Инженерный вестник Дона, №2 (2016)
ivdon.ru/ru/magazine/archive/n2y2016/3621

Применение алгоритмов пчелиных колоний для реализации криптоанализа блочных методов шифрования

А.С.Сергеев[1], *А.Н.Рязанов*[2], *Е.О. Дубров*[3]

[1]*Донской государственный технический университет, Ростов-на-Дону*
[2] *Открытое акционерное общество «711 Военпроект», г Ростов-на-Дону*
[3] *Федеральное государственное унитарное предприятие «Ростовский НИИ радиосвязи»*

Аннотация: Рассматривается возможность применения алгоритмов пчелиных колоний для реализации криптоанализа блочных шифров в предположении, что наличие информации об исходном тексте и шифртексте обеспечивает определение секретного ключа, и реализация алгоритма шифрования сводится к реализации операций шифров перестановок. Для решения данной оптимизационной задачи применяются известные методы пчелиных колоний, относящихся к сравнительно новому классу биоинспирированных оптимизационных методов, имитирующих процессы, протекающие в живой природе. Показано, что задача криптоанализа может быть интерпретирована как задача о назначениях и может быть решена с помощью алгоритма пчелиных колоний, в поведении которых основной является самоорганизация, благодаря чему имеет место достижение общей цели всего роя. Приведены: алгоритм поиска оптимальной комбинации символов с максимальным значением целевой функции, а также формула для определения значения целевой функции. Приведена структурная схема основных этапов алгоритма пчелиных колоний (формирование множества областей–источников, а также исследование с помощью рабочих пчел-фуражиров окрестностей данных областей), также приведен пример работы алгоритма.
Ключевые слова: криптоанализ, биоинспирированные методы, блочное шифрование, пчелы-фуражиры, пчелы-разведчики, секретный ключ.

Введение

Известно, что научное направление «биоинспирированные методы и алгоритмы» в последние годы получает все более широкое распространение для решения широкого круга оптимизационных задач, к которым относятся и задачи криптоанализа. В данных методах и оптимизационных моделях основным моментом является построение начальной структуры и определение совокупности правил, по которым она должна изменяться и оптимизироваться [1,3]. В течение последних лет были предложены различные схемы биоинспирированных вычислений, среди последних разработок эвристических методов, используемых для решения задачи параметрической оптимизации технических объектов, можно

отметить стохастический алгоритм, основанный на модели поведения роя светлячков, рассмотренный в [2]. В этой связи можно отметить также новые подходы, посвященные применению биоинспирированных методов (метода «роя частиц») для решения задачи моделирования распределительных процессов [19], а также генетических алгоритмов для решения нелинейной транспортной и распределительной задач [20]. Следует отметить, что в [20] авторами сделаны выводы в пользу использования генетических алгоритмов, основанных на эволюционном моделировании по сравнению с существующими методами. В [4,5] авторами рассматривались методы решения задачи криптоанализа, относящейся к переборным задачам с экспоненциальной временной сложностью, на классические симметричные и асимметричные криптосистемы на основе биоинспирированных методов, а в [6,7] также и на блочные криптосистемы. Поскольку данные задачи криптоанализа в большинстве случаев являются NP-полными и имеют комбинаторную сложность, то основным мотивом для разработок новых алгоритмов решения комбинаторных задач являются возникшие потребности в решении задач большой размерности [8].

Тем не менее, как отмечено в [10], недостатком методов эволюционной оптимизации является наличие «слепого» поиска, что в общем случае может привести к значительным временным затратам, формированию большого количества одинаковых и плохих решений и попаданию в локальный оптимум. Одной из последних разработок роевого интеллекта является алгоритм колонии пчел, используемый для нахождения глобальных экстремумов функций [8, 14].

Как отмечено в [9, 11], первые публикации, посвященные алгоритмам колонии пчел для определения экстремумов функций, относятся к 2005 году ([12,13]). Описание данного алгоритма приводится в [11, 15, 16, 18]. Отметим, что в [9] приводится обзор некоторых публикаций, посвященных

применению алгоритмов пчелиных колоний для решения комбинаторных задач (теоретико-графовые задачи, сравнение с другими «природными» методами), решению задачи размещения, задачи разложения составных чисел на простые сомножители, используемой при криптоанализе асимметричных алгоритмов.

Постановка задачи

В настоящее время задача криптоанализа блочных криптосистем является, несомненно, актуальной, так как переход к блочному шифрованию создает новые дополнительные возможности для повышения стойкости криптоалгоритмов. Ранее в [6,7] рассматривались возможные методы реализации криптоанализа блочных криптосистем с использованием методов генетического поиска, а в [17] - с использованием алгоритма муравьиных колоний путем сведения данной проблемы к задаче о назначениях. Отметим, что отличительные особенности алгоритмов муравьиных и пчелиных колоний описаны в [5]. В данной работе рассматривается подход для реализации криптоанализа блочных алгоритмов шифрования с помощью сведения данной проблемы к задаче нахождения экстремума немонотонной функции, решаемой с помощью алгоритма пчелиного роя.

Отметим, что в [5,9,17] приведено описание оптимизационной модели, применяемой для решения задачи криптоанализа. Данная модель использует параметры: C_{ij} — вероятность того, что символ в позиции i+1 должен следовать за символом в позиции i; Q_i, показывающий осмысленность фрагмента текста из i символов, то есть его совпадение со словарным запасом. В соответствии с [5,9,17] оптимизационная модель имеет вид:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} Q_i C_{ij} X_{ij} \to \max$$

Элементы C_{ij} могут быть заданы в виде матрицы размера nxn, где n — число символов текста.

Вычисление на каждой итерации значения Q – оценки оптимальности фрагмента текста, получаемого при криптоанализе с использованием каждого варианта сформированного ключа - является основным моментом при реализации описанного алгоритма. В [4,5,17] для решения данной проблемы (оценки оптимальности фрагмента текста, получаемого при криптоанализе) описано применение целевой функции Якобсена. Данная функция использует информацию о распределении частот биграмм в исходных текстах и представляет собой сумму разностей по модулю между среднестатистическим количеством биграмм и их реальным количеством в тексте.

Таким образом, задача криптоанализа является комбинаторной задачей, при этом целью поиска является определение варианта ключа, обеспечивающего получение исходного текста (либо, в ряде случаев, вариантов текста, обеспечивающих получение секретного ключа) с максимальным значением целевой функции R (аналогично [17]).

