Інформаційні технології і засоби навчання. – 2011. – № 6 (26). – Режим доступу: https://journal.iitta.gov.ua/index.php/itlt/article/view/551.

2. Золотухин С.А. Преимущества и недостатки массовых открытых онлайн-курсов // Дискуссия. – 2015. – № 4 (56). – С. 97-102.

3. Bates T. What's right and what's wrong about Coursera-style MOOCs: [online] // Online Learning and Distance Education Resources. – 05.08.2012 – URL: https://www.tonybates.ca/2012/08/05/whats-right-and-whats-wrong-about-coursera-style-moocs/.

Деякі методичні аспекти навчання генетичних алгоритмів

Кузьміна Наталія Миколаївна
кандидат фіз.-мат. наук, професор кафедри теоретичних основ інформатики
Національний педагогічний університет імені М.П. Драгоманова

Анотація. Розглядаються основні поняття генетичних алгоритмів і пропонуються методичні аспекти їх навчання і реалізації у педагогічних університетах.

Ключові слова. Генетичні алгоритми, функція пристосованості, відбір, схрещування, мутація.

Метою і завданнями навчання нормативної дисципліни «Основи теорії і методів оптимізації» студентів-магістрів інформатичних спеціальностей є ознайомлення і оволодіння основними теоретичними принципами та класичними методами оптимізації, набуття практичних навичок доцільного, ефективного і педагогічно виваженого використання сучасних інформаційних технологій при розв'язуванні задач оптимізації в різних галузях науки і техніки. До предмету навчання відносять основні відомості про класичні методи оптимізації функцій однієї та багатьох змінних, огляд основних постановок, методів дослідження і розв'язування задач безумовної та умовної оптимізації, прикладних задач оптимізації; а також сучасні інформаційні системи і технології, що використовують під час розв'язування оптимізаційних задач [5].

Але доцільним також є розгляд у даному курсі таких оптимізаційних задач, для яких класичні методи і алгоритми оптмізації непридатні.

Одним з універсальних методів оптимізації є генетичні алгоритми, за допомогою яких розв'язують широкий спектр оптимізаційних задач [2-4]. *Генетичний алгоритм* – це алгоритм, за допомогою якого знаходять придатні (за певними критеріями, вимогами) розв'язки аналітично нерозв'язних або складно розв'язних проблем, застосовуючи послідовний добір і комбінування невідомих параметрів (величин) з використанням механізмів, що нагадують біологічну еволюцію.

Термін «генетичний алгоритм» і його класичну схему вперше було запропоновано у 1975 году Дж. Холландом у роботі «Adaptation in Natural and Artificial Systems» [1].

Для застосування генетичних алгоритмів задачу оптимізації розглядають як задачу знаходження максимуму деякої невід'ємної на замкненій області визначення функції $f(x_1,x_2,\ldots,x_n)$, яку називають *функцією пристосованості* (fitness function), для якої не вимагаються неперервність и диференційованість. Кожний параметр функції пристосовності кодується рядком бітів. $(x_1, x_2, \ldots, x_n)$ =>(101,11001,…,1110). Універсальність генетичних алгоритмів полягає в тому, що від постановки конкретної задачі залежать тільки такі параметри, як функція пристосованості і кодування розв'язків. Всі інші кроки алгоритмів для всіх задач виконуються однаково.

Генетичні алгоритми оперують із сукупністю особин (популяцій), які є рядками – кодами можливих розв'язків задачі, тоді як більшість інших алгоритмів оптимізації оперують лише з одним розв'язком, покращуючи його.

За допомогою функції пристосованості серед всіх особин популяції (розв'язків) виокремлюють: найбільш пристосовані, які можуть схрещуватись и давати потомство, та найгірші, які вилучаються з популяції і не дають потомства. Таким чином, пристосованість нового покоління в середньому вище за попереднє.

У класичному генетичному алгоритмі початкова популяція формується випадковим чином; розмір популяції (кількість особин) фіксується і не змінюється під час роботи алгоритму; кожна з особин генерується як випадковий k-бітний рядок, де k – довжина кодування особини – є однаковою для всіх особин.

Схема роботи будь-якого генетичного алгоритму складається зі створення початкової популяції, відбору, схрещування, мутації, переходу до нового покоління і результату. Кожен крок алгоритму має такі стадії:

- генерацию проміжної популяції (*intermediate generation*) шляхом *відбору* (*selection*) біжучого покоління. У класичному генетичному алгоритмі використовується пропорційний відбір (*proportional selection*) попадання особини до проміжної популяції;
- *схрещування* (*recombination*) особин проміжної популяції шляхом *кроссовера* (*crossover*), що забезпечує формування нового покоління. Особини проміжної популяції випадковим чином розбиваються на пари, потім з деякою ймовірністю схрещуються, в результаті чого отримують двох потомків, які записуються у нове покоління, або вони не схрещуються, тоді у нове покоління записується сама пара;
- *мутація* нового покоління. До отриманого в результаті відбору і схрещування нового покоління використовують оператор мутації, який є необхідним для виходу популяції з локального екстремуму, а також сприянню захисту від передчасної збіжності алгоритму.

Критерієм зупинки алгоритму використовують задану кількість поколінь або *сходження* (*convergence*) популяції (всі рядки популяції знаходяться в області деякого екстремума і майже однакові), а це означає, що досягнуте рішення близьке до оптимального.

Під час навчання генетичних алгоритмів доцільно розглянути такі моделі генетичних алгоритмів:

- Genitor (Whitley), де використовуються специфічна стратегія відбору – на кожному кроці тільки одна пара випадкових батьків створює тільки одну дитину, яка замінює не батька, а одну з гірших особин популяції;
- CHC (Eshelman) (Cross generational elitist selection, Heterogenous recombination, Cataclysmic mutation). Для нового покоління обирають N кращих різних особин серед батьків і дітей. При схрещуванні кожному потомку переходить рівно половина бітів кожного з батьків;
- Hybrid algorithm (Davis). У гібрідному алгоритмі об’єднуються переваги генетичних алгоритмів з перевагами класичних методів;
- Island Models (Острівна модель) – модель паралельного генетичного алгоритму та інші.

Студентам інформатичних спеціальностей доцільно давати завдання стосовно програмної реалізації генетичних алгоритмів з подальшим їх тестуванням як на відомих задачах оптимізації, які можна досліджувати і розв’язувати класичними методами оптимізації з використанням, наприклад, систем комп’ютерної математики, так і на таких задачах як оптимізація або апроксимація складних функцій, різноманітні задачі на графах, налаштування і навчання штучних нейронних мереж, складання розкладів, знаходження різних ігрових стратегій, задачі біоінформатики тощо.

Список використаних джерел

1. 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.

2. Курейчик В.М. Генетические алгоритмы и их применение / В.М. Курейчик. – Таганрог: Изд-во ТРТУ, издание второе, дополненное, 2002, 242 с.

3. Авторский сайт Ю. Цоя [Електронний ресурс]. Режим доступу: http://www.qai.narod.ru/.

4. Исаев С.А. Популярно о генетических алгоритмах [Електронний ресурс]. Режим доступу: http://algolist.manual.ru/ai/ga/ga1.php.

5. Кузьміна Н.М. Основи теорії і методів оптимізації: програма навчальної дисципліни для підготовки студентів спеціальностей 7.04030201, 8.04030201 «Інформатика*» Інституту інформатики НПУ імені М.П. Драгоманова// К.: Вид-во НПУ імені М.П. Драгоманова, 2014. – 28 с.