

### Про деякі аспекти навчання еволюційних алгоритмів

**Анотація.** В статті розглянуто основні поняття еволюційних алгоритмів, їх класифікації та методи розв'язування задач. Проаналізовано сучасні програмні засоби, в яких реалізовано інструментарій еволюційних алгоритмів, наведено приклад їх використання. Навчання і використання еволюційних алгоритмів з комп'ютерною підтримкою сприяє розвитку математичних та інформатичних компетентностей майбутніх фахівців.

**Ключові слова:** еволюційний алгоритм, генетичний алгоритм, задача комівояжера.

Важливою особливістю, властивою практично всім науковим і технічним проблемам, є їхній оптимізаційний характер. На сучасному етапі перед людством стоять оптимізаційні задачі, які потребують розв'язування і прийняття виважених та ефективних рішень. Процес прийняття таких рішень може бути доволі різноманітним. Але в будь-якому випадку прийняти виважене ґрунтовне рішення складно без використання математичних моделей і методів. Для того, щоб використати адекватну модель, необхідно насамперед сформулювати проблему, з'ясувати, до якого класу задач вона відноситься і які математичні моделі й методи доречно застосовувати для її розв'язування.

За допомогою еволюційних методів проводять аналіз і адаптацію вже створених популяцій (рішень) до нових умов середовища, тим самим скорочують час роботи алгоритмів, їх машинної реалізації та навчання.

Питаннями оптимізації та еволюційними алгоритмами займалися такі вчені як Растрігін Л. А. [8], Холанд Дж. [13], Голденберг Д. [12], Курейчик В. М. [2, 6], Погорілий С. Д., Білоус Р. В. [2], Самотній В., Дзелендзяк У. [3], Дубровін В. І., Федорченко Є.М. [1], Рутковська Д., Пілінський М., Рутковський Л. [9], Погорілий С. Д. [6], Субботін С. О., Олійник А. О., Олійник О. О. [4] та інші.

Історія еволюційних алгоритмів починається з розробок різних незалежних моделей. Перші публікації «Симбіогенетичні еволюційні процеси, реалізовані штучними методами» («Symbiogenetic evolution processes realised by artificial methods» (1957)) та «Цифрова перевірка еволюційних теорій» («Numerical testing of evolution theories" (1962)), які належать Барічеллі Н. А., були спрямовані насамперед на розуміння механізмів природного феномену спадковості. Відомі початківці штучного моделювання еволюційних процесів і генетичних систем А.Фрейзер, Г.Бремерманн, Г.-Ф.Швефель, І.Рехенберг досліджували еволюційні стратегії. У 1966 році Л.Фогель, А.Оуенз та М.Волш запропонували і провели дослідження еволюції простих автоматів, у яких були передбачені символи у числових послідовностях, заснувавши таким чином еволюційне програмування [10].

Системи моделювання еволюційних процесів можна розподілити на дві категорії:

1. *Системи, в яких використовуються лише еволюційні принципи.* Їх використовують для опису математичних моделей функціональної оптимізації. До таких систем відносять еволюційні алгоритми, такі як еволюційне програмування, генетичні алгоритми, еволюційні стратегії.

2. *Системи, які є реалістичнішими з біологічної точки зору, але які не мають широкого прикладного застосування.* Їх менше використовують для аналізу технічних завдань. За допомогою таких систем моделюють складні, наближені до реальних природних процесів, дії, до них відносять так зване «штучне життя».

Наразі назвою *еволюційні алгоритми* охоплюють усі обчислювальні і оптимізаційні методи, в основу яких покладена еволюційна метафора:

- еволюційні стратегії,
- еволюційне програмування,
- генетичні алгоритми,
- генетичне програмування.

Ці методи вирізняються здебільшого представленнями можливих розв'язків задач. В еволюційних стратегіях набори хромосом представляють векторами дійсних чисел, в генетичних алгоритмах - векторами двійкових чисел, в еволюційному програмуванні – скінченними автоматами, а в генетичному програмуванні – деревами (списками).

