УДК 003.26

АНАЛИЗ АЛГОРИТМА ПЧЕЛИНОЙ КОЛОНИИ И ЕГО ПРИМЕНЕНИЕ В КРИПТОГРАФИИ

Т. Э. Бахтигозин, М. Д. Пивоваров, О. А. Сафарьян
Донской государственный технический университет (г. Ростов-на-Дону, Российская Федерация)

В статье изучен и проанализирован один из биоинспирированных методов оптимизации — алгоритм пчелиной колонии, его задачи и этапы работы, особенности применения в криптографии. Представлена концепция собственного разрабатываемого программного продукта, предложен способ применения данного алгоритма в криптоанализе, позволяющий провести атаку на основе открытых текстов и соответствующих шифртекстов. Результатом данной статьи является анализ алгоритма пчелиной колонии и его применения в криптографии, а также созданный работающий алгоритм пчелиной колонии, который находит ключ для преобразования открытого текста в шифртекст на основе шифра перестановки.

***Ключевые слова*:** алгоритм, пчелиная колония, наемные пчелы, пчелы-фуражиры, пчелы-разведчики, криптоанализ, шифры перестановки.

ANALYSIS OF THE BEE COLONY ALGORITHM AND ITS APPLICATION IN CRYPTOGRAPHY

T. E. Bakhtigozin, M. D. Pivovarov, O. A. Safaryan
Don State Technical University (Rostov-on-Don, Russian Federation)

The article studies and analyzes one of the bioinspired optimization methods — the bee colony algorithm, its idea and stages of work, methods of application in cryptography. The concept of a proprietary software product being developed is presented, a method for using this algorithm in cryptanalysis is proposed, which allows an attack based on open texts and corresponding ciphertexts. The result of this article is an analysis of the bee colony algorithm and its application in cryptography, as well as a working bee colony algorithm that finds a key for converting plaintext into ciphertext based on a permutation cipher.

Keywords: algorithm, bee colony, hired bees, foraging bees, scout bees, cryptanalysis, permutation ciphers.

Введение. Поиск эффективных решений при наименьшей затрате ресурсов является актуальной задачей для современной науки. Деятельность, направленная на решение данной задачи, называется оптимизацией [1].

Методы оптимизации применяются на этапе проектирования разработки для установления входных характеристик и управляющих процедур, предоставляющих оптимальные качества какой-либо функции системы, решая задачи структурного и параметрического синтеза, а также минимизации расхождений расчетных и экспериментальных условий. В ходе решения данных задач определяются такие параметры, при которых итог вычислений будет соответствовать

требованиям: экономическим, технологическим, социальным и так далее (зависит от проектируемой системы).

До второй половины XX века методы оптимизации применялись нечасто, так как для этого требовалась большая вычислительная мощность. В наше время они решают задачи криптоанализа. Широкую популярность получили так называемые «природные алгоритмы».

Отличительная особенность таких методов криптоанализа — это возможность использования самого алгоритма шифрования, как целевой функции для оценки ключа, разнообразие которого должен обеспечивать биоинспирированный алгоритм [2].

Целью данной статьи является программная реализация алгоритма пчелиной колонии и его применение в криптографии.

Основная часть. Искусственная пчелиная колония — это метод оптимизации, который имитирует то, как ведут себя пчелы, специфическое применение кластерного интеллекта, главной отличительной чертой которого является то, что ему не обязательно понимать проблемы, главное — её оптимизировать [3].

Пчелы — социальные насекомые, которые могут добывать нектар с большой эффективностью в любом окружающем пространстве, и в то же время им не составляет труда подстроиться ко всем его изменениям [4].

Идея и работа алгоритма. Идея алгоритма состоит в следующем: из улея выдвигаются пчелы-разведчики для того, чтобы найти участки, на которых находится нектар. Спустя некоторое время пчелы-разведчики возвращаются в улей и дают информацию другим пчелам, в каком месте и, в каком количестве они нашли нектар. После этого на данные участки отправляются пчелы-фуражиры, а пчелы разведчики продолжают вести поиск других участков [5].

Таким образом, работа алгоритма зависит от следующих основных параметров:

– общее количество исследуемых участков;

– количество лучших участков;

– количество перспективных участков;

– количество пчёл на лучших участках;

– количество пчёл на перспективных участках;

– исходный размер участков;

– максимальное количество переборов.

Все пчёлы на каждом этапе выбирают не только элитные участки, но и участки, находящиеся в окрестности элитных. Это позволяет разнообразить количество решений в последующих итерациях и увеличить возможность нахождения, близкую к оптимальной [6].

