

## ФОРМУВАННЯ ПРОТОТИПІВ ДЛЯ КОМП'ЮТЕРНОЇ СИСТЕМИ РОЗПІЗНАВАННЯ ЕМОЦІЙ ЛЮДИНИ ПО МІМІЦІ ЗА ДОПОМОГОЮ ЕВОЛЮЦІЙНОГО АЛГОРИТМУ

Цап В.Й.

**Цап В. Й. Формування прототипів для комп'ютерної системи розпізнавання емоцій людини по міміці за допомогою еволюційного алгоритму**

*Стаття присвячена проблемі побудови комп'ютерних автоматичних систем розпізнавання емоційних станів людини по її міміці і застосування їх на практиці в різних галузях. Описуються підходи та алгоритми, що були застосовані для подолання труднощів при створенні такої комп'ютерної програми. Зокрема для формування прототипів експресії пропонується застосовувати еволюційний алгоритм.*

**Ключові слова:** автоматичні системи розпізнавання, емоційні стани, міміка, прототипи експресії, еволюційний алгоритм.

**Цап В. Й. Формирование прототипов для компьютерной системы распознавания эмоций человека по мимике с помощью эволюционного алгоритма**

*Статья посвящена проблеме создания компьютерных автоматических систем распознавания эмоциональных состояний человека по его мимике и использования их на практике в разных областях. Описываются подходы и алгоритмы, которые применялись для преодоления трудностей при создании такой компьютерной программы. В частности для формирования прототипов экспрессии предлагается использовать эволюционный алгоритм.*

**Ключевые слова:** автоматические системы распознавания, эмоциональные состояния, мимика, прототипы экспрессии, эволюционный алгоритм.

**Tsap V. I. Forming of prototype models for computer-based system intended for recognition of human emotions by facial expressions and using the evolutionary algorithm**

*The article deals with the problem of development of computer-based automated systems aimed at the recognition of human emotional states by the individual's facial expressions. Possibilities for practical use of such systems in different spheres of life are analyzed. Approaches and algorithms are described, which were used for overcoming difficulties in the process of design and development of the program. To form the prototype models of expressions, the author suggests that the evolutionary algorithm should be used.*

**Key words:** automated pattern recognition systems, emotional states, facial expressions, prototype models of expressions, evolutionary algorithm.

Останнім часом спостерігається зростаючий інтерес до досліджень ідентифікації міміки людини. Ці проблеми привертають передусім увагу психологів, які вивчають емоційну сферу, та фахівців з автоматичного розпізнавання облич або емоційного стану людини. У першу чергу досягнення в цих галузях необхідне для вдосконалення взаємодії індивіда з комп'ютером. Комп'ютери майбутнього будуть взаємодіяти з нами майже як люди. Ключовий аспект такої взаємодії – можливість розпізнавання наших облич і сприйняття їх виразів. Це найпривабливіші і економічно значущі напрями досліджень у сфері інформаційних технологій. Найефективніший комунікативний канал до споживача – персоналізований спосіб спілкування з комп'ютером. При цьому моделюються новітні взаємини між учасниками даного комунікативного процесу. Відбувається обмін інформацією між комп'ютером та людиною, мінімізуючи протистояння між ними, оскільки відсутність порозуміння між партнерами може призвести до труднощів у спілкуванні, так званих комунікативних бар'єрів. Спілкування впливає на емоційний стан людини. Крім цього, можливості розпізнавання агресивних намірів допомагають підвищити нашу безпеку.

Зокрема, розкриття потенційних можливостей емоційної сфери в навчально-виховному процесі є однією з найактуальніших проблем сучасної педагогіки. Теоретичні та експериментальні дослідження в цій галузі відкривають широку перспективу вивчення та застосування результатів об'єктивно-психологічного аналізу щодо емоційних проявів та оптимізації спілкування комп'ютерної навчальної системи з учнем.

В руслі цих досліджень нами була створена комп'ютерна програма МІМІКА\_k3 – система розпізнавання міміки людини [4]. Ця програма призначена виконувати експертні функції у визначенні емоційних станів пред'явленого зображення обличчя. Зображення вводиться у комп'ютер як цифровий фотопортрет, під яким мається на увазі фотографія особи однієї людини у фас, бажано без додаткових атрибутів, що приховують якісь частини обличчя (окуляри, вуса, борода і т. ін.). Бажано також, щоб фотопортрети були приблизно одного масштабу, отримані в умовах рівномірного освітлення й з мінімальними варіаціями ракурсу.

