

УДК 519.86

**Макаренко О.І.***кандидат фізико-математичних наук,  
професор кафедри вищої математики  
Київського національного економічного університету імені Вадима Гетьмана***Кіщенко О.В.***аспірант кафедри вищої математики  
Київського національного економічного університету імені Вадима Гетьмана*

## ВИКОРИСТАННЯ ГЕНЕТИЧНОГО АЛГОРИТМУ ПРИ ПРОЕКТУВАННІ ЕЛЕКТРОННОГО ДОКУМЕНТООБІГУ НА ОСНОВІ МЕРЕЖІ ПЕТРІ

У статті розглядається метод мінімізації ризиків, які виникають на етапах життєвого циклу електронного документу з використанням імітаційної моделі на основі мереж Петрі, параметри якої визначаються генетичним алгоритмом. Формулюється цільова функція для застосування генетичного алгоритму при розв'язуванні даного типу задач.

**Ключові слова:** мережа Петрі, ризик, електронний документообіг, генетичний алгоритм, популяція.

### **Макаренко А.И., Кищенко О.В. ИСПОЛЬЗОВАНИЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА ПРИ ПРОЕКТИРОВАНИИ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА НА ОСНОВЕ СЕТИ ПЕТРИ**

В статье рассматривается метод минимизации рисков, возникающих на этапах жизненного цикла электронного документа с использованием имитационной модели на основе сетей Петри, параметры которой определяются генетическим алгоритмом. Формулируется целевая функция для применения генетического алгоритма при решении данного типа задач.

**Ключевые слова:** сеть Петри, риск, электронный документооборот, генетический алгоритм, популяция.

### **Makarenko O.I., Kishchenko O.V. DESIGNING AN ELECTRONIC DOCFLOW BASED ON PETRI NETS WITH USING THE GENETIC ALGORITHM**

The article describes a method to minimize the risks arising on the stages of the life cycle of an electronic document using a simulation model based on Petri nets, the parameters of which are determined by the genetic algorithm. The author formulates the objective function for the application of genetic algorithm for solving this type of problem.

**Keywords:** Petri net risk, electronic document management, genetic algorithm, population.

**Постановка проблеми.** Сьогодні актуальним і перспективним з наукової та практичної точки зору є вирішення проблеми проектування системи з необхідною поведінкою на основі заданих компонентів. Тобто на основі певних правил програма повинна перетворити вхідні дані у вихідні. Причиною цього є мінливе бізнес-середовище, яке вносить свій відбиток і на існуючий документообіг компанії – для прикладу, зміни в законодавстві змушують переглядати бізнес-процеси та змінювати параметри документообігу. Таким чином, необхідним стає використання синтезу електронних систем: задані вхідні сигнали система повинна перетворити у вихідні за певними правилами; треба автоматично синтезувати в електронну систему документообігу з потрібною поведінкою та з певним набором компонентів. Більшість існуючих програмних комплексів систем електронного документообігу не здатні в автоматичному режимі вирішувати ці проблеми.

**Аналіз останніх досліджень і публікацій.** Генетичний алгоритм є стратегічним підходом до вирішення проблеми, зокрема мінімізації ризиків, який можливо адаптувати фактично до будь-якої предметної області. У той же час для аналізу проблемних зон особлива увага приділяється методам моделювання, одним з яких є мережі Петрі. Актуальність і перспектива використання генетичних алгоритмів на основі мереж Петрі підтверджується сучасними науковими дослідженнями. Так, науковці часто для аналізу процесу використовують графові моделі. В них задача зводиться до розфарбовування графа. Різні етапи процесу часто вирішуються за допомогою алгоритмів математичного програмування і різних евристик, що моделюють роботу об'єкта моделювання. В останнє десятиліття все більшої популярності для вирішення різних задач набули генетичні алгорит-

ми. При пошуку рішення вони реалізують спрощені моделі основних факторів еволюції, таких як спадкова мінливість, природний відбір, дрейф генів.

**Постановка завдання.** На основі викладеного можна сформулювати дослідження, яке полягає в адаптації генетичного алгоритму до вирішення проблеми – мінімізації ризиків, які виникають при проходженні електронним документом його життєвого циклу. Моделі життєвого циклу документообігу доцільно представляти у вигляді мереж Петрі.

