

Ю. Рашкевич, А. Ковальчук, Д. Пелешко  
Національний університет “Львівська політехніка”,  
кафедра автоматизованих систем управління

## **ЗБІЛЬШЕННЯ РОЗДІЛЬНОЇ ЗДАТНОСТІ ЗОБРАЖЕНЬ ПОЄДНАННЯМ ЗАСОБІВ ТЕОРІЇ НЕЧІТКИХ МНОЖИН ТА ГЕНЕТИЧНИХ АЛГОРИТМІВ**

© Рашкевич Ю., Ковальчук А., Пелешко Д., 2008

**На основі поєднання методів теорії нечітких множин і генетичних алгоритмів запропоновано метод збільшення роздільної здатності з одночасною фільтрацією зображення.**

**There is proposed method of image dpi increase. The proposed method is based on the cross-using of the fuzzy logic theory and genetic algorithms.**

### **Вступ**

Набір підходів до обробки зображень можна розділити на два великі класи: методи обробки в просторовій області (просторові методи) і методи обробки в частотній області (частотні методи) [1]. Термін “просторова область” належить до площини зображення, а просторові методи ґрунтуються на прямому маніпулюванні значеннями пікселів. Методи обробки в частотній області ґрунтуються на модифікації сигналу, сформованого шляхом застосування до зображення перетворення Фур’є.

У частотній області найпоширенішими та ефективними є: методи реконструкції з усуненням зв’язків; використання техніки рекурсивних найменших квадратів; техніка повністю рекурсивних найменших квадратів. До переваг методів обробки зображень у частотній області належить насамперед простота (пояснюється в термінах базової теорії Фур’є); обчислювальна складність (дозволяють паралельні обчислення);

У просторовій області найпопулярнішими і широко вживаними є: інтерполяція зразків; алгебраїчна фільтрація зі зворотним зв’язком; ймовірнісні методи [1]. Перевагами методів обробки в просторовій області є: моделі руху (використання лінійної моделі спостереження); моделі деградації; механізми екстраполяції.

У випадку одиничного зображення застосовуються в основному різноманітні методи інтерполяції, наприклад, білінійну, кубічними В-сплайнами тощо.

Найпростішим методом інтерполяції є повторення тону чи кольору сусідніх пікселів [2]. Цей метод використовують, коли необхідно зберегти декоративну структуру зображення.

Інші алгоритми використовуються для створення проміжкових тонів чи кольорів, які створюють деяку шкалу. Наприклад, якщо вихідні кольори чорний і білий, то методом інтерполювання забезпечується серія сірих тонів від темного до світлого.

### **Постановка задачі**

Метою даної статті є розроблення методу збільшення роздільності, який би після трансформації максимально зберігав інформативні ознаки для подальшої прикладної обробки, наприклад, методами DataMining.

Для вирішення поставленого завдання пропонується використовувати ідеї і методи теорії нечітких множин, які дають можливість зберігати кольорові тренди в зображеннях, генетичні алгоритми (операція кросинговеру) та елементи лінійної алгебри, що в сукупності дає можливість здійснити фільтрацію зображення з мінімальними втратами інформаційних ознак. Із застосуванням

кросинговеру не замінюють одне згенероване рішення на інше, а отримують нове рішення через обмін між ними інформацією.

### 1. Базові припущення

Для вирішення задачі збільшення роздільності на основі теорії нечітких множин адаптовано алгоритм прогнозування [4], який на основі двох векторів кольорів дає можливість знайти новий вектор.