*Еволюційне програмування* винайдене Лоуренсом Фогелем у 1960-1965 роках. Він помітив можливість ще одного, альтернативного підходу до проблем штучного інтелекту – моделювання не кінцевого результату еволюції, а моделювання самого процесу еволюції як засобу «відпрацювання розумної поведінки» і можливостей передбачення різних явищ у середовищі. Л. Фогель виконав низку експериментів, у яких скінченні автомати представляли собою особин у популяції розв'язків задач. У цих скінченних автоматах було передбачено використання символів у цифрових послідовностях, які, еволюціонуючи, ставали все придатнішими до розв'язування поставленої задачі.

В еволюційному програмуванні популяцію скінченних автоматів розглядають в експериментальному середовищі та отримують певну послідовність символів. Для кожного автомата-батька виконується процедура звірення кожного наступного символу з відповідним йому прогнозованим вихідним символом з автомата і оцінюється значення функції втрат для цього виходу. Після останнього прогнозу обчислюється «життєздатність» такого автомата чи програми. Автомати-нащадки утворюють випадковою мутацією автоматів-батьків і теж оцінюють. Найкращі автомати

відбираються для наступного покоління, і процес повторюється. У разі необхідності прогнозу нових символів, нове спостереження додається до старих [11].

Таким чином, в еволюційному програмуванні, на відміну від генетичних алгоритмів, моделюють еволюцію більш як процес пристосувальної поведінки особин популяції або виду, аніж процес адаптації генів.

*Еволюційні стратегії* у багатьох аспектах подібні і до генетичних алгоритмів, і до еволюційного програмування, бо в них теж імітують процеси природної еволюції. Однак, вони мають суттєву відмінність на прикладному рівні. У той час як генетичні алгоритми створені для оптимізації дискретних або цілочислових розв'язків задач, еволюційні стратегії застосовують для неперервних значень, які є типовішими для експериментальних задач.

Основні відмінності еволюційних стратегій від інших оптимізаційних методів і еволюційних алгоритмів [14]: пошук від однієї популяції до іншої на протигагу від однієї особини до іншої; використання даних про саму функцію, а не її похідних; використання ймовірнісних, а не детермінованих, методів за переходу від популяції до популяції; кодування розв'язків векторами дійсних чисел; зосередження уваги на вплив генетичних операторів на зміни фенотипу; адаптивний крок мутації – крок мутації еволюціонує разом з розв'язком, оскільки цей параметр пов'язаний з хромосомами; невелика різниця між батьками та нащадками в процесі схрещування, що обумовлено сильною взаємопов'язаністю – невеликі зміни у перших відображаються невеликими змінами в останніх.

Технологія еволюційних стратегій була винайдена І. Рехенбергом, а згодом розвинута Г.-П. Швefeldem та іншими вченими.

*Генетичне програмування* було розроблене у Сполучених Штатах у 90-их роках, серед перших дослідників – Дж. Коза. Особливістю цього виду еволюційних алгоритмів є використання нелінійних хромосом (графи, дерева). Здебільшого використовують або мутацію, або схрещування. Мутація полягає в заміні випадковим деревом деякого піддерева розв'язку, а схрещування - в обміні батьківських хромосом піддеревами. На відміну від лінійного подання даних, подано через дерева дає можливість отримувати більш гнучкі зміни у розв'язках, а також оперувати хромосомами різного розміру (деревами з різною глибиною чи величиною).

*Генетичні алгоритми* використовуються для розв'язування задач оптимізації і моделювання шляхом послідовного добору, комбінування і варіацій шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію.

Батьком сучасної теорії генетичних алгоритмів вважається Дж. Е. Голланд. Він розпочав роботу над алгоритмами, пов'язаними з діями над послідовностями двійкових чисел, які за аналогією до біологічних понять назвали хромосомами. В його алгоритмах виконувались операції над цими хромосомами, імітуючи процеси природного відбору і еволюції. В 1975 році Дж. Голланд опублікував найвідомішу свою працю «Адаптація у природних і штучних системах» («Adaptation in Natural and Artificial Systems»). Після цієї публікації інтерес до новоствореної галузі безупинно зростає. У книзі вводиться власне ідея генетичного алгоритму («репродуктивний план Голланда» [7, с. 6]) і пропонується схема такого алгоритму, який згодом назвали класичним або канонічним генетичним алгоритмом (canonical GA).