Описание алгоритма представлено на рис. 1.

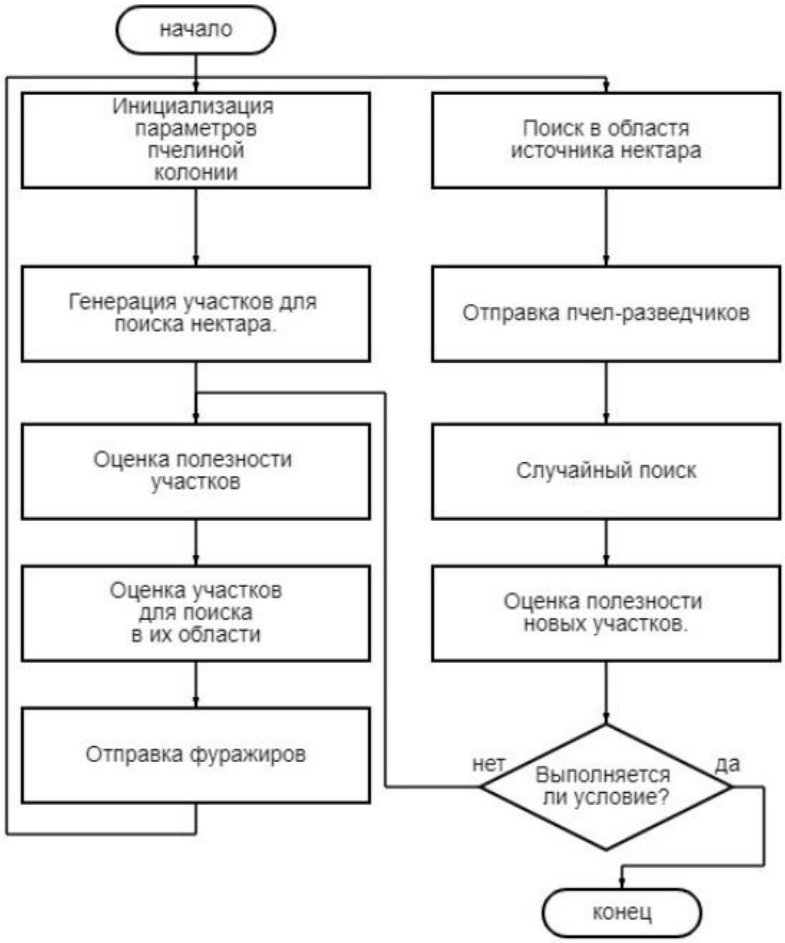

Рис. 1. Схема алгоритма пчелиной колонии

Программная реализация. Рассмотрим программную реализацию применения алгоритма пчелиной колонии для криптоанализа шифра «Магический квадрат». В ходе программной реализации использовалось функциональное программирование на языке программирования Python. В данной работе рассмотрено использование алгоритма пчелиной колонии в одном из методов криптоанализа — атаке на основе открытых текстов и соответствующих шифртекстов, задачей которой является поиск ключа.

Входные параметры:

– number_of_bees (общее количество исследуемых участков) = 10;

– best_area (количество лучших участков) = 2;

– other_area (количество перспективных участков) = 3;

– best_bees (количество пчёл на лучших участках) = 5;

– other_bees (количество пчёл на перспективных участках) = 2;

– size (исходный размер участков) = 16;

– iteration (максимальное количество переборов) = 25.

Для рассматриваемого метода криптоанализа необходимо знать открытый текст и соответствующий ему шифртекст.

Функция encrypt() шифрует открытый текст, а функция decrypt() — дешифрует. Результаты записываются в переменные «enc_text» и «dec_text» соответственно. Вывод работы функций представлен на рис. 2.

```
Enc. text: яерюеен мнаямстВ
Dec. text: Времена меняются
```

Рис. 2. Результат работы функций encrypt(), decrypt()

