

3. Крячок С. Д. Побудова полігонометричних мереж зустрічними ходами / С. Д. Крячок, Л. С. Мамонтова // Вісник геодезії та картографії. – 2014. – № 5. – С. 9–12.

4. Островський Ф. Л. Геодезія : підручник / А. Л. Островський, О. І. Мороз, В. Л. Гарновський. – Львів : Львівська політехніка, 2007. – Ч. II. – 508 с.

5. Селиханович В. Г. Практикум по геодезії / В. Г. Селиханович, В. Н. Козлова, Г. П. Логинова. – М. : Недра, 1970. – 288 с.

УДК 681.3.082.5

**С.О. Нестеренко**, канд. техн. наук

Чернігівський національний технологічний університет, м. Чернігів, Україна

### АЛГОРИТМ ЗШИВАННЯ ПОЛІГОНАЛЬНИХ СІТОК

**С.А. Нестеренко**, канд. техн. наук

Черниговский национальный технологический университет, г. Чернигов, Украина

### АЛГОРИТМ СШИВАНИЯ ПОЛИГОНАЛЬНЫХ СЕТЕЙ

**Serhii Nesterenko**, PhD in Technical Sciences

Chernihiv National University of Technology, Chernihiv, Ukraine

### THE ALGORITHM FOR POLYGONAL NETWORKS OF CROSS-LINKING

*Стаття містить аналітичний виклад головних кроків алгоритму поєднання двох відкритих полігональних сіток третьою додатковою. Вирішені задачі пошуку вільного краю відкритої полісітки як просторової полілінії, визначення підмножин з'єднаних вершин на обох вільних краях та побудови проміжної поєднальної сітки, яка описується способом Б. Баумгарта.*

**Ключові слова:** комп'ютерне геометричне моделювання, тривимірні поверхневі моделі, полігональна сітка, уявлення Баумгарта, цілісність.

*Статья содержит аналитическое изложение основных этапов алгоритма соединения двух открытых полигональных сетей третьей дополнительной. Решены задачи поиска свободного края открытой полисетки как пространственной полилинии, определения подмножеств соединяемых вершин на обоих свободных краях и построения промежуточной соединяющей сети, которая описывается способом Б. Баумгарта.*

**Ключевые слова:** компьютерное геометрическое моделирование, трехмерные поверхностные модели, полигональная сеть, представление Баумгарта, целостность.

*The article contains an analytical summary of the main stages of the algorithm for connecting two open polygonal networks with third one. The problems of the search of the open polymesh free edge as 3D polyline, a specific subset of connected vertices on both free ends and building the intermediate connecting network which is described in the B. Baumgart's representation are solved.*

**Key words:** computer geometric modeling, 3D surface models, polygonal network, representation by Baumgart, integrity.

**Постановка проблеми.** Побудова просторових поверхневих моделей на основі полігональних сіток (мереж) для реальних тіл нерідко виконується з використанням лазерних технологій [1]. Реальне тривимірне тіло освітлюється променем лазера, координати яскравої точки на поверхні тіла від лазерного променя фіксуються тим чи іншим способом і на базі масиву координат подібних точок будується математичний опис поверхні тіла у формі полігональної сітки (полісітки). Як показує практика [2], полігональні сітки, одержані таким способом, нерідко є розривними, тобто неоднозв'язними. Типовим варіантом результату є такий, коли замість єдиної замкненої полісітки, яка описувала б все тіло, ми маємо кілька незамкнених сегментів (відсіків) полісіток.

Звідси виникає загальна задача ремонту полісіток, що мають описані вище дефекти. Під цим розуміється одержання коректної моделі об'єкта у вигляді замкненої полігональної сітки. Цю роботу можна успішно виконати лише за умови наявності регулярного алгоритму з'єднання двох незамкнених полісіток. Дано формальний аналітичний виклад подібного алгоритму.

