УДК 004.056.5:519.8

А.Д. КОЖУХІВСЬКИЙ, О.О. НАМОФІЛОВА

ЗАСТОСУВАННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ У ЗАДАЧІ ПРО УКЛАДАННЯ РАНЦЯ

Розглядається застосування генетичних алгоритмів у криптології. Описуються основні поняття генетичного алгоритму, представляється його цикл у вигляді блок-схеми та приведена послідовність етапів роботи алгоритму. Формулюється задача про укладання ранця та загальний алгоритм її розв'язання. На прикладі задачі про укладання ранця розглядається можливість застосування генетичних алгоритмів при шифруванні. Показується, що застосування генетичних алгоритмів – один із найкращих методів для пошуку «прийнятних» рішень.

1. Вступ

Ідея генетичних алгоритмів запозичена з живої природи. Вона полягає в машинній організації еволюційного процесу створення, модифікації і відбору кращих розв'язків. У загальному значенні *генетичні алгоритми* – це тип алгоритмів, інспірованих механізмами еволюції живої природи, які застосовуються до задач глобальної оптимізації, для комбінування шаблонів з правил індукції, що були відкриті до цього, навчання нейромереж, пошуку зразків у даних, відкриття шаблонів у тексті тощо. Методологічна основа генетичних алгоритмів ґрунтується на гіпотезі селекції, яка в загальному виді може бути сформульована так: чим вища пристосованість особини, тим вища ймовірність того, що в потомстві, отриманому з її участю, ознаки, які визначають пристосованість, будуть виражені ще сильніше. Генетичні алгоритми застосовуються в багатьох задачах, в тому числі і в задачах криптології.

Шифрування інформації в наш час стало чи не основним методом її захисту. Доступність обчислювальної техніки й стрімкий прогрес у її розвитку привели до вдосконалювання давно відомих шифрів і застосування в масовому масштабі нових високонадійних схем шифрування. Однак цей прогрес має й інший бік: збільшені можливості обчислювальної техніки успішно застосовуються не тільки для шифрування, але й для дешифрування тих шифрів, які ще зовсім недавно гарантували повний захист інформації.

Основними завданнями криптології є розробка надійних схем шифрування (завдання криптографії) і знаходження ефективних методів дешифрування існуючих схем (завдання криптоаналізу). Криптографічний спосіб захисту інформації передбачає таке її перетворення, при якому вона стає доступною для прочитання лише власникові секретного ключа. Надійність цього способу захисту визначається стійкістю використовуваної схеми шифрування до криптоаналізу. При криптоаналізі конкретного шифру передбачається, що сама схема шифрування відома, а невідомим є тільки секретний ключ і/або його довжина. Іншими словами, завдання розкриття шифру полягає в знаходженні єдиного справжнього секретного ключа серед безлічі всіх можливих ключів, тобто є завданням пошуку. При цьому простір пошуку великий, а критерій «якості» знайденого розв'язку, як правило, не піддається строгій формалізації. В наш час у криптоаналізі успішним є застосування генетичних алгоритмів.

Метою дослідження є розробка альтернативного способу розв'язання задачі про укладання ранця.

2. Основні поняття генетичного алгоритму

Застосування генетичних алгоритмів вперше запропоновано Джоном Холландом. Ці алгоритми являють собою модифікацію так званого «еволюційного програмування». Ідея Холланда полягала в тому, щоб розробити алгоритми на основі «спрямованого» випадкового пошуку або на основі механізмів природного відбору, відомого з біології. На етапі ініціалізації цієї процедури створюється популяція можливих розв'язків. У результаті з цієї популяції виводиться нове покоління розв'язків, яке, у свою чергу, служить «вихідним

матеріалом» для чергового покоління [1]. Представимо цикл генетичного алгоритму у вигляді блок-схеми (рис. 1), який включає стадії відбору, схрещування й мутації. Аналізуючи представлену блок-схему, відмітимо, що кращі представники покоління відбираються для відтворення популяції. Таким чином, можна припустити, що кожне нове покоління повинне містити кращі розв'язки, ніж попереднє.