**Виклад основного матеріалу дослідження.** Основним результатом роботи є формалізація і алгоритмізація процесу проектування системи електронного документообігу шляхом формування рішення з фіксованою структурою з заданих компонентів. Хоча ця задача і належить до найпростішого класу задач проектування електронних систем, однак її рішення істотно ускладнюється величезною кількістю можливих варіантів реалізації, у зв'язку з присутністю ризиків на кожному етапі виконання процесу, з яких необхідно вибрати один, що задовольняє заданій умові. Для подолання цієї проблеми пропонується використовувати генетичні алгоритми на основі побудованих мереж Петрі. Метою даної роботи є вирішення більш складної задачі формалізації і алгоритмізації процесу проектування систем електронного документообігу шляхом формування рішення, спираючись на мінливу структуру процесу документообігу через прояви ризиків.

**Постановка задачі.** Розглянемо наступний клас задач проектування систем електронного документообігу. Дано:

$$S = (In, Out, (S_k)_{k=1}^K, (f_\alpha)_{\alpha=1}^A, (f_\beta)_{\beta=1}^B),$$

де  $S$  – система, яку необхідно спроектувати;

$In$  – множина вхідних даних системи  $S$ ;

$Out$  – множина вихідних даних системи  $S$ ;

$S_k$  –  $k$ -та підсистема системи  $S$ ;  
 $f_a$  – функція, що визначає, які вихідні дані відпо-  
 відають вхідним даними:  $f_a : In \rightarrow Out$ ;  
 $f_\beta$  – бінарне відношення на множині  
 $(S_k)_{k=1}^K : f_\beta \subset (S_k)_{k=1}^K \times (S_k)_{k=1}^K$ .



Рис. 1. Контекстна модель системи

Потрібно для заданої функції  $f_a$  підібрати бінарне відношення  $F_\beta$  таке, щоб безліч підсистем  $(S_k)_{k=1}^K$  забезпечувало обробку системою  $S$  вхідних даних у вихідні дані відповідно до функції  $f_a$ .

**Формалізація мережами Петрі.** Входи системи  $S$  будуть моделюватися множиною позицій  $P^{in} = (p_m^{in})_{m=1}^M$ , де  $M$  – кількість входів системи, а її виходи – множиною позицій  $P^{out} = (p_n^{out})_{n=1}^N$ , де  $N$  – кількість виходів системи (рис. 1).

Входи і виходи підсистем системи  $S$  також будуть моделюватися множиною позицій. Входи підсистеми  $S_k$  – це множина  $P_k^{in} = (p_{k,m}^{in})_{m=1}^{M(k)}$ , де  $M(k)$  – кількість входів підсистеми  $S_k$ , а її виходи – множина  $P_k^{out} = (p_{k,n}^{out})_{n=1}^{N(k)}$ , де  $N(k)$  – кількість виходів підсистеми  $S_k$  (рис. 2).

Підсистеми  $(S_k)_{k=1}^K$  можуть бути пов'язані як між собою, так і з входами і виходами системи  $S$ . Ці зв'язки будуть моделюватися безліччю переходів  $T = (t_q)_{q=1}^Q$ , де  $Q$  – кількість переходів. Кожний перехід  $t_q$  характеризується своїми вхідними і вихідними позиціями. Входами переходу  $t_q$  можуть бути як будь-які входи системи  $S$ , так і будь-які виходи підсистем. Виходами переходу можуть бути як будь-які виходи системи, так і будь-які входи підсистем  $(S_k)_{k=1}^K$ .

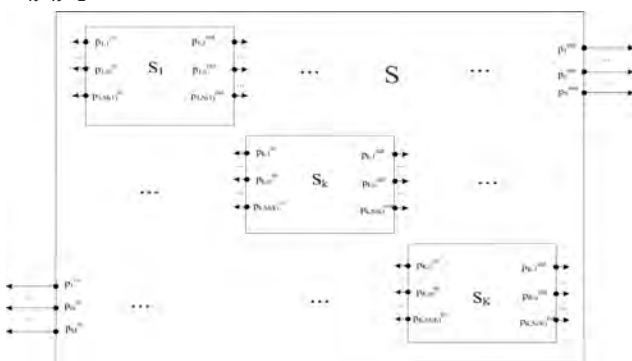


Рис. 2. Множина підсистем системи

Позначимо входи переходу  $t_q$  через  $In_q$ , а виходи – через  $Out_q$ . Тоді, виходячи з вищесказаного,

$$In_q \in P_{in} \cup \left( \bigcup_{k=1}^K P_k^{out} \right) \text{ і}$$

$$Out_q \in P_{out} \cup \left( \bigcup_{k=1}^K P_k^{in} \right).$$

Відповідно, для кожного входу системи  $S$  повинен існувати перехід  $t_q$ , з'єднаний з цим входом. Крім