Від традиційних методів оптимізації генетичні алгоритми вирізняються тим, що в них використовують не значення параметрів задачі, а їх закодовані форми (кодування параметрів); здійснюють пошук розв'язку не з єдиної вихідної точки, а відштовхуючись від деякої популяції (операції над популяціями); використовують лише цільову функцію, а не її похідні чи інші допоміжні дані; застосовують ймовірнісні, а не детерміновані правила вибору (рандомізація операцій); їх застосовують до задач, які раніше розв'язувалися лише через повне комбінаторне перебирання, або лише за допомогою нейронних мереж, а також до раніше нерозв'язних задач.

Генетичні алгоритми застосовуються для розв'язування таких задач: оптимізація функцій, оптимізація запитів в базах даних, різноманітні задачі на графах, налаштування і навчання штучної нейронної мережі, задачі компоновки, створення розкладів, ігрові стратегії, штучне життя, біоінформатика, створення дизайну за допомогою комп'ютера, складання порядку розв'язування завдань, тощо. Генетичні алгоритми є одним з найперспективніших напрямів розвитку комп'ютерних технологій, оскільки охоплюють широке коло оптимізаційних (і не тільки) задач.

Еволюційні алгоритми широко застосовують у сучасній теорії оптимізації. Однією з найпоширеніших галузей їх застосування є комбінаторна оптимізація. Так, еволюційні алгоритми з успіхом можна використовувати для розв'язування класичних NP (Nonlinear Programming)-повних проблем, таких як задача комівояжера, задача пакування рюкзака, розбиття чисел, знаходження максимальної незалежної множини, розфарбовування графів тощо.

До інших некласичних, але важливих, задач, для розв'язування яких застосовано еволюційні алгоритми, належать планування, складання розкладів, обчислення маршрутів, задачі розташування та транспортування.

Також еволюційні алгоритми використовують для оптимізації структур та електронних схем, в медицині та в економіці.

На сьогоднішній день існує ряд прикладних програмних продуктів, в яких реалізовано інструментарій еволюційних алгоритмів. Залежно від сфери використання, ступеня автономності та призначення, їх можна класифікувати, зокрема, наступним чином:

– спеціалізоване програмне забезпечення – створені для розв’язування вузького кола прикладних задач. Нескладні для освоєння та користування, але обмежені для використання широким колом користувачів через свою вузьконаправленість;

– додатки до математичних та аналітичних пакетів – через них надаються можливості розв’язувати широке коло задач, але більшість математичних пакетів не є вільно поширюваними та їх використання потребує певного рівня знань та навичок від користувачів;

– фреймворки – є безкоштовним програмним забезпеченням. За їх допомогою користувач може будувати власні програмні засоби на основі існуючого коду, але це також потребує певних знань і навичок у програмуванні.

Розглянемо приклади окремих програмних продуктів, які відносяться до кожного з наведених класів.

#### *Спеціалізоване програмне забезпечення (ПЗ):*

1. NeuroShell Trader – ПЗ для створення торгової системи. Це інструмент, за допомогою якого користувач створює торгові моделі, поєднуючи штучний інтелект і традиційні методи. Можна будувати моделі для акцій, товарів, FOREX, індексів, та ін. Можна створювати моделі для фондових бірж у всьому світі, таких як NYSE (New York Stock Exchange Index), FTSE (Financial Times Stock Exchange Index), DAX (Deutscher Aktienindex (German stock index)) та ін. Створені моделі автоматично тестуються та надають сигнали-прогнози з надходженням нових даних;

2. StrategyQuant – потужна платформа для розробки торгових систем для будь-яких ринків та часових періодів. Не потребує безпосереднього програмування та автоматично тестує згенеровані стратегії;

3. Genetic System Search for Technical Analysis – програмний засіб, використання якого допомагає користувачеві визначати правила роботи та розробляти власні інвестиційні торгові системи.