Функция gen_keys() генерирует возможные ключи (генерация участков для поиска нектара) на первой итерации. На второй и последующих итерациях для генерации ключей служит функция gen_keys_2(), входными параметрами которой являются лучшие и перспективные участки: из каждого лучшего участка получается пять новых участков (количество пчел на лучших участках), из каждого перспективного — два новых участка (количество пчел на перспективных участках). Результат их работы представлен на рис. 3.

```
+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+
|  3 |  8 |  6 |  2 |  9 |  7 |  1 | 13 | 12 | 11 |  4 |  5 | 15 | 14 | 10 | 16 |
|  8 |  3 | 13 |  5 | 11 | 16 |  4 | 15 |  9 |  6 | 14 | 10 |  2 |  7 |  1 | 12 |
|  7 | 14 | 12 |  1 | 16 | 15 |  2 | 13 |  4 |  3 |  6 | 10 |  8 |  9 | 11 |  5 |
|  2 |  3 | 10 | 16 | 12 |  7 |  1 |  9 | 13 |  4 |  6 | 15 |  8 | 11 | 14 |  5 |
|  9 |  6 | 16 | 12 | 13 | 14 |  5 |  8 | 10 |  2 |  7 | 11 | 15 |  1 |  3 |  4 |
|  8 |  7 | 16 |  3 | 13 | 10 | 14 |  4 | 11 |  6 | 12 |  1 |  9 |  5 |  2 | 15 |
|  3 |  4 |  1 |  7 | 14 |  9 | 15 |  8 |  6 | 16 |  5 |  2 | 13 | 10 | 12 | 11 |
| 14 |  3 |  5 |  4 |  8 |  9 | 10 | 15 |  2 | 13 | 16 | 11 |  6 |  1 | 12 |  7 |
|  1 | 11 |  8 | 10 |  4 |  6 |  2 | 12 |  5 |  7 | 13 | 16 | 14 |  3 | 15 |  9 |
|  5 |  7 |  4 | 14 | 11 |  6 | 10 | 12 |  8 |  9 |  2 | 13 | 16 |  1 |  3 | 15 |
+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+
```

Рис. 3. Генерация участков для поиска нектара

Функция area() выполняет дешифрование шифртекста, используя в качестве ключа найденные участки, сравнивает количество совпадающих символов дешифрованного шифртекста и открытого текста, на основе этого даёт оценку полезности участка. Функция remove_mismatches() заменяет символы, которые дали неверный результат на 0. В результате работы этих двух функций мы знаем, какие участки являются лучшими, а какие перспективными. Вывод представлен на рис. 4, 5.

Найденные участки:

Кол-во совпадений	Ключи	Dec. text
1	[3, 8, 6, 2, 9, 7, 1, 13, 12, 11, 4, 5, 15, 14, 10, 16]	нюяаяреееетнм смВ
3	[8, 3, 13, 5, 11, 16, 4, 15, 9, 6, 14, 10, 2, 7, 1, 12]	тменюнсямяеВра е
1	[7, 14, 12, 1, 16, 15, 2, 13, 4, 3, 6, 10, 8, 9, 11, 5]	юннмВаямсятр еее
2	[2, 3, 10, 16, 12, 7, 1, 9, 13, 4, 6, 15, 8, 11, 14, 5]	няенВаем рсемтяю
2	[9, 6, 16, 12, 13, 14, 5, 8, 10, 2, 7, 11, 15, 1, 3, 4]	снтВнеа ямяюеемр
3	[8, 7, 16, 3, 13, 10, 14, 4, 11, 6, 12, 1, 9, 5, 2, 15]	ятю снеямемаенВр
1	[3, 4, 1, 7, 14, 9, 15, 8, 6, 16, 5, 2, 13, 10, 12, 11]	ряяеамю есВтменн
1	[14, 3, 5, 4, 8, 9, 10, 15, 2, 13, 16, 11, 6, 1, 12, 7]	смеюрмВеенятня а
1	[1, 11, 8, 10, 4, 6, 2, 12, 5, 7, 13, 16, 14, 3, 15, 9]	янсеменрВюе амтя
0	[5, 7, 4, 14, 11, 6, 10, 12, 8, 9, 2, 13, 16, 1, 3, 15]	сатряеемнне яюВм

Рис. 4. Найденные участки

Лучшие участки:

Кол-во совпадений	Ключи	Dec. text
4	[0, 0, 0, 2, 8, 0, 0, 0, 0, 0, 15, 0, 0, 0, 0, 14]	тюсмеяре ннемВая
3	[0, 0, 0, 1, 0, 0, 0, 8, 0, 0, 0, 12, 0, 0, 0, 0]	ютВммер няеяснеа

Перспективные участки:

Кол-во совпадений	Ключи	Dec. text
3	[0, 0, 0, 0, 10, 0, 0, 0, 0, 11, 6, 0, 0, 0, 0, 0]	мянВеаяеюенмср т
3	[0, 14, 0, 0, 0, 0, 0, 0, 0, 11, 6, 0, 0, 0, 0, 0]	ярм санмюенВееят
3	[12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 6, 9, 0, 0, 0, 0]	Всмера няюнямеет

Рис. 5. Лучшие и перспективные участки

Алгоритм повторяется заданное количество раз (максимальное количество переборов). Как правило, пятидесяти итераций достаточно, чтобы получить ключ размером 16 символов.

