АДАПТАЦІЙНІ МЕХАНІЗМИ В ГЕНЕТИЧНИХ АЛГОРИТМАХ

Скобцов Ю.О., Закусило С.А., Іванов Д.Є.

Інститут прикладної математики та механіки НАН України, м. Донецьк
E-mail: skobtsov@iamm.ac.donetsk.ua, myf2001@rax.ru, ivanov@iamm.ac.donetsk.ua

Abstract

Skobtsov Yu., Zakusilo S., Ivanov D. An adaptation mechanisms for genetic algorithms. *An easy built in adaptation mechanism for genetic algorithm is considered. A method for maintaining genetic algorithm parameters is proposed that based on the appropriate changing of the crossingover and mutation probability depending on the individual position in the population. The results of the comparison between the simple genetic algorithm and adaptive genetic algorithm are presented that show the advantage of the proposed adaptive mechanism.*

Генетичні алгоритми відомі та добре зарекомендували себе як потужний засіб розв'язку пошукових та оптимізаційних задач. Вони мають як переваги, так і недоліки.

До переваг застосування ГА на практиці слід віднести незалежність від специфіки розв'язуваної задачі. Загальна схема ГА залишається незмінною для будь-якої задачі, що розв'язується за його допомогою. Змінюються тільки відповідні дані та деякі параметри ГА. З іншого боку, паралелізм ГА дозволяє обробляти набагато більше інформації за одну ітерацію ГА, ніж стандартні алгоритми, навіть при розгортанні ГА в стандартний алгоритм. До переваг ГА слід також віднести їх дуже легку реалізацію на паралельних обчислювальних комплексах.

До недоліків ГА слід віднести те, що застосування таких алгоритмів для розв'язку певної задачі не гарантує отримання найкращого рішення даної задачі (хоча знайдене ГА рішення, як правило, досить близьке до оптимального). Іншим недоліком ГА є те, що, якщо такий алгоритм не знаходить розв'язку, то це не означає, що його взагалі не існує.

Перші моделі генетичних алгоритмів [1] мали досить просту структуру. В подальшому вони отримали назву простих генетичних алгоритмів. Псевдокод простого генетичного алгоритму представлено на Схемі. 1.

створення стартової популяції);
доки (не досягнуто умови закінчення) виконувати
{
операція вибору;
операція кросинговер (P_c);
операція мутації (P_m);
}
вивести <найкращу особу>;

Схема. 1. Схема простого генетичного алгоритму.

Виконання цього алгоритму залежить від наступних параметрів:

- P_c – ймовірність кросинговеру;
- P_m – ймовірність мутації;
- N_p – потужність (розмір) популяції,

які визначають частоту застосування генетичних операторів до осіб в перехідній популяції

(батьків). Для простого генетичного алгоритму ці величини є наперед заданими, та

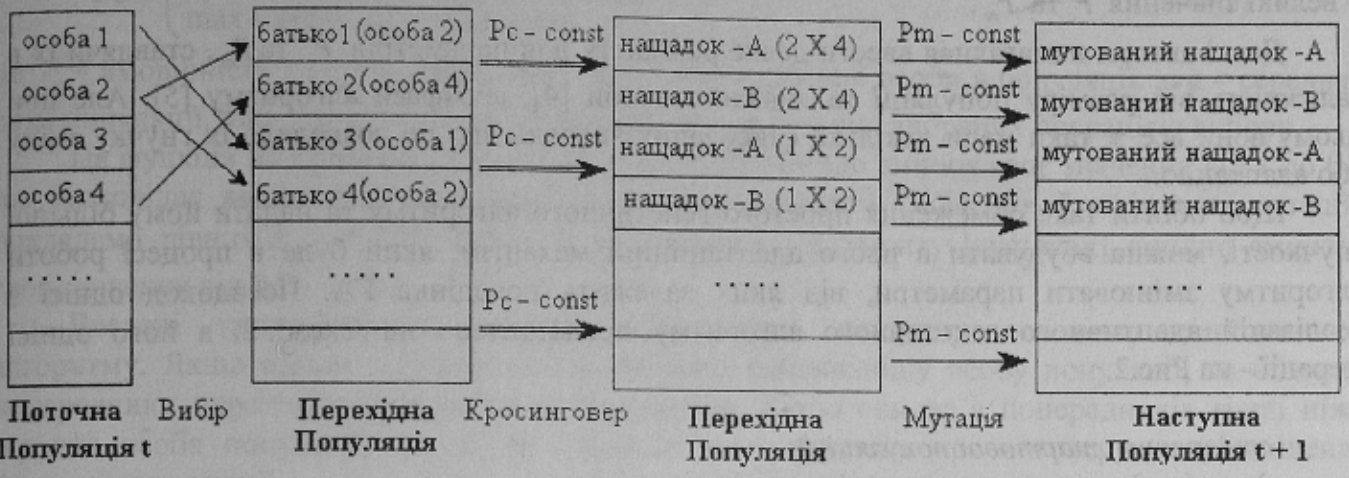

Рис.1. Одна ітерація простого генетичного алгоритму.