Складність завдання, а також характер опису використовуваних знань при розробці цієї програми [5] навели на думку, що це має бути гібридна система. На першому етапі один модуль системи має розрізняти експресію обличчя, а на другому, використовуючи його вихідні дані, інший модуль визначає емоційний стан людини. Другий модуль являє собою експертну систему, що керується знаннями. Для

розробки правил, згідно яким можна визначити будь які емоції, була вибрана система, створена П. Екманом та У. Фрізенем FACS (Facial Action Coding System) [5]. Розроблена система MIMIKA\_k3 базується на правилах, згідно яких можна визначити такі емоційні стани як радість, подив, гнів, відраза, презирство, страх, сум, гнів + відраза, гнів + презирство, гнів + страх, гнів + сум, відраза + страх, презирство + сум, страх + сум, страх + подив. Кожному стану відповідає окремий фрейм експертної системи, які містять унікальні набори правил, записаних як коди FACS. А кожному правилу відповідає коефіцієнт "упевненості", що визначає, наскільки точно це правило відповідає даній емоції. Усього використовувалося більш 3200 правил. Ці правила і є акумульованими знаннями фахівців-психологів.

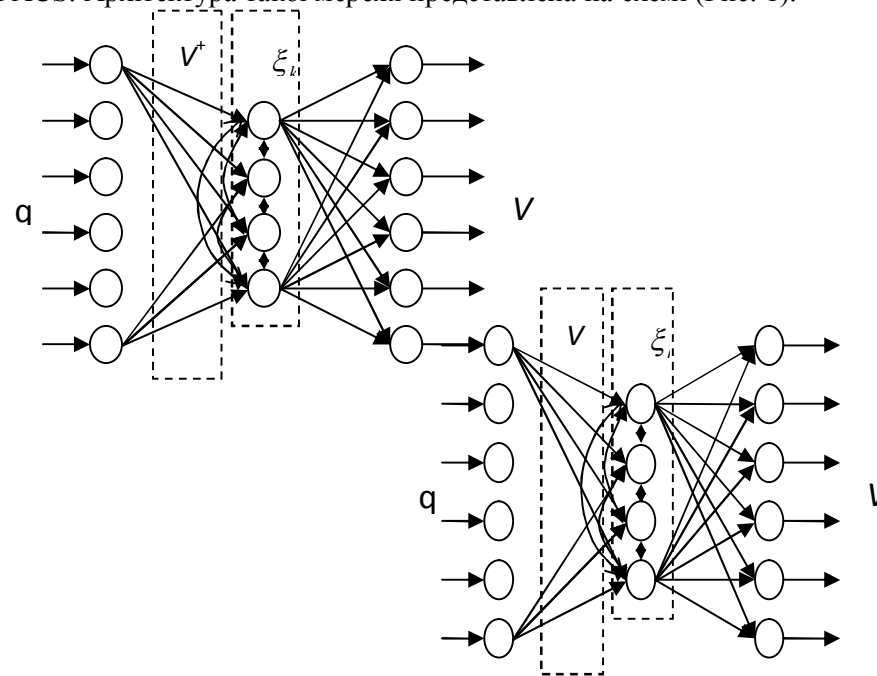
Робота програми визначення емоцій полягає в наступному. Зображення з виразом обличчя, що піддається експертизі, вводиться в комп'ютер. На виході система видає свій висновок і, при необхідності, пояснює причину даного вибору. Призначення першого модулю – перетворювати зображення фотопортрета на код FACS. Інакше кажучи, він реалізує своєрідний діалог між користувачем й експертною системою для введення завдання. Це можна здійснювати трьома способами:

1. введенням безпосередньо коду FACS;
2. відтворенням експресії особи за допомогою вмонтованого фоторобота;
3. введенням безпосередньо фотографії людини (найбільш зручний і перспективний спосіб введення).

