УДК 004.272

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

ПОРІВНЯЛЬНИЙ АНАЛІЗ СПОСОБІВ МАТРИЧНОГО ПОДАННЯ ГРАФУ АЛГОРИТМУ

Порівнюються способи матричного подання графа алгоритму та виявлено, що завдяки забезпеченню впорядкованого запису розподілу вершин за ярусами та зв’язків між ними структурна матриця займає менший об’єм пам’яті, необхідний для збереження графа алгоритму, ніж інші матриці.

In this paper the comparison of methods for algorithm graph matrix representation is performed and is found that the structural matrix needs less memory to save the algorithm graph than the other matrixes as it allows the ordered saving of the nodes of graph layers and connections between them.

Вступ. Аналіз сучасних способів проектування операційних пристроїв показав, що прискорення продуктивності обробки даних досягається завдяки наближенню структури алгоритмічного операційного пристрою (АОП) [1, 2] до структури виконуваного алгоритму. Чим більше адаптована структура АОП до структури ГА, тим більше покращуються його характеристики. Така адаптація досягається шляхом перекладення обчислювальних процедур з програмного забезпечення на апаратну частину.

Повне апаратне відображення потокового графу виконуваного алгоритму (ПГА) комбінаційними схемами (КС), що виконують операції, задано функціональними операторами (ФО) алгоритму і з’єднано між собою відповідно до ПГА, надає такі переваги: використовується максимальна паралельна форма алгоритму; операцію можна виконувати незалежно від стану інших операцій; обмін даними між операціями чітко визначений; управління операціями здійснюється за допомогою передачі даних між ними; алгоритм виконується за один такт, що надає багаторазового прискорення виконанню конкретної задачі [1].

Отже, оскільки для проектування АОП необхідно володіти інформацією про обчислювальні та просторово-часові параметри алгоритму, то актуальним питанням є пошук зручного представлення алгоритму на рівні зв’язків між окремими операціями або блоками операцій.

Огляд літератури. Аналіз способів подання алгоритмів показує, що завдяки своїй наочності найпоширеніший графічний спосіб, тобто подання алгоритму у вигляді графу, а також його варіант – потоковий граф алгоритму (ПГА) [1]. Такі графи ще називають граф-машина [2], граф-схема паралельного алгоритму, ярусно-паралельна форма алгоритму [3], інформаційний граф задачі [3], але суть їх побудови однакова і означає розподіл всіх його вершин за ярусами так, що в і-му ярусі розміщені тільки функціональні оператори (ФО), які залежать від ФО попередніх ярусів і не залежать від ФО наступних ярусів (рис. 1, б). У межах одного ярусу між вершинами такого ГА зв’язків немає. Всі вхідні дані переміщуються по дугах ПГА одночасно. Якщо на шляху даних, які переміщуються дугами, зустрічається вершина, то дані беруть участь в операції, що задана вершиною ПГА – ФО, інакше – просто передаються на наступний ярус. У результаті перетікання даних між ФО ПГА і виконання операцій, заданих алгоритмом, на виході ПГА отримується результат виконання алгоритму. Така форма подання алгоритму визначає ступінь паралелізму w ПГА (максимальну кількість вершин найширшого ярусу ПГА), висоту l паралельної форми алгоритму (кількість ярусів) та обчислювальні характеристики, що задаються множиною F_i функціональних операторів. Виникає питання його формального опису для зручного зберігання в комп'ютерах, дослідження, аналізу та модифікації.

Для зберігання в комп'ютері графу алгоритму та його опрацювання кращим способом подання алгоритму є матричний [4]. Але матриці інциденцій та суміжності не зберігають розподіл вершин ПГА за ярусами та вимагають для збереження багато пам'яті.