залишаються незмінними протягом всього терміну роботи генетичного алгоритму. Одну ітерацію простого генетичного алгоритму представлено на Рис. 1.

Ефективність та дієвість ГА такого типу доведена теоремою схем [1,2]. У модифікованому вигляді цю теорему можна використовувати для визначення оптимальних параметрів при реалізації генетичних алгоритмів [3].

Розглянута вище модель простого генетичного алгоритму передбачає хоча і випадкову, але цілком визначену поведінку, яка зумовлена наперед заданими параметрами. Ці параметри залишаються незмінними протягом всього періоду роботи алгоритму.

Роботу (функціонування) генетичного алгоритму можна розглядати як збалансовану комбінацію розширення (освоєння) нових регіонів в просторі пошуку та використання (експлуатації) вже перевірених областей. Баланс між розширенням та експлуатацією таких областей суттєво впливає на характеристики генетичного алгоритму і визначається правильним вибором керуючих параметрів: ймовірністю кросинговеру $-P_c$, ймовірністю мутації - P_m, розміром (кількістю осіб) популяції $-N_p$. При цьому потрібно враховувати наступні фактори:

- збільшення ймовірності кросинговеру P_c приводить з одного боку до підвищення рекомбінації "будівельних блоків", а з іншого – до руйнування вже існуючих хороших розв'язків;
- збільшення ймовірності мутації P_m перетворює генетичний пошук в сліпий випадковий пошук і, в той же час, приводить до більшого введення нового генетичного матеріалу в популяцію;
- збільшення розміру популяції приводить до різноманітності в популяції і знижує ймовірність попадання в пастку передчасного збігу до локального екстремуму. З іншого боку, збільшується час згущення популяції в перспективних областях простору пошуку.

Без сумніву, керуючі параметри слід вибирати з врахуванням взаємодії основних операторів генетичного алгоритму. Оптимальні керуючі параметри суттєво залежать від виду цільової функції. Існує два протилежних підходи до розв'язку даної проблеми. При першому підході використовується великий розмір (потужність) популяції N_p та відносно невеликі ймовірності кросинговеру P_c та мутації P_m (типові значення P_c=0.6, P_m=0.001, N_p=100). Другий підхід використовує популяції меншого розміру, але більші ймовірності кросинговеру та мутації (характерні значення P_c=0.9, P_m=0.01, N_p=30). При використанні першого підходу мутація відіграє другорядну роль порівняно з другим. Для другого підходу,

який використовує малий розмір популяції, бажана більша мінливість генетичного матеріалу – великі значення P_c та P_m.

Деякі автори намагалися ввести деяке розмаїття для параметрів P_c та P_m, ставлячи їх в залежність від розміру популяції та довжини особи [4], або фази алгоритму [5]. Але при цьому вони все ж таки мали декілька фіксованих значень, що не дозволяло їх гнучку зміну або адаптацію.

Щоб обійти таке обмеження простого генетичного алгоритму та надати йому більшої гнучкості, можна вбудувати в нього адаптаційний механізм, який буде в процесі роботи алгоритму змінювати параметри, від яких залежить поведінка ГА. Псевдокод однієї з реалізацій адаптивного генетичного алгоритму представлено на Схемі. 2, а його однієї ітерації - на Рис.2.

створення стартової популяції();
доки (не досягнуто умови закінчення) виконувати
{
операція вибору;
визначити $P_c = f_{Pc}(сила\_батькаA, сила\_батькаB)$;
визначити $P_m = F_{Pm}(сила\_батькаA, сила\_батькаB)$;
операція кросинговера (P_c);
операція мутації (P_m);
}
вивести <найкращу особу>;

Схема. 2. Схема адаптивного генетичного алгоритму.

Розглянемо один з механізмів адаптації. Нехай адаптація буде виявлятися наступним чином. Після вибору з популяції двох попередників саме для вибраних визначимо ймовірність кросинговеру та мутації за наступними правилами:

- у випадку, коли сильніший з попередників гірший ніж середня особа популяції, то застосуємо фіксовані значення ймовірності кросинговеру та мутації $P_c = 1, P_m = 0.5$;
- чим ближче сильніший з попередників до найсильнішої особи, тим менший рівень ймовірності кросинговеру та мутації будемо застосовувати до цієї пари. Для визначення значення P_c застосуємо формулу:

Рис.2. Одна ітерація адаптивного генетичного алгоритму.

$$Pc = \left(\frac{\max}{\max - avg}\right) - \left(\frac{\max\_mate\_fit}{\max - avg}\right), \quad (1)$$

де P_c – ймовірність кросинговеру, max – сила найсильнішої особи в популяції, avg – середня сила популяції, max_mate_fit – сила найсильнішого з батьків, вибраних операцією вибору.