Описание пчелиного алгоритма

При описании алгоритма криптоанализа воспользуемся методами и терминологией, используемыми в [8, 9, 14]. Следует заметить, что процесс криптоанализа может быть реализован аналогично [9], при этом целевая функция R определяется для каждого варианта текста, сформированного на каждом шаге с помощью алгоритма пчелиных колоний. Таким образом, ключ, обеспечивающий получение исходного текста с максимальным значением функции R, является искомым.

Как отмечено в [8,9], основу поведения пчелиного роя составляет двухуровневая стратегия поиска, при которой обеспечивается достижение общих целей роя. Множество перспективных областей–источников

формируется с помощью пчел-разведчиков на первом этапе, исследование окрестностей данных областей производится с помощью пчел-фуражиров на втором этапе. Поиск источника с максимальным количеством нектара является основной целью всей колонии.

Как и ранее в [9], будем предполагать, что каждое решение является позицией в пространстве поиска, которая содержит определенное количество нектара. При этом значение целевой функции в данной точке определяется данным количеством нектара. Решение задачи криптоанализа представляет собой последовательность символов алфавита $x_1, x_2, ..., x_k$, пройденных агентом–пчелой в пространстве поиска (вариант ключа). Целью поиска, таким образом, является определение оптимальной комбинации (последовательности прохождения) символов, соответствующих оптимальному варианту текста, с максимальным значением R.

Таким образом, процесс поиска решений состоит в передвижении агентов–пчел в пространстве поиска, формировании соответствующих вариантов текста с последующей проверкой их оптимальности, а также выборе соответствующего оптимального (или квазиоптимального) варианта ключа.

В соответствии с [8,9,14,18] основными операциями алгоритма колонии пчел являются следующие.

1. Формирование пространства поиска и популяции пчел.
2. Оценка целевой функции (ЦФ) пчел путем определения ЦФ, определяющей оптимальность исходного текста.
3. Формирование перспективных участков для поиска.
4. Отправка пчел-разведчиков, а также поиск агентами-разведчиками перспективных позиций.
5. Выбор пчел, имеющих лучшие значения ЦФ на каждом участке.

6. Отправка пчел-фуражиров для случайного поиска, а также оценка их ЦФ.

7. Создание новой пчелиной популяции.

8. Проверка условий остановки алгоритма. Если они выполняются, переход к 9, иначе к 2.

9. Конец работы алгоритма.

Отметим, что структурная схема алгоритма колонии пчел для организации поисковых процедур, а также оценки временной сложности алгоритма пчелиных колоний приведены в [14]. В соответствии с [14] в лучшем случае временная сложность пчелиных алгоритмов T составляет $T \approx O(n^{\lg n})$, в худшем случае $T \approx O(n^3)$.

Рассмотрим описание данного алгоритма для реализации криптоанализа, при котором на основе заданного блока шифртекста необходимо определить блок исходного текста, а также секретный ключ. Как и ранее в [17], будем использовать допущения, что каждый бит шифртекста зависит от каждого бита исходного текста и каждого бита ключа и что шифртект и исходный текст определяются символами из одного и того же алфавита (то есть применение секретного ключа осуществляет реализацию шифров перестановок). В этом случае, определяя с помощью алгоритма пчел исходный текст (аналогично [9]), можно, очевидным образом, определить соответствующий секретный ключ.

На первом этапе пчелиного алгоритма осуществляется формирование пространства поиска. Будем далее предполагать, что каждая позиция a_s представляет собой размещенный элемент алфавита текста, и при этом каждая пчела–агент содержит в памяти последовательный список $E_s=\{e_{si}, i=1,2,\ldots,n\}$ посещенных символов. Этот список E_s, поставленный в соответствие каждому символу, посещенному пчелой, в пространстве поиска,

фактически представляет решение — текст, для которого могут быть определены секретный ключ и ЦФ.

Следующим этапом пчелиного алгоритма является формирование перспективных участков и поиск в их окрестности. Как и в [9], будем далее предполагать, что пространство, в котором размещено m символов алфавита шифртекста, является квадратной матрицей A размером mxm. Для каждой позиции a_s определена окрестность размера λ для поиска (множество позиций a_{si} на расстоянии, не превышающем λ, от позиции a_s)

В соответствии с [18] алгоритм колонии пчел можно описать следующим образом. На начальном этапе работы алгоритма N пчёл располагаются случайно на m участках. ЦФ участков определяются на следующем шаге. Участки с большими значениями ЦФ (элитные участки) выбираются для поиска решений в их окрестностях, и на эти участки отправляется большее количество пчёл. На следующем шаге проводится оценка ЦФ и выбираются лучшие пчёлы (в соответствии со значениями ЦФ участков, исследуемых ими). Из этих пчёл формируется новая популяция решений, которая используется в следующей итерации алгоритма. Далее с помощью пчёл-фуражиров осуществляется случайный поиск в окрестностях элитных участков для поиска новых решений. Данная последовательность операций продолжается, пока не будет выполнено условие остановки алгоритма.

Применительно к решению задачи криптоанализа этапы данного алгоритма реализуются в следующей форме. Начальными параметрами алгоритма являются: общее количество пчел–агентов N, количество итераций L, количество агентов–разведчиков n_r, количество агентов–фуражиров n_f, значение максимального размера окрестности для поиска $\lambda_{макс}$.

На l=1 итерации алгоритма производится размещение случайным образом n_r пчел-разведчиков в пространстве поиска (осуществляется выбор произвольным образом n_r символов матрицы A). На начальном этапе значение ЦФ R полагается равным малому положительному числу.

Далее в соответствии с [8] выбирается n_b лучших (базовых) решений, имеющих значения ЦФ R не хуже, чем у любого другого решения. На начальной итерации это может быть осуществлено произвольным образом. В пространстве поиска осуществляется формирование множества базовых позиций $A_b=\{a_{bi}\}$, которые соответствуют базовым решениям.

Далее заданное число рабочих пчел (фуражиров), имитирующих поиск нектара, направляется в окрестности всех базовых позиций в соответствии с методикой, описанной в [9].

После выбора рабочей пчелой (фуражиром) n_{fi} базовой позиции a_i осуществляется выбор случайным образом позиции a_s в окрестности базовой позиции a_i. При этом значение окрестности λ определяется в границах $1\leq\lambda\leq\lambda_{макс}$ случайным образом.

Таким образом, каждой пчеле–агенту можно поставить в соответствие список E_s символов пространства поиска с ЦФ, определенной для этого списка, и данная последовательность может быть поставлена в соответствие последней посещенной пчелой–агентом позиции.