цього, для кожного виходу кожної підсистеми  $S_k$  повинен існувати перехід  $t_q$ , з'єднаний з цим виходом. Формально ці вимоги можна записати у вигляді рівності:

$$\bigcup_{q=1}^Q In_q = P_{in} \cup \left( \bigcup_{k=1}^K P_k^{out} \right).$$

Аналогічно, для кожного виходу системи  $S$  повинен існувати перехід  $t_q$ , з'єднаний з цим виходом, і для кожного входу кожної підсистеми  $S_k$  повинен існувати перехід  $t_q$ , з'єднаний з цим входом. Формально ці вимоги можна записати у вигляді рівності.

$$\bigcup_{q=1}^Q Out_q = P_{out} \cup \left( \bigcup_{k=1}^K P_k^{in} \right).$$

Різні переходи можуть мати деякі загальні вхідні і вихідні позиції, але повністю співпадаючих переходів бути не повинно. Ця вимога формально записується наступним чином:

$$In_{q_1} = In_{q_2} \Rightarrow Out_{q_1} \neq Out_{q_2} \text{ (при } q_1 \neq q_2 \text{);}$$

$$Out_{q_1} = Out_{q_2} \Rightarrow In_{q_1} \neq In_{q_2} \text{ (при } q_1 \neq q_2 \text{).}$$

Множина підсистем  $(S_k)_{k=1}^K$  разом з підмножиною множини переходів  $T = (t_q)_{q=1}^Q$  повністю визначають поточну структуру системи  $S$ . Необов'язково всі підсистеми повинні бути задіяні в поточній структурі. При цьому перебудова структури системи  $S$  визначається зміною підмножини задіяних в даний момент переходів. Приклад структури системи показаний на рис. 3.

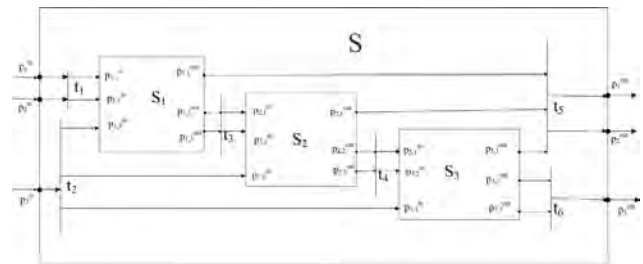


Рис. 3. Приклад структури системи

Перебудова функцій (зміна алгоритму функціонування системи  $S$ ) відбувається в підсистемах  $(S_k)_{k=1}^K$ . У загальному випадку кожній підсистемі  $S_k$  можна співставити множину мереж Петрі, які будуть моделями програм обробки даних. Позначимо цю множину через  $PN_k$  і визначимо її наступним чином

$$PN_k = (PN_{k,r})_{r=1}^{R(k)},$$

де  $PN_{k,r}$  –  $r$ -й алгоритм обробки даних підсистемою  $S_k$ , змодельований у вигляді мережі Петрі.

Формально необхідно для кожної мережі Петрі  $PN_{k,r}$  описати її позиції, переходи і дуги. Але в загальному випадку мережа  $PN_{k,r}$  моделює деяку дію з перетворення вхідних даних у вихідні дані. Тому далі для простоти в якості моделей алгоритмів будуть розглядатися не мережі Петрі  $PN_{k,r}$ , а переходи («вироджені» мережі Петрі). Таким чином, множина  $PN_k$  буде виглядати так:

$$PN_k = (t_{k,r})_{r=1}^{R(k)}.$$

Входами  $In_{k,r}$  кожного переходу  $t_{k,r}$  можуть бути будь-які входи підсистеми  $S_k$ :  $In_{k,r} \in P_k^{in}$ , а виходами  $Out_{k,r}$  кожного переходу  $t_{k,r}$  можуть бути виходи підсистеми  $S_k$ :  $Out_{k,r} \in P_k^{out}$ .

Виходячи з усього вищесказаного, модель системи  $S$  у вигляді мережі Петрі може виглядати, наприклад, так, як показано на рис. 4. У цій моделі

напівжирним виділено шлях, по якому вхідні дані, що надходять на верхні два входи системи S (на малюнку у вигляді двох міток), проходять через її підсистеми і з'являються на нижньому виході системи S. Пунктиром виділені незадіяні переходи і дуги.