Це досить складне завдання належить до класу автоматичного розпізнавання візуальних зображень, зокрема обличчя людини. Цій проблемі присвячено досить багато досліджень, що показали вражаючі досягнення. Вже досить багато компаній розробників програмних продуктів, здебільшого зарубіжних, пропонують свої відповідні комерційні рішення. На сайті [http://habrahabr.ru/blogs/image\\_processing/133686/](http://habrahabr.ru/blogs/image_processing/133686/) зроблено огляд цих пропозицій. Наша система відрізняється від інших тим, що має гібридну побудову і на першому етапі визначає не окрему емоцію, а описує вираз обличчя як код FACS. Основна складність полягає у тому, що необхідно розпізнати експресію обличчя за зображеннями різних людей незалежно від зміни ракурсу, масштабу й умов освітлення в момент зйомки, а також від різних змін, пов'язаних із віком, зачіскою тощо. Для розв'язання цього завдання використано синергетичний підхід [4, 3, 6, 7]. Синергетичний підхід обґрунтовує зв'язок між розпізнаванням образів і формуванням структур, якими керують так звані параметри

порядку. Інакше кажучи, розпізнавання зображення є не що інше, як формування образу прототипу. Формування для запам'ятовування образу прототипу призводить до появи нового параметра порядку, що згодом несе відповідальність за впізнавання цього образу, тобто за утворення структури, яка це реалізує.

Зображення в комп'ютері описуються масивом так званих пікселів, що зберігають коди кольорів відповідної точки екрану монітора. Кожному пікселю приписується певне значення сірості  $v_j$ . Таким чином, зображення являє собою вектор  $V = (v_1, v_2, \dots, v_N)$  із сукупністю значень сірого. Множина зображень становить множину векторів  $V_k$ , при цьому кожному відповідає індекс  $k = 1, \dots, M$ . Це ті образи, які ми збираємося розпізнавати. Реалізується процес розпізнавання на структурі, яка зветься синергетичний комп'ютер, що можна розглядати як новий різновид нейронної мережі [4]. Розпізнавання зводиться до конкуренції мод параметрів порядку  $\xi_k$ , у результаті якої всі моди, крім деякої  $k$ -ї, прагнуть до нульового значення, а пропонований образ  $q(0)$  при  $t \rightarrow \infty$  прагне до  $V_{k0}$ .  $V_{k0}$  - є відгуком системи на пред'явлене зображення і відповідає одному з кодів FACS. Архітектура такої мережі представлена на схемі (Рис. 1).



### Рис. 1. Архітектура нейронної мережі для розпізнавання зображення.

Щоб синергетичний комп'ютер мав змогу вправно виконувати свої функції, його спочатку треба навчити-налаштувати це робити. Саме алгоритм навчання формує вигляд кінцевої структури системи розпізнавання. Для навчання формується так звана навчальна вибірка тестових зображень, які відповідають образам, які ми хочемо розпізнати. Кожному образу можуть відповідати зображення, зовсім не схожі один на одного. Чим більш різноманітних зображень будуть складати навчальну вибірку, тим, можна сподіватись, буде точнішою наша система. Але процес навчання в цьому випадку буде протікати важче. Процес навчання полягає в тому, щоб так налагодити вхідні коефіцієнти нейронної мережі та змінити її структуру, аби на виході була як можна менша кількість помилкових відповідей. Для цього навчальна вибірка тестових зображень вважається фіксованою, а образи-прототипи й сполучені з ними вектори, розпізнаванню яких потрібно навчитися, як ті, що потрібно «підігнати». Якщо кожному прототипу відповідає тільки одне зображення, то жодних труднощів не виникає. Гірше, коли необхідно сформуванню узагальнюючий прототип, тобто коли йому відповідає численність зразків. Це досягається за допомогою усереднення зображень, що відносяться до кожного прототипу. Таким чином, навчання набуває форми процесу оптимізації. Як видно з рисунка (Рис. 1), архітектура нейронної мережі може складатися кількох рівнів. Це залежить від обсягу навчальної вибірки. Прототипи відшукуються за кілька циклів навчання. У кожному циклі після формування прототипів перевіряється правильність розпізнавання. З неправильно розпізнаних зразків утворюється нова навчальна вибірка, й для неї формуються нові параметри порядку, які додаються до вже отриманих. Таким чином формується перший рівень нейронної системи. Такі цикли повторюються доти, поки розширення першого рівня відповідають чинникам оптимізації. Після цього, якщо в навчальній вибірці ще є зображення, які система неправильно розпізнає, формується другий рівень. Налаштування другого рівня відбувається так само. Процес навчання триває доти, поки не залишиться жодного нерозпізнаного зображення. Таким чином, система може здійснювати розпізнавання зі 100-відсотковою точністю на навчальній вибірці, якою б великою вона не була. У такий спосіб підвищується ймовірність розпізнавання й абсолютно незнайомих зображень.