Аналогично [8,9] определим область D_i, представляющую собой $D_i=a_i \bigcup O_i$, где O_i — множество позиций, выбранных рабочими пчелами (фуражирами) в окрестности позиции a_i. В каждой области D_i определяется оценка области - позиция a^* с лучшим значением ЦФ R_i^*. Из всех оценок областей R_i^* определяется лучшая оценка R_i* и соответствующее решение (список E_s). Далее определяется вариант текста с лучшим значением ЦФ, и производится переход к следующей итерации.

На последующих итерациях алгоритма на поиск новых позиций отправляется множество n_{rl} разведчиков ($n_{rl}<n_r$). При этом множество базовых позиций $A_b(l)$ содержит две части $A_{b1}(l)$ и $A_{b2}(l)$, часть $A_{b1}(l)$ содержит n_{b1} лучших решений a^*, найденных в каждой из областей на итерации l-1, часть $A_{b2}(l)$ содержит n_{b2} лучших решений из n_{rl} позиций, найденных пчелами–разведчиками на итерации l.

Таким образом, $n_{b1}+n_{b1}=n_b$. Далее, как и на первой итерации, определяется число агентов–фуражиров, которые должны быть отправлены в окрестности базовых позиций. Каждым агентом–фуражиром n_{fi} выбираются позиции: базовая позиция $a_i(l)$, и позиция $a_s(l)$ в ее окрестности. Далее определяются области $D_i(l)$. Лучшая позиция a_i^* с лучшей оценкой ЦФ R_i^* выбирается в каждой области, далее среди оценок R_i^* выбирается лучшая оценка R^*. Если $R^*(l)$ оптимальней, чем $R^*(l-1)$, то запоминается соответствующее решение, и происходит переход к следующей итерации.

Отметим, что пошаговое описание данного алгоритма приведено в [9]. Структурная схема данного алгоритма криптоанализа представлена на рис. 1.

Демонстрационный пример

Приведем пример реализации представленного алгоритма криптоанализа, в котором на основе блока шифрованного текста определяется блок исходного текста с помощью пчелиного алгоритма (аналогично примеру, приведенному в [9, 17]), и на их основе соответствующий секретный ключ. Пусть задан блок шифртекста, представляющий строку символов русского алфавита **ИОСАКБ.** Требуется определить строку символов исходного текста и секретный ключ при отмеченных выше допущениях (отметим, что в [17] было показано, как аналогичная задача может быть решена с использованием

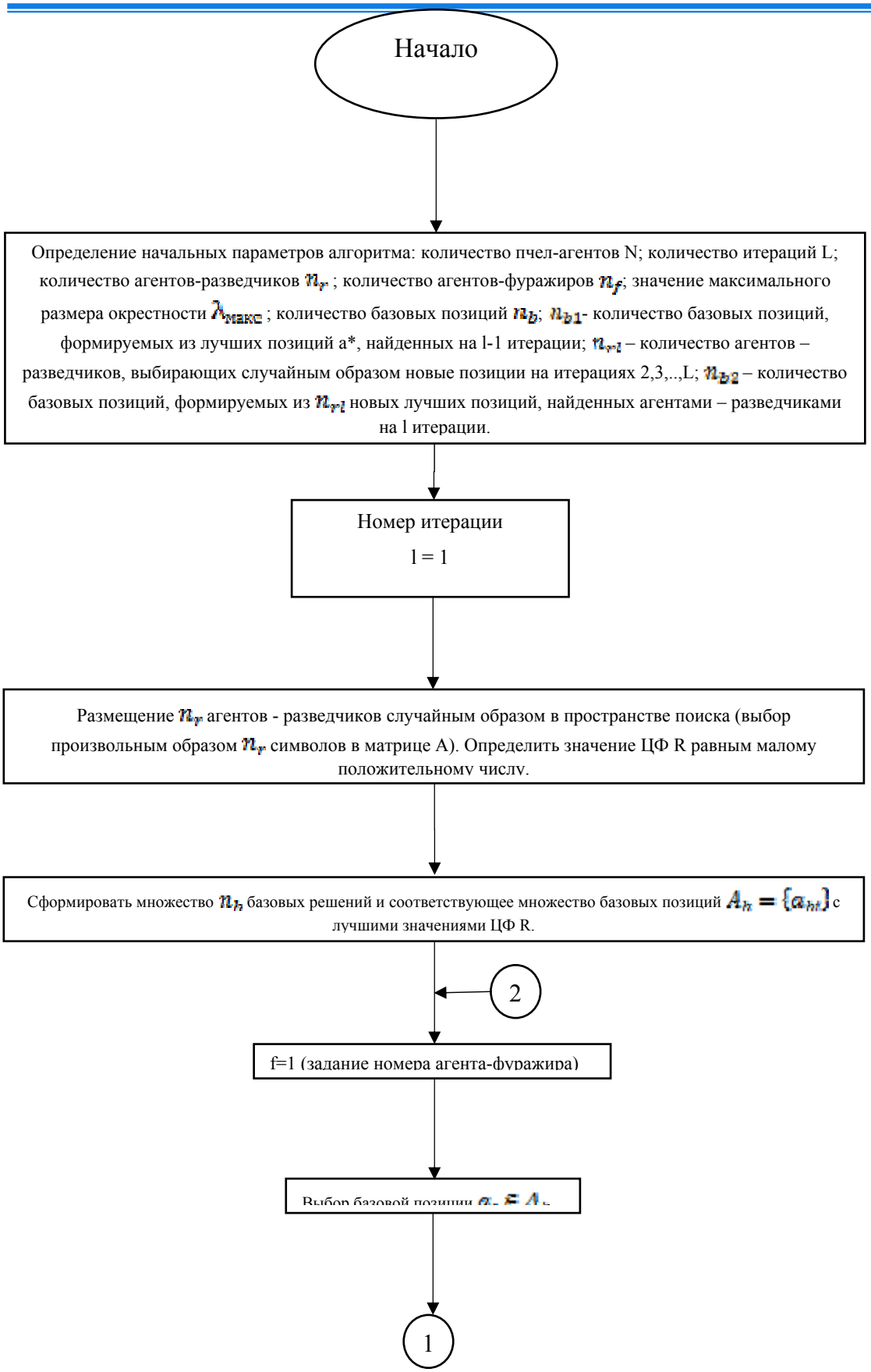

Рис. 1. Структурная схема криптоанализа на основе алгоритма колонии пчел.