Нехай задано рисунок  $P$  з ширини  $l$  і висоти  $h$ . Його можна розглядати як матрицю пікселів

$$\langle dtp_{ij} \rangle_{1 \leq i \leq n, 1 \leq j \leq m}, \quad (1)$$

де  $dtp_{ij}$  – піксел з координатами  $i$  та  $j$ ,  $n$  і  $m$  – кількість точок по ширині  $l$  та висоті. В загальному випадку  $n$  і  $m$  є залежними від  $l$  та  $h$ , а тому коректнішим є запис

$$n = n(l) \text{ і } m = m(h). \quad (2)$$

Матриці (1) відповідає матриця кольорів

$$C = \begin{pmatrix} c_{1,1} & \dots & c_{1,m} \\ \dots & \dots & \dots \\ c_{n,1} & \dots & c_{n,m} \end{pmatrix}, \quad (3)$$

де  $c_{ij}$  – значення інтенсивності у напівтонових зображень піксела  $dtp_{ij}$ . Тобто існує відповідність

$$P = P_{l,h} = [dtp_{ij}]_{1 \leq i \leq n(l), 1 \leq j \leq m(h)} \rightarrow C. \quad (4)$$

У випадку кольорових зображень  $c_{ij}$  треба розглядати як вектор основних характеристик кольорової палітри. Наприклад, якщо задано зображення у 24-бітному форматі палітри RGB, то

$$c_{i,j} = \{c_{i,j}^R, c_{i,j}^G, c_{i,j}^B\},$$

де  $c_{i,j}^R, c_{i,j}^G, c_{i,j}^B$  – значення червоного, зеленого та синього кольорів піксела  $dtp_{ij}$  відповідно. Тоді наведений нижче алгоритм треба застосувати до кожної характеристики окремо.

Якщо, наприклад, по ширині ввести в розгляд вектор

$$\mathcal{C}_j = \{c_{i,j} \mid i = \overline{1, n}\}, \quad (5)$$

то (3) можна записати у вигляді

$$C = [\mathcal{C}_j]_{j=\overline{1, m}}. \quad (6)$$

### 2. Збільшення роздільної здатності

Основою пропонованого алгоритму збільшення роздільності зображення є використання методів теорії нечітких множин.

Беручи до уваги (3) та (4), завдання збільшення роздільності полягає в тому, щоб сформувати нове зображення  $P^\Psi$  таке, що

$$P^\Psi = \begin{pmatrix} c_{1,1} & \dots & c_{1,m^\Psi} \\ \dots & \dots & \dots \\ c_{n^\Psi,1} & \dots & c_{n^\Psi,m^\Psi} \end{pmatrix}, \quad (7)$$

де  $(n^\Psi, m^\Psi)$  – розмірність матриці,  $n^\Psi > n$ ,  $m^\Psi > m$ , з мінімальними втратами якості зображення.

### 2.1. Опис методу збільшення роздільної здатності

Основним завданням цієї частини є опис методу, який давав би змогу збільшувати роздільну здатність зображення. Для розв’язання цієї задачі побудовано алгоритм, складниками якого є:

1. Генетичний алгоритм (операція кросинговеру) [5,6];
2. Об’єкти лінійної алгебри;
3. Метод побудови нечіткої системи з логічною структурою [4].

Вказані три частини послідовно можуть застосовуватися до всіх рядків матриці (3) для збільшення розмірів заданого зображення за висотою, і – до всіх стовпців матриці (3) для збільшення зображення за шириною.

### 2.2. Опис складових частин алгоритму

#### 2.2.1. Операція кросинговеру

Розв’язуючи задачі оптимізації, можна моделювати процеси рекомбінації. У цьому випадку будь-яке рішення можна подати у вигляді вектора, здатного до оновлення внаслідок введення елементів іншого рішення. Кожний рядок чи стовпець матриці  $\mathbf{P}$  має якість функціональне навантаження. Тому бажано створити такий результуючий вектор - рішення, який би був найкращим з можливих. Для цього можна застосувати операцію кросинговеру [5].