Ця функція на проміжку аргументу [avg; max] лінійно змінює своє значення від 1 до 0, що відповідає введеному нами раніше правилу адаптації. Значення ймовірності мутації покладемо рівним $P_m = P_c / 4$. Тобто P_m на проміжку аргументу [avg; max] лінійно змінює своє значення від 0,25 до 0.

Введені таким чином правила забезпечують наступну поведінку генетичного алгоритму. Якщо одним з попередників вибрано найсильнішу особу популяції, то обидва попередники переносяться в наступну популяцію. Якщо обидва з попередників гірші ніж середня особа популяції, то до них завжди застосовується кросинговеру та підвищена ймовірність мутації.

Як проводилося порівняння. В якості тестової задачі були вибрані модифіковані функції Де Йонга [2]. Модифікація стосувалася того, що функція замінювалася на її симетричний образ, який мав один глобальний максимум, а також додатні значення на всій області, на якій відбувався пошук. Таке перетворення забезпечувалося наступною формулою: $F_1 = Max(F) - F$, де F_1 – модифікована функція Де Йонга, $Max(F)$ – максимальне значення оригінальної функції Де Йонга на області пошуку або більша за його величина, F – оригінальна функція Де Йонга. Модифіковані функції Де Йонга наведені нижче.

1) Модифікована функція Де Йонга F_1:

$$100 - \sum_{i=1}^{3} x_i^2, \text{ де } -5.12 \le x_i \le 5.12. \quad (2)$$

Функція має глобальний максимум 100 в точці $(x_1,x_2,x_3) = (0,0,0)$.

2) Модифікована функція Де Йонга F_2:

$$10000 - 100\left(x_1^2 - x_2\right)^2 + \left(1 - x_2\right)^2, \text{ де } -2.048 \le x_i \le 2.048. \quad (3)$$

Функція має глобальний максимум 10000 в точці $(x_1,x_2) = (1,1)$.

3) Модифікована функція Де Йонга F_3:

$$30 - \sum_{i=1}^{5} \text{int}(x_i), \text{ де } -5.12 \le x_i \le 5.12. \quad (4)$$

Функція має глобальний максимум 60 для всіх $-5.12 <= x_i <= -5$.

4) Модифікована функція Де Йонга F_4:

$$1000 - \sum_{i=1}^{30} i \cdot x_i^4 + Gauss(0,1), \text{ де } -1.28 \le x_i \le 1.28. \quad (5)$$

Функція (без шуму Гауса) має глобальний максимум 1000 в точці $(x_1, x_2, ..., x_{30}) = (0,0,...,0)$.

Далі для знаходження максимуму тестових функцій застосовувалися простий та адаптивний генетичні алгоритми. Для реалізацій обох алгоритмів максимальна кількість параметрів вибиралася однаковою: спосіб кодування інформації в генах, розмір популяції, початкова популяція, максимальна кількість поколінь, генетичні оператори, умова зупинки пошуку. В якості умови зупинки пошуку вибиралася наступна: лічильник прогресу = 0 або досягнуто покоління з максимальним номером. На початку роботи генетичного алгоритму лічильник прогресу приймався рівним 1/4 максимальної кількості поколінь і в процесі роботи ГА змінював своє значення наступним чином: якщо при переході до наступного покоління відбувалося покращення найсильнішої особи в популяції, то лічильник прогресу збільшувався на 1, якщо ж покращення не відбувалося то він зменшувався на 1. Як в

простому так і в адаптивному ГА використовувалися наступні генетичні оператори: вибір – roulette wheel, кросинговер – одномісцевий, мутація – зміна значення гену з ймовірністю P_m. Розмір популяції в обох випадках був рівний 40. В простому генетичному алгоритмі ймовірності кросинговеру та мутації були наступні $P_c = 1/4$, $P_m = 1/(3 * L_{chrom})$, де L_{chrom} – кількість генів в хромосомі особи (це означає одна мутація на три хромосоми). Кодування аргументів функцій виконувалося згідно з Таблицею 1.

Таблиця 1. Характеристики функцій Де Йонга.