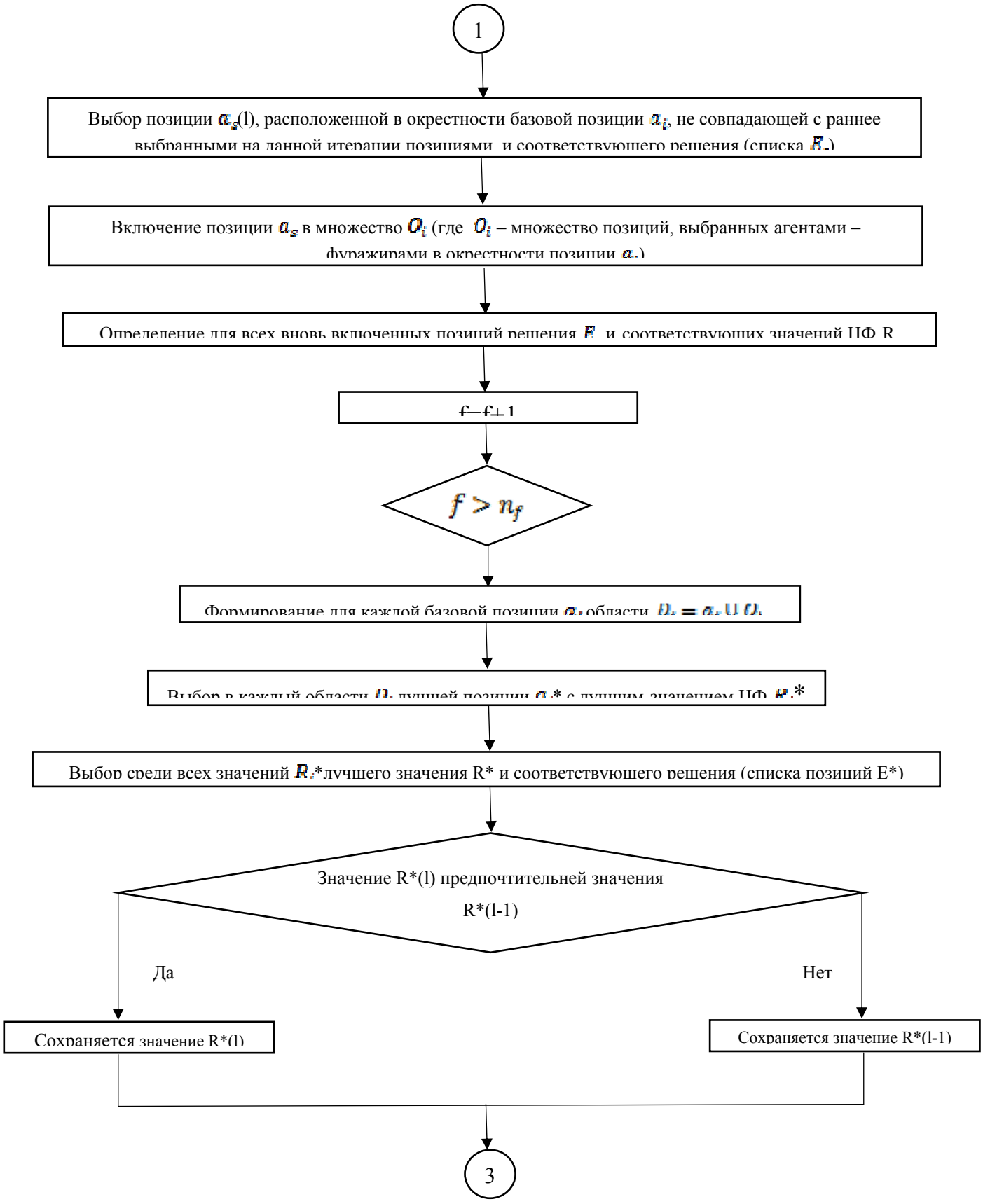

Рис. 1 (продолжение)

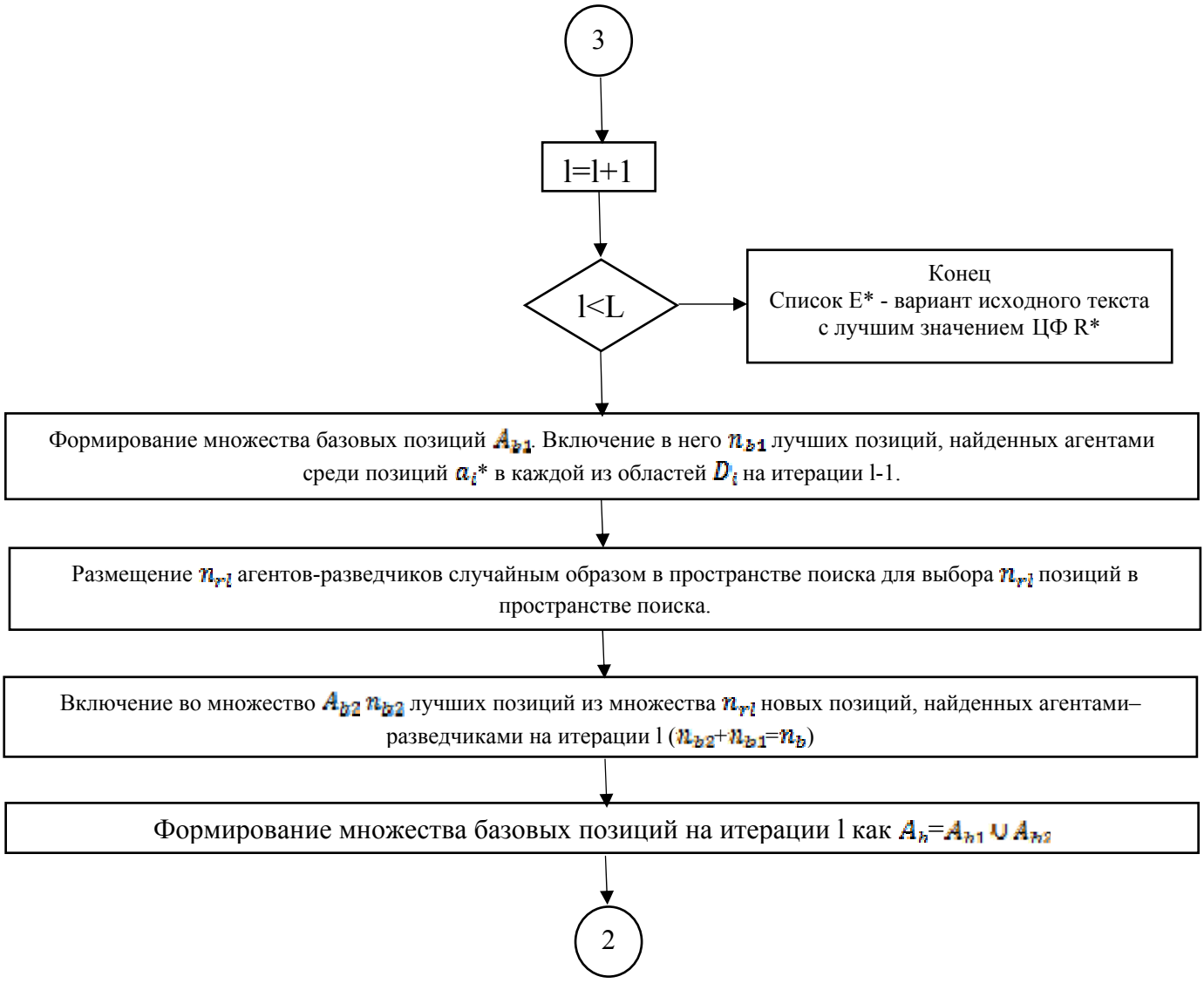

Рис. 1 (окончание)

алгоритма муравьиных колоний). Введем матрицу C вероятности появления биграмм, приведенную в [17] и на рис. 2.

Пространство поиска определим аналогично [9] как матрицу A размером 11x11, содержащую символы из алфавита шифртекста, размещенные случайным образом (рис. 3).

	Б	О	С	К	А	И
Б	0,01	0,5	0,1	0,01	0,6	0,6
О	0,6	0,02	0,5	0,3	0,1	0,1
С	0,05	0,6	0,05	0,08	0,3	0,3
К	0,01	0,5	0,01	0,01	0,4	0,4
А	0,6	0,1	0,6	0,6	0,01	0,01
И	0,6	0,1	0,6	0,6	0,01	0,01

Рис. 2. Матрица С, элемент C_{ij} которой определяет вероятность соседства в тексте символов i и j

11	О	А	О	А	С	О	О	К	А	Б	О
10	И	Б	О	А	Б	И	С	А	О	К	И
9	А	И	С	С	И	О	К	А	К	О	Б
8	К	А	С	К	И	***Б***	А	О	К	И	А
7	С	Б	Б	А	А	К	И	С	И	А	Б
6	И	Б	О	К	И	А	О	А	К	Б	К
5	Б	А	Б	К	О	А	И	Б	С	И	Б
4	И	Б	А	С	А	А	И	А	О	И	С
3	И	А	С	Б	***К***	Б	Б	К	Б	***О***	Б
2	О	К	О	К	А	***С***	И	О	Б	С	А
1	Б	К	Б	О	И	К	А	Б	С	А	И
	1	2	3	4	5	6	7	8	9	10	11

Рис. 3. Матрица А - пространство поиска для пчелиного алгоритма.

Итерация 1.

1. Выберем количество агентов–разведчиков n_r=5 и зададим их размещение случайным образом в пространстве поиска (выберем произвольные n_r символов в матрице А). Пусть это будут символы К(5,3), Б(6,8), С(6,2), О(10,3), К(11,6). Определим значение ЦФ R для всех позиций как малое положительное число R=0,001.

2. Произвольным образом определим множество базовых решений n_b=4 и базовые позиции с лучшими значениями ЦФ. Пусть это будут позиции A_b ={К(5,3), Б(6,8), С(6,2), О(10,3)}. На рис. 3 они выделены курсивом

3. Определим число агентов–фуражиров n_f=5 и размер окрестности $\lambda_{макс}$=3. Пусть порядок выбора базовых позиций следующий: С, О, К, Б, С, и им соответствуют следующие позиции a_s: С→К(4,2); О→Б(9,3); К→С(4,4); Б→А(5,7); С→А(6,4). Таким образом, мы получим на данном этапе следующий список позиций, решений и значений ЦФ: позиции К(5,3), Б(6,8), С(6,2), О(10,3), R=0,001, список Е содержит один символ; позиция К(4,2), Е={СК}, R=0,08; позиция Б(9,3), Е={ОБ}, R=0,6; позиция С(4,4), Е={КС}, R=0,01; позиция А(5,7), Е={БА}, R=0,6; позиция А(6,4), Е={СА}, R=0,3. Области D_i имеют следующий вид: D_1={С(6,2), К(4,2), А(6,4)}; D_2={О(10,3), Б(9,3)}; D_3={К(5,3), С(4,4)}; D_4={Б(6,8), А(5,7)}.

4. В каждой области D_i выбирается лучшая позиция a_i^* с оптимальным значением ЦФ R_i^*. Таким образом, D_1→А(6,4), R_1^*=0,3; D_2→Б(9,3), R_2^*=0,6; D_3→С(4,4), R_3^*=0,01; D_4→А(5,7), R_4^*=0,6.

5. Выбрав среди всех значений R_i^* лучшие значения, получим, что $R^*(2)$=0,6; $R^*(4)$=0,6; $E^*(2)$={ОБ}; $E^*(4)$={БА}.

6. l=2.

Итерация 2.

1. Зададим число базовых позиций n_{b1}=2. Множество A_{b1} будет включать n_{b1} лучших позиций среди позиций a_i^*, определенных в каждой из

областей D_i на итерации 1. Следовательно, A_{b1}={Б(9,3), А(5,7)}. Списки, поставленные в соответствие данным позициям, показаны на рис. 4.

2. Как и на предыдущей итерации, определим количество агентов-разведчиков n_r=5 и разместим их в произвольных позициях С(3,3), И(1,3), К(4,5), О(10,9), А(8,10).

3. Включим в множество A_{b2} n_{b2}=2 оптимальных позиций из множества n_{rl} новых позиций, определенных агентами–разведчиками на итерации 2. Пусть A_{b2}=={И(1,3), К(4,5)}. Получим A_b={Б(9,3), А(5,7), И(1,3), К(4,5)}.

11	О	А	О	А	С	О	О	К	А	Б	О
10	И	Б	О	А	Б	И	С	А	О	К	И
9	А	И	С	С	И	О	К	А	К	О	Б
8	К	А	С	К	И	Б	А	О	К	И	А
7	С	Б	Б	А	***БА***	К	И	С	И	А	Б
6	И	Б	О	К	И	А	О	А	К	Б	К
5	Б	А	Б	К	О	А	И	Б	С	И	Б
4	И	Б	А	***КС***	А	***СА***	И	А	О	И	С
3	И	А	С	Б	К	Б	Б	К	***ОБ***	О	Б
2	О	К	О	***СК***	А	С	И	О	Б	С	А
1	Б	К	Б	О	И	К	А	Б	С	А	И
	1	2	3	4	5	6	7	8	9	10	11

Рис. 4. Матрица А - пространство поиска
для пчелиного алгоритма после 1 итерации.

4. Как и ранее, определим число агентов–фуражиров n_f=5 и размер окрестности $\lambda_{макс}$=3. Предположим, что выбор базовых позиций осуществляется в последовательности Б, А, К, Б, И, и в их окрестности им ставятся в соответствие следующие позиции: Б(9,3)→А(2,3); А(5,7)→К(4,8); К(4,5)→И(7,5); Б(9,3)→О(8,2); И(1,3)→Б(2,4). На данной итерации будет сформирован следующий список позиций, решений и значений ЦФ: позиция Б(9,3), E={ОБ}, R=0,6; позиция А(5,7), E={БА}, R=0,6; позиции К(4,5), И(1,3), R=0,001, список E состоит из одного символа; позиция А(2,3), E={ОБА}, R=1,2; позиция К(4,8), E={БАК}, R=1,2; позиция И(7,5), E={КИ}, R=0,4; позиция О(8,2), E={ОБО}, R=1,1; позиция Б(2,4), E={ИБ}, R=0,6. Как и ранее в [9,17], фрагменты текста, содержащие три и более символа, умножим на значения Q_i, определяемые в зависимости от частоты встречаемости. Для фрагментов текста ОБА, ОБО, БАК определим Q=0,9, Q=0,6, Q=0,9. Таким образом, для позиции А(2,3) R=1,08; для позиции О(8,2) R=0,66, для позиции К(4,8) R=1,08. Области D_i определятся следующим образом: D_1={Б(9,3), А(2,3), О(8,2)}; D_2={А(5,7), К(4,8)}; D_3={К(4,5), И(7,5)}; D_4={И(1,3), Б(2,4)}.

5. Как и на предыдущей итерации, в каждой области D_i выбирается лучшая позиция a_i^* с оптимальным значением ЦФ. В этом случае получим: D_1→А(2,3), R_1^*=1,08; D_2→К(4,8), R_2^*=1,08; D_3→И(7,5), R_3^*=0,4; D_4→Б(2,4), R_4^*=0,6.

6. Получим, что на данной итерации $R^*(1)$=1,08; $R^*(2)$=1,08; $E^*(1)$={ОБА}; $E^*(2)$={БАК}.

7. l=3.

Итерация 3.

1. Как и ранее, число базовых позиций определим n_{b1}=2. Множество A_{b1} будет содержать n_{b1} оптимальных позиций, найденных

агентами среди позиций a_i^* в каждой из областей D_i на итерации 2. Таким образом, A_{b1}={А(2,3), К(4,8)}. Списки, поставленные в соответствие данным позициям, показаны на рис. 5.

2. Определим количество агентов-разведчиков n_r=5 и разместим их в произвольных позициях С(6,2), О(3,6), А(6,6), Б(9,3), И(1,10).

3. В множество A_{b2} включаем n_{b2}=2 оптимальных позиций из множества n_{rl} позиций, определенных агентами–разведчиками на итерации 2. В этом случае пусть A_{b2}={Б(9,3), С(6,2)}. В этом случае A_b={А(2,3), К(4,8), Б(9,3), С(6,2)}.

11	О	А	О	А	С	О	О	К	А	Б	О
10	И	Б	О	А	Б	И	С	А	О	К	И
9	А	И	С	С	И	О	К	А	К	О	Б
8	К	А	С	***БАК***	И	Б	А	О	К	И	А
7	С	Б	Б	А	***БА***	К	И	С	И	А	Б
6	И	Б	О	К	И	А	О	А	К	Б	К
5	Б	А	Б	К	О	А	***КИ***	Б	С	И	Б
4	И	***ИБ***	А	***КС***	А	***СА***	И	А	О	И	С
3	И	***ОБА***	С	Б	К	Б	Б	К	***ОБ***	О	Б
2	О	К	О	***СК***	А	С	И	***ОБО***	Б	С	А
1	Б	К	Б	О	И	К	А	Б	С	А	И
	1	2	3	4	5	6	7	8	9	10	11

Рис. 5. Матрица А - пространство поиска для пчелиного алгоритма после 2 итерации.

4. Полагаем n_f=5 и размер окрестности $\lambda_{макс}$=3. Пусть базовым позициям поставлены в соответствие следующие позиции из их окрестностей (позиции выбираются в последовательности С, А, К, Б, С): С(6,2)→О(5,5); А(2,3)→К(4,5); К(4,8)→И(5,6); Б(9,3)→А(8,4); С(6,2)→К(8,3). Таким образом, будет получен список позиций, решений и значений ЦФ: позиция С(6,2), R=0,001, список Е содержит один символ, позиция А(2,3), Е={ОБА}, R=1,2; позиция К(4,8), Е={БАК}, R=1,2; позиция Б(9,3), Е={ОБ}, R=0,6; позиция О(5,5), Е={СО}, R=0,6; позиция К(4,5), Е={ОБАК}, R=1,8; позиция И(5,6), Е={БАКИ}, R=1,6; позиция А(8,4), Е={ОБА}, R=1,2; позиция К(8,3), Е={СК}, R=0,08. Фрагменты текста, содержащие три и более символа, как и ранее, умножаются на значения Q_i. Для списков ОБА, БАК, ОБАК, БАКИ определим соответственно Q=0,9. В этом случае для позиции А(2,3) R=1,08; для позиции К(4,8) R=1,08; для позиции К(4,5) R=1,62; для позиции И(5,6) R=1,44, для позиции А(8,4) R=1,08. Области D_i определятся следующим образом: D_1={С(6,2), О(5,5), К(8,3)}; D_2={А(2,3), К(4,5)}; D_3={К(4,8), И(5,6)}; D_4={Б(9,3), А(8,4)}.

5. Выбирая лучшие позиции a_i^* с лучшим значением ЦФ в данных областях, получим: D_1→О(5,5), R_1^*=0,6; D_2→К(4,5), R_2^*=1,62; D_3→И(5,6), R_3^*=1,44; D_4→А(8,4), R_4^*=1,08.

6. Таким образом, на данной итерации $R^*(2)$=1,62; $E^*(2)$={ОБАК}.

7. l=4.

Итерация 4.

1. Число базовых позиций, как и ранее, выберем n_{b1}=2, множество A_{b1} будет содержать n_{b1} лучших позиций, определенных агентами в каждой из областей D_i на итерации 3. Таким образом, A_{b1}={К(4,5), И(5,6)}. Соответствующие списки показаны на рис. 6.

11	О	А	О	А	С	О	О	К	А	Б	О
10	И	Б	О	А	Б	И	С	А	О	К	И
9	А	И	С	С	И	О	К	А	К	О	Б
8	К	А	С	***БАК***	И	Б	А	О	К	И	А
7	С	Б	Б	А	***БА***	К	И	С	И	А	Б
6	И	Б	О	К	***БАКИ***	А	О	А	К	Б	К
5	Б	А	Б	***ОБАК***	***СО***	А	***КИ***	Б	С	И	Б
4	И	***ИБ***	А	***КС***	А	***СА***	И	***ОБА***	О	И	С
3	И	***ОБА***	С	Б	К	Б	Б	***СК***	***ОБ***	О	Б
2	О	К	О	***СК***	А	С	И	***ОБО***	Б	С	А
1	Б	К	Б	О	И	К	А	Б	С	А	И
	1	2	3	4	5	6	7	8	9	10	11

Рис. 6. Матрица А - пространство поиска
для пчелиного алгоритма после 3 итерации.

2. Количество агентов-разведчиков, как и ранее, зададим n_r=5 и разместим их в произвольных позициях С(3,3), К(8,11), О(5,5), К(1,8), Б(10,11).

3. Множество A_{b2} будет содержать n_{b2}=2 оптимальных позиций из множества n_{rl} новых позиций, определенных агентами–разведчиками на итерации 2. Пусть A_{b2}={С(3,3), О(5,5)}. Таким образом, A_b={К(4,5), И(5,6), С(3,3), О(5,5)}.

4. Определим число агентов–фуражиров n_f=5 и размер окрестности $\lambda_{макс}$=3. Пусть последовательность выбора базовых позиций следующая: С, И, О, С, К. Пусть базовым позициям поставлены в соответствие следующие позиции из окрестностей: С(3,3)→К(4,5); И(5,6)→А(5,7); О(5,5)→И(5,6);

С(3,3)→А(2,3); К(4,5)→И(1,6). В этом случае будет сформирован следующий список позиций, решений и значений ЦФ: позиция С(3,3), R=0,001, список Е содержит один символ; позиция О(5,5), Е={СО}, R=0,6; позиция К(4,5), Е={СОБАК}, R=2,4; позиция А(5,7), Е={БАКИБА}, R=2,8; позиция И(5,6), Е={СОБАКИ}, R=2,8; позиция А(2,3), Е={СОБА}, R=1,6; позиция И(1,6), Е={ОБАКИ}, R=2,2.

Таким образом, на данной итерации позициям И(5,6) и А(5,7) соответствует текст длиной 6 символов. Поскольку длина полученного блока исходного текста совпадает с длиной исходного текста, то для данных блоков может быть определен секретный ключ (в соответствии с отмеченными выше допущениями), обеспечивающий преобразование шифртекста в исходный (и наоборот). Поскольку строка, соответствующая позиции И(5,6), является осмысленной строкой с максимальным значением R, то ключ, определенный для данной строки и приведенный в [17], очевидно, является искомым (одним из вариантов ключа, определенного в [17], является, например, строка FCBADE).

Заключение

Таким образом, в данной работе была рассмотрена возможность применения метода пчелиной колонии для реализации криптоанализа блочного шифрования, при котором применение секретного ключа осуществляет реализацию шифров перестановок для блока текста, а также наличие исходного и полученного текста позволяет определить секретный ключ.

Приведенный пример иллюстрирует, как с помощью пчелиного алгоритма строка шифртекста может быть преобразована в строку исходного текста (аналогично [9]), для которой может быть определен секретный ключ шифрования.

Как и ранее в [9], отметим, что при реализации алгоритма существенным является тот момент, что в задаче криптоанализа имеет место поиск экстремума немонотонной функции, (построение списка с оптимальным значением ЦФ в общем случае не означает его оптимальность на дальнейших итерациях). Отличительные особенности алгоритма, возникающие при реализации в связи с этим, отмечены в [9] (такие как: достаточно большое пространство поиска и применение операций, используемых в эволюционном моделировании для предотвращения попадания в локальный оптимум; реализация алгоритма как аналога генетического алгоритма при достаточно большом числе итераций и достаточно большом числе списков). В заключение также отметим, что поскольку задача криптоанализа является оптимизационной задачей и в общем случае может интерпретироваться как задача формирования упорядоченных списков, то, как отмечено в [8,9], алгоритмы пчелиных колоний могут являться эффективным способом поиска рациональных решений для данного класса задач.

Работа выполнена при финансовой поддержке РФФИ (проект 14-01-00634).

Литература

1. Курейчик В. В., Курейчик В.М., Родзин С.И. Концепция природных вычислений, инспирированных природными системами // Известия ЮФУ. 2009. № 4. С. 16–24.

2. Курейчик В.В., Заруба Д.В., Запорожец Д.Ю. Алгоритм параметрической оптимизации на основе модели поведения роя светлячков // Известия ЮФУ. 2015. № 6(167). С. 6–15.

3. Курейчик В.М., Родзин С.И. Эволюционные алгоритмы: генетическое программирование (обзор) // Известия РАН. Теория и системы управления. 2002. № 1. С. 127-137.

4. Чернышев Ю.О., Сергеев А.С., Дубров Е.О., Крупенин А.В., Третьяков О.П. Криптографические методы и генетические алгоритмы решения задач криптоанализа: монография. Краснодар: ФВАС, 2013, 138 с.

5. Чернышев Ю.О., Сергеев А.С., Дубров Е.О., Крупенин А.В., Капустин С.А., Рязанов А.Н. Биоинспирированные алгоритмы решения задач криптоанализа классических и асимметричных криптосистем: монография. Краснодар: КВВУ, 2015, 132 с.

6. Чернышев Ю.О., Сергеев А.С., Венцов Н.Н., Рязанов А.Н. Исследование возможности применения генетических алгоритмов для реализации криптоанализа блочных криптосистем // Вестник Донского государственного технического университета. 2015. № 3(82). С. 65-72.

7. Чернышев Ю.О., Сергеев А.С., Капустин С.А., Рязанов А.Н. Исследование возможности применения методов эволюционной оптимизации для реализации криптоанализа блочных методов шифрования // Изв. СПбГЭТУ "ЛЭТИ". 2015. № 10. С. 32-40.

8. Лебедев В. Б. Модели адаптивного поведения колонии пчел для решения задач на графах // Известия ЮФУ. 2012. № 7. С. 42–49.

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

10. Лебедев О. Б. Трассировка в канале методом муравьиной колонии // Известия ЮФУ. 2009. № 4. С. 46–52.

11. Алгоритм пчел для оптимизации функции. URL: jenyay.net/Programming/Bees (дата обращения 08.06.2016).

12. Pham D.T., Ghanbarzadeh A., Koc E. The Bees Algorithm //Technical Note, Manufacturing Engineering Centre. Cardiff University UK/. – 2005. Pp. 1-57

13. Karaboga D. An idea based on honey bee swarm for numerical optimization, technical report-tr06 // Erciyes University, Engineering Faculty, Computer Engineering Department. 2005. 10 p.

14. Курейчик В. В., Жиленков М.А. Пчелиный алгоритм для решения оптимизационных задач с явно выраженной целевой функцией //Информатика, вычислительная техника и инженерное образование. 2015. № 1(21). С. 1-8.

15. Алгоритм пчел для оптимизации функции. URL: lit999.narod.ru/soft/ga/index.html (дата обращения 08.06.2016).

16. Курейчик В. В., Запорожец Д.Ю. Роевой алгоритм в задачах оптимизации // Известия ЮФУ. 2010. № 7(108). С. 28–32.

17. Чернышев Ю.О., Сергеев А.С, Дубров Е.О., Рязанов А.Н. Применение метода муравьиных колоний для реализации криптоанализа блочных криптосистем // Программные продукты и системы: международный научно-практический журнал. 2014. № 1(105). С. 10-19.

18. Курейчик В. В., Полупанова Е.Е. Эволюционная оптимизация на основе алгоритма колонии пчел // Известия ЮФУ. 2009. № 12(101). С. 41–46.

19. Венцов Н.Н. Эволюционный подход к моделированию распределительных процессов // Инженерный вестник Дона, 2013, № 4, URL: ivdon.ru/ru/magazine/archive/n4y2013/1886.

20. Нетесов А.С. Эволюционно-генетический подход к решению задач оптимизации. Сравнительный анализ генетических алгоритмов с

традиционными методами оптимизации // Инженерный вестник Дона, 2011, № 3, URL: ivdon.ru/ru/magazine/archive/n3y2011/459.

References

1. Kurejchik V. V., Kurejchik V.M., Rodzin S.I. Izvestija JuFU. 2009. № 4. pp. 16-24.

2. Kurejchik V.V., Zaruba D.V., Zaporozhec D.Ju. Izvestija JuFU. 2015. № 6(167). pp. 6-15

3. Kurejchik V.M., Rodzin S.I. Izvestija RAN. Teorija i sistemy upravlenija. 2002. № 1. pp. 127-137.

4. Chernyshev Ju.O., Sergeev A.S., Dubrov E.O., Krupenin A.V., Tret'jakov O.P. Kriptograficheskie metody i geneticheskie algoritmy reshenija zadach kriptoanaliza: monografija [Cryptographic methods and genetic algorithms of the solution of problems of cryptanalysis: monograph.]. Krasnodar: FVAS, 2013, 138 p.

5. Chernyshev Ju.O., Sergeev A.S., Dubrov E.O., Krupenin A.V., Kapustin S.A., Rjazanov A.N. Bioinspirirovannye algoritmy reshenija zadach kriptoanaliza klassicheskih i asimmetrichnyh kriptosistem: monografija [The bioinspired algorithms of the solution of problems of cryptanalysis of classical and asymmetric cryptosystems: monograph.]. Krasnodar: KVVU, 2015, 132 p.

6. Chernyshev Ju.O., Sergeev A.S., Vencov N.N., Rjazanov A.N. Vestnik Donskogo gosudarstvennogo tehnicheskogo universiteta. 2015. № 3(82). pp. 65-72.

7. Chernyshev Ju.O., Sergeev A.S., Kapustin S.A., Rjazanov A.N. Izv. SPbGJeTU "LJeTI". 2015. № 10. pp. 32-40.

8. Lebedev V. B. Izvestija JuFU. 2012. № 7. pp. 42-49.

9. Chernyshev Ju.O., Sergeev A.S., Dubrov E.O., Rjazanov A.N. Vestnik Donskogo gosudarstvennogo tehnicheskogo universiteta. 2014. T. 14. № 1(76). pp. 62-75.

10. Lebedev O. B. Izvestija JuFU. 2009. № 4. pp. 46-52.

11. Algoritm pchel dlja optimizacii funkcii [Algorithm of bees for function optimization]. URL: jenyay.net/Programming/Bees (data obrashhenija 08.06.2016).

12. Pham D.T., Ghanbarzadeh A., Koc E. The Bees Algorithm //Technical Note, Manufacturing Engineering Centre. Cardiff University UK/. – 2005. Pp. 1-57

13. Karaboga D. An idea based on honey bee swarm for numerical optimization, technical report-tr06. Erciyes University, Engineering Faculty, Computer Engineering Department. 2005. 10 p.

14. Kurejchik V. V., Zhilenkov M.A. Informatika, vychislitel'naja tehnika i inzhenernoe obrazovanie. 2015. № 1(21). pp. 1-8.

15. Algoritm pchel dlja optimizacii funkcii [Algorithm of bees for function optimization]. URL: lit999.narod.ru/soft/ga/index.html (data obrashhenija 08.06.2016).

16. Kurejchik V. V., Zaporozhec D.Ju. Izvestija JuFU. 2010. № 7(108). pp. 28-32.

17. Chernyshev Ju.O., Sergeev A.S, Dubrov E.O., Rjazanov A.N. Programmnye produkty i sistemy: mezhdunarodnyj nauchno-prakticheskij zhurnal. 2014. № 1(105). pp. 10-19.

18. Kurejchik V. V., Polupanova E.E. Izvestija JuFU. 2009. № 12(101). pp. 41-46.

19. Vencov N.N. Inženernyj vestnik Dona (Rus), 2013, № 4, URL: ivdon.ru/ru/magazine/archive/n4y2013/1886.

20. Netesov A.S. Inženernyj vestnik Dona (Rus), 2011, № 3, URL: ivdon.ru/ru/magazine/archive/n3y2011/459.