#### *Додатки та надбудови до математичних та аналітичних пакетів:*

1. XL VIT – додаток до MS Excel, універсальний інструмент для оптимізації та прогнозування. До задач, які можна розв’язувати за його допомогою відносяться опрацювання зображень, складання розкладів, вибір акцій, розпізнавання підозрюваних, управління задачами, торгівля на Forex. Особливостями даного програмного засобу є те, що у роботі з еволюційним алгоритмом, зокрема генетичними, можна використовувати до 100 популяцій, три методи схрещування на вибір та два види масштабування функції пристосовуваності, доступна можливість налаштування рівнів схрещування та мутацій під час роботи з програмою та відстежування значення змінних, а кількість оптимізованих змінних залежить тільки від швидкодії та об’єму запам’ятовуваних пристроїв комп’ютера; є можливість налаштування випадкових вихідних даних та генерації детального звіту і графіка пристосовуваності;

2. Excel Solver – надбудова Microsoft Excel. За допомогою засобу пошуку рішення знаходять розв’язку оптимізаційних задач, зокрема за еволюційним методом.

3. Global Optimization Toolbox є додатком для програмного пакету MATLAB. За його допомогою знаходять глобальні розв’язки многокритеріальних задач. Цей засіб можна використовувати для розв’язування оптимізаційних задач, в яких цільова функція є скінченною, нескінченною, стохастичною, не містить похідних, або включає симуляції чи закриті функції з неявно заданими значеннями для параметрів у налаштуваннях. До особливостей даного інструменту можна віднести вибір оптимального компоненту шляхом використання змішано-цілочислового генетичного алгоритму; обмежену мінімізацію та многокритеріальну оптимізацію, можливість налаштування генетичного алгоритму та застосування гібридних схем.

#### *Фреймворки:*

1. фреймворк Pyevolve – розроблений з метою побудови цілісного фреймворку для генетичних алгоритмів. Його особливостями є мультиплатформне використання, просте для використання API, надання можливості користувачеві створювати нові представлення та генетичні оператори і використовувати еволюційну статистику; висока швидкодія; багатий набір стандартних функцій та параметрів за замовчуванням; відкритий код.

Перелік зазначених вище програмних засобів не є вичерпним, його можна доповнити такими програмними засобами: Evolver, Excel Genetic Algorithm Tool, GeneHunter. Крім того, в Інтернеті також можна знайти багато корисних сайтів, присвячених реалізації еволюційних алгоритмів. Зокрема, на сайті [www.basegroup.ru](http://www.basegroup.ru) запропоновано бібліотеку компонентів "Delphi GeneBase", на сайтах [www.generation5.org](http://www.generation5.org) та [www.sourceforge.net](http://www.sourceforge.net) є достатня кількість прикладів реалізації генетичного алгоритму різними мовами програмування.

Теорія еволюційних алгоритмів дала поштовх для створення нових підходів і до розв’язування класичних задач. Розглянемо задачу комівояжера, яка є однією із найбільш відомих задач комбінаторики. Умова задачі була сформульована в 1934 році, і перед багатьма математиками, які розв’язували її, поставали досить складні проблеми.

#### *Постановка задачі комівояжера.*

На площині задано координати 22 міст України (Рис. 1). Потрібно побудувати шлях подорожі комівояжера до всіх міст так, щоб його довжина була мінімальною, а також за умов, що комівояжер має побувати один раз у кожному з 22 міст (у довільному порядку) і повернутися до міста, з якого починав свою подорож. В якому порядку потрібно відвідати міста, щоб замкнутий шлях комівояжера був найкоротшим?

Назва міста	Ужгород	Львів	Івано-Франківськ	Рівне	Чернівці	Ковель	Хмельницький	Могилів-Подільський	Тернопіль	Житомир	Київ	Чернігів	Суми	Біла Церква	Кіровоград	Миколаїв	Херсон	Запоріжжя	Донецьк	Луганськ	Харків	Кривий Ріг
$x_i$	1	5	6	9	8	6	10	12	8	14	18	18	25	17	21	20	22	27	31	34	28	23
$y_i$	11	15	11	17	9	20	13	9	13	16	16	20	17	14	9	4	2	7	8	10	15	7

Рис. 1.

Розв'язування:

Дану задачу будемо розв'язувати за допомогою методу *Еволюційний пошук розв'язку* додатку до MS Excel *Пошук розв'язку*.

