МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет “Києво-Могилянська академія”

Факультет інформатики

Кафедра інформатики

**ЗВІТ**

**до практичного завдання**

**на тему “Експерементальний аналіз ефективності роботи методу стандартного витиснення”**

**Виконав:**

студент ІУСТ-2

Добрянський Андрій Олександрович

**Перевірив:**

канд. фіз.-мат. наук, доцент

Гулаєва Наталія Михайлівна

Київ – 2018

**Зміст**

1 Постановка задачі ........................................................................................ 2

2 Опис інструментарію……............................................................................ 5

Висновки…....................................................................................................... 5

**1 Постановка задачі**

Задачею практичного завдання була реалізація та дослідження методу стандартного витиснення.

*Метод стандартного витиснення,* або *метод витиснення,* або *метод витискаючого множника* (*Standard crowding, Crowding, Crowding factor model*) ґрунтується на припущенні, що різноманітність популяції зберігатиметься, якщо нові особини заміщатимуть схожих на нихчленів популяції. В методі стандартного витиснення використовується стійкий режим репродукції, тобто тільки частина популяції народжується та вмирає на кожному поколінні. Кожна новонароджена особина заміщує існуючу особину популяції, причому перевага надається найбільш схожій на неї особині.

Алгоритм стандартного витиснення працює наступним чином. З популяції за допомогою пропорційного (ймовірнісного) відбору відбираються особини, до яких застосовуються оператори кросинговеру (як правило, одноточковий або однорідний) та мутації. Кількість таких особин визначається параметром *GG* – *розривом покоління* (*generation gap*). Цей параметр задається як частка від загальної кількості елементів популяції. В оригінальній версії алгоритму кількість породжених на кожній ітерації особин складала 20% від розміру популяції.

Далі необхідно визначити множину особин (кількість визначається параметром *GG*), які будуть заміщені новонародженими особинами. Для цього використовується наступний підхід. Для кожного нащадка, якого слід додати в популяцію, випадковим чином вибирають групу з *cf* особин поточної популяції; серед цих особин знаходять найбільш схожу на нащадка особину, яку і видаляють з популяції. Параметр *cf* називають *витискаючим множником* (*crowding factor*); частопокладають *cf*=2 або *cf*=3 (в оригінальній версії *cf*=2). Зазначимо, що за такого підходу новий нащадок теоретично може витиснути з популяції щойно додану особину, втім такий сценарій є малоймовірним. Для визначення схожості особин використовується міра відстані в просторі генотипів або в просторі фенотипів; найчастіше використовується міра відстані в просторі генотипів, зокрема, відстань Геммінга.

Необхідно зазначити, що, розв’язуючи задачі мультимодальної оптимізації, метод стандартного витиснення не завжди дає задовільні результати. Причиною є помилки заміщення, коли особина з однієї ніші витискає особину з іншої. Через помилки заміщення деякі піки можуть втратити всіх своїх особин. Дійсно, нехай елемент з піку 1 має бути доданий в популяцію. Метод витиснення вибирає *cf* кандидатів з популяції та зміщає найближчого з них. Очевидно, якщо в множині з *cf* кандидатів немає елементів з піку 1, то з популяції буде виштовхнуто елемент з якогось іншого піку. Легко бачити, що у випадку, коли сумарна кількість елементів на всіх піках, відмінних від піку 1, перевищує *cf*, ці піки ризикують втратити свої елементи. Експерименти показали, що найчастіше алгоритм знаходить лише два оптимуми [122]. Зауважимо, що ефективність алгоритму суттєво залежить від значення параметру *cf*.

У зв’язку із зазначеним були запропоновані різні модифікації методу стандартного витиснення, більшість з яких спрямовано на зменшення кількості помилок заміщення. Прикладом таких модифікацій є введення додаткового тиску відбору на кроці заміщення:

1. Стратегія заміщення *найближчий-серед-найгірших-1* (*closest-of-the-worst-1*) обирає *cf* кандидатів з популяції зворотно пропорційно до їхнього здоров’я та заміщає найближчого до нового елемента кандидата [291].
2. Стратегія заміщення *найближчий-серед-найгірших-2* (*closest-of-the-worst-2*) заміщає найближчу особину з нижньої третини відсортованої за спаданням здоров’я популяції [292]. Таким чином, для кожної нової особини треба обчислювати відстані до *N*\*1/3 особин.

*NFE* = *N*+*G*\*(*N*\**GG*), де *G* – кількість поколінь

***Набори параметрів.***

Кодування: бінарне - код Грея - вузлами дискретизації з точністю 3 знаки після коми.

Схожість особин визначається в просторі генотипів: відстань Геммінга.

Кросинговер: одноточковий. *Pc* = 1.

Мутація: одноточкова. *Pm*=0.2.

*GG* (розрив покоління) =0.2, *cf* (витискаючий множник)=3

Стратегія заміщення – конфігурації:

P1) Стратегія заміщення: найближчий-серед-найгірших-1.

P2) Стратегія заміщення: найближчий-серед-найгірших-2.