**Опис генотипу.** Генетичний алгоритм необхідно адаптувати до конкретної предметної області шляхом визначення параметрів і операторів генетичного алгоритму. При використанні генетичного алгоритму альтернативні рішення представляються у вигляді рядка символів, що має фіксовану довжину і називається генотипом. Тому особливу увагу слід приділити формальному опису генотипу. В нашому випадку генотип G повинен визначати:

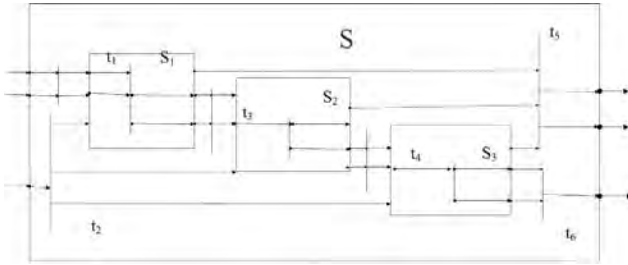


Рис. 4. Приклад моделі системи S

- які переходи з множини  $T = (t_q)_{q=1}^Q$  будуть представлені в моделі системи S;
- які переходи з множини  $PN_k = (t_{k,r})_{r=1}^{R(k)}$  будуть представлені в моделі кожної підсистеми  $S_k$ .

Таким чином, генотип G матиме наступний вигляд:

$$G = (g_1, \dots, g_q, \dots, g_Q, h_1, \dots, h_k, \dots, h_K),$$

де  $g_q \in (0,1)$  і  $h_k \in (0,1, \dots, R(k))$ .

Якщо  $g_q = 0$ , то перехід  $t_q$  відсутній в моделі системи S, інакше ( $g_q = 1$ ) перехід  $t_q$  присутній в моделі системи S. Якщо  $h_k = 0$ , то в моделі підсистеми  $S_k$  відсутній алгоритм обробки даних (не завантажена програма). Якщо  $h_k = r$  ( $r \in \{1, \dots, R(k)\}$ ), то в моделі підсистеми  $S_k$  представлений перехід  $t_{k,r}$ .

Таким чином, число C всіх гіпотетично можливих моделей системи S визначається наступною формулою:

$$C = 2^Q \prod_{k=1}^K (R(k) + 1).$$

Тому навіть при невеликій кількості зв'язків, підсистем і алгоритмів обробки даних кількість моделей системи S дуже велика. Наприклад, при  $Q=10, K=10$   $R(1)=R(2)=\dots=R(10)=10$  і отримуємо  $C=2^{10} \times 11^{10} = 22^10 = 26559922791424$ .

**Цільова функція.** Серед величезної кількості усіх гіпотетично можливих моделей системи S необхідно знайти саме ту, яка вирішує поставлену задачу: з наявної множини підсистем  $(S_k)_{k=1}^K$  та системи переходів  $T = (t_q)_{q=1}^Q$  побудувати таку систему S, яка на вхідний вектор реагувала би потрібним вихідним вектором. Тому чим ближче вихідний вектор до необхідного, тим побудована система краще.

Для оцінки цієї близькості введемо позначення для необхідного вектора:  $V = (v_1, \dots, v_p, \dots, v_N)$ , де  $v_n$  – планова ймовірність виникнення того чи іншого ризику в позиції  $p_n^{in}$  системи S. Зрозуміло, що  $v_n \in (0..1)$ . Вектор, який вийшов в результаті роботи системи S, позначимо через  $W = (w_1, \dots, w_p, \dots, w_N)$ , де  $w_n$  – ймовірність виникнення ризику, які реально знаходяться в позиції  $p_n^{out}$  системи S. Аналогічно,  $w_n \in (0..1)$ .

Різницю (відстань) між необхідним і реальним вектором (плановою і фактичною ймовірністю) можна оцінювати, наприклад, за формулою [1]:

$$p(V, W) = \sum_{n=1}^N |v_n - w_n|.$$

Вибір цільової функції впливає на ефективність роботи генетичного алгоритму. Тому на практиці можна використовувати і формулу, яка визначає евклідову відстань

$$p(V, W) = \sqrt{\sum_{n=1}^N (v_n - w_n)^2}.$$

Змінюючи параметр p, можна буде впливати на ефективність роботи генетичного алгоритму. Чим менше p, тим ближче система S до необхідної конфігурації. При  $p=0$  система повністю відповідає заданій вимозі. Але, тим не менш, немає гарантії того, що з заданої множини підсистем  $(S_k)_{k=1}^K$ , їх моделей  $PN_k = (t_{k,r})_{r=1}^{R(k)}$  і зв'язків  $T = (t_q)_{q=1}^Q$  можна побудувати необхідну систему S.