Оскільки подання ПГА матрицею спрощує його зберігання та опрацювання, а відомі матричні способи не забезпечують збереження розподілу вершин за ярусами та вимагають великих об'ємів пам'яті, в [5] запропоновано спосіб формального опису потокових графів, інваріантних до зсуву алгоритмів, використовуючи подання ПГА структурною матрицею.

Постановка задачі. Оскільки ПГА – орієнтований ациклічний граф із джерелами (вектор вхідних даних) та стоками (вектор результату), де кожна вершина лежить на одному з шляхів із джерела в стік, то вирішувати задачу формального опису, дослідження, аналізу та модифікації можна за допомогою відомих матричних способів опису графів. Але реальні алгоритми, що потребують аналізу, містять велику кількість операцій та змінних, що може сягати тисяч та робить графи алгоритмів достатньо великими. Виникає питання визначення кращого способу подання графу алгоритму, що зберігає розподіл вершин ПГА за ярусами та займає менше пам'яті.

1. Подання графу алгоритму матрицею інциденцій

Матриця інциденцій – це матриця розміру $N \times M$, де N – кількість вершин ГА, M – кількість дуг ГА [4]. Заповнення матриці інциденцій здійснюють за правилом: в елемент матриці з індексами *(j,i)* ($j = \overline{1,M}$, $i = \overline{1,N}$) записують одиницю, якщо вершина інцидентна дузі і є її початком, в елемент *(j,i)* – мінус одиницю, якщо вершина інцидентна дузі і є її кінцем, та нуль, якщо вершина та дуга не інцидентні.

Недоліком цього способу є те, що він вимагає найбільшого об'єму для зберігання матриці (розмір пропорційний кількості вершин та дуг ПГА) і тому незручний для зберігання. Загальна кількість елементів S_i матриці інциденцій дорівнює

$$S_i = N \times M , \tag{1}$$

а об'єм пам'яті V_i для зберігання такої матриці залежить від кількості елементів матриці: $V_i = \Theta(S_i) = \Theta(N \times M)$, де функція Θ – асимптоматична форма запису і використовується для спрощення аналізу та запису темпу росту досліджуваної функції [5].

$V_i = \Theta(N \times M)$ тоді і тільки тоді, коли існують додатні константи c_1 та c_2 та n_0 такі, що $c_1(N \times M) \le V_i \le c_2(N \times M)$ при $N \ge n_0$. Функції V_i та $(N \times M)$ зростають з однаковою швидкістю при всіх $N \ge n_0$.

Крім того, матриця інциденцій доволі рідко використовується, оскільки для графів з великою кількістю ребер вона має дуже велику кількість стовпців.

Також недоліком подання ГА в формі матриці інциденцій є неможливість збереження ПГА, оскільки він не забезпечує збереження розподілу вершин за ярусами.

2. Подання графу алгоритму матрицею суміжності

Матриця суміжності – це квадратна матриця порядку N, де N – кількість вершин графу [4]. Заповнюють матрицю суміжності за правилом: в елемент матриці з індексами *(i,j)* ($i = \overline{1,N}$, $j = \overline{1,N}$) записують кількість дуг a ($a = \overline{1,n}$, n - кількість вхідних дуг графу), що з'єднують вершини i та j, які є суміжними й вершина i є початком для даних дуг та нуль в інших випадках.

Загальна кількість елементів матриці суміжності дорівнює квадрату кількості вершин ГА:

$$S_{сум} = N^2, \tag{2}$$

Недоліком цього способу є те, що для представлення ГА за допомогою матриці суміжності загальну кількість вершин потрібно знати заздалегідь. Якщо граф повинен створюватися та (або) змінюватися, то для кожного додавання або знищення вершини необхідно будувати нову матрицю. Крім того, навіть якщо граф містить невелику кількість дуг, і матриця суміжності складається переважно з нулів, пам'ять повинна бути виділена для всіх можливих дуг, незалежно від того, чи вони існують. Об'єм пам'яті для збереження матриці квадратично збільшується при збільшенні кількості вершин, оскільки матриця суміжності – квадратна матриця порядку N. Загальна кількість елементів матриці суміжності – N^2 і вимагає пам'яті незалежно від кількості дуг у графі:

$$V_{сум} = \Theta(N^2). \tag{3}$$

Отже, недоліком способу подання ГА у формі матриці суміжності є потреба у великому обсязі пам'яті та неможливість збереження ПГА, оскільки він не забезпечує збереження розподілу вершин за ярусами.

