УДК 004.023; 511.52 DOI 10.18522/2311-3103-2021-6-115-123

Е.Е. Полупанова, П.Е. Усов

ЭВРИСТИЧЕСКИЙ ГЕНЕТИЧЕСКИЙ АЛГОРИТМ РЕШЕНИЯ ДИОФАНТОВЫХ УРАВНЕНИЙ

Рассматривается задача решения диофантовых уравнений, которая может применяться в криптографии и криптоанализе. Кратко излагается описание генетического алгоритма решения диофантовых уравнений. Определяется правило вычисления значения целевой функции для хромосомы, описывается система кодирования в генетическом алгоритме. Упоминаются генетические операторы, используемые в алгоритме, определяются условия их выполнения. Описывается критерий останова генетического алгоритма. Анализируется один из недостатков генетического алгоритма – попытки решения любого диофантова уравнения, в том числе и такого, которое заведомо не имеет решений. Предлагается способ, позволяющий устранить этот недостаток в некоторых случаях, и, основанный на теории чисел. Даётся пояснение, в каких случаях этот способ будет работать. Перед описанием этого способа даётся определение вычета и невычета заданной степени по заданному модулю. После описания этого способа подробно описывается программная реализация алгоритма решения диофантовых уравнений и их систем. Затем приводятся результаты экспериментальных исследований времени и качества работы генетического алгоритма. Затем представляется результат работы алгоритма для уравнения, которое заведомо не имеет решений, и для системы уравнений, которая также заведомо не имеет решений, но в которой общее число неизвестных слишком велико для работы предлагаемого метода. Сравнивается время работы алгоритма при решении уравнения и при решении системы уравнений. Делается вывод о полезности применения предложенного способа при решении диофантовых уравнений и систем диофантовых уравнений.

Эвристика; генетический алгоритм; диофантово уравнение; вычет степени n по модулю m; невычет степени n по модулю m.

E.E. Polupanova, P.E. Usov

HEURISTIC GENETIC ALGORITHM FOR DIOPHANTINE EQUATIONS SOLVING

The problem of diophantine equations solving is considered in this article. This problem can be applied in cryptography and cryptanalysis. The description of the genetic algorithm solving diophantine equations is stated briefly in the article. The rule of calculation the value of fitness function of chromosome is determined, the coding system in the genetic algorithm is described. The genetic operators used in the algorithm are mentioned and the conditions for their execution are determined. The criterion for stopping the genetic algorithm is described. One of the shortcomings of the genetic algorithm is analyzed. The shortcoming of the algorithm lies in its attempts to solve any diophantine equation, including one that has no solutions. A method eliminating this shortcoming in some cases is proposed. This method is based on number theory. An explanation is given in which cases this method will be used. The definition of residue and nonresidue of fixed power for fixed modulus is given before describing this method. After describing this method the implementation of the algorithm for solving diophantine equations and systems of them is described in detail. Then the results of experimental studies of the time and quality of the genetic algorithm are presented. Then the result of the algorithm is presented for an equation that has no solutions and for a system of equations that also has no solutions, but in which the total number of unknowns is too large for the proposed method to work. The algorithm running time is compared when solving an equation and when solving a system of equations. The conclusion is made about the usefulness of the proposed method in solving diophantine equations and systems of diophantine equations.

Heuristics; genetic algorithm; diophantine equation; residue of power n for modulus m; nonresidue of power n for modulus m.

Постановка задачи решения диофантова уравнения. Диофантовым уравнением (ДУ) называется уравнение вида (1) [12, с. 7; 8; 9, 14].

$$D(x_1, \dots, x_n) = 0, \quad (1)$$

где D – полином с целыми коэффициентами.

Задача решения ДУ хорошо известна и может применяться, например, при криптоанализе систем защиты информации, содержащих диофантовы трудности [15].

Решение ДУ состоит в нахождении всех наборов из n чисел $x_1, x_2, \dots, x_n$, при которых верна формула (1). Не существует универсального алгоритма решения в целых числах произвольного ДУ. Это следует из DPRM-теоремы [10, с. 46].

Решение диофантовых уравнений генетическим алгоритмом. Разработанный генетический алгоритм [1, 2, 3] состоит из последовательности шагов:

случайная генерация первого поколения хромосом [4, 7, 11];

1) применение последовательности одноточечных операторов мутации;

2) формирование второго поколения хромосом путём попарного скрещивания хромосом первого поколения;

3) формирование третьего и всех последующих поколений путём попарного скрещивания первых $n1$ хромосом предыдущего поколения;

4) проверка критериев останова алгоритма.

Рассмотрим данную последовательность шагов подробнее.

Гены [13, 17, 20] хромосомы первого поколения являются случайными целыми числами из отрезка $[-10; 10]$. Генов в каждой хромосоме каждого поколения столько, сколько неизвестных в уравнении. Одно неизвестное кодируется одним геном. Генерируется всего $n1$ хромосом в первом поколении. Параметр $n1$ задаётся пользователем.

После генерации хромосом первого поколения над ними производится до трёх одноточечных операторов мутации. Для этого вначале для каждой хромосомы вычисляется значение целевой функции. Целевая функция имеет вид $D(x_1, x_2, \dots, x_n)$, где $D(x_1, x_2, \dots, x_n) = 0$ – решаемое ДУ. Если значение целевой функции для данной хромосомы не равно нулю, то над ней выполняется первый оператор мутации. Он заключается в изменении значения случайно выбранного гена на случайное целое число из отрезка $[-1000; 1000]$. Если после первой мутации значение целевой функции для этой хромосомы не стало равным нулю, то над ней выполняется второй