алгоритму, та вершини-ФО, що виконують операції алгоритму) наочним є такий метод побудови усічених на виході алгоритмів [5].

Нехай задано діапазон [a,b] вихідних вершин, які беруть участь у формуванні вихідних даних. Для синтезу усіченого на виході ПГА залежно від правила виведення спочатку необхідно визначити вершини-ФО, які пов'язані дугами із вихідними вершинами діапазону [a, b], і позначити їх як такі, що потребують виконання, надавши ознаку «виконувати». На наступному кроці визначити усі решта вершин-ФО, що пов'язані дугами із вершинами-ФО, які мають ознаку «виконувати». Після аналізу всіх вершин-ФО видалити із ПГА вершини-ФО, що не мають ознаки «виконувати». Отриманий усічений на виході ПГА міститиме тільки ті вершини, які відповідають за отримання результуючого вектора вихідних даних.

Недоліком цього методу є те, що із збільшенням розмірності алгоритмів ускладнюється їхнє опрацювання.

У [6] запропоновано виконувати запис ПГА у формі структурної матриці (СМ), яка зберігає структуру алгоритму у зручній для опрацювання формі. Цей метод подання ПГА у формі СМ передбачає виконання: етапу маркування ПГА та етапу запису інформації.

На етапі маркування ПГА здійснюють нумерацію вершин – функціональних операторів (ФО) числами від 1 до N_f, де N_f – кількість вершин-ФО ПГА та нумерація дуг ПГА. Нумерацію дуг розпочинають з першого ярусу і кожній вхідній дузі цього ярусу присвоюють натуральне число j ($j=\overline{1,n}$), де n – кількість вхідних дуг графа, а від нього вниз за правилом: вихідним дугам присвоюють довільне значення із множини натуральних чисел N, що присвоєні дугам, які входять у цю вершину. Причому кількість вхідних дуг будь-якої вершини графа k ($k=\overline{1,N_f}$) дорівнює a ($a=\overline{1,n}$), а кількість вихідних дуг будь-якої вершини k дорівнює b ($b=\overline{1,n}$). Для кожної вершини повинна виконуватися нерівність $a \geq b$.

На етапі запису ПГА здійснюють перегляд усіх дуг j ($j=\overline{1,n}$) кожного ярусу i ($i=\overline{1,l}$) та в елемент f_{ij} СМ записують номер вершини k, в яку надходить ця дуга j, або *0*, якщо у цьому i-му ярусі вона не надходить у жодну з вершин, а перетинає цей ярус; в усі інші елементи СМ, виділені для цього ярусу, записують число, яке не входить у множину номерів вершин, та продовжують виконання цих дій до того часу, поки не будуть переглянуті усі дуги графа.

Процес відображення ПГА із структурної матриці полягає у зчитуванні вмісту елемента f_{ij} СМ та залежно від зчитаних значень відображення вершини k, що є вмістом елемента f_{ij} та дуги j між портами вершин в i-му ярусі доти, поки не буде зчитано вміст усіх елементів СМ і відображено усі дуги та вершини графа. Відновлення ПГА із СМ виконується за одноразовим аналізом СМ.

Використання СМ дає змогу досліджувати ПГА не із графічного подання, а з його формального подання СМ, що дає можливість автоматизувати цей процес.

Постановка задачі. Оскільки синтез усічених на виході алгоритмів з аналітичного представлення є складною задачею, актуальним є розроблення методу побудови усічених на виході алгоритмів із структурної матриці, яка зберігає структуру алгоритму у зручній для опрацювання формі.

Побудова усічених на виході алгоритмів. Побудова усічених на виході алгоритмів виконується модифікацією СМ та полягає у фіксації номерів елементів останнього рядка *i* СМ, другий індекс яких належить діапазону усіченого вектора вихідних даних та присвоєнню нуля іншим елементам цього рядка, і наступному визначенню елементів в попередньому рядку *(i-1)* з таким самим другим індексом, як і в зафіксованих елементах *i*-го рядка і пари до нього, та присвоєнню нуля іншим елементам цього рядка і виконанню цих дій доти, доки не буде переглянуто усю СМ. Після виконання усіх операцій СМ міститиме тільки номери тих операцій, які беруть участь у формуванні заданого діапазону вихідних даних. Цей метод змінює правило обробки даних алгоритму і приводить до отримання усіченого на виході алгоритму з меншою обчислювальною складністю.