Більш економним по відношенню до пам'яті (особливо у випадку, коли кількість дуг $|M|$ набагато менша за кількість вершин $|N^2|$) порівняно з матрицею суміжності, є спосіб представлення графу за допомогою структури суміжності [5], яка є списком пар, що відповідають його дугам. Кожна дуга у цьому списку дуг задається двома числами – номерами вершин цієї дуги. Список дуг зручніший для реалізації алгоритмів на графах порівняно із матрицею суміжності.

Кількість комірок пам'яті, необхідних для збереження ПГА цим способом, залежить від суми кількості вершин та дуг ГА $V_{спис\_сум} = \Theta(max(N,M)) = \Theta(N+M)$.

Недоліком способу збереження в пам'яті графу алгоритму в формі списку суміжності є неможливість збереження ПГА, оскільки він не забезпечує збереження розподілу вершин за ярусами ПГА та виконання великої кількості кроків (порядку M у найгіршому випадку) для отримання множини вершин, які є суміжними з цією вершиною. У випадку, коли кількість дуг набагато більша за кількість вершин, інформація про вершини стає важкодоступною, отже, і цей спосіб не є доцільним для використання.

3. Подання графу алгоритму символьною матрицею

Символьна матриця – це матриця порядку N, де N – кількість вершин графу [6]. Символьну матрицю заповнюють за правилом: в елемент матриці з індексом *(i,i)* ($i = \overline{1,N}$) записують одиницю, а в елементи *(i,j)* ($i = \overline{1,N}$, $j = \overline{1,N}$) – символьне позначення дуги $a(i,j)$, якщо вершина і зв'язана дугою $a(i,j)$ з вершиною *j* і є її початком, заперечення символьного позначення дуги $a(i,j)$, що позначається рискою згори ($\overline{a(i,j)}$), якщо вершина *i* зв'язана дугою $a(i,j)$ з вершиною *j* і є її кінцем, та записують в (*i,j*) нуль, якщо вершини *i* та *j* не зв'язані.

Кількість елементів матриці, необхідних для збереження ГА цим способом, дорівнює квадрату кількості вершин ГА, а об'єм пам'яті залишається таким самим високим, як і в матриці суміжності (3) $V_c = \Theta(N^2)$.

Недоліком способу збереження в пам'яті ГА в формі символьної матриці є потреба у великому об'ємі пам'яті та неможливість збереження ПГА, оскільки він не забезпечує збереження розподілу вершин за ярусами ГА.

Отже, недоліком цих способів є великий об'єм пам'яті для зберігання ГА в комп'ютері, складність запису та необхідність виконання додаткових операцій для визначення розподілу вершин за ярусами ПГА.

Оскільки символьна матриця має таку саму кількість елементів, як і матриця суміжності, далі виконуємо порівняння тільки з матрицею суміжності.

4. Подання графу алгоритму структурною матрицею

Подання ПГА структурною матрицею [7] полягає в нумерації вершин та дуг ПГА за правилом: 1) у найширшому ярусі привсвоїти кожній дузі ПГА число $j, j = \overline{1,n}$, де n – кількість вхідних дуг ПГА; 2) виконати нумерацію дуг всіх решти ярусів за правилом: номери вхідних та вихідних дуг для будь-якої вершини ПГА однакові і збігаються; та в записі інформації про ПГА в СМ, кількість стовпців якої дорівнює кількості дуг у найширшому ярусі ПГА, а кількість рядків – загальній кількості ярусів ПГА. Елементами такої матриці є номери ФО, які записують в елемент ij СМ, де i – номер ярусу, а j – номер дуги, яка надходить у цей ФО.