**Аналіз останніх досліджень і публікацій.** Найбільш детальний огляд алгоритмів та методів роботи з просторовими полігональними сітками наведено в [3]. Разом з тим слід зазначити, що ці автори апріорі вважають просторові полігональні моделі цілісними та внутрішньо несуперечними. Проблема ремонту просторової сітки, який має на меті надання останній цілісності, не розглядається взагалі.

Деякі технічні аспекти надійного одержання множини координат просторових точок при лазерному скануванні реальних об'єктів обговорюються в [1] та [2]. Яким чином результати замірів мають перетворюватись в адекватну 3D-модель, придатну для комп'ютерного моделювання, автори не зазначають.

Таким чином, задача опису регулярного алгоритму відновлення цілісності полігональних сіток поки не знайшла висвітлення в літературі і тому є актуальною.

**Мета статті.** Головною метою цієї роботи є розроблення і формальний опис алгоритму поєднання двох відкритих полігональних сіток за допомогою додаткової третьої.

### Виклад основного матеріалу.

#### *Побудова просторової полілінії вільного краю відкритої полісітки.*

Припустимо, що дана відкрита (незамкнена) полігональна сітка, тобто така, яка має вільний край. Першою підзадачею алгоритму, що описується, є одержання опису просторової полілінії, яка і є, власне, вільним краєм. Умовою приналежності ребра до вільного краю є те, що у нього з одного боку розташована грань полісітки, а з протилежного – нічого, тобто точки вільного модельного простору. Таким чином, для пошуку полілінії вільного краю нам потрібен спосіб опису полісіток, адекватний поставленій задачі. З досвіду побудови поверхневих моделей тривимірних тіл найбільш функціональною виглядає [3] модель Б. Баумгарта, яка відома також як «крилата» модель.

Згідно з [4], така модель спирається на класичне використання таблиць вершин, ребер та граней полігональної сітки. Таблиця вершин є масивом тривимірних точок

$$V = \{V_i(x_i, y_i, z_i) \mid i \in (1 \dots n)\}, \quad (1)$$

де  $x_i, y_i, z_i$  – декартові координати вершини.

Другою, найбільш інформативною, частиною моделі в уявленні Б. Баумгарта є таблиця ребер, доповнена топологічною інформацією. Зміст останньої пояснимо таким чином. Традиційна репрезентація ребра зводиться до простої пари інцидентних вершин, кожна з яких однозначно визначається індексом  $i \in (1 \dots n)$  з формули (1). Б. Баумгарт доповнив це уявлення топологічними даними, для використання яких були введені додаткові обмеження логічної моделі полігональної сітки:

– ребро  $a$  (рис. 1) є орієнтованим від початкової вершини  $M$  до кінцевої  $N$ . Порядок вершин неявно задається порядком запису термінальних вершин ребра в елементі масиву ребер;

– порядок обходу ребер кожної грані – за годинниковою стрілкою, дивлячись ззовні тіла.

Додаткові топологічні дані, що були запропоновані [4] Б. Баумгартом, пояснюються схемою на рис. 1.



Рис. 1. Ребро полісітки в уявленні Б. Баумгарта

Відносно традиційного варіанта, коли ребро  $a \leftrightarrow \{M; N\}$  в цьому випадку суттєвими доповненнями є:

– з орієнтованим ребром  $MN$  пов'язуються грані  $R$  та  $L$ , розташовані відповідно справа та зліва від ребра. На практиці вказуються або індекси  $i$  цих граней ( $i \in [1; G_{max}]$ , ( $G_{max}$  – кількість граней полісітки), або вказівники на них. Домовимось використовувати константу NULL як значення вказівника на будь-який відсутній елемент полігональної сітки;

– з ребром  $MN$  зв'язуються інцидентні йому сусідні ребра правої та лівої граней, які передують йому ( $d$  та  $c$ ) або слідують за ним ( $b$  та  $e$ ) в порядку обходу відповідної грані.