Блок-схему процесу побудови усічених на виході алгоритмів із структурної матриці показано на рис. 1.

Вхідними даними для запропонованого методу є структурна матриця *F*, яка описує структуру ПГА, та діапазон *[a,b]*. Результат побудови знаходитиметься в усіченій СМ F_u , де ФО, які не використовуються для отримання вихідних даних із заданого діапазону, будуть замінені нульовими значеннями.

Процес побудови усічених на виході алгоритмів із СМ передбачає виконання таких кроків:

1) створити матрицю F_u з розмірами матриці *F*, заповнену нулями;

2) починаючи з останнього рядка СМ *F*, визначити елементи, що є останніми в стовпці матриці і задовольняють умові: $f_{ij} = f_{iz} = k$, $j \in [a,b]$ або $z \in [a,b]$;

3) переписати елементи СМ F $f_{ij} = f_{iz} = k$ в СМ Fu з тими самими індексами: $f_{u\,ij} \leftarrow f_{ij}$; $f_{u\,i,z} \leftarrow f_{iz}$. (Знак $\leftarrow$ означає присвоювання).

Отже, після виконання цих кроків в СМ F_u будуть записані значення, що дорівнюють номеру вершини-ФО ПГА, результат якої бере участь у формуванні вектора вихідних даних, що заданий діапазоном $[a,b]$.

На наступному кроці з СМ F_u визначити елементи СМ F $f_{i+1,j} = f_{i\,i+1,z} = k$, що не дорівнюють нулю, та елементи СМ F $f_{ij} = f_{iz} = k$ записати в СМ Fu з тими самими індексами: $f_{u\,i} \leftarrow f_{ij}$; $f_{u\,i,z} \leftarrow f_{iz}$. І так виконувати до того часу, доки не буде опрацьовано всю СМ.

В такий спосіб отримується СМ F_u, що містить тільки ті номери вершин ГА, що беруть участь у формуванні вектора вихідних даних, виконується зміна правила обробки даних і отримання усіченого на виході алгоритму.

Отримані результати. ШПФ [7] є основою сучасних систем цифрової обробки сигналів і часто зустрічається на практиці там, де швидкість роботи доволі важлива (під час побудови засобів аналізу та розпізнавання образів, кодування сигналів і зображень, ідентифікації систем). Характерною ознакою ШПФ є багаторазове повторення простих базових операцій "метелик". ПГА ШПФ має *N* входів. Вхідні дані поступають в інверсному двійковому порядку. ПГА містить $log_2 N$ ярусів. На кожному ярусі виконується *N/2* незалежних перетворень „метелик". Обчислювальна складність алгоритму швидкого перетворення Фур'є дорівнює $N/2 \log_2 N$ базових операцій. ПГА ШПФ для *N*=32 показано на рис. 2.

Побудований на основі СМ усічений на виході до діапазону [1,10] ПГА ШПФ показано на рис. 3.

За усічення до діапазону [1, 10] відбулося зменшення кількості базових операцій «метелик» з 80 до 43. Обчислювальна складність алгоритму зменшилася майже на 50 %.

Дослідження усічених на виходів алгоритмів ШПФ для N від 128 до 4096 точок відліку для вектора вихідних даних від максимальної величини до одиниці дало змогу отримати дані та побудувати графіки зменшення обчислювальної складності алгоритму.

Отримані результати показали зменшення обчислювальної складності алгоритму у найкращих випадках від 70 до 80 % порівняно з неусіченим алгоритмом (рис. 4), а за *N* = 65563 – до 90 %.