При збереженні ПГА СМ об’єм пам’яті залежить від добутку кількості ярусів на кількість дуг ширшого ярусу – $l \times n$, де l – кількість ярусів, а n – кількість дуг ширшого ярусу $V_{стр} = \Theta(S_{стр}) = \Theta(l \times n)$.

Використання СМ дає змогу зберегти розподіл ФО за ярусами ПГА та зв’язки між ними у зручному для опрацювання комп’ютером матричному вигляді, що дає можливість автоматизувати процес збереження, дослідження та аналізу ПГА, але виникає питання ефективності застосування структурної матриці для опису потокових графів алгоритмів порівняно із відомими способами.

5. Порівняння матричних способів подання графів алгоритмів

Оскільки розмір пам’яті для збереження ГА залежить від кількості елементів матриць, порівняємо ці величини. Для порівняння кількості елементів матриць необхідно подати їх розмір в однакових величинах. Оскільки матриці суміжності та інциденцій виражені через загальну кількість вершин та дуг, визначимо розмір СМ також у цих величинах.

Нехай граф алгоритму має M дуг та N вершин. Розглянемо найгірший випадок, коли кількість ярусів l є найбільшою, тобто в кожному ярусі знаходиться тільки одна вершина (вироджений граф). Тоді кількість елементів СМ дорівнює $S_{стр} = N \times n$.

У роботах [8–10] досліджували структуру випадкового графу і визначили, що N вершин можуть бути зв’язані максимум

$$M = \frac{N(N-1)}{2} \tag{4}$$

ребрами. Така ситуація відповідає випадку, коли всі вершини зв’язані між собою.

Припустімо, що і в нашому випадку кількість дуг є максимальною і всі вершини пов’язані між собою. Тоді, виразивши в (1) кількість ребер через кількість вершин (4), отримаємо кількість елементів матриці інциденцій: $S_i = N \times \frac{N(N-1)}{2} = \frac{N^2(N-1)}{2}$, що в $\frac{(N-1)}{2}$ разів більше ніж кількість елементів матриці суміжності (2).

Якщо в ярусі ПГА міститься тільки одна вершина, що зв’язана із рештою (N -1) вершинами дугами, то в такому ярусі міститься не більше (N -1) дуг. А оскільки за умови N вершин розподілені за N ярусами, то кожен ярус містить не більше (N -1) дуг. З цього випливає, що найбільша кількість дуг n в ярусі дорівнює (N-1). Тоді кількість елементів СМ у найгіршому випадку дорівнює $S_{стр} = N \times n = N \times (N-1)$, що завжди менше за кількістю елементів матриці суміжності (2). Отже, кількість елементів СМ завжди менша за кількість елементів матриць суміжності та інциденцій: $\left(N \times (N-1)\right) < N^2 < \frac{N^2(N-1)}{2}$.

Оскільки об’єм пам’яті, що необхідний для збереження ПГА, залежить від кількості елементів матриць, то можна зробити висновок, що об’єм пам’яті СМ завжди менший за об’єм матриць суміжності та інциденцій: $V_{стр} < V_{сум} < V_i$. Крім того, СМ зберігає розподіл вершин за ярусами ПГА.

6. Результати експериментальних досліджень

Отримані результати показали, що кількість дуг ПГА для досліджених алгоритмів не наближається до свого максимального значення (4), а кількість ярусів – до кількості вершин N. Це значно зменшує кількість елементів СМ, а отже, і розмір пам'яті для її зберігання.

Для алгоритмів швидкого перетворення Фур'є (ШПФ) сортування за модифікованим способом Бетчера, парно-непарної перестановки, способом Бетчера та двійкового множення різної кількості вхідних значень отримані матриці, що описують їх ГА. Результати наведені у таблиці.