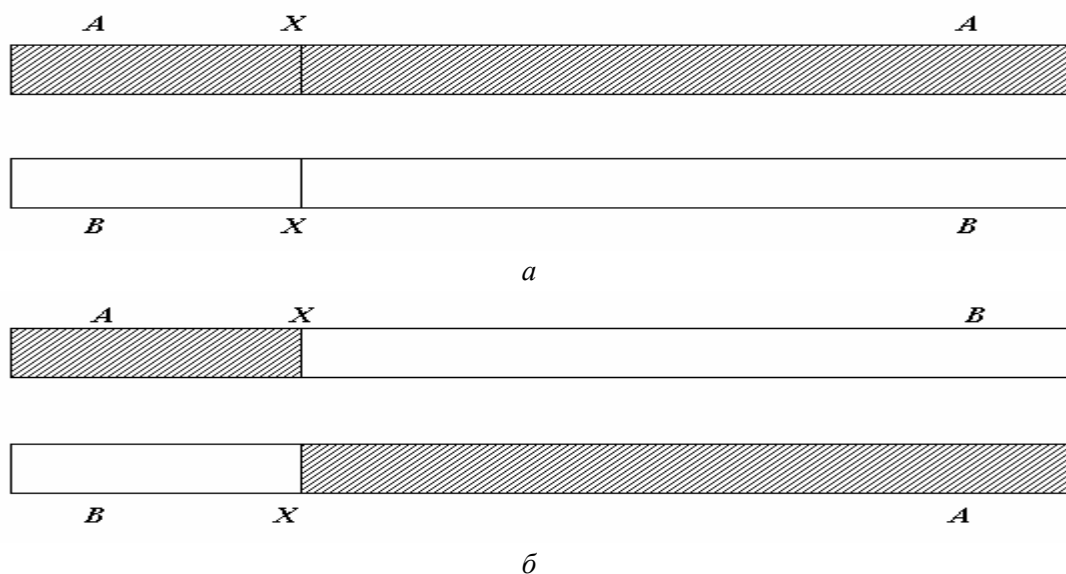


Рис. 1. Схема класичного кросинговеру:  $a$  – до кросинговеру;  $b$  – після кросинговеру

На рис. 1 показана загальна схема класичного кросинговеру, коли у двох векторів  $\mathbf{A}$  та  $\mathbf{B}$  в точці розриву  $X$  відбувається обмін частинами цих векторів.

Математично для пари векторів  $\mathbf{A}$  та  $\mathbf{B}$  операція кросинговеру має вигляд [6]

$$\mathbf{A}c = c \cdot \mathbf{A} + (1 - c) \cdot \mathbf{B}, \quad \mathbf{B}c = (1 - c) \cdot \mathbf{A} + c \cdot \mathbf{B}, \quad (8)$$

де  $0 < c < 1$ .

Для задачі збільшення роздільної здатності в (8) як вектор  $\mathbf{A}$  вибирають рядки матриці (3) при збільшенні роздільної здатності за висотою, а вектором  $\mathbf{B}$  слугує вектор  $\mathbf{A}$ , записаний у зворотному порядку. Роздільна здатність збільшується за шириною аналогічно вибором стовпців матриці (3).

Нехай  $\mathbf{A}' = \{a_j, j = \overline{1, m}\}$ ,  $\mathbf{B}' = \{b_j, j = \overline{1, m}\}$ . Розглянемо вектор  $\mathbf{D}' = \{d_j, j = \overline{1, m}\}$ , елементи якого визначаються так

$$d_j = \sqrt{\min(a_j, b_j)^2 + \max(a_j, b_j)^2}, \quad (9)$$

де  $a_j = c_{i,j}$ ,  $i = 1, n$ .

Або  $\mathbf{A}' = \{a_i, i = \overline{1, n}\}$ ,  $\mathbf{B}' = \{b_i, i = \overline{1, n}\}$ . Тоді  $\mathbf{D}' = \{d_i, i = \overline{1, n}\}$ , де

$$d_i = \sqrt{\min(a_i, b_i)^2 + \max(a_i, b_i)^2}, \quad i = a_{i,j}, \quad j = 1, m. \quad (10)$$

### 2.2.2. Побудова матриці-оператора

У цьому пункті описано побудову квадратної матриці  $\tilde{\mathbf{N}}$ , яка є основним об'єктом і використовується для отримання нового вектора  $\mathbf{D}'$ , який разом з вектором  $\mathbf{A}$  дає розширення матриці (3). Вказану квадратну матрицю  $\tilde{\mathbf{N}}$  отримуємо так:

$$\tilde{\mathbf{N}} = \{\delta_{k,l} \mid k = \overline{1, n}, l = \overline{1, n}\}, \quad \delta_{k,l} = \frac{d_k}{d_l} \quad (11)$$

для розширення за висотою або

$$\tilde{\mathbf{N}} = \{\delta_{k,l} \mid k = \overline{1, m}, l = \overline{1, m}\}, \quad \delta_{k,l} = \frac{d_k}{d_l} \quad (12)$$

для збільшення за шириною зображення.