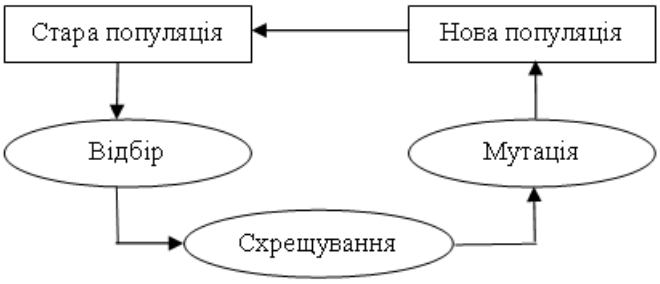

Рис. 1. Блок-схема еволюції популяції

Розглянемо принцип роботи блок-схеми, представленої на рис. 1.

Стара популяція складається з набору бінарних рядків. *Кожний бінарний рядок* представляє розв'язок проблеми й *називається хромосомою*. Першою стадією генетичного алгоритму є відбір. У процесі відбору визначаються рядки, які будуть використовуватися при створенні нової популяції (нового покоління). «Батьки» вибираються довільно, однак «кращі особини» популяції мають більший шанс виявитися обраними. Таким чином, алгоритм «просувається» в перспективному напрямку пошуку.

Наступна стадія – схрещування, яке полягає в тому, що для «батьків» (пари відібраних рядків – пари хромосом) довжини r (r – кількість бітів у хромосомі) кожна наступна пара «нащадків» вибирається довільним чином. «Нащадки» формуються з частини бітів s, де $s \in \{1,...,r\}$, і бітів s' та s'', де s' і s'' – це частина бітів від $s+1$ до r, якими обмінюються «батьки» (рис. 2).

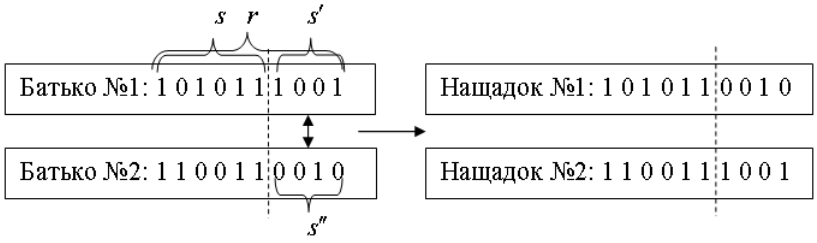

Рис. 2. Передача генетичної інформації від «батьків» до «нащадків»

Заключна стадія – мутація. При ініціалізації алгоритму встановлюється ймовірність мутації, якої зазнають новоутворені хромосоми. Для прикладу, наведеного на рис. 2, розглянемо таку мутацію: «нащадок №1» отримав мутацію 8-го біта, а «нащадок №2» отримав мутацію 2-го і 10-го біта (рис. 3).

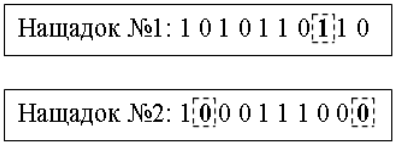

Рис. 3. Процес мутації серед «нащадків»

Схема еволюції популяції (див. рис. 1) буде виконана як мінімум один раз. Стадії еволюції повторюються до досягнення умови виходу із циклу (такою умовою може бути, наприклад, перевищення максимальної кількості популяцій).

3. Шифрування за допомогою задачі про укладання ранця

Один із перших шифрів на основі задачі про укладання ранця був запропонований Мерклі й Хеллманом в 1978 [2]. Це була одна з перших спроб створення системи шифрування з

відкритим ключем. Незважаючи на те, що проблема укладання ранця відноситься до класу NP–повних [3], було показано, що більшість версій алгоритму є нестійкими. В 1983 р. Брікел запропонував спосіб злому криптосистеми на основі ранця низької щільності [4]. Рік по тому Шамір розробив поліноміальний алгоритм для атаки на вихідну «рюкзачну» криптосистему [5]. Після цього було запропоновано безліч інших систем на основі алгоритму укладання ранця: кілька послідовних рюкзаків, рюкзаки Грем–Шаміра [6]. Для всіх цих систем були розроблені методи розкриття. У статті [7] пропонується ще один метод криптоаналізу шифрів на основі алгоритму укладання ранця; відмінною рисою такого підходу є його універсальність, тобто можливість застосування до будь-якої версії «рюкзачної» криптосистеми, а також простота роботи. Метод базується на використанні генетичних алгоритмів.