Розміри матриць ГА

Параметри ГА				Загальна кількість елементів матриці		
к-ть вхідних даних n	вершин, N	дуг, M	ярусів, l	інциденцій, V_i (NxM)	суміжності, V_c (NxN)	структурної V_s (lxn)
1.	2.	3.	4.	5.	6.	7.
ПГА						
сортування за модифікованим способом бульбашки						
8	112	232	13	25984	12544	104
64	8064	16192	125	130572288	65028096	8000
256	130560	261376	509	34125250560	17045913600	130304
1024	2095104	4191232	2045	8781066928128	4389460770816	2094080
сортування за способом парно-непарної перестановки						
8	112	232	8	25984	12544	64
64	8064	16192	64	130572288	65028096	4096
256	130560	261376	256	34125250560	17045913600	65536
1024	2095104	4191232	1024	8781066928128	4389460770816	1048576
сортування за способом Бетчера						
8	19	46	6	874	361	48
64	543	1150	21	624450	294849	1344
256	3839	7934	36	30458626	14737921	9216
1024	24063	49150	55	1182696450	579027969	56320
ШПФ						
8	12	32	3	384	144	24
64	192	448	6	86016	36864	384
256	1024	2304	8	2359296	1048576	2048
1024	5120	11264	10	57671680	26214400	10240
двійкового множення з горизонтальним розповсюдженням перенесення						
8	120	304	21	36480	14400	1344
64	8128	20352	189	165421056	66064384	774144
256	130816	327168	765	42798809088	17112825856	50135040
1024	2096128	5240832	3069	10985454698496	4393752592384	3218079744
двійкового множення з діагональним розповсюдженням перенесення та за алгоритмом Дадда						
8	120	304	16	36480	14400	1024
64	8128	20352	128	165421056	66064384	524288
256	130816	327168	512	42798809088	17112825856	33554432
1024	2096128	5240832	2048	10985454698496	4393752592384	2147483648

Із таблиці видно, що із збільшенням кількості вхідних даних кількість елементів для збереження ГА задопомогою матриць інциденцій та суміжності експоненційно зростають, тоді як кількість елементів для збереження структурної матриці збільшується лінійно, що для ШПФ показано на рис. 1.

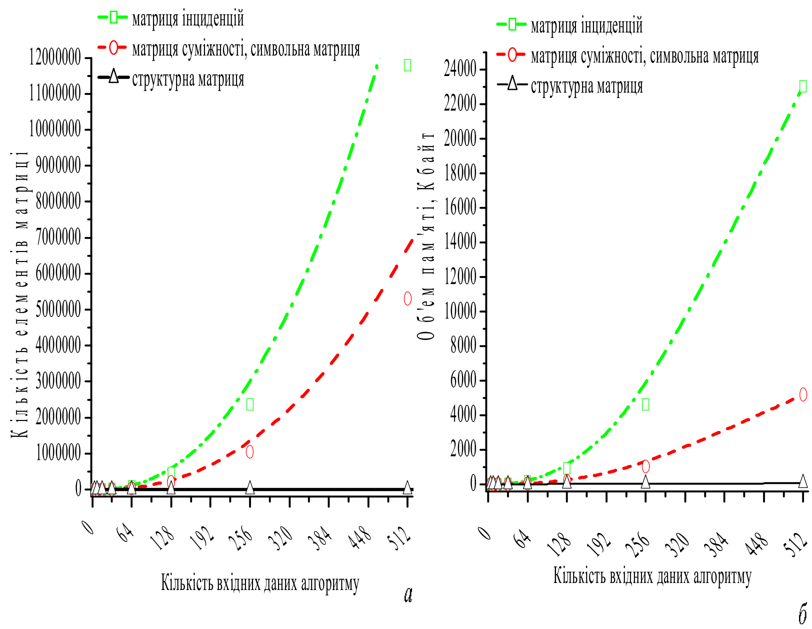

Рис. 1. Порівняння розмірів матриць для подання ГА ШПФ

Якщо V – це об'єм пам'яті, що необхідний для збереження ГА матрицею інциденцій V_i або матрицею суміжності V_c і дорівнює 100%, то ефективність використання пам'яті, що показує, наскільки менше пам'яті займає СМ відносно матриць інциденцій та суміжності, розраховується за формулою:

$$e = \Theta\left((1 - \frac{V_{стр}}{V}) \times 100\% \right). \quad (5)$$

Підставивши в (5) із табл. 1 розміри матриць, що описують ГА ШПФ, сортування та двійкового множення, отримали, що СМ займає майже у сто разів менше пам'яті, ніж матриці інциденцій та суміжності (рис.2).

Отримані результати для різних алгоритмів підтверджують ефективність використання СМ порівняно з іншими матрицями та показують, що із збільшенням розмірів ГА СМ займають пам'яті майже в сто разів менше, ніж матриці інциденцій та суміжності.

Для точнішого оцінювання об'ємів пам'яті для збереження ГА необхідно врахувати розмір комірки пам'яті для запису одного елемента матриці. Для матриці суміжності розмір комірки становить один біт, оскільки необхідно виконати запис 0 або1, для матриці інциденцій – два біти, оскільки необхідно мати можливість запису множини $\{0,1,-1\}$ та для СМ розмір комірки дорівнює кількості бітів, що необхідні для запису значення, яке дорівнює загальній кількості вершин ГА. З таблиці видно, що із збільшенням величини вхідного вектора даних загальна кількість вершин ГА збільшуються лінійно, тоді як розмір матриць суміжності та інциденцій має експоненційну залежність. Тому для оцінювання об'ємів пам'яті із збільшенням розміру матриць розмір комірки стає неістотним (рис. 3).

