

¹ студент 5 курсу фізико-математичного факультету, СДПУ

² доцент кафедри алгебри, СДПУ

e-mail: vladislav.velichko@ukr.net

ВИКОРИСТАННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧІ КОМІВОЯЖЕРА

Пошук розв'язання задач, в яких точним розв'язком може бути тільки повний перебор, викликає великі труднощі, коли кількість вхідних даних досить великий. А тому, інколи дуже корисними бувають наближені методи, до яких і відноситься генетичний алгоритм.

Ключові слова: задача комівояжера, генетичний алгоритм.

Вступ

Будь-яка задача безумовної оптимізації виглядає наступним чином:

$$\max(\min)f(\vec{x}), \text{ де } \vec{x} = (x_1, x_2, \dots, x_n) \quad x_i \in [a, b] \quad i = \overline{1, n} \quad (1)$$

де $f(\vec{x})$ – цільова функція, що має один глобальний екстремум. Передбачається, що про функцію $f(\vec{x})$ відомо лише те, що вона визначена в будь-якій точці області пошуку. Будь-яка додаткова інформація про характер функції та її властивості (диференційовність, безперервність, властивості Ліпшиця і т.ін.) передбачається невідомою та не враховується у процесі пошуку.

Під розв'язком задачі (1) будемо розуміти вектор $\vec{x} = (x_1, x_2, \dots, x_n)$. Оптимальним розв'язком задачі (1) будемо вважати вектор \vec{x} , при якому цільова функція $f(\vec{x})$ приймає максимальне (мінімальне) значення.

Практично завжди функція яка оптимізується володіє деякою властивістю (властивостями): багатоекстремальності, складна конфігурація допустимої області, наявність декількох типів змінних. Це призводить до необхідності застосування спеціалізованих методів, до яких і відносяться еволюційні і генетичні алгоритми, які добре зарекомендували себе в ситуаціях, коли застосування стандартних методів оптимізації вкрай утруднено.

1. Обмеження при реалізації генетичних алгоритмів

Використання генетичних алгоритмів базується на трьох основних принципах: кодування, оцінювання та відтворення, але з практичної точки зору

мають сенс інші якості, які, однак, анітрошки не скасовують основні паралелі з еволюційними механізмами.

Під *кодуванням* розуміється спосіб представлення даних в генетичному вигляді. Тут важливо, щоб була можливість отримати розв'язання у вигляді хромосоми, а також, щоб у генотипі міг бути записаний будь-який коректний варіант, який більш-менш претендує на те, щоб виявитися відповіддю на поставлене завдання. У більшості випадків проблем з цим не виникає, однак для однієї і тієї ж задачі може існувати кілька способів генетичного подання параметрів, які можуть істотно впливати на швидкість генетичного пошуку і якість розв'язку.

Оцінювання є ще одним важливим принципом. Сенс оцінювання полягає в тому, щоб розрізняти особин в залежності від того, наскільки «успішні» відповідні їм закодовані рішення. При цьому не повинно виникати колізій, коли дві практично рівноцінні особини мають суттєво різні значення пристосованості, і, навпаки, коли якісно різні особини оцінюються однаково.

Основна мета *відтворення* – отримання нових варіантів кандидатів на розв'язок із вже існуючих. Тут дуже бажано, щоб при схрещуванні батьківських особин виходили коректні в рамках поставленої завдання нащадки. У ряді випадків це умова вимагає використання «нестандартних» генетичних операторів і/або специфічного кодування. Наприклад, при вирішенні задачі комівояжера в маршруті не повинна два рази і частіше зустрічатися одна і та сама вершина, що часто виходить в результаті застосування традиційних операторів одно-і двоточкового і однорідного кросоверу. Тому для даної проблеми розроблені спеціальні оператори схрещування та мутації.

Так само важливим параметром генетичних алгоритмів є розмір популяції. При практичній реалізації можливі дві крайності:

- занадто малий розмір популяції (< 10). Даний вибір у більшості випадків годиться тільки для дуже простих завдань. В іншому випадку буде спостерігатися швидке виродження популяції;

- занадто великий розмір популяції (> 1000). Зрозуміло, розв'язок швидше за все, буде знайдено за менше число поколінь, проте часто ціною зайвих обчислювальних витрат. У деяких випадках, коли просто треба знайти розв'язок, це не критично. Проте буває так, що необхідно продемонструвати переваги (якщо є) генетичного підходу для вирішення обраної проблеми перед уже існуючими методами і алгоритмами.