### 2.2.3. Побудова додаткових векторів

У [4] описано наступний метод побудови додаткових векторів для розширення матриці (3).

Для зображення  $\mathbf{P}^\Psi$  згідно з (3) формується набір векторів за усіма точками  $i \in [1, n]$  та  $j \in [1, m]$

$$\mathbf{L} = \{\mathbf{L}_{i,j}\}_{i \in [1, n]} = \left\{ \left[ \begin{array}{c} c_{i,j} \\ \dots \\ c_{i,j} \end{array} \right]_{i \in [1, n]} \right\}_{j \in [1, m]}. \quad (13)$$

За векторами (13) будується матриця декартових добуток

$$\mathbf{X} = \{\mathbf{X}_j\}_{j \in [1, m]} = \left\{ \left[ \begin{array}{c} \mathbf{X}_{1,1} \\ \dots \\ \mathbf{X}_{n-1,1} \end{array} \right], \dots, \left[ \begin{array}{c} \mathbf{X}_{1,m} \\ \dots \\ \mathbf{X}_{n-1,m} \end{array} \right] \right\}, \quad (14)$$

де при фіксованому  $j$   $\mathbf{X}_{i,j}$  визначається так

$$\mathbf{X}_{i,j} = \mathbf{L}_{i,j} \times \mathbf{L}_{i+1,j}. \quad (15)$$

Декартовий добуток матриць  $\mathbf{X}_{i,j}$  визначається так

$$\forall j \in [1, m]: \quad \mathbf{D}_{i,j} = F(\mathbf{X}_{i,j} \times \mathbf{X}_{i+1,j}), \quad i \in [1, n-1]. \quad (16)$$

У (16)  $F$  створює матрицю  $\mathbf{D}_{i,j}$  такими діями:

□ з кожної пари елементів кожного декартового добутку  $\mathbf{X}_{i,j} \times \mathbf{X}_{i+1,j}$  вибирається мінімальне значення;

□ з усіх отриманих мінімальних значень вибирається максимальне значення, яке і записується у матрицю  $\mathbf{D}_{i,j}$ .

Тоді набір додаткових точок для  $\mathbf{P}^\Psi$  визначають так:

$$\forall j \in [1, m]: \quad \mathbf{D}_{?(i),j} = \mathbf{L}_{i+2,j} \delta \mathbf{D}_{i,j}, \quad i \in [1, n-1]. \quad (17)$$

У результаті для формування  $\{\Psi_n\}$  при фіксованому  $j$  отримують набір з чотирьох векторів

$$(\mathbf{L}_{i,j}, \mathbf{L}_{i+1,j}, \mathbf{L}_{i+2,j}, \mathbf{D}_{\gamma(i),j})_{i \in [1, n-1]}. \quad (18)$$

#### 2.2.4. Схема уточнення методу збільшення роздільної здатності

Адаптований алгоритм на основі теорії нечітких множин дає змогу знайти новий вектор  $(\mathbf{D}_{\gamma(i),j})_{i \in [1, n-1]}$  на основі відомих значень векторів  $(\mathbf{L}_{i,j}, \mathbf{L}_{i+1,j}, \mathbf{L}_{i+2,j})_{i \in [1, n-1]}$ , але разом з тим нічого не відомо про впорядкування цих векторів. Тому наступним кроком є впорядкування послідовності (18), тобто розташування  $\mathbf{D}_{\gamma(i),j}$  серед  $\mathbf{L}_{i,j}, \mathbf{L}_{i+1,j}, \mathbf{L}_{i+2,j}$ . Для впорядкування (18) можна використати такі варіанти:

1.  $\mathbf{D}_{\gamma(i),j}, \mathbf{L}_{i,j}, \mathbf{L}_{i+1,j}, \mathbf{L}_{i+2,j}$ ;
2.  $\mathbf{L}_{i,j}, \mathbf{D}_{\gamma(i),j}, \mathbf{L}_{i+1,j}, \mathbf{L}_{i+2,j}$ ;
3.  $\mathbf{L}_{i,j}, \mathbf{L}_{i+1,j}, \mathbf{D}_{\gamma(i),j}, \mathbf{L}_{i+2,j}$ ;
4.  $\mathbf{L}_{i,j}, \mathbf{L}_{i+1,j}, \mathbf{L}_{i+2,j}, \mathbf{D}_{\gamma(i),j}$ .

Для вибору одного із варіантів (19) можна використовувати таку методику впорядкування послідовності векторів (18).

При фіксованому  $j$  для кожного вектора з (18) знаходимо мінімум норм

$$P_i = \min_{x \in [i, i+2]} \|\mathbf{L}_{x,j} - \mathbf{L}_{\gamma(i),j}\| = \min_{x \in [i, i+2]} \left( \sqrt{\sum_{n=1}^l (c_{x,0}^n - c_{\gamma(i),j}^n)^2} \right), \quad (20)$$

де  $c_{\gamma(i),j}^n$  – значення кольору вектора  $\mathbf{D}_{\gamma(i),j}$ .

За знайденим  $P_i$  визначаємо такий вектор  $x$  серед  $\mathbf{L}_{i,j}, \mathbf{L}_{i+1,j}, \mathbf{L}_{i+2,j}$ , в околі якого буде розміщено  $\mathbf{D}_{\gamma(i),j}$ . Знайдене значення  $x$  позначимо через  $x'$ .

Для уточнення розміщення  $\mathbf{D}_{\gamma(i),j}$  (тобто перед чи за вектором  $\mathbf{L}_{x,j}$ ) знайдемо довжини векторів  $\mathbf{D}_{\gamma(i),j}$  і  $\mathbf{L}_{x,j}$

$$r_{x',j} = \|\mathbf{L}_{x',j}\| = \sqrt{\sum_{n=1}^l (c_{x',j}^n)^2}, \quad i \leq x' \leq i+2; \quad r_{\gamma(i),j} = \|\mathbf{L}_{\gamma(i),j}\| = \sqrt{\sum_{n=1}^l (c_{\gamma(i),j}^n)^2}, \quad (21)$$

Надалі достатньо порівняти  $r_{x',j}$  і  $r_{\gamma(i),j}$ . Якщо  $r_{\gamma(i),j} < r_{x',j}$ , то  $\mathbf{L}_{\gamma(i),j}$  розміщується перед  $\mathbf{L}_{x,j}$ , у протилежному випадку – за  $\mathbf{L}_{x,j}$ . Так формуються усі  $n^\Psi$  векторів  $\{\mathbf{L}_{i,j}\}_{i \in [1, n^\Psi]}^{j-\text{фікс}}$  при фіксованому  $j$ . Аналогічно відносно вихідного набору (3) формується набір при фіксованому  $i$ .

Для уникнення об'ємних обчислень за формулами (14) – (21) вибирається як матриця  $\mathbf{L}_{i,j}$  з (16) матриця  $\tilde{\mathbf{N}}$ , отримана з (11) або (12). Результуюча матриця  $\mathbf{P}^\Psi$  будується послідовним внесенням вектора  $\mathbf{D}$ , координати якого обчислюються по (9), або (10), і вектора  $\mathbf{D}' = \mathbf{D}_{\gamma(i),j}$ , отриманого з (17).

### 3. Приклади практичного застосування методу

Результати збільшення роздільної здатності вхідного зображення I (за висотою – II, за шириною – III, за шириною – IV) наведено в таблиці.