Сформуємо матрицю відстаней обчисленнями значень її елементів  $a_{ij}$ ,  $i, j = 1, 2, 3, 4, \dots, 22$  за формулою

$$a_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad i, j = 1, 2, 3, 4, \dots, 22,$$

усі діагональні елементи  $(a_{ii}) = 0$ ,  $i = 1, 2, 3, 4, \dots, 22$  (рис. 2).

Нехай  $i$  – поточний номер шуканої змінної (співпадає з номером міста),  $i = 1, 2, 3, 4, \dots, 22$

$X = \{x_i\}, (i = 1, 2, 3, 4, \dots, 22)$  – шуканий вектор невідомих,  $X = \{x_1, x_2, x_3, \dots, x_{22}\}$ ;

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
2					1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
3					Ужгород	Львів	Івано-Франківськ	Рівне	Чернівці	Ковель	Хмельницький	Могилів-Подільський	Тернопіль	Житомир	Київ	Чернігів	Суми	Біла Церква	Кіровоград	Миколаїв	Херсон	Запоріжжя	Донецьк	Луганськ	Харків	Кривий Ріг
4			X		1	5	6	9	8	6	10	12	8	14	18	18	25	17	21	20	22	27	31	34	28	23
5			Y		11	15	11	17	9	20	13	9	13	16	16	20	17	14	9	4	2	7	8	10	15	7
6	1	Ужгород	1	11	0	5,7	5,0	10,0	7,3	10,3	9,2	11,2	7,3	13,9	17,7	19,2	24,7	16,3	20,1	20,2	22,8	26,3	30,1	33,0	27,3	22,4
7	2	Львів	5	15	5,7	0	4,1	4,5	6,7	5,1	5,4	9,2	3,6	9,1	13,0	13,9	20,1	12,0	17,1	18,6	21,4	23,4	26,9	29,4	23,0	19,7
8	3	Івано-Франківськ	6	11	5,0	4,1	0	6,7	2,8	9,0	4,5	6,3	2,8	9,4	13,0	15,0	19,9	11,4	15,1	15,7	18,4	21,4	25,2	28,0	22,4	17,5
9	4	Рівне	9	17	10,0	4,5	6,7	0	8,1	4,2	4,1	8,5	4,1	5,1	9,1	9,5	16,0	8,5	14,4	17,0	19,8	20,6	23,8	26,0	19,1	17,2
10	5	Чернівці	8	9	7,3	6,7	2,8	8,1	0	11,2	4,5	4,0	4,0	9,2	12,2	14,9	18,8	10,3	13,0	13,0	15,7	19,1	23,0	26,0	20,9	15,1
11	6	Ковель	6	20	10,3	5,1	9,0	4,2	11,2	0	8,1	12,5	7,3	8,9	12,6	12,0	19,2	12,5	18,6	21,3	24,1	24,7	27,7	29,7	22,6	21,4
12	7	Хмельницький	10	13	9,2	5,4	4,5	4,1	4,5	8,1	0	4,5	2,0	5,0	8,5	10,6	15,5	7,1	11,7	13,5	16,3	18,0	21,6	24,2	18,1	14,3
13	8	Могилів-Подільський	12	9	11,2	9,2	6,3	8,5	4,0	12,5	4,5	0	5,7	7,3	9,2	12,5	15,3	7,1	9,0	9,4	12,2	15,1	19,0	22,0	17,1	11,2
14	9	Тернопіль	8	13	7,3	3,6	2,8	4,1	4,0	7,3	2,0	5,7	0	6,7	10,4	12,2	17,5	9,1	13,6	15,0	17,8	19,9	23,5	26,2	20,1	16,2
15	10	Житомир	14	16	13,9	9,1	9,4	5,1	9,2	8,9	5,0	7,3	6,7	0	4,0	5,7	11,0	3,6	9,9	13,4	16,1	15,8	18,8	20,9	14,0	12,7
16	11	Київ	18	16	17,7	13,0	13,0	9,1	12,2	12,6	8,5	9,2	10,4	4,0	0	4,0	7,1	2,2	7,6	12,2	14,6	12,7	15,3	17,1	10,0	10,3
17	12	Чернігів	18	20	19,2	13,9	15,0	9,5	14,9	12,0	10,6	12,5	12,2	5,7	4,0	0	7,6	6,1	11,4	16,1	18,4	15,8	17,7	18,9	11,2	13,9
18	13	Суми	25	17	24,7	20,1	19,9	16,0	18,8	19,2	15,5	15,3	17,5	11,0	7,1	7,6	0	8,5	8,9	13,9	15,3	10,2	10,8	11,4	3,6	10,2
19	14	Біла Церква	17	14	16,3	12,0	11,4	8,5	10,3	12,5	7,1	7,1	9,1	3,6	2,2	6,1	8,5	0	6,4	10,4	13,0	12,2	15,2	17,5	11,0	9,2
20	15	Кіровоград	21	9	20,1	17,1	15,1	14,4	13,0	18,6	11,7	9,0	13,6	9,9	7,6	11,4	8,9	6,4	0	5,1	7,1	6,3	10,0	13,0	9,2	2,8
21	16	Миколаїв	20	4	20,2	18,6	15,7	17,0	13,0	21,3	13,5	9,4	15,0	13,4	12,2	16,1	13,9	10,4	5,1	0	2,8	7,6	11,7	15,2	13,6	4,2
22	17	Херсон	22	2	22,8	21,4	18,4	19,8	15,7	24,1	16,3	12,2	17,8	16,1	14,6	18,4	15,3	13,0	7,1	2,8	0	7,1	10,8	14,4	14,3	5,1
23	18	Запоріжжя	27	7	26,3	23,4	21,4	20,6	19,1	24,7	18,0	15,1	19,9	15,8	12,7	15,8	10,2	12,2	6,3	7,6	7,1	0	4,1	7,6	8,1	4,0
24	19	Донецьк	31	8	30,1	26,9	25,2	23,8	23,0	27,7	21,6	19,0	23,5	18,8	15,3	17,7	10,8	15,2	10,0	11,7	10,8	4,1	0	3,6	7,6	8,1
25	20	Луганськ	34	10	33,0	29,4	28,0	26,0	26,0	29,7	24,2	22,0	26,2	20,9	17,1	18,9	11,4	17,5	13,0	15,2	14,4	7,6	3,6	0	7,8	11,4
26	21	Харків	28	15	27,3	23,0	22,4	19,1	20,9	22,6	18,1	17,1	20,1	14,0	10,0	11,2	3,6	11,0	9,2	13,6	14,3	8,1	7,6	7,8	0	9,4
27	22	Кривий Ріг	23	7	22,4	19,7	17,5	17,2	15,1	21,4	14,3	11,2	16,2	12,7	10,3	13,9	10,2	9,2	2,8	4,2	5,1	4,0	8,1	11,4	9,4	0