Отже, множина ребер полігональної сітки згідно з Баумгартом може бути описана як

$$E = \{\{a_i; M_i; N_i; b_i; c_i; d_i; e_i; L_i; R_i\}_i \mid i \in [1; G_{max}]\}, \quad (2)$$

де сенс згаданих змінних ясний із викладеного вище.

Множина (2) програмно відображається на таблицю ребер, елементами якої є записи зі структурою, що відповідає параметрам ребра в уявленні Баумгарта. Ми використаємо таку логічну модель полісітки як найбільш адекватну поставленій задачі.

Тепер припустимо, що дана відкрита (незамкнена) однозв'язна полігональна сітка, тобто така, яка має вільний край (рис.2).

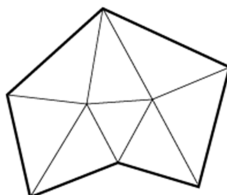


Рис. 2. Полісітка з відкритим краєм

Для пошуку множини вершин просторової полілінії вільного краю можна запропонувати таку послідовність дій.

Початок алгоритму 1.

Крок 1. Знайти будь-яке ребро  $E_0$ , в якого відсутня грань чи справа, чи зліва. Критерій пошуку такого ребра в нотації мови програмування С:

$$((L == NULL) \wedge (R == NULL)) == TRUE. \quad (3)$$

Запам'ятати початкову вершину  $V_0$  цього ребра і записати її до масиву  $M_{vfree}$  вершин полілінії вільного краю полісітки. Очевидно, що масив  $M_{vfree}$  є фізичною реалізацією множини вершин вільного контуру

$$V_{free} = \{V_{free\ i} \mid i \in [1... k]\}, \quad (4)$$

де  $k$  – кількість вершин вільного контуру.

Прийняти  $V_0$  як поточну початкову вершину  $V_k$  поточного ребра, яке розглядається.

Крок 2. Перейти на поточному ребрі до протилежної вершини ребра  $V_{k+1}$  і записати його до масиву  $M_{vfree}$  вершин полілінії вільного краю полісітки. Якщо  $V_{k+1} = V_0$ , вийти з циклу на точку 4. Якщо ні, то перебором усіх ребер, які інцидентні цій вершині, знайти ребро, що відповідає умові (3).

Крок 3. Прийняти  $V_{k+1}$  як  $V_k$  наступної ітерації і перейти на крок 2.

4. Точка виходу з циклу.

Кінець алгоритму 1.

Запропонований алгоритм забезпечує послідовний обхід відкритої полісітки ребрами вільного краю та формує масив  $M_{vfree}$  вершин полілінії вільного краю полісітки. Пари вершин цього масиву відповідають кінцевим вершинам ребер вільного краю. Таким чином, задача побудови полілінії останнього вирішена.

#### **Виділення множини з'єднаних вершин у парі крайових контурів.**

Постановка задачі. Розглянемо пару вільних країв двох полісіток, що мають бути поєднані описуваним алгоритмом (рис. 3). З очевидних практичних міркувань можна сформулювати обмеження на множину вершин вільних країв першої та другої полісітки, що будуть поєднуватись додатковою полісіткою: нею мають «зшиватись» вершини контурів, просторово найближчі одні до одних. Такі вершини показані на рис. 3 білими точками, на відміну від тих, які в поєднанні задіяні не будуть, і вони показані чорними точками.

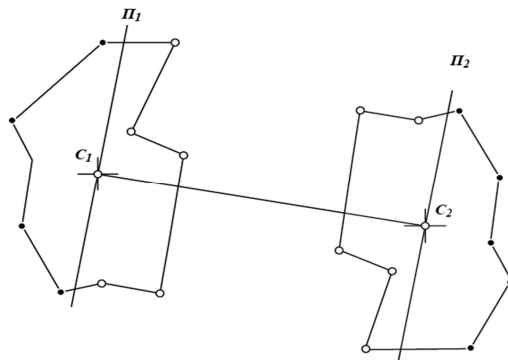


Рис. 3. Пара вільних контурів (проекція на площину малюнка):  $\circ$  – вершини, які будуть поєднуватись;  $\bullet$  – вершини, які не будуть поєднуватись

Наведемо евристичний алгоритм, за яким у множинах  $V_{free}^1$  та  $V_{free}^2$  вершин вільних контурів 1 і 2 виділяються підмножини  $V_{free}^{1c}$  та  $V_{free}^{2c}$  вершин, придатних для подальшого поєднання (показані на рис. 3 білими точками). Ідея алгоритму полягає в наступному.