Порівняння зображень

I. Початкове зображення	II. Розширення удвічі за висотою	III. Розширення удвічі за шириною	IV. Розширення в чотири рази за шириною

Інший приклад збільшення розмірів зображення показано на рис. 2 – 3.



*Рис. 2. Вхідне зображення*



*Рис.3. Зображення, збільшене за шириною*



Рис. 4. Зображення, збільшене за висотою

Для ілюстрації роботи алгоритму рішення опишемо на трьох точках. На їх основі побудуємо четверту. Аналогічно, пересуваючись по точках, в ітераційному циклі можна побудувати решту наборів  $D_{x(i),j}$ . Згідно з (3) маємо такі вхідні три вектори

$$L_{1,1} = \begin{pmatrix} 25 \\ 35 \\ 30 \\ 45 \\ 42 \end{pmatrix}; \quad L_{2,1} = \begin{pmatrix} 15 \\ 25 \\ 32 \\ 10 \\ 30 \end{pmatrix}; \quad L_{3,1} = \begin{pmatrix} 22 \\ 34 \\ 23 \\ 32 \\ 25 \end{pmatrix}. \quad (22)$$

За векторами (22) можна обчислити декартові добутки (3.5)  $X_{1,1}$  і  $X_{2,1}$

$$X_{1,1} = \begin{bmatrix} (25,15) & (25,25) & (25,32) & (25,10) & (25,30) \\ (35,15) & (35,25) & (35,32) & (35,10) & (35,30) \\ (30,15) & (30,25) & (30,32) & (30,10) & (30,30) \\ (45,15) & (45,25) & (45,32) & (45,10) & (45,30) \\ (42,15) & (42,25) & (42,32) & (42,10) & (42,30) \end{bmatrix}. \quad (23)$$

$$X_{2,1} = \begin{bmatrix} (15,22) & (15,34) & (15,23) & (15,32) & (15,25) \\ (25,22) & (25,34) & (25,23) & (25,32) & (25,25) \\ (32,22) & (32,34) & (32,23) & (32,32) & (32,25) \\ (10,22) & (10,34) & (10,23) & (10,32) & (10,25) \\ (30,22) & (30,34) & (30,23) & (30,32) & (30,25) \end{bmatrix}. \quad (24)$$

Використовуючи (16), (23) і (24), можна обчислити матрицю  $D_{1,1}$

$$D_{1,1} = \begin{bmatrix} 15 & 25 & 25 & 15 & 25 \\ 25 & 25 & 32 & 25 & 30 \\ 22 & 32 & 30 & 32 & 30 \\ 15 & 25 & 32 & 10 & 30 \\ 22 & 30 & 32 & 30 & 30 \end{bmatrix}. \quad (25)$$

На вхід матриці  $D_{1,1}$  подаємо дію, описану нечіткою множиною  $D_{3,1}$  (22) і отримуємо прогнозоване значення:

$$L_{\gamma(1),1} = [22, 34, 23, 32, 25] \delta \begin{bmatrix} 15 & 25 & 25 & 15 & 25 \\ 25 & 25 & 32 & 25 & 30 \\ 22 & 32 & 30 & 32 & 30 \\ 15 & 25 & 32 & 10 & 30 \\ 22 & 30 & 32 & 30 & 30 \end{bmatrix} = [15, 25, 32, 10, 30]. \quad (26)$$

Аналогічно знаходимо решту  $L_{\gamma(i),j}$ ,  $i > 1$ . Наостанок впорядковуємо набір  $L_{1,1}, L_{2,1}, L_{3,1}, D_{\gamma(1),1}$ .