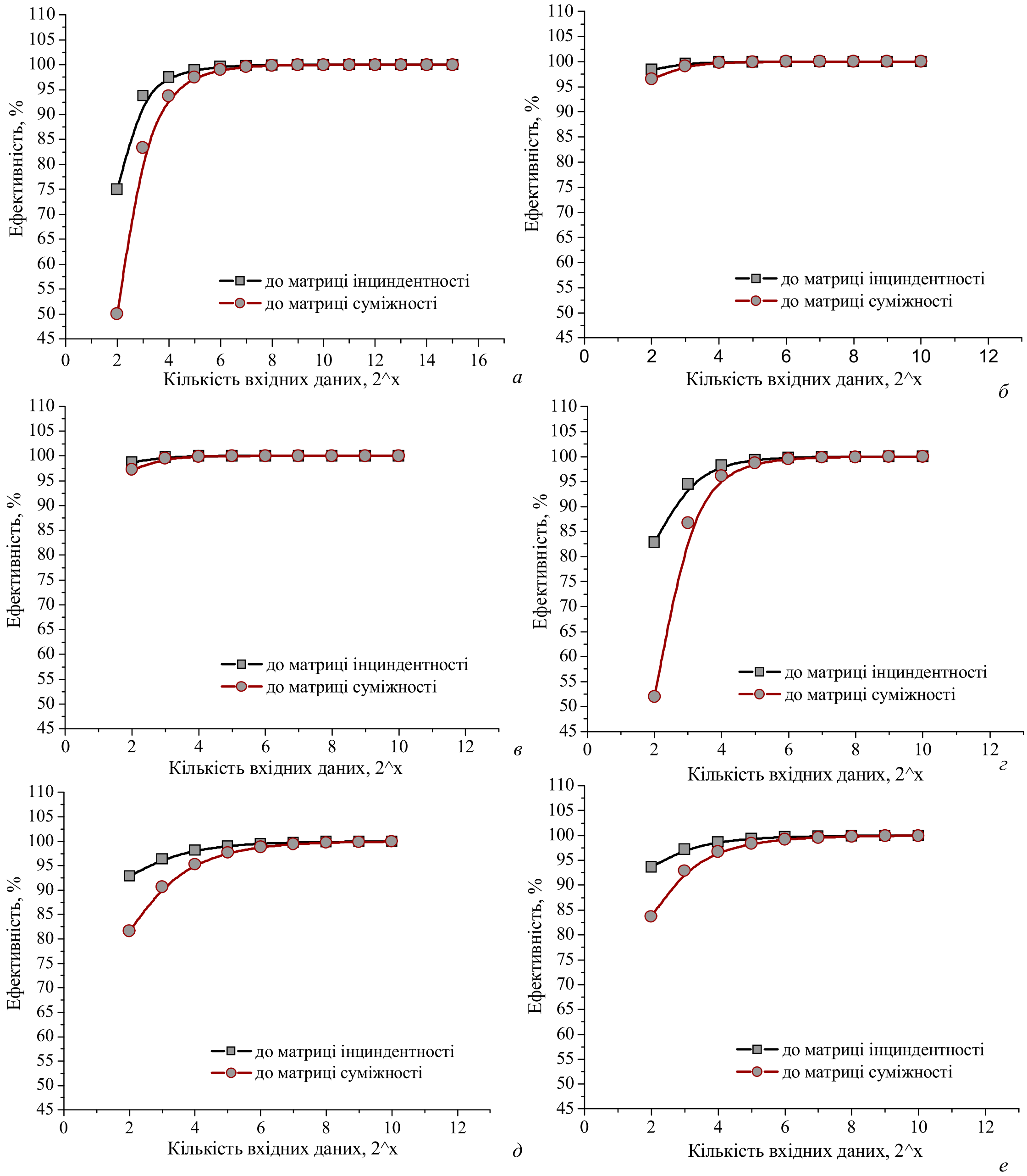

Рис. 2. Ефективність використання пам'яті СМ для: а – ПГА ШПФ; б – ПГА сортування за модифікованим способом бульбашки; в – ПГА сортування за способом парно-непарної перестановки; г – ПГА сортування за способом Бетчера; д – ПГА двійкового множення з горизонтальним розповсюдженням переносу; е – двійкового множення з діагональним розповсюдженням переносу та за алгоритмом Дадда

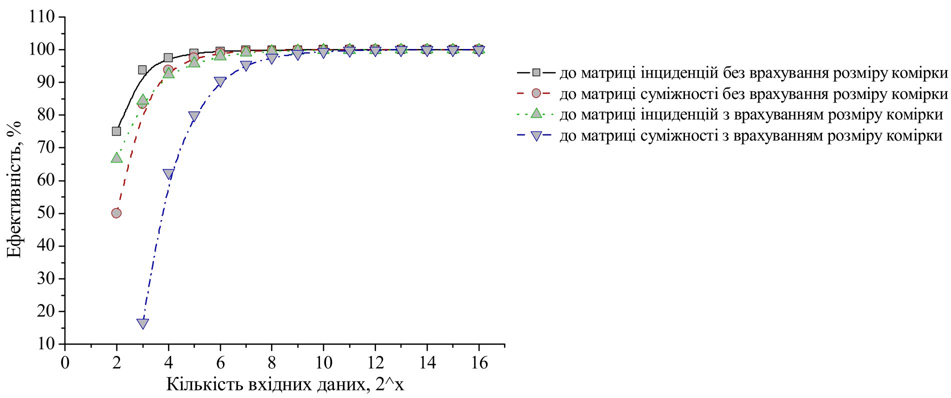

Рис. 3. Ефективність використання пам'яті СМ для ГА ШПФ

Висновки. Виконане порівняння способів матричного подання ГА виявило, що СМ за рахунок впорядкованого запису розподілу вершин за ярусами та зв'язків між ними, крім того, що зберігає ПГА, зменшує об'єм пам'яті, що необхідний для збереження ПГА, оскільки розмір СМ залежить не від загальної кількості дуг та вершин ПГА, як у відомих матриць, а від кількості ярусів, за якими ці вершини закріплені та від кількості дуг найширшого ярусу.