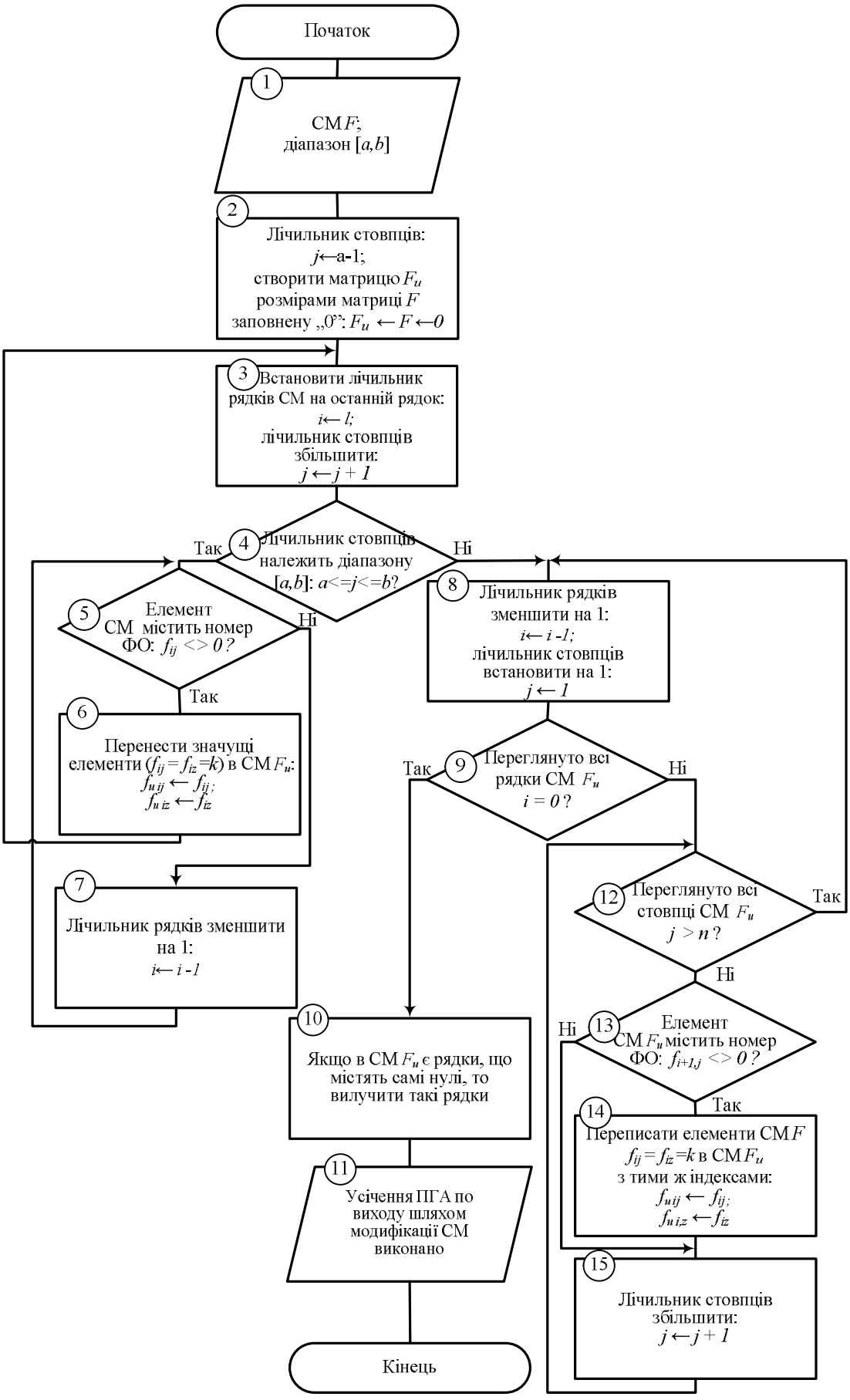

Рис. 1. Блок-схема процесу побудови усічених на виході алгоритмів із структурної матриці