Розглянемо формулювання задачі про укладання ранця. Задана множина предметів різної ваги; запитується, чи можна покласти деякі із цих предметів у ранець так, щоб його вага стала дорівнювати певному значенню? Більш формально задача формулюється так: даний набір значень $M_1, M_2, ..., M_n$ і сумарне значення S, потрібно обчислити значення b_i $\left(i = \overline{0,n}\right)$ такі, що: $S = b_1M_1 + b_2M_2 + ... + b_nM_n$.

Тут b_i може бути або нулем, або одиницею. Значення $b_i = 1$ означає, що і–й предмет кладуть у рюкзак, а $b_i = 0$ – не кладуть. Звідси випливає уявлення про вміст рюкзака у вигляді хромосом, біти яких відповідають значенням b_i. Функція вибору «кращих хромосом» оцінює близькість ваги конкретного рюкзака до заданого числа. Значення функції розташовуються в діапазоні [0, 1], де 1 означає точний збіг із шуканою вагою. Якщо вага одного рюкзака перевищує цільове значення S на деяке число x, а вага іншого, навпаки, менша за потрібну на те ж число x, то «кращим» вважається останній рюкзак. Опишемо цю задачу більш формально:

1. Потрібно обчислити максимальну розбіжність, яка може виникнути між довільною хромосомою й цільовим значенням S: $\Delta_{max} = \max\left(S, \widetilde{S} - S\right)$, де $\widetilde{S}$ – сума всіх компонентів, які можна використовувати при укладанні рюкзака.
2. Обчислити вагу рюкзака, відповідного до поточної хромосоми, і позначити S'.
3. Якщо $S' \le S$, то «якість» хромосоми оцінюється значенням: $\alpha = 1 - \sqrt{\frac{|S' - S|}{S}}$.
4. Якщо $S' > S$, то «якість» хромосоми оцінюється значенням: $\alpha = 1 - \sqrt{\frac{|S' - S|}{\Delta_{max}}}$.

Загальний алгоритм задачі про укладання ранця можна представити так:

1. Створюється випадкова популяція двійкових хромосом.
2. Для кожної хромосоми обчислюється значення α (функція оцінки).
3. На основі отриманих коефіцієнтів відбувається природний відбір.
4. До обраних на 3-му етапі особин застосовується схрещування.
5. Нащадки зазнають мутації.
6. Нова популяція аналізується, виділяються «кращі хромосоми».

Процес перерветься, коли кількість поколінь перевищить певне задане число. «Кращі хромосоми» нового покоління будуть використані для «злому» шифру [8, 9].

Розглянемо приклад шифрування за допомогою задачі про ранець.

Рюкзачний вектор $A = \left(a_1, a_2, ..., a_n\right)$ – впорядкований набір із n предметів. Повідомлення шифрується як розв'язок набору задач про ранець. Для шифрування відкритого тексту у двійковому представленні його розбивають на блоки довжини n (наприклад, (1 1 1 0 0) відповідає 5-ти предметам у рюкзаку). Вважається, що одиниця вказує на наявність предмета в рюкзаку, а нуль – на його відсутність. Якщо для заданого рюкзачного вектора $A = (3,4,5,6,7,8)$ довжини $n = 6$ за даним алгоритмом шукати шифротекст, то усі криптосистеми не перевищуватимуть 33 – сумарну вагу всіх предметів у рюкзачному векторі (таблиця).