Даний алгоритм навчання суттєво відрізняється від раніше запропонованих та описаних в літературі, присвяченій цій темі, оскільки відбувається налаштування не тільки вхідних коефіцієнтів

фіксованої структури нейронної мережі, але й відбувається зміна її структури. В нашій практиці у самому складному випадку навчальна вибірка становила більш ніж 17500 зображень, що відповідали понад 1200 образам прототипів, які потрібно було розпізнати, і ми задоволилися двома рівнями. Але в складніших випадках рівнів може бути й більше. Щоб полегшити процес навчання (тобто скоротити його тривалість), ми вирішили спочатку кластеризувати навчальну вибірку для тих прототипів, які налічують надто великий обсяг несхожих зображень. Кожний кластер відповідає образу прототипу, що треба розпізнати. На першому рівні створюються структури, здатні розрізняти зображення в кожному кластері, а також структура, що навчається розпізнавати, якому саме кластеру відповідає тестоване зображення. Ця структура є своєрідним «перемикачем», що спрямовує тестований образ до потрібного кластера, де й видається остаточне рішення.

Для кластеризації була створена спеціальна програма, яка працює з такими об'єктами, як зображення. Кластерний аналіз вирішує завдання побудови *класифікації*, тобто поділу вихідної множини об'єктів на групи (класи, кластери). Такий підхід особливо важливий для дослідника, який немає вихідних припущень ні про склад класів, ні про їхню відмінність один від одного, і т.ін. Приступаючи до кластерного аналізу, дослідник має лише інформацію про характеристики (ознаки) для об'єктів, що дозволяє йому судити про їх подібність (відмінність), або покладатися тільки на дані про їхню попарну подібність. Варіанти кластерного аналізу – це численність простих обчислювальних процедур, використовуваних для класифікації об'єктів. А *класифікація* об'єктів – це групування їх у класи таким чином, щоб вони в кожному класі були більш схожими один на одного, ніж на об'єкти з інших класів. Точніше, кластерний аналіз – це процедура упорядкування об'єктів у порівняно однорідні класи на основі попарного їх порівняння за попередньо визначеним і вимірним критерієм.

Існує багато варіантів кластерного аналізу, але найбільше широко використовуються методи, об'єднані загальною назвою ієрархічний кластерний аналіз. Але якщо треба кластеризувати досить велику кількість об'єктів, як це було в нашій практиці, підвищуються вимоги до обчислювальної техніки. Зростає потреба в пам'яті та часі, оскільки пошук рішення ведеться послідовно. Це змусило звернутися до інших методів, а саме до еволюційних алгоритмів [1]. За допомогою еволюційних алгоритмів пошук рішення ведеться паралельно, тобто є множина альтернативних рішень, які конкурують між собою, і

оцінюються за певним правилом - критерієм. У результаті вибирається найоптимальніше рішення, яке вважається результатом роботи.

Еволюційні алгоритми зацікавили нас тим, що вони придатні для пошуку в складному просторі рішень великої розмірності, відсутності обмежень на вид цільової функції, присутності ясної схеми і базових принципів, інтегрованості їх з іншими неklasичними парадигмами штучного інтелекту. Зокрема, такими, як штучні нейронні мережі та нечітка логіка. Еволюційні алгоритми мають не лише переваги, але й недоліки. Одним з головних недоліків є те, що вони не гарантують оптимальності отриманого рішення. Але може бути знайдено одне або кілька квазіоптимальних рішень, які отримані за прийнятний час.