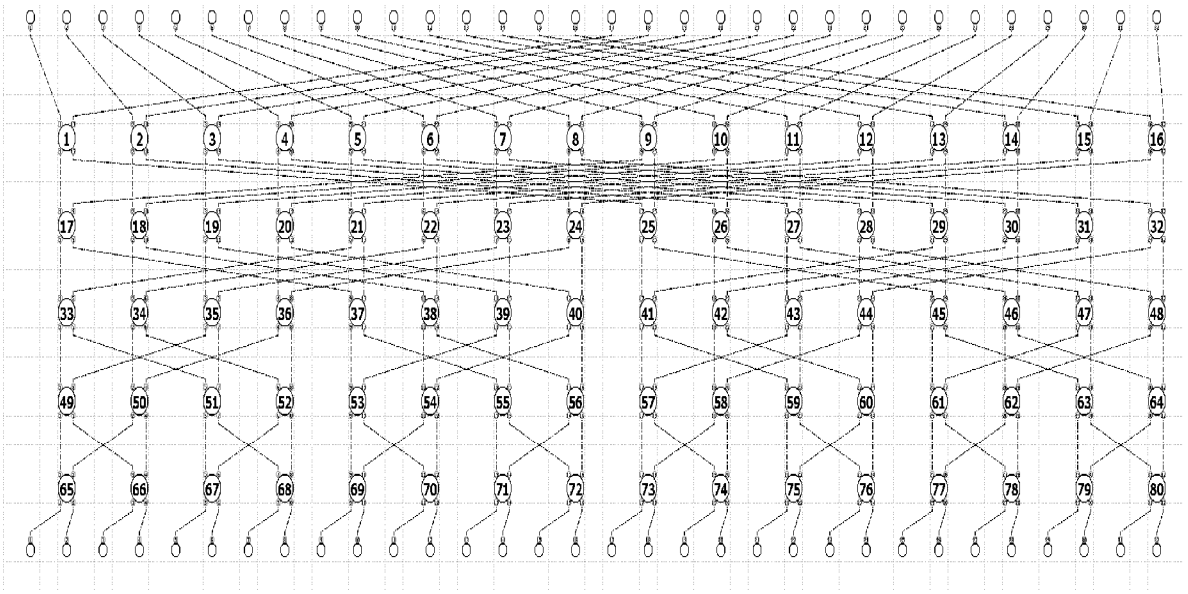

Рис. 2. ПГА ШПФ для N=32

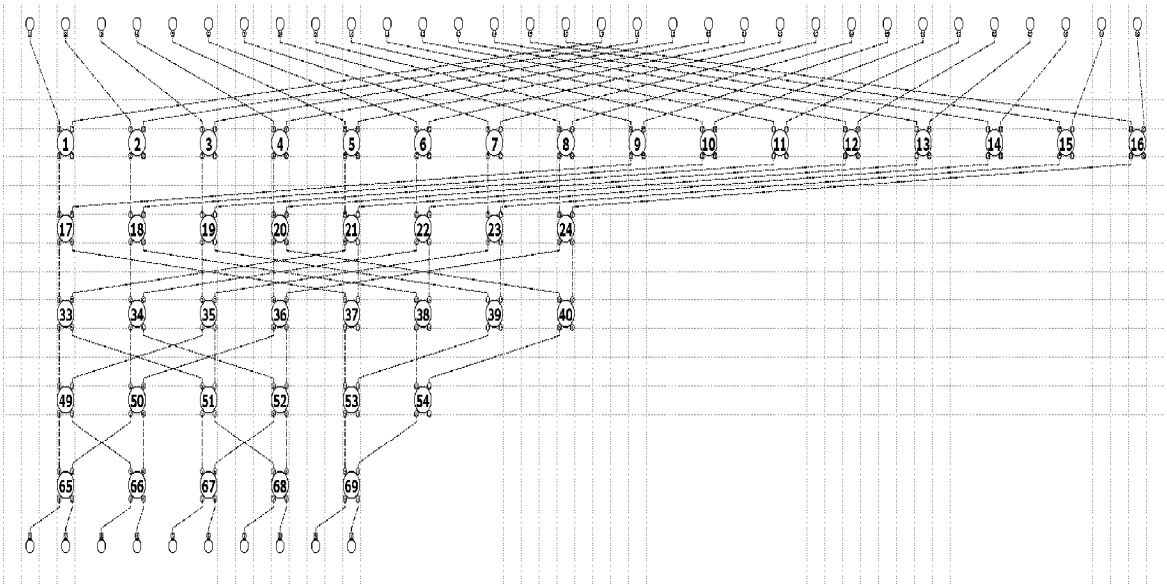

Рис. 3. Усічений на виході до діапазону [1,10] ПГА ШПФ для N=32

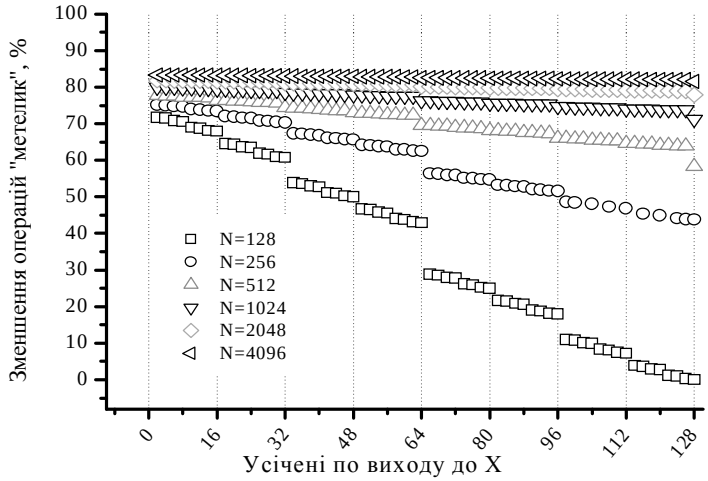

Рис. 4. Обчислювальна складність усічених на виході алгоритмів ШПФ

Розроблений метод реалізований у графічній системі дослідження й опрацювання структури алгоритмів “ОСА” [5].

Висновки. Запропоновано метод побудови усічених на виході алгоритмів, який полягає в модифікації структурної матриці, за допомогою якої подано структуру потокового графа алгоритму, який призводить до зміни правила обробки даних та зменшення обчислювальної складності. За цим методом синтезовано усічені на виході алгоритми ШПФ зі зменшеною обчислювальною складністю.

1. Hoeven J. The truncated fourier transform and applications / Joris van der Hoeven // International Conference on Symbolic and Algebraic Computation // Proceedings of the 2004 international symposium on Symbolic and algebraic computation table of contents.– Santander, Spain, 2004 .– P. 290–296. 