Рис. 2.

$D = \{d(i, j)\}$  ( $i, j = 1, 2, 3, 4, \dots, 22$ ) – вектори відстаней між парами сусідніх міст у контурі обходу розміром з 22 елементів, де  $i < j$  ( $i, j = 1, 2, 3, 4, \dots, 22$ ) (рис. 2).

Потрібно знайти план обходу  $X = \{x_i\}, (i = 1, 2, 3, 4, \dots, 22)$  такий, щоб довжина шляху  $K = \sum_{i,j=1}^{22} d_{ij} \rightarrow \min$  за обмежень:  $x_i \in \{\text{всі різні}\}, (i = 1, 2, 3, 4, \dots, 22)$ .

Обчислимо відстані між містами за допомогою функції – INDEX(), за допомогою якої отримують значення елемента таблиці або масиву за індексами номерів рядків і стовпців (Рис. 3).

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
29																									
30	План		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	1
31	Відстань			5,66	4,12	6,71	8,06	11,18	8,06	4,47	5,66	6,71	4,00	4,00	7,62	8,54	6,40	5,10	2,83	7,07	4,12	3,61	7,81	9,43	22,36

Рис. 3.

В якості змінних даної задачі візьмемо номери міст.

У клітину AA31 введемо формулу для підрахунку значень цільової функції: =SUM(E31:Z31) – сума відстаней між містами (Рис. 5).