Робота еволюційних алгоритмів ґрунтується на принципах, які запозичені з генетики. Основна ідея полягає у створенні популяції істот (масивів об'єктів), кожна з яких представляється у вигляді хромосоми. Хромосома містить в собі набір властивостей і являє можливе рішення будь-якої розв'язуваної оптимізаційної задачі. При пошуку кращих рішень необхідно тільки значення цільової функції, або, як її називають, функції пристосованості. Значення функції пристосованості істоти (хромосоми) показує, наскільки добре підходить вона для виконання завдання з врахуванням можливих обмежень. До популяції застосовуються такі основні оператори, як схрещування, мутації, інверсії та ін. У процесі еволюції діє відомий принцип генетики, коли виживає найбільш пристосована істота. Популяція постійно оновлюється за допомогою генерації нових істот і знищення старих або менш пристосованих індивідів.

Розробка еволюційного алгоритму складалася з етапів:

1. конструкція хромосом в залежності від характеру обраної задачі;
2. вибір функції пристосованості для всіх істот в популяції;
3. генерація початкової популяції істот;
4. вибір оператора репродукції (селекції);
5. розробка оператора схрещування;
6. розробка оператора мутації;
7. розробка критеріїв зупинки еволюційного алгоритму.

Проблема кластеризації полягає в тому, щоб сукупність зображень, які стосуються одного прототипу, розбити найкращим чином на порівняно невелику кількість однорідних, в певному сенсі, груп або класів (так звані «кластери»). Сукупність зображень представлена у вигляді матриці, де кожен рядок – вектор  $V = (v_1, v_2, \dots, v_N)$  відповідає одному зображенню (об'єкту). Таке представлення

дозволяє інтерпретувати кожне зображення в якості точки у відповідному ознаковому просторі. І завдання кластеризації полягає в розбитті множини заданих точок на області, в яких вони близькі один одному. В якості міри близькості, чи подібності, зображень застосовувався коефіцієнт кореляції. Щоб можна було оцінювати різні рішення, треба вибрати критерій якості розбиття, який задає спосіб зіставлення з кожним можливим розбиттям заданої сукупності зображень на класи деякого числа  $q$ , що оцінює ступінь оптимальності даного розбиття. Тоді завдання пошуку кращого розбиття зводиться до вирішення оптимізаційної задачі. Критерій якості розбиття обирався таким чином, щоб на кожному кластері він забезпечував максимізацію сумарної міри подібності між зображеннями цього кластеру, і навпаки, мінімізував міру подібності між різними кластерами.

Як вище вже зазначалося, в еволюційних алгоритмах першим кроком є кодування рішень у вигляді хромосом, яке залежить від характеру розв'язуваної задачі. Тому, перш ніж використовувати еволюційний алгоритм, спочатку необхідно конструювати розв'язування задачі у вигляді хромосоми. Виходячи з характеру розв'язуваної задачі, хромосому в популяції представляємо у вигляді рядка  $Y = (y_1, y_2, \dots, y_n)$  довжиною  $n$ , де:

$y_i$ - визначає номер кластера і приймає значення з множини  $\{1, 2, 3, \dots, q\}$ ;

$n$ - кількість зображень, що відносяться до даного прототипу;

$q$ - кількість кластерів окремого розбиття;

$i$ - номер зображення (позиція  $y_i$  відповідає номеру зображення).

Кожне зображення може бути віднесено тільки до одного з кластерів і не може бути пустих кластерів, тобто таких, що не містять жодного зображення.

Визначимо ще таке поняття як центр кластера. Центр кластера задається так само як зображення - вектором  $C = (c_1, c_2, \dots, c_N)$ . До кластеру з центром  $C_q$  відносяться ті зображення, що мають з ним більшу подібність, ніж з іншими центрами. Під „радіусом” кластера  $r_q$  будемо розуміти найменше значення міри подібності між зображенням, що належить цьому кластеру і його центром. Міра подібності між кластерами  $r_{ij}$  визначається як міра подібності між їх центрами.