P3) Стратегія заміщення: найближчий-серед-випадково-обраних.

P4) Стратегія заміщення: найгірший-серед-найближчих. Таким чином, для кожної нової особини треба обчислювати відстані до *N* особин.

Для кожного прогону алгоритму обчислюватимемо такі ***критерії***.

1. *NFE* (*number of fitness function evaluations*) – кількість обчислень функції пристосованості. Мається на увазі кількість, задекларована алгоритмом (див. свій варіант).
2. *NSeeds* (*number of Seeds*) – кількість піків (реальних та хибних), знайдених за один прогін алгоритму.
3. *NP* (*number of peaks*) – кількість реальних (глобальних та локальних) піків, знайдених за один прогін алгоритму.
4. *GP* (*number of global peaks*) – кількість реальних глобальних піків, знайдених за один прогін алгоритму.
5. *LP* (*number of local peaks*) – кількість реальних локальних піків, знайдених за один прогін алгоритму.
6. *PR* (*peak ratio*) – відношення кількості знайдених алгоритмом реальних піків до загальної кількості піків, які необхідно локалізувати.
7. *GPR* (*global peak ratio*) – відношення кількості знайдених алгоритмом реальних глобальних піків до загальної кількості глобальних піків, які необхідно локалізувати.
8. *LPR* (*local peak ratio*) – відношення кількості знайдених алгоритмом реальних локальних піків до загальної кількості локальних піків, які необхідно локалізувати.
9. *FPR* (*fake peak ratio*) – частка хибних піків – відношення кількості хибних піків до загальної кількості сформованих алгоритмом ніш: .

**2 Опис інструментарію**

Виконання практичного завдання виконувалося на мові програмування JavаScript з середовищем розробки Intellij Idea.

Для візуалізації роботи генетичного алгоритму та побудови фінальних графіків функції використовувалася бібліотека Plotly.

З урахуванням великої кількості статистичних даних, які необхідно було заповнити, використовувався запис даних у Excel файл з програмного коду за допомогою npm пакету node-xlsx.

Також для розробки даного коду, використовувалася система контролю версій Git з використанням платформи GitHub.

**Висновки**

P1) Стратегія заміщення: найближчий-серед-найгірших-1.

1. Розмірність 1, deba1: глобальні максимуми знайдені, присутня незначна частина хибних піків
2. Розмірність 1, deba2: знаходить глобальний та найвищий локальний піки. Присутня незначна частина хибних піків
3. Розмірність 1, deba3: знаходить всі глобальні піки. Присутня невелика кількість хибних піків.
4. Розмірність 1, deba4: знаходить глобальний максимум та найвищий локальний пік (майже завжди). Присутня невелика кількість хибних піків.

P2) Стратегія заміщення: найближчий-серед-найгірших-2.

1. Розмірність 1, deba1: глобальні максимуми знайдені, присутня значна кількість хибних піків (особливо там, де значення функції мінімальне)
2. Розмірність 1, deba2: глобальний максимум знайдений, присутня значна кількість хибних піків. На відміну від P1, спостерігається краще виявлення локальних піків.
3. Розмірність 1, deba3: знаходить всі глобальні піки, присутня велика кількість хибних піків.
4. Розмірність 1, deba4: знаходить глобальний пік та 2-3 локальні піки. Присутня велика кількість хибних піків.

P3) Стратегія заміщення: найближчий-серед-випадково-обраних.

1. Розмірність 1, deba1: сходження не спостерігалось. Довелося модифікувати стратегію заміщення: дитина витискає батька, лише, якщо має краще здоров’я. Всі піки знайдені, присутні незначна частина хибних піків.
2. Розмірність 1, deba2: знаходить глобальний пік та 1-2 локальних. Присутня невелика кількість хибних піків.
3. Розмірність 1, deba3: знаходить майже всі глобальні піки. Присутня невелика кількість хибних піків.
4. Розмірність 1, deba4: знаходить один глобальний пік та 2 локальні (в середньому)

P4) Стратегія заміщення: найгірший-серед-найближчих. Таким чином, для кожної нової особини треба обчислювати відстані до *N* особин.

1. Розмірність 1, deba1: глобальні максимуми знайдені, присутня значна кількість рівномірно розподілених фейкових піків.
2. Розмірність 1, deba2: знаходить всі глобальні та локальні піки. Дуже багато хибних піків.
3. Розмірність 1, deba3: знаходить усі глобальні піки, велика кількість хибних піків.
4. Розмірність 1, deba4: знаходить всі глобальні та локальні піки. Дуже багато хибних піків.

Експерименти показали, що стратегія заміщення: найгірший-серед-найближчих знаходить найбільшу кількість локальних та глобальних піків, але не зрозуміло, як їх потрібно буде виокремлювати серед хибно знайдених. Натомість, стратегія заміщення: найближчий-серед-найгірших-1знаходить усі глобальні піки та незначну частину локальних піків (здоров’я яких близьке до глобального) з невеликою кількістю хибних піків.