На рисунках (4), (5) наведено відповідно параметри використання засобу Пошук розв'язку і результат розв'язування задачі.

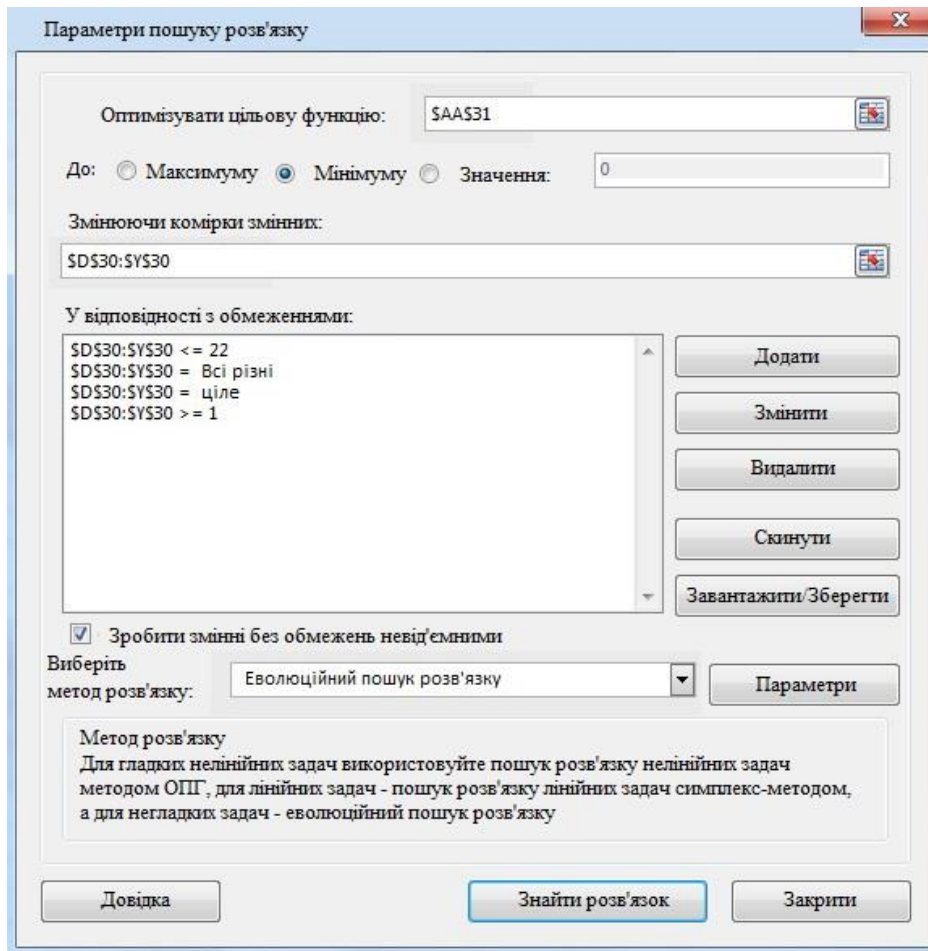


Рис. 4.

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
29																										
30	План		14	11	12	13	21	20	19	18	17	16	22	15	8	5	7	9	3	1	2	6	4	10	14	
31	Відстань			2,24	4,00	7,62	3,61	7,81	3,61	4,12	7,07	2,83	4,24	2,83	9,00	4,00	4,47	2,00	2,83	5,00	5,66	5,10	4,24	5,10	3,61	101,0
32																										

Рис. 5.

Отже, комівояжер пройде найкоротший шлях (довжиною 101 ум. од.), коли буде відвідувати міста в такій послідовності:

14→11→12→13→21→20→19→18→17→16→22→15→8→5→7→9→3→1→2→6→4→10→14.

Поява подібних програмних продуктів на ринку, за умови їх якісної реалізації та доступної ціни, стає не тільки у нагоді користувачам, а і стимулює дослідження у даній сфері та приводить до нових розробок покращених методів оптимізації та розширення сфери використання еволюційних алгоритмів.

Навчання і використання еволюційних алгоритмів з комп'ютерною підтримкою сприяє розвитку математичних та інформатичних компетентностей майбутніх фахівців.