Виходячи з даних рекомендацій, оптимальним розміром популяції є 20 - 30 особин, проте в деяких завданнях потрібно 50-100 особин. Дослідження показують, що розмір популяції багато в чому залежить від розміру хромо-

сом. Так, для алгоритму з 32-бітовими хромосомами розмір популяції буде більше, ніж для алгоритму з 16-бітовими. [1]

Так як у генетичних алгоритмів є характеристика, яка не оцінюється чисельно, що описує їх пошукові здібності, вони залежать від усього, але більшою мірою від стратегій селекції і генетичних операторів. Звідси:

- використання більш агресивних варіантів відбору укупі з досить великою ймовірністю мутації в багатьох випадках дозволяє домогтися більш гарних результатів, в порівнянні з канонічним генетичним алгоритмом. Агресивними стратегіями відбору можна вважати відбір урізанням з досить великим порогом (тобто коли до відтворення допускається менша кількість особин), а також турнірний відбір з розміром турніру 4 і більше;

- популяція більшого розміру працює стабільніше і часто краще. Якщо ж необхідно вкластися в деяку кількість обчислень цільової функції, то краще пошукати оптимальний розмір, при якому і рішення може бути знайдено, і обчислювальні витрати цілком прийнятні;

- двоточковий і однорідний оператори кросоверу, як правило, працюють краще, ніж одноточковий;

- планомірне вистежування і ліквідація диверсійних елементів в особі дублікатів в популяції підвищують якість результатів і є корисним проти передчасної збіжності;

- застосування стратегії елітарності – дозволяє гарантовано залишити в популяції найкращих особин;

- велика ймовірність мутації в деяких випадках здатна поліпшити роботу алгоритму (особливо для малих популяцій), але небажана, в силу внесення великої хаотичності в еволюційний процес, що може негативно позначитися на стабільності роботи алгоритму. [2]

2. Розв'язання задачі комівояжера за допомогою генетичного алгоритму

Розглянемо переваги і недоліки стандартних і генетичних методів на прикладі класичної задачі комівояжера (TSP – traveling salesman problem), яка є однією з найбільш відомих задач дискретної оптимізації. Вона формулюється наступним чином: даний повний зважений граф $G(X, V)$ порядку n , де $X = \{x_1, x_2, \dots, x_n\}$ – множина вершин; $V \subseteq X \times X$ – множина ребер. У даному графі потрібно знайти Гамільтонів цикл, який має найменшу сумарну вагу ребер, які входять до нього.

Або формально:

$$\begin{cases} Q(x) = \sum_{i=1}^N \sum_{j=1}^N c_{ij}x_{ij} \rightarrow \min \\ \sum_{i=1}^N x_{ij} = 1, \forall j = \overline{1, N} \\ \sum_{j=1}^N x_{ij} = 1, \forall i = \overline{1, N} \\ x_{i,j} \in \{0, 1\}, \end{cases} \quad (5)$$

де c_{ij} – вага ребра (i, j) ,

$$x_{ij} = \begin{cases} 1, & \text{якщо є перехід від } i \text{ до } j \\ 0, & \text{якщо переходу від } i \text{ до } j \text{ немає} \end{cases} \quad (6)$$

Очевидно, що розв'язком задачі є перестановка з N вершин, кількість можливих перестановок рівно $N!$, однак кількість різних розв'язків задачі з урахуванням напрямку обходу і зсуву початкової вершини буде $\frac{(N-1)!}{2}$. [3]

Дана задача відноситься до класу NP-повних задач, тобто час роботи алгоритму, який розв'язує задачу комівояжера, істотно залежить від розміру вхідних даних, а отже від кількості населених пунктів.

Всі ефективні (що скорочують повний перебір) методи розв'язання задачі комівояжера – евристичні. У більшості евристичних методів знаходиться не найефективніший маршрут, а наближений розв'язок. Найчастіше затребувані так звані any-time алгоритми, які поступово покращують деякі поточні наближені розв'язки. На практиці застосовуються різні модифікації більш ефективних методів: метод гілок і меж, метод генетичних алгоритмів, а також алгоритм мурашиної колонії.

Для того щоб завдання можна було вирішити за допомогою генетичних алгоритмів, потрібно з'ясувати, що саме є вирішенням цієї задачі, закодувати рішення у вигляді хромосоми і скласти функцію пристосованості для таких хромосом. Тільки після цього можна вирішувати це завдання засобами генетичних алгоритмів.

З'ясуємо, що можна вважати розв'язком задачі комівояжера. Очевидно, що будь-яким розв'язком буде деякий маршрут між населеними пунктами, що задовольняє наступним умовам: він перетинає всі без винятку населені пункти і жоден з них не перетинає більше одного разу. Закодувати такий маршрут можна в вигляді послідовності номерів населених пунктів, починаючи з самого першого, в кінці послідовності номер передостаннього міста, так як маршрут замкнений і останнім буде місто, з якого він починався.