**Оператори генетичного алгоритму.** Після вибору цільової функції можна описати оператори генетичного алгоритму. Оператор відбору повинен відбирати для схрещування ті генотипи, які найближчі до необхідної конфігурації. Організувати цей процес можна, наприклад, наступним чином: упорядкувати всі наявні генотипи за якістю (від кращого до гіршого) і схрещувати сусідні пари. Можна також дозволити найкращим генотипам брати участь в декількох схрещуваннях.

Оператор схрещування можна описати таким чином. Розглянемо два генотипу

$$G_1 = (g_1^1, \dots, g_q^1, g_{q+1}^1, \dots, g_Q^1, h_1^1, \dots, h_k^1, h_{k+1}^1, \dots, h_K^1)$$

$$G_2 = (g_1^2, \dots, g_q^2, g_{q+1}^2, \dots, g_Q^2, h_1^2, \dots, h_k^2, h_{k+1}^2, \dots, h_K^2)$$

Випадковим чином вибираємо два числа: q з діапазону (множини)  $\{1, 2, \dots, Q\}$  і k з діапазону  $\{1, 2, \dots, K\}$ . А потім міняємо відповідні ділянки генотипів. Таким чином, з батьків  $G_1$  і  $G_2$  виходять нащадки  $G_3$  і  $G_4$ , які успадковують властивості батьків:

$$G_3 = (g_1^1, \dots, g_q^1, g_{q+1}^2, \dots, g_Q^2, h_1^1, \dots, h_k^1, h_{k+1}^2, \dots, h_K^2)$$

$$G_4 = (g_1^2, \dots, g_q^2, g_{q+1}^1, \dots, g_Q^1, h_1^2, \dots, h_k^2, h_{k+1}^1, \dots, h_K^1)$$

Можна обмежитися і вибором якогось одного числа (q або k). Скільки чисел буде вибрано (одне або два) і якщо одне, то яке саме, можна також визначити випадковим образом. Можна також вибрати для кожного діапазону  $\{1, 2, \dots, Q\}$  і  $\{1, 2, \dots, K\}$  по дві точки і обмінюватися «серединками».

Оператор мутації можна описати таким чином. Розглянемо один генотип

$$G = (g_1, \dots, g_q, \dots, g_Q, h_1, \dots, h_k, \dots, h_K).$$

Випадково вибираємо два числа: q з діапазону  $\{1, 2, \dots, Q\}$  і k з діапазону  $\{1, 2, \dots, K\}$ . А потім міняємо значення  $g_q$  та  $h_k$  в такий спосіб, що якщо  $g_q = 0$ , міняємо його на одиницю:  $g_q = 1$ . І навпаки: якщо  $g_q = 1$ , то стане  $g_q = 0$ . Якщо  $h_k = r$ , то міняємо його на будь-яке інше з діапазону  $\{0, 1, \dots, r-1, r+1, \dots, R(k)\}$ . Зрозуміло, при  $r=0$  це буде діапазон  $\{1, \dots, R(k)\}$ , а при  $r=R(k)$  це буде діапазон  $\{0, 1, \dots, R(k)-1\}$ . Оператор редукції повинен видаляти слабкі генотипи. Для цього після схрещування необхідно визначити характеристики всіх нащадків. Потім нащадків і батьків треба об'єднати в одну множину генотипів і впорядкувати їх за характеристиками від кращого до гіршого. Хто виявився в гіршій половині, того відкидаємо.

**Рішення задачі.** Перед початком роботи операторів генетичного алгоритму необхідно задати початкову популяцію генотипів. Якщо є які-небудь гіпотези про структуру і функціонування системи S, то генотипи

може описати сам проектувальник. Інакше можна випадково згенерувати кілька генотипів. Розмір популяції також визначається проектувальником.