#### Список використаних джерел

1. Дубровін В.І., Федорченко Є.М. Дослідження та розроблення генетичних алгоритмів та операторів схрещування // Інформаційні системи та мережі: Вісник Національного університету «Львівська політехніка». – №673. – Львів, 2010. – С. 97-104.
2. Погорілий С.Д., Білоус Р.В. Генетичний алгоритм розв'язання задачі маршрутизації в мережах // Проблеми програмування. – 2010. – №2–3: Спец. вип. – С. 171–178.
3. Сомотній В., Дзелендзяк У. Використання генетичних алгоритмів для апроксимації функції дійсними числами // Комп'ютерні науки та інформаційні технології: Вісник Національного університету «Львівська політехніка». – №694. – Львів, 2011. – С. 313-318.
4. Субботін С.О., Олійник А.О., Олійник О.О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей: Монографія / За заг. ред. С.О. Субботіна. – Запоріжжя: ЗНТУ, 2009. – 374 с.
5. Курейчик В.М. Генетические алгоритмы и их применение / В.М. Курейчик. – Таганрог: Изд-во ТРТУ, издание второе, дополненное, 2002, 242 с.
6. Курейчик В.М. Поисковая адаптация: теория и практика / В.М. Курейчик, Б.К. Лебедев, О.К. Лебедев. – М.: Физматлит, 2006. – 272 с.
7. Панченко, Т. В. Генетические алгоритмы [Текст] : учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань : Издательский дом «Астраханский университет», 2007. — 87 [3] с.
8. Растрингин Л.А. Статистические методы поиска / Л.А. Растрингин. – М.: наука, 1968. – 76с.
9. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы / Пер. с польск. И.Д. Рудинского. – М.: Горячая линия – Телеком, 2007. – 452 с.
10. Eiben A. E. Introduction to Evolutionary Computing / Eiben A. E., Smith J. E. //Natural Computing Series. – Springer. – 2007. Available at: <http://www.cs.vu.nl/~gusz/ecbook/ecbook.html>
11. Fogel J. A. Retrospective View and Outlook on Evolutionary Algorithms / Fogel J., Lawrence J. A. // Proc. of the Int. Conf. On Computational Intelligence Theory and Applications: Lecture Notes In Computer Science. – 1997. – Vol. 1226. – P. 337-342.
12. Goldberg David E. Genetic algorithms in search, optimization, and machine learning / David E. Goldberg – USA: Addison-Wesley publishing company, Inc., 1989. – 482 p.
13. Holland John H. Adaptation in natural and artificial systems: an introductory analysis with application to biology, control and artificial intelligence. / John H. Holland – USA: University of Michigan, 1975. – 211 p.
14. Kureichik V. M. Evolutionary Computations: Genetic and Evolutionary Programming / Kureichik V. M., Rodzin S.I. // Novosti Iskusstvennogo Intellecta. – 2003. – №5(59). – P. 13–20. – РАИ, Москва. – Available at: [http://www.raai.org/library/ainews/2003/5/kureichik\\_rodzin.doc](http://www.raai.org/library/ainews/2003/5/kureichik_rodzin.doc)

#### О некоторых аспектах изучения эволюционных алгоритмов

*Ишук Я.Ю.*

**Аннотация:** В статье рассмотрены основные понятия эволюционных алгоритмов, их классификации и методы решения задач. Проанализированы современные программные средства, в которых реализован инструментарий эволюционных алгоритмов, приведен пример их использования. Обучение и использование эволюционных алгоритмов с компьютерной поддержкой способствует развитию математических и информатических компетентностей будущих специалистов.

**Ключевые слова:** эволюционный алгоритм, генетический алгоритм, задача коммивояжера.

#### About some aspects of teaching evolutionary algorithms

*Ischuk Yaroslav*

**Resume:** The article describes the basic concepts of evolutionary algorithms, their classification and the methods of problem solving. The author has analysed the modern software which implemented the tools of evolutionary algorithms, and has given the example of their use. The training and use of evolutionary algorithms with computer support contributes to the development of mathematical and informatical competence of future specialists.

**Keywords:** evolutionary algorithm, genetic algorithm, salesman task.