Результати досліджень показали, що із збільшенням кількості вхідних даних алгоритму кількість комірок пам'яті для збереження потокових графів різних алгоритмів за допомогою матриці інциденцій, суміжностей та символьної матриці експоненційно зростають, тоді як кількість комірок пам'яті для збереження структурної матриці збільшується лінійно, та підтвердили ефективність використання СМ порівняно з матрицями інциденцій та суміжності, оскільки для різних алгоритмів із збільшенням розмірів ПГА СМ займають пам'яті майже в сто разів менше ніж інші матриці.

1. Мельник А. О. Спеціалізовані комп'ютерні системи реального часу / А. О. Мельник – Львів: НУ «Львівська політехніка», 2002. – 60 с. 2. Глушков В. М. Синтез цифровых автоматов / В.М. Глушков – М. : Физматгиз, 1962–- 476 с. 3. Шпаковский Г.И. Архитектура параллельных ЭВМ / Г.И Шпаковский. – Мн.: Университетское, 1989. – 192 с. 4. Оре О. Теория графов / О. Оре. – М.: Наука. 1980. – 336 с. 5. Миллер Р. Последовательные и параллельные алгоритмы / Р. Миллер, Л. Боксер – М.:БИНОМ. Лаборатория знаний, 2006. – 960 с. 6. Рабкин Е.Л. Дискретная математика. Булевы функции и элементы теории графов [Електронний ресурс] / Е.Л. Рабкин, Ю.Б. Фарфоровская // Библиотека учебно-способических пособий – Режим доступу: http://dvo.sut.ru/libr/himath/w163rabk/11.htm. 7. Мельник А. О. Подання потокового графа алгоритму структурною матрицею / А. О. Мельник, І. Д. Яковлєва // Науковий журнал "Технічні науки" – Хмельницький: Хмельницький національний університет, 2008. – №4 – С. 124 – 129. 8. Erdos P. On random graphs / P. Erdos, A. Renyi // Publ. Math. (Debrecen), 1959. – №6 – P.290 – 297. 9. Erdos P. On the evolution on random graphs / P. Erdos, A. Renyi // Publ. Math. Inst .Hung. Acad. Sci ., 1960. – № 5A – P. 17–61. 10. Erdos P. On the sizength of connectedness of a random graph / P. Erdos, A. Renyi // Acta Math. Acad. Sci. Hung., 1961– №6 – 261-267.

УДК 681.3

О.Ю. Мельников, О.В. Пустова
Донбаська державна машинобудівна академія,
кафедра інтелектуальних систем прийняття рішень

РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМІ ДЛЯ ТОРГОВЕЛЬНОЇ КОМПАНІЇ З ПРОДАЖУ КОМП'ЮТЕРІВ І КОМПЛЕКТУЮЧИХ

Розглянуто проблему автоматизації функціонування торговельної компанії з продажу комп'ютерів та комплектуючих. Мовою моделювання UML розроблено модель інформаційної системи, здійснено її комп'ютерну реалізацію.

The problem of automation of the trading company to sell computers and accessories is considered. In the language of UML the model of information system is developed, its computer implementation is carried out.

Вступ. Сьогодні конкурентоспроможність, адекватність і рентабельність бізнесу сильно залежать від того, наскільки повно і оперативно дані про бізнес-процеси надходять до керівництва. високої ефективності управління здатні досягти тільки ті компанії, де застосовуються найсучасніші інформаційні технології і організовано замкнутий цикл передавання даних усіма інформаційними каналами. Такі компанії вигідно виділяються серед конкурентів високою якістю управління і