Шифрування тексту у двійковому представленні

Відкритий текст						Речі в рюкзаку	Шифротекст
0	0	0	0	0	0	0·3 + 0·4 + 0·5 + 0·6 + 0·7 + 0·8	0
0	0	0	0	0	1	0·3 + 0·4 + 0·5 + 0·6 + 0·7 + 1·8	8
0	0	0	0	1	0	0·3 + 0·4 + 0·5 + 0·6 + 1·7 + 0·8	7
0	0	0	0	1	1	0·3 + 0·4 + 0·5 + 0·6 + 0·7 + 0·8	15
0	0	0	1	0	0	0·3 + 0·4 + 0·5 + 0·6 + 1·7 + 1·8	6
0	0	0	1	0	1	0·3 + 0·4 + 0·5 + 1·6 + 0·7 + 1·8	14
0	0	0	1	1	0	0·3 + 0·4 + 0·5 + 1·6 + 1·7 + 0·8	13
0	0	0	1	1	1	0·3 + 0·4 + 0·5 + 1·6 + 1·7 + 1·8	21
0	0	1	0	0	0	0·3 + 0·4 + 1·5 + 0·6 + 0·7 + 0·8	5
0	0	1	0	0	1	0·3 + 0·4 + 1·5 + 0·6 + 0·7 + 1·8	13
0	0	1	0	1	0	0·3 + 0·4 + 1·5 + 0·6 + 1·7 + 0·8	12
0	0	1	0	1	1	0·3 + 0·4 + 1·5 + 0·6 + 1·7 + 1·8	20
0	0	1	1	0	0	0·3 + 0·4 + 1·5 + 1·6 + 0·7 + 0·8	11
0	0	1	1	0	1	0·3 + 0·4 + 1·5 + 1·6 + 0·7 + 1·8	19
0	0	1	1	1	0	0·3 + 0·4 + 1·5 + 1·6 + 1·7 + 0·8	18
0	0	1	1	1	1	0·3 + 0·4 + 1·5 + 1·6 + 1·7 + 1·8	26
0	1	0	0	0	0	0·3 + 1·4 + 0·5 + 0·6 + 0·7 + 0·8	4
0	1	0	0	0	1	0·3 + 1·4 + 0·5 + 0·6 + 0·7 + 1·8	12
0	1	0	0	1	0	0·3 + 1·4 + 0·5 + 0·6 + 1·7 + 0·8	11
0	1	0	0	1	1	0·3 + 1·4 + 0·5 + 0·6 + 1·7 + 1·8	19
0	1	0	1	0	0	0·3 + 1·4 + 0·5 + 1·6 + 0·7 + 0·8	10
0	1	0	1	0	1	0·3 + 1·4 + 0·5 + 1·6 + 0·7 + 1·8	18
0	1	0	1	1	0	0·3 + 1·4 + 0·5 + 1·6 + 1·7 + 0·8	17
0	1	0	1	1	1	0·3 + 1·4 + 0·5 + 1·6 + 1·7 + 1·8	25
0	1	1	0	0	0	0·3 + 1·4 + 1·5 + 0·6 + 0·7 + 0·8	9
0	1	1	0	0	1	0·3 + 1·4 + 1·5 + 0·6 + 0·7 + 1·8	17
0	1	1	0	1	0	0·3 + 1·4 + 1·5 + 0·6 + 1·7 + 0·8	16
0	1	1	0	1	1	0·3 + 1·4 + 1·5 + 0·6 + 1·7 + 1·8	24
0	1	1	1	0	0	0·3 + 1·4 + 1·5 + 1·6 + 0·7 + 0·8	15
0	1	1	1	0	1	0·3 + 1·4 + 1·5 + 1·6 + 0·7 + 1·8	23
0	1	1	1	1	0	0·3 + 1·4 + 1·5 + 1·6 + 1·7 + 0·8	22
0	1	1	1	1	1	0·3 + 1·4 + 1·5 + 1·6 + 1·7 + 1·8	30
1	0	0	0	0	0	1·3 + 0·4 + 0·5 + 0·6 + 0·7 + 0·8	3
1	0	0	0	0	1	1·3 + 0·4 + 0·5 + 0·6 + 0·7 + 1·8	11
1	0	0	0	1	0	1·3 + 0·4 + 0·5 + 0·6 + 1·7 + 0·8	10
1	0	0	0	1	1	1·3 + 0·4 + 0·5 + 0·6 + 1·7 + 1·8	18
1	0	0	1	0	0	1·3 + 0·4 + 0·5 + 1·6 + 0·7 + 0·8	9
1	0	0	1	0	1	1·3 + 0·4 + 0·5 + 1·6 + 0·7 + 1·8	17
1	0	0	1	1	0	1·3 + 0·4 + 0·5 + 1·6 + 1·7 + 0·8	16
1	0	0	1	1	1	1·3 + 0·4 + 0·5 + 1·6 + 1·7 + 1·8	24
1	0	1	0	0	0	1·3 + 0·4 + 1·5 + 0·6 + 0·7 + 0·8	8
1	0	1	0	0	1	1·3 + 0·4 + 1·5 + 0·6 + 0·7 + 1·8	16
1	0	1	0	1	0	1·3 + 0·4 + 1·5 + 0·6 + 1·7 + 0·8	15
1	0	1	0	1	1	1·3 + 0·4 + 1·5 + 0·6 + 1·7 + 1·8	23

Для кожного вихідного тексту існує єдиний унікальний криптотекст.