Функція	Кількість аргументів	біт на аргумент
Де Йонг F_1	3	20
Де Йонг F_2	2	11
Де Йонг F_3	5	20
Де Йонг F_4	30	5

Для кожної з тестових функцій виконувалося 20 запусків простого та адаптивного ГА при різних значеннях максимальної кількості поколінь (строка MaxGen). Як результати роботи бралися середні арифметичні значення наступних величин: кількість поколінь, виконаних ГА (avg gen) та максимальне значення функції (avg max), знайдене ГА на даній області пошуку. Крім того фіксувалося, скільки разів ГА знайшов єдиний максимум (find max). Результати наведено в Таблицях 2-5.

Із наведених даних можна бачити, що простий ГА знайшов максимум тестових функцій у 4-х випадках, а адаптивний ГА у 6 із 380 запусків програми. Але у 89% випадків адаптивний ГА знайшов максимальне значення, яке знаходиться ближче до глобального оптимуму, ніж значення, знайдене простим ГА. Також адаптивний ГА завершував роботу приблизно вдвічі швидше ніж простий ГА, будуючи в середньому лише 49% від числа популяцій, що їх побудував простий ГА.

Таблиця 2. Результати роботи простого та адаптивного генетичних алгоритмів для модифікованої функції Де Йонга F_1.

MaxGen		50	100	200	400	1000
Простий ГА	avg gen	50.00	100.00	200.00	400.00	1000.00
	avg max	97.86	97.96	97.75	98.28	98.59
	find max	0	0	0	0	0
Адаптивний ГА	avg gen	20.60	34.90	64.30	117.60	263.30
	avg max	99.08	99.44	99.61	99.73	99.26
	find max	0	0	0	0	0

Таблиця 3. Результати роботи простого та адаптивного генетичних алгоритмів для модифікованої функції Де Йонга F_2.

MaxGen		50	100	200	400	1000
Простий ГА	avg gen	37.50	78.60	163.70	316.00	804.60
	avg max	9999.28	9998.76	9998.66	9998.69	9998.95
	find max	2	0	1	1	0
Адаптивний ГА	avg gen	16.60	30.90	61.10	110.10	269.10
	avg max	9999.08	9999.19	9999.68	9999.18	9999.21
	find max	1	1	2	2	1

Таблиця 4. Результати роботи простого та адаптивного генетичних алгоритмів для модифікованої функції Де Йонга F_3.

MaxGen		50	100	200	400	1000
простий ГА	avg gen	19.30	40.40	75.50	137.40	321.70
	avg max	45.05	46.45	46.20	49.20	50.05
	find max	0	0	0	0	0
Адаптивний ГА	avg gen	17.50	32.80	56.80	108.30	258.10
	avg max	46.40	47.45	47.35	47.70	47.40
	find max	0	0	0	0	0

Таблиця 5. Результати роботи простого та адаптивного генетичних алгоритмів для модифікованої функції Де Йонга F_4.

MaxGen		50	100	200	400	1000
простий ГА	avg gen	42.40	90.60	185.60	370.90	–
	avg max	902.08	919.20	928.52	930.17	–
	find max	0	0	0	0	0
Адаптивний ГА	avg gen	26.80	45.70	69.70	117.40	–
	avg max	927.30	929.02	933.85	938.03	–
	find max	0	0	0	0	0

Слід відзначити, що розглянутий метод адаптації параметрів ГА не завжди приводить до єдиного максимуму, який існує на даній області аргументів функції. І, хоча адаптивний генетичний алгоритм працює набагато швидше, він не завжди дає кращий результат ніж звичайний генетичний алгоритм.

Висновки

Розглянуто механізми адаптації параметрів для генетичних алгоритмів, а саме ймовірностей кросинговеру та мутації. Експериментально показано, що адаптивні ГА показують більшу ефективність порівняно з простими ГА. Простий спосіб вбудовування адаптації в генетичний алгоритм робить подібний механізм можливим для застосування майже у всіх відомих реалізаціях генетичних алгоритмів. Значні варіації в виборі адаптації відкривають широкий простір для пошуку більш ефективних методів адаптації.

Література

1. Holland J.P. Adaptation in Natural Artificial systems. University of Michigan Press, Ann Arbor, MC(USA), 1975.
2. Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs. Springer Verlag, Berlin (D), 1992.
3. Goldberg D.E. Kumara S. A Practical Schema Theorem for Genetic Algorithms Design and Tuning. IlliGAL Report №2001017, January 2001.
4. Elizabet M. Rudnick, Janak H. Patel, Gary S. Greenstain, Thomas N. Niermann, Sequential Circuits Test Generation in a Genetic Algorithm Framework // Proceedings of the ACM/IEEE Design Automation Conference, pp.698-704, June 1994.
5. Corno F., Prinetto P., Sonza Reorda M., Squillero G. A new approach for Initialization Sequences Computation for Synchronous Sequential Circuits // Proceedings of the ICCD'97, pp.381-387, 1997.