2. Melnyk A.. Parameters of Algorithm / A. Melnyk, V. Melnyk // Proceedings of Conference “Modern Problems in Electrical Engineering and Informatics”, Politechnika Świętokrzyska, Ameliówka, Poland, 17–18.06.2005. – P. 115-121. 3. Мельник А.О. Архітектура комп’ютера / А. Мельник. – Луцьк: Волинська обласна друкарня, 2008. – 470 с. 4. Мельник А.О. Спеціалізовані комп’ютерні системи реального часу / А.О. Мельник – Львів: Вид-во Національного університету "Львівська політехніка", 1996. – 60 с. 5. Мельник А. Графічна система для дослідження та опрацювання структури алгоритмів / А. Мельник, І. Яковлєва // Матеріали 4-ї Міжнар. конф. “Сучасні комп'ютерні системи та мережі: Розробка та використання”. – Львів: Вид-во Львівської політехніки, 2009. – С. 67-71. 6. Мельник А. О. Подання потокового графа алгоритму структурною матрицею / А.О. Мельник, І.Д. Яковлєва // Технічні науки. – Хмельницький: Хмельницький національний університет, 2008. – № 4 – С. 124–129. 7. Рабинер Л. Теория и применение цифровой обработки сигналов / Л. Рабинер, Б. Гоулд. – М.: Мир, 1978. – 848 с.