Для кожного вільного контуру знаходиться умовний центр  $C$  і тим самим визначається пара міжцентрових векторів  $\overline{C_1C_2}$  та  $\overline{C_2C_1}$ . Вектор  $\overline{C_1C_2}$  визначає перпендикулярну йому площину  $\Pi_1$ , яка проходить через точку  $C_1$ . Аналогічно через вектор  $\overline{C_2C_1}$  і точку  $C_2$  визначається площина  $\Pi_2$ . На рис. 3 показані сліди цих площин на площині рисунка. Подальші дії стосуються визначення того, з якої сторони відповідної площини знаходяться точки множин  $V_{free}^1$  та  $V_{free}^2$ . У підмножини  $V_{free}^{1c}$  та  $V_{free}^{2c}$  мають входити тільки точки, розташовані з доданої сторони відповідної площини.

Початок алгоритму 2.

Крок 1. Обчислити координати усереднених точок (умовних центрів)  $C_1$  та  $C_2$  вільних контурів за формулою (наводиться для одного контуру):

$$C = \frac{1}{k} \sum_{i=1}^k V_i, i \in [1...k], \tag{5}$$

де  $V_i$  – радіус-вектор  $i$ -ї вершини.

Крок 2. Обчислити міжцентрові вектори

$$a_1 = C_2 - C_1; a_2 = C_1 - C_2. \tag{6}$$

Крок 3. Визначити рівняння площин  $\Pi_1$  і  $\Pi_2$  :

– для площини  $\Pi_1$

$$(P_{\Pi_1} - C_1) \cdot a_1 = 0; \tag{7a}$$

– для площини  $\Pi_2$

$$(P_{\Pi_2} - C_2) \cdot a_2 = 0, \tag{7б}$$

де  $P_{\Pi_1}$  – довільна точка площини  $\Pi_1$  ;

$P_{\Pi_2}$  – довільна точка площини  $\Pi_2$ .

Крок 4. Перебором точок множини  $\{V_{free}^1\}$  сформуувати підмножину  $\{V_{free}^{1c}\}$  за умовою  $(V_{free}^{1c} - C_1) \cdot a_1 > 0$ .

Крок 5. Перебором точок множини  $\{V_{free}^2\}$  сформуувати підмножину  $\{V_{free}^{2c}\}$  за умовою  $(V_{free}^{2c} - C_2) \cdot a_2 > 0$ .

Кінець алгоритму 2.

**Побудова поєднувальної полігональної сітки**

Попередній розділ як результат мав дві просторові полілінії (опорні лінії)  $A$  та  $B$  (рис. 4, а), на основі яких ми можемо побудувати проміжну полігональну сітку (ППС), що є метою цього дослідження. Без втрати геометричної загальності з метою побудови простого алгоритму в подальшому викладі будемо спиратись на спрощену топологічно

тотожну [5] геометричну модель цих ліній (рис. 4, б), в якій вершини рівномірно розташовані на прямій, паралельній довільній осі  $x$ , а повна модельна довжина ліній  $A$  та  $B$  умовно приймається однаковою. Лінії, що поєднують вершини ліній  $A$  та  $B$ , наприклад  $A_1-B_1$ , умовно назовемо «меридіани» полісітки, яку ми проектуємо. Також у подальшому завжди вважатимемо лінією  $A$  ту, яка має кількість вершин  $N$  більшу, або рівну кількості вершин  $M$  лінії  $B$ .