На рис. 6 видно, что декодированный текст при помощи найденного пчёлами ключа совпадает с открытым текстом и количество совпадений равно длине ключа. Это свидетельствует о том, что ключ найден.

Лучшие участки:

Кол-во совпадений	Ключи	Dec. text
16	[16, 3, 2, 13, 5, 10, 11, 8, 9, 6, 7, 12, 4, 15, 14, 1]	Времена меняются
16	[16, 3, 2, 13, 5, 10, 11, 8, 9, 6, 7, 12, 4, 15, 14, 1]	Времена меняются

Перспективные участки:

Кол-во совпадений	Ключи	Dec. text
16	[16, 3, 2, 13, 5, 10, 11, 8, 9, 6, 7, 12, 4, 15, 14, 1]	Времена меняются
16	[16, 3, 2, 13, 5, 10, 11, 8, 9, 6, 7, 12, 4, 15, 14, 1]	Времена меняются
16	[16, 3, 2, 13, 5, 10, 11, 8, 9, 6, 7, 12, 4, 15, 14, 1]	Времена меняются

Рис. 6. Результат работы программы

Заключение. В результате проделанной работы был проанализирован один из биоинспирированных методов оптимизации — алгоритм пчелиной колонии, его применение в криптографии, а также реализовано собственное программное средство, позволяющее проводить атаку на основе открытого текста и соответствующего шифртекста. Данный алгоритм можно использовать для нахождения значений ключей в шифрах перестановки, обладая знаниями об открытом тексте и соответствующем шифртексте.

Библиографический список

1. Бабенко, Л. К. Современные алгоритмы блочного шифрования и методы их анализа / Л. К. Бабенко, Е. А. Ищукова. — Москва : Гелиос АРВ, 2006. — 376 с.

2. Сергеев, А. С. Разработка генетического метода криптоанализа блочных криптосистем и исследование возможности их параллельной реализации в системах защиты информации на примере стандарта DES / А. С. Сергеев // Системный анализ в проектировании и управлении: тр. 10 междунар. науч.-практ. конф. — Санкт-Петербург, 2006. — С. 258–265.

3. Морозенко, В. В. Генетический алгоритм для криптоанализа шифра Вижинера / В. В. Морозенко, Г. О. Елисеев // Вестник Пермского государственного университета. Серия: Математика. Механика. Информатика. — 2010. — № 1. — С. 75– 80.

4. Исследование возможности применения бионических методов пчелиных колоний для реализации криптоанализа классических шифров перестановок / Ю. О. Чернышев, А. С. Сергеев, Е. О. Дубров, А. Н. Рязанов // Вестник Донского государственного технического университета. — 2014. — Т. 14, № 1(76). — С. 62–75.

5. Криптографические методы и генетические алгоритмы решения задач криптоанализа / Ю. О. Чернышев, А. С. Сергеев, Е. О. Дубров, О. П. Третьяков. — Краснодар: ФВАС, 2013. — 138 с.

6. Курейчик, В. В. Концепция эволюционных вычислений, инспирированных природными системами / В. В. Курейчик, В. М. Курейчик, С. И. Родзин. // Известия ЮФУ. Технические науки. — 2009. — № 4 (93). — С. 16—24.

Об авторах:

Бахтигозин Тамирлан Энверович, студент кафедры «Информатика и вычислительная техника» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), bakhtigozin76@gmail.com

Пивоваров Максим Дмитриевич, студент кафедры «Информатика и вычислительная техника» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), pivowarof@mail.ru

Сафарьян Ольга Александровна, доцент кафедры «Компьютерная безопасность» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), кандидат технических наук, доцент, ORCID, safari_2006@mail.ru

About the Authors:

Bakhtigozin, Tamirlan E., Student, Department of Informatics and Computer Engineering, Don State Technical University (1 Gagarin Square, Rostov-on-Don, 344003, RF), bakhtigozin76@gmail.com

Pivovarov, Maksim D., Student, Department of Informatics and Computer Engineering, Don State Technical University (1 Gagarin Square, Rostov-on-Don, 344003, RF), pivowarof@mail.ru

Safaryan, Olga A., Associate professor, Department of Computer Security, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), Cand.Sci. (Eng.), associate professor, ORCID, safari_2006@mail.ru