Оскільки ідея кластеризації полягає у віднесенні схожих зображень в один кластер, то звідси випливає, що при правильній кластеризації радіуси кластерів повинні бути як можна більш великими. Отже, при кластеризації необхідно максимізувати радіуси кожного кластера. Звідси середнє значення радіуса конкретного розбиття буде

максимальним. Середнє значення радіуса  $R_{cp}$  обчислюється за формулою:

$$R_{cp} = \frac{1}{q} \sum_{i=1}^q r_i \rightarrow \max \quad (1)$$

де  $q$  – кількість кластерів;  $r_i$  – радіус  $i$ -того кластеру.

З іншого боку, міри подібності між кластерами повинні бути мінімальними. Значить, необхідно мінімізувати середнє значення міжкластерної міри подібності  $R_{mk}$ , яке обчислюється за формулою:

$$R_{mk} = \frac{2}{q(q-1)} \sum_{i=1}^{q-1} \sum_{j=i+1}^q r_{ij} \rightarrow \min \quad (2)$$

де  $q$  – кількість кластерів;  $r_{ij}$  – міра подібності між кластерами  $i$  та  $j$ .

Таким чином, завдання кластеризації зводиться до одночасної максимізації середнього значення радіусу і мінімізації середньої міжкластерної міри подібності. Функція пристосованості для всіх істот в популяції має такий вигляд:

$$F = R_{cp} - R_{mk} \rightarrow \max \quad (3)$$

Еволюційний алгоритм починається з генерації початкової популяції, яка містить кінцевий набір можливих рішень. Як вже було сказано, початкова популяція складається з хромосом. Хромосоми формувалися наступним чином. Для кожної хромосоми задавалася своя кількість кластерів і координати їх центрів в просторі ознак. Центри обиралися випадковим чином. Далі обчислювались попарні міри подібності кожного зображення і центрів кластерів. Шляхом порівняння мір подібностей кожне зображення відносилось до того чи іншого кластеру. Значення номеру кластера присвоювалося відповідній координаті вектору хромосоми  $Y = (y_1, y_2, \dots, y_n)$ . Оскільки бажане рішення повинно містити як можна менше число кластерів, але ніяких припущень з цього приводу не було, до популяції залучалися хромосоми з різною кількістю заданих кластерів. Розмір популяції залежав від довжини списку, що налічував зображення того чи іншого прототипу.

В процесі роботи алгоритму популяція постійно збільшується шляхом залучення нових об'єктів та знову зменшується до первинного розміру завдяки селекції. Основними операторами генерації нових об'єктів є оператори схрещування та мутації. В еволюційних алгоритмах оператори схрещування майже не використовуються. Ми також відмовилися від їх застосування. В якості одного з операторів мутації застосовувався відомий метод кластеризації  $k$ -середніх.

Основна ідея цього методу полягає в тому, що для хромосоми на кожній ітерації переобчислюються координати центрів кластерів для кожного з них, отриманого на попередньому кроці. Потім формується нова хромосома таким чином, що масив зображень розбивається на кластери знову відповідно до того, який з нових центрів виявився кращим з обраного критерію. Переобчислення відбувається наступним чином. Обчислюються центри важкості кластерів. Кожен центр важкості – це вектор, елементами якого є середні значення ознак, обчислені за всіма зображеннями кластера. Потім центр кластера зміщується в його центр важкості. Слід зазначити, що після такої мутації батьківська хромосома втрачає здатність до мутації, тому що нові хромосоми будуть дублювати самі себе.

Хромосоми оцінюються з використання функції пристосованості (3), в результаті кожній з них присвоюється певне значення (fitness-значення). Від fitness-значення залежить ймовірність виживання хромосоми. Якщо fitness-значення має від'ємний знак, то застосовуються ще інші оператори мутації. Це оператори об'єднання кластерів та поділу кластерів. Для першого відшукуються кластери з найбільшою міжкластерною мірою подібності. Зображення, що містяться в цих кластерах об'єднуються в один новий кластер, центр якого обчислюється як центр важкості. Таким чином, кількість кластерів в новій хромосомі зменшується на один. Для оператора поділу кластерів відшукується зображення, що відповідає найменшому радіусу кластера. Вектор цього зображення призначається як центр нового кластеру, що додається до існуючих. Таким чином, кількість кластерів в новій хромосомі збільшується на один. Отримані в результаті мутації нові хромосоми додаються до популяції, і вона перевіряється з метою вилучення істот з однаковими хромосомами.