Задача про ранець відноситься до класу NP–повних задач [3], для неї немає поліноміального алгоритму, що вирішує її за розумний час. Тому при розв'язуванні задачі про ранець завжди потрібно вибирати між точними алгоритмами, які не застосовні для «великих» рюкзаків, і наближеними, які працюють швидко, але не забезпечують оптимального розв'язку задачі. Природно, створення швидкого й достатньо точного алгоритму становить великий інтерес.

4. Висновки

Наведені вище основні особливості застосування генетичних алгоритмів, які імітують процеси еволюції живої природи, свідчать про ефективність їх застосування для розв'язування задач криптології. «Сила» генетичного алгоритму полягає в його здатності оперувати одночасно багатьма параметрами, які використовується в сотнях прикладних програм. У деяких випадках потрібно знайти параметри, при яких досягається точне значення результату. В інших випадках точний оптимум не потрібний – рішенням може бути будь-яке значення, краще за певну задану величину. У цьому випадку генетичні алгоритми – найкращий метод для пошуку «прийнятних» рішень.

Список літератури: 1. *Holland J.H.* Adaptation in Natural and Artificial Systems / J.H. Holland. Ann Arbor, MI: The University of Michigan Press, 1975. 2nd edn. Boston, MA: MIT Press, 1992. **2.** *Merkle R.C.* Hiding Information and Signatures in Trapdoor Knapsacks / R.C. Merkle, M.E. Hellman // IEEE transactions on Information Theory. Sep 1978. V. 24. N. 5. P. 525–530. **3.** *Axo A.* Построение и анализ вычислительных алгоритмов / А. Ахо, Д. Хопкрофт, Д. Ульман. М.: Мир, 1979. С. 404–446. **4.** *Brickell E.* Solving Low Density Knapsacks / E. Brickell // Advances in Cryptology: Proceedings of crypto. New York: Plenum Press, 1984. P. 25–37. **5.** *Shamir A.* A Polynomial – Time Algorithm for Breaking the Basic Merkle – Hellman Cryptosystem / A. Shamir // Proceedings of the 23rd IEEE Symposium on the Foundations of Computer Science. 1982. P. 145–152. **6.** *Schneier B.* Applied Cryptography Second Edition: protocols, algorithms and source code in C. John Wiley & Sons Inc., 1996. (Русский перевод: Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.: ТРИУМФ, 2002). **7.** *Cryptanalysis* of knapsack ciphers using genetic algorithms / R. Spillman // Cryptologia. 1993. V. 17. N. 4. P. 367–377. **8.** *Matthews R.* The use of genetic algorithms in cryptanalysts / R. Matthews // Cryptologia. 1993. V. 17. N. 2. P. 187–201. **9.** *Use* of a Genetic Algorithm in the Cryptanalysis of Simple Substitution Ciphers / [R. Spillman, M. Janssen, B. Nelson, M. Kepner] // Cryptologia. 1993. V. 17. N. 1. P. 31–44.

Надійшла до редколегії 23.12.2015

Кожухівський Андрій Дмитрович, д-р техн. наук, професор кафедри інформатики та інформаційної безпеки Черкаського державного технологічного університету. Наукові інтереси: аналіз і моделювання складних систем. Адреса: Україна, 18006, Черкаси, бульвар Шевченка, 460, тел. 0472730217. E-mail: andrejdk@mail.ru

Намофілова Ольга Олексіївна, аспірантка кафедри інформатики та інформаційної безпеки Черкаського державного технологічного університету. Наукові інтереси: математичне моделювання, генетичні алгоритми. Адреса: Україна, 18006, Черкаси, бульвар Шевченка, 460, тел. 0472730217. E-mail: olga_namofilova@rambler.ru