Результати впорядкування усіх можливих варіантів (3.10) наведено на рис.4

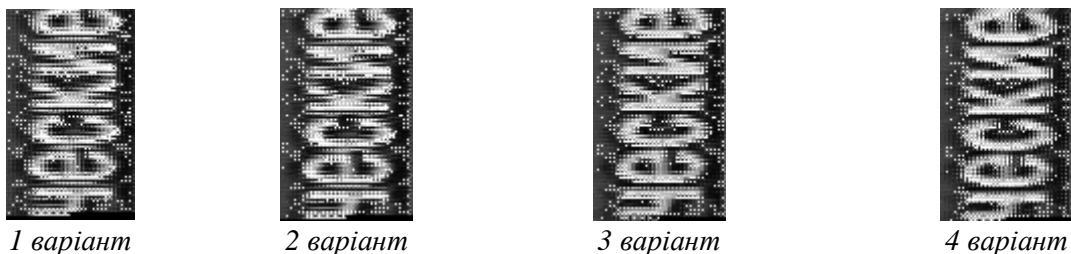


Рис. 4. Зображення розширене удвічі з використанням варіанта впорядкування

Результати роботи алгоритму вказують на те, що за критерієм PSNR (відношення сигналу до шуму) – варіант впорядкування наборів з використанням норми є найкращий для візуального сприйняття.



## Висновки

Метод підвищення роздільної здатності, побудований на основі теорії нечітких множин, дає можливість отримати якісніші (на основі експертних оцінок) результати порівняно з іншими. Використання операції кросингверу в процесі збільшення роздільної здатності дає можливість здійснити заміну зображення з мінімальними втратами його інформативності.

Метод підвищення роздільної здатності зберігає колірні градієнти, що робить можливим використання інформаційних трендів у подальших прикладних задачах видобутку інформації.

1. Павлидис Т. Алгоритмы машинной графики и обработки изображений. – М.: Радио и связь, 1986. – 399 с. 2. Уоссермен Ф. Нейрокомпьютерная техника. – М.: Мир, 1992. – 259 с. 3. Воробель Р.А., Журавель И.М. Повышение контраста изображений с помощью модифицированного метода кусочного растяжения // Отбор и обработка информации. – 2000. – № 14 (90). – С. 116–121. 4. Нечеткие множества и теория возможностей. Последние достижения: Пер. с англ. / Под ред. Р.Р. Ягера. – М.: Радио и связь, 1986. – 407 с. 5. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. – М.: Физматлит, 2006. – 320 с. 6. Баодин Лю. Теория и практика неопределенного программирования. – М.: Бином, 2005. – 416 с.

УДК 004.852; 004.942

**П. Кравець**

Національний університет “Львівська політехніка”,  
кафедра інформаційних систем та мереж

## ІГРОВЕ ПРИЙНЯТТЯ РІШЕНЬ В МУЛЬТИАГЕНТНИХ СИСТЕМАХ

© Кравець П., 2008

**Досліджується проблема прийняття рішень у мультіагентних системах за допомогою методів стохастичних ігор. Сформульовано ігрову задачу в умовах невизначеності. Розглянуто ігрові моделі без обміну та з обміном інформацією, без відмов та з відмовами гравців. Розроблено рекурентні методи для розв’язування ігрової задачі. Визначено умови збіжності ігрових методів до колективних станів рівноваги.**

**The problem of decision making in multiagent systems with aid of stochastic game methods is investigated. The game problem formulation in uncertainty conditions is executed. The game models without (and with) exchange information are considered. The game models without (and with) failure of players are considered. The recurrence methods for the game task solving are developed. The convergence conditions of game methods to collective equilibrium states is defined.**

### **Проблема прийняття рішень в мультіагентних системах**

Проблема ефективного прийняття рішень є однією з найважливіших в системах керування та штучного інтелекту [1]. Процес вироблення та прийняття рішень відповідно до сформульованої мети визначається взаємодією інтелектуального агента та проблемного середовища [2]. Під прийняттям рішення у дискретній системі розуміємо вибір агентом одного із альтернативних варіантів дій, який ґрунтується на спостереженні відповідних реакцій середовища та апріорних знаннях агента про проблемну область.

У розподілених системах процес прийняття рішень здійснюється множиною агентів [3]. Це значно ускладнює вибір ефективних рішень, оскільки кількість їх комбінованих варіантів зростає експоненційно до зростання кількості агентів. У зв’язку з цим побудова моделей та методів, що