До збільшеної популяції застосовується селекція. Вона відбувається за змагальним принципом [2], згідно з якого кожна хромосома попарно порівнюється з деякими її противниками. Кількість противників задавалася як певний параметр, який дорівнював приблизно 60% від числа хромосом з однаковою кількістю кластерів. Противники вибираються випадково за програмою генерації випадкових чисел з числа хромосом, які налічують таку саме кількість кластерів. Переможець визначається шляхом попарного порівняння fitness-значень. Істота перемагає в змаганні, якщо її fitness-значення щонайменше не гірше, ніж у її противника. Кількість перемог для кожної істоти підсумовується окремо. Після цього список, що містить всі об'єкти збільшеної популяції, сортується за спаданням числа перемог (а не за fitness-значенням). При однаковому числі перемог

перевагу отримує істота з кращим fitness-значенням. Далі цей список скорочується знизу до розміру початкової популяції. Тобто кращі істоти утворюють нову популяцію.

В якості критеріїв зупинки еволюційного алгоритму було обрано декілька. Алгоритм припиняє роботу у випадках: якщо число ітерацій дорівнювало 50; якщо всі хромосоми популяції втрачали здатність до мутації; якщо найкраща хромосома не змінювалася 5 ітерацій поспіль і якщо fitness-значення найкращої хромосоми не змінюється 5 ітерацій поспіль, а також користувач може зупинити дію алгоритму примусово, надавши відповідну команду.

Таким чином, еволюційний алгоритм має наступний вигляд:

1. Формування хромосом початкової популяції.
2. Початок циклу.
3. Застосування, якщо це можливо, операторів мутації.
4. Формування збільшеної популяції.
5. Застосування змагання хромосом та їх селекція.
6. Формування нової популяції з кращих хромосом.
7. Перевірка критеріїв зупинки алгоритму (якщо

виконуються, то кінець циклу; якщо ні, то новий цикл).

Дана стаття присвячена проблемі побудови комп'ютерних автоматичних систем розпізнавання емоційних станів людини по її міміці і застосування їх на практиці в різних галузях. Для цього застосовуються різні підходи і алгоритми. Загальна архітектура системи має дуже складний вигляд і має гібридну структуру. До неї входить модуль автоматичного розпізнавання експресії обличчя і модуль експертної оцінки емоційного стану. Для формування прототипів експресії пропонується застосовувати еволюційний алгоритм, для чого була створена спеціальна програма. Цю програму можна використовувати для кластеризації об'єктів і в інших задачах.

## ЛІТЕРАТУРА

1. Алгулиев Р. М. Быстрый генетический алгоритм решения задачи кластеризации текстовых документов / Р. М. Алгулиев, Р. М. Алыгулиев // Искусственный интеллект. – 2005. – № 3. – С. 698–707.
2. Курейчик В. М. Эволюционные вычисления: генетическое и эволюционное программирование / В. М. Курейчик, С. И. Родзин // Теория и системы управления / РАН. – М. – 2002. – № 1. – С. 127–138.
3. Хакен Г. Принципы работы головного мозга. Синергетический подход к активности мозга, поведению и когнитивной деятельности / Г. Хакен. – М. : ПЕР СЭ, 2001. – 351 с.

4. Цап В. Й. Комп'ютерна система психологічної експертизи міміки емоцій / В. Й. Цап // Збірник наукових праць Інституту психології ім. Г. С. Костюка АПН України / за ред. С. Д. Максименка. – К., 2005. – Т. VII, вип. 5. – С. 358–369.

5. Ekman P., Friesen W. Facial Actions Coding System. – Palo Alto: Consulting psychologists Press, 1978.

6. Vanger P., Hönlinger R., Haken H. Applications of the synergetic computer in decoding complex facial patterns // Proceedings of the First International Conference on Applied Synergetic and Synergetic Engineering. 1994. June 21-23, Erlangen. – P. 111-117.

7. Vanger P., Hönlinger R., Haken H. Die Anwendung der Synergetik bei der Erkennung von Emotionen in Gesichtsausdruck. //Selbstorganisation in Psychologie und Psychiatrie. G. Shiepek und W. Tschacher (eds.). – Braunschweig: Vieweg, 1997. – S. 85-101.