Очевидно, що в цій послідовності не буде повторюваних значень. Нехай для

простоти прикладу кількість у населених пунктів $N = 8$, тоді однією з можливих послідовностей буде шлях, зображений на рис. 1.

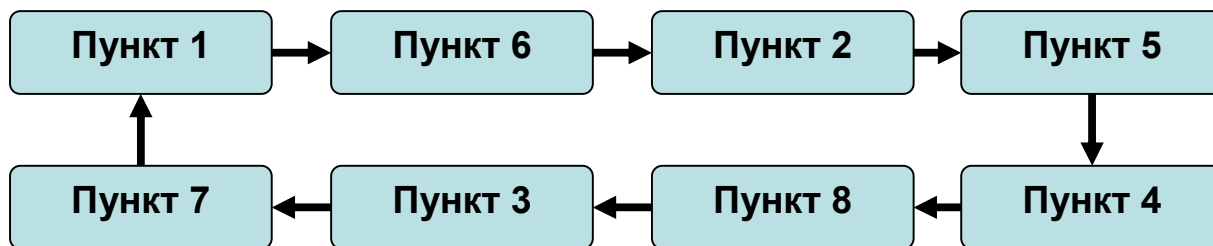


Рис. 1: Приклад маршруту комівояжера при обході 8 населених пунктів

Закодуємо міста числами від 1 до 8. Тоді той же самий шлях набуде вигляду: 1-6-2-5-4-8-3-7.

Тепер нам потрібно представити розв'язок у вигляді хромосоми. Вище ми вже закодували розв'язок у вигляді послідовності номерів населених пунктів, тепер залишилося перекодувати його в хромосому. Для визначеності будемо вважати, що ми кодуємо в хромосому у вигляді бітового вектора. Очевидно, що довжина гена в бітах в хромосомі буде дорівнювати:

$$L = \log_2 N \quad (7)$$

Для нашого прикладу $L = \log_2 8 = 3$, тобто для кодування одного гена знадобитися 3 біта. Кодуємо послідовність за допомогою двійкового кодування (табл. 1):

000	101	001	100	011	111	010	110
1	6	2	5	4	8	3	7

Табл. 1: Кодування послідовності міст за допомогою двійкового кодування

Однак представивши розв'язок таким чином, ми не врахували кілька істотних факторів:

1. при випадковій генерації початкової популяції може виникнути хромосома, в якій будуть повторювані значення генів: 000 000 010 011 100 101 110 010.
2. хромосоми з повторюваними генами може дати кросовер або мутація.

Є кілька способів вирішення цього недоліку кодування, але всі вони ведуть до зайвого споживання обчислювальних ресурсів, тому що треба додатково перевіряти хромосоми.

Один зі способів – перевіряти на повторювані значення усередині функції пристосованості, і, зустрівши такі, замінювати їх на ті значення, яких немає в хромосомі.

Другий спосіб – нічого не перевіряти, а привласнити таким хромосомам дуже низьке значення функції пристосованості, але в цьому випадку генетичний алгоритм починає вкрай неефективно працювати.

Взагалі кажучи, для генетичних алгоритмів дуже важливе питання кодування рішень в послідовність генів. Від того, наскільки воно вдало, залежить якість роботи алгоритму. Найголовніше, і обов'язкове, вимога до кодування – хромосома повинна однозначно представляти деяке рішення, щоб не було можливості трактувати одну і ту ж хромосому по-різному. Бажано, щоб хромосоми займали якомога менше біт, були коротші. Так само важливою умовою є простота кодування. Від цього залежить швидкість роботи. Після кодування запускається генетичний алгоритм з бажаними параметрами.

Література

- [1] *Панченко Т. В.* Генетические алгоритмы: учебно-методическое пособие / Т.В. Панченко. — Астрахань: Изд. дом «Астраханский университет», 2007. — 87 с.
- [2] Генетические алгоритмы [Электронный ресурс] // Генетические алгоритмы и не только. — Электрон. дан. — [б.м.], 2003-2007. — URL: <http://qai.narod.ru/GA/> (Дата обращения: 01.04.2011).
- [3] *Батищев Д.И., Неймарк Е.А., Старостин Н.В.* Применение генетических алгоритмов к решению задач дискретной оптимизации / Д.И. Батищев, Е.А. Неймарк, Н.В. Старостин. — Н. Новгород: 2007. — 85 с.