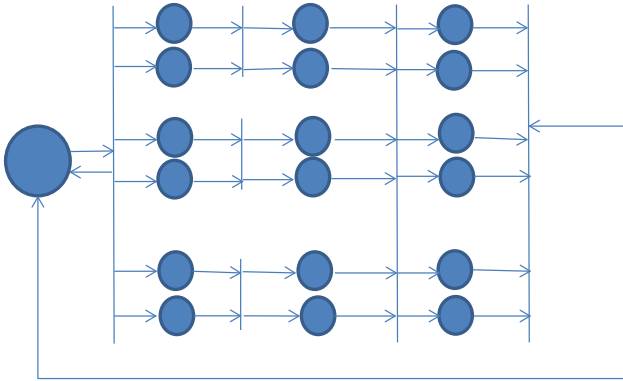


Рис. 5. Опис генетичного алгоритму мережею Петрі

Нижче приведений покроковий процес генетичного алгоритму [2]:

- 1) початкова популяція генотипів  $G_1, G_2, \dots, G_{2D}$  розміщується в позиціях мережі Петрі;
  - 2) оператор здійснює відбір генотипів для схрещування;
  - 3) оператори здійснюють схрещування генотипів, які потім піддаються мутації;
  - 4) цикл роботи завершує оператор, що видаляє слабкі генотипи;
  - 5) потім процес повторюється.
- Зупинка цього циклічного процесу може визначитися різними способами:

- визначенням певної кількості циклів;
- визначенням нижньої межі якості всієї популяції;
- визначенням нижньої межі якості частини популяції (наприклад, половини).

**Висновки з проведеного дослідження.** З наведеного вище можна зробити наступні висновки:

1. Сформульована задача мінімізації ризиків, що виникають на етапах руху електронного документа в системі, в термінах генетичного алгоритму має практичне застосування в системах управління документообігом.
2. Отримана мережева модель системи, в структурі якої окремо виділені підмножини, що описують різні категорії документів.
3. Запропонований підхід до вирішення завдання мінімізації ризиків, що виникають на етапах руху електронного документа в системі, може бути рекомендований для систем різного рівня складності.
4. Запропонована модель електронного документообігу на основі мережі Петрі може використовуватися для систем з великим потоком документів.
5. Даний метод можливо адаптувати для вирішення інших задач електронного документообігу, змінивши цільову функцію генетичного алгоритму.

#### БІБЛІОГРАФІЧНИЙ СПИСОК:

1. Колмогоров А.Н. Элементы теории функционального анализа : учеб. для вузов / А.Н. Колмогоров, С.В. Фомин. – 6-е изд., испр. – М. : Наука, 1989. – 624 с.
2. Ельчанинов Д.Б. Применение генетических алгоритмов при проектировании компьютерной техники / Д.Б. Ельчанинов, Д.А. Петросов, Сами Механа // Вестник Херсонского государственного университета. – 2003. – № 2(18). – С. 45-49.

УДК 004.738.5:658.8

Панасенко Л.М.

старший викладач кафедри туризму та готельно-ресторанної справи Черкаського державного технологічного університету

## ПОЗИЦІОНУВАННЯ ТЕРИТОРІЇ НА ОСНОВІ ВИКОРИСТАННЯ МОЖЛИВОСТЕЙ МЕРЕЖІ ІНТЕРНЕТ

У статті обґрунтовується підхід до використання інтернет-брендингу. Розглянуто основні напрями онлайн-просування бренду та його особливості. Представлено рекомендації з підвищення ефективності використання можливостей мережі Інтернет щодо брендингу території.

**Ключові слова:** бренд території, інтернет-брендинг, інструменти брендингу, геомаркетинг, територіальний маркетинг, брендовані сніппети, вірусний маркетинг.

### Панасенко Л.М. ПОЗИЦИОНИРОВАНИЕ ТЕРРИТОРИИ НА ОСНОВЕ ИСПОЛЬЗОВАНИЯ ВОЗМОЖНОСТЕЙ СЕТИ ИНТЕРНЕТ

В статье обосновывается подход к использованию интернет-брендинга. Рассмотрены основные направления онлайн-продвижения бренда и его особенности. Представлены рекомендации по повышению эффективности использования возможностей сети Интернет для брендинга территории.

**Ключевые слова:** бренд территории, интернет-брендинг, инструменты брендинга, геомаркетинг, территориальный маркетинг, брендированные снippets, вирусный маркетинг.

### Panasenko L.M. THE TERRITORY POSITIONING ON THE INTERNET OPPORTUNITIES BASIS

The article explains the campaign to use Internet branding. The main directions of online branding and its features are considered. The recommendations for Internet opportunities improving for territory branding are presented.

**Keywords:** brand territory, online branding, branding tools, geomarketing, territorial marketing, branded snippets, virus marketing.

**Постановка проблеми.** Інтернет-брендинг є відносно новим напрямом брендингу, який застосовується при просуванні території. Він передбачає викорис-

тання можливостей інтернет-мережі для створення і просування нових чи вже існуючих брендів так званих геопродуктів, тобто країн, регіонів, міст, ло-