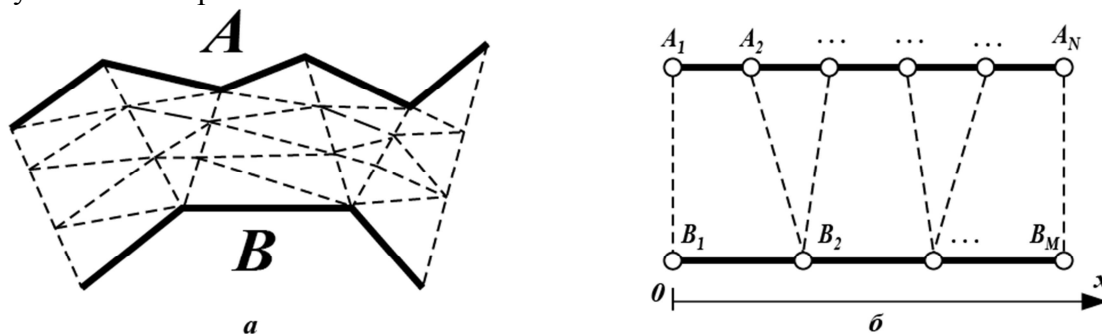


Рис. 4. Схеми до побудови проміжної поєднувальної сітки: а – бажаний результат; б – топологічно тотожна модель для побудови проміжної полісітки; показані меридіани

Першою підзадачею в побудові ППС є визначення пар вершин ліній  $A$  та  $B$ , поєднання яких формує множину меридіанів ППС.

Як видно з рис. 4, б, меридіани мають поєднувати всі вершини  $A_i / i \in [1...N]$  з усіма вершинами  $B_j / j \in [1...M]$ . Очевидно, що при  $N > M$  виконання цієї умови приводить до того, що кожній вершині  $A_i / i \in [1...N]$  завжди інцидентний рівно один меридіан, а на лінії  $B$  будуть існувати вершини, інцидентні кільком меридіанам. Для визначення матриці інцидентності меридіанів введемо модельну метрику розташування вершин вздовж осі  $x$  (рис. 4, б). За цією метрикою будемо вважати повну модельну довжину ліній  $A$  та  $B$  рівній  $(N-1) \cdot (M-1)$ , а  $x$ -координати вершин  $A_i$  становлять арифметичну прогресію  $(M-1) \cdot (i-1) / i \in [1...N]$ . Аналогічно,  $x$ -координати вершин  $B_j$  становлять арифметичну прогресію  $(N-1) \cdot (j-1) / j \in [1...M]$ . Як критерій існування меридіана, який поєднує вершину  $A_i$  з вершиною  $B_j$ , приймемо мінімальність модельної відстані між ними. Іншими словами, при фіксованій вершині  $A_i / i \in [1...N]$  інцидентний їй меридіан буде проводитись до вершини лінії  $B$  з таким індексом  $j \in [1...M]$ , за якого модельна відстань

$$D_{ij} = |x_{A_i} - x_{B_j}| = |(M-1) \cdot (i-1) - (N-1) \cdot (j-1)|$$

буде мінімальною.

Назвемо матрицею меридіанів  $H$  таку матрицю  $\{H_{ij}\} / i \in [1...N], j \in [1...M]$ , в якій  $H_{ij} = 1$ , якщо існує меридіан між  $A_i$  та  $B_j$ , в іншому випадку  $H_{ij} = 0$ . Очевидно, що побудова такої матриці може бути виконана двома вкладеними циклами (в нотатції мови C):

// Початок програмного фрагмента.

Всі елементи  $\{H_{ij}\} = 0$ ;

for(  $i=1$ ;  $i < N$ ;  $i++$  ) {

$D_{min} = (N-1) \cdot (M-1)$ ;

for( $j=1$ ;  $j < M$ ;  $j++$ )

$D_{ij} = \text{abs}((M-1) \cdot (i-1) - (N-1) \cdot (j-1))$ ;

If ( $D_{ij} \leq D_{min}$ )

$D_{min} = D_{ij}$ ;

Else {  $j - -$ ; break; }

$H_{ij} = 1$ ; // меридіан буде проведений між  $A_i$  та  $B_j$

}

// Кінець програмного фрагмента.

Таким чином, підзадачу побудови множини меридіанів проміжної з'єднувальної полісітки можна вважати вирішеною.

Другою підзадачею будемо вважати побудову матриці вершин  $P$  проміжної сітки. Оскільки канонічна поверхнева модель у вигляді полігональних сіток завжди оперує [6], крім іншого, таблицею вершин  $\{V_k\}=\{x_k, y_k, z_k\}$  (де  $k$  – індекс вершини полісітки), то елементами  $P$  можуть бути просто індекси тих вершин, які входять до ППС.

Побудову  $P$  пояснимо на конкретному прикладі ППС з  $M=6$  і  $N=4$  (рис. 5).

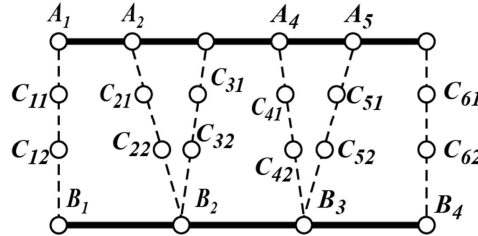


Рис. 5. Схема до побудови матриці вершин  $P$  проміжної полісітки

Спочатку в цій матриці  $P$  визначені тільки перший та останній рядки, в яких перелічуються вершини поліліній  $A$  та  $B$ :

$A_1 A_2 A_3 A_4 A_5 A_6$   
 ? ? ? ? ? ?  
 ... ..  
 ? ? ? ? ? ?  
 **$B_1 B_2 B_3 B_4$**

Треба розділити меридіани на менші відрізки додатковими точками  $C$ , індекси яких ми будемо розміщувати в додаткових проміжних рядках матриці  $P$ .

Кількість додаткових проміжних рядків матриці  $P$  дорівнює кількості додаткових точок на кожному меридіані, і вона може призначатись з практичних міркувань. Одним з можливих критеріїв може бути намагання забезпечити довжину часткового відрізка меридіана (скажімо,  $A_1-C_{11}$  на рис. 5) можливо близькою до середньої довжини ребра полісіток, що поєднуються.

Просторові координати проміжних вершин розраховуються, виходячи з відомих просторових координат крайніх точок відповідного меридіана та визначеної кількості точок ділення  $C$ . Кожна розрахована нова вершина вноситься до таблиці вершин  $\{V_k\}$ , а її індекс – до матриці  $P$ .

Таким чином, друга підзадача може вважатись вирішеною.

Третя підзадача – формування нових ребер та граней ППС. Вона досить тривіальна і вирішується на основі побудованої матриці вершин  $P$ .

Формування нових ребер ППС можна виконати процедурою обходу матриці вершин з послідовним розглядом пар сусідніх вершин як вздовж меридіанів ППС, так і в перпендикулярному напрямку. Кожна з названих пар вершин є основою для внесення запису про нове ребро в таблицю ребер  $E$  (див. (2)). Треба зазначити, що на цьому етапі роботи елементи  $L_i$  та  $R_i$  в записах про нове ребро, які є індексами правої та лівої граней, тимчасово є невизначеними. Причина цього в тому, що в таблиці граней полігональної моделі об'єднаної полісітки поки що нема записів про згадані грані. Після внесення цих нових записів у таблицю граней потрібно повернутись до таблиці ребер та доповнити записи нових ребер відповідними даними.

Формування нових граней ППС також виконується послідовним розглядом четвірок сусідніх компонентів матриці  $P$ . Наприклад (рис. 5), четвірка вершин  $A_1-A_2-C_{21}-C_{11}$  дозволяє сформуванню новий запис про першу нову грань ППС, потім четвірка  $A_2-A_3-C_{31}-C_{21}$  – про сусідню направо і т. ін.

Таким чином, усі кроки побудови проміжної полісітки розглянуто, що відкриває можливість подальшої програмної реалізації цього алгоритму в системах тривимірної комп'ютерної графіки.

#### **Висновки і пропозиції.**

1. Поєднання («зшивання») двох полігональних сіток через побудову третьої, проміжної полісітки може бути виконана за регулярним алгоритмом, основні кроки якого викладені у статті.

2. Алгоритм використовує як складові частини прийоми теорії множин та векторної алгебри, тому для його успішної програмної реалізації потрібно забезпечити доступ до відповідних математичних бібліотек.

3. У випадку програмної реалізації цей алгоритм може бути використаний як складова частина технології автоматизованої побудови поверхневих моделей реальних об'єктів з використанням тривимірного сканування останніх.

#### **Список використаних джерел**

1. *HDS Laser Scanners & SW*. Leica Geosystems Products [Електронний ресурс]. – Режим доступу : [http://www.leica-geosystems.com/en/HDS-Laser-Scanners-SW\\_5570.htm](http://www.leica-geosystems.com/en/HDS-Laser-Scanners-SW_5570.htm).

2. *Bloomenthal J. Convolution Surfaces* / J. Bloomenthal, K. Shoemake // SIGGRAPH'91, Computer Graphics. – 1991. – Vol. 25, № 4. – P. 251–256.

3. *Шикин Е.* Компьютерная графика. Полигональные модели / Е. Шикин, А. Боресков. – М. : Диалог-МИФИ, 2000. – 464 с.

4. *Baumgart, B.* Winged Edge Polyhedron Representation. Technical Report. – Stanford University, Stanford, CA, USA. – 1972.

5. *Бронштейн И. Н.* Справочник по математике для инженеров и учащихся втузов / И. Н. Бронштейн, К. А. Семендяев. – М. : Наука, 1981. – 720 с.

6. *Фоли Д. Ж.* Основы интерактивной машинной графики : в 2 кн. Кн. 2 / Д. Ж. Фоли, Андрис вэн Дэм ; пер. с англ. В. А. Галактионова, Ю. М. Лазутина, О. Н. Родинко. – М. : Мир, 1985. – 368 с.

УДК 004.056.5:004.057.42

**В.В. Соломаха**, старш. викладач

**М.В. Верьовко**, аспірант

**І.В. Соломаха**, канд. екон. наук

Чернігівський національний технологічний університет, м. Чернігів, Україна

#### **ДОСЛІДЖЕННЯ АЛГОРИТМІВ СИМЕТРИЧНИХ КРИПТОСИСТЕМ**

**В.В. Соломаха**, ст. преподаватель

**М.В. Веревко**, аспирант

**И.В. Соломаха**, канд. экон. наук

Черниговский национальный технологический университет, г. Чернигов, Украина

#### **ИССЛЕДОВАНИЕ АЛГОРИТМОВ СИММЕТРИЧНЫХ КРИПТОСИСТЕМ**

**Valerii Solomakha**, senior teacher

**Mariia Verovko**, PhD student

**Iryna Solomakha**, PhD in Economics

Chernihiv National University of Technology, Chernihiv, Ukraine

#### **THE STUDY OF ALGORITHMS FOR SYMMETRIC CRYPTOSYSTEMS**

*Наведено результати практичних порівняльних досліджень сучасних алгоритмів симетричних криптосистем DES і IDEA по швидкодії під час роботи з різним об'ємом інформації і в різних режимах.*

*Ключові слова:* алгоритм шифрування, секретність ключа, криптосистема, швидкодія.

*Приведены результаты практических сравнительных исследований современных алгоритмов симметричных криптосистем DES и IDEA по быстродействию при работе с различным объемом информации и в различных режимах.*

*Ключевые слова:* алгоритм шифрования, секретность ключа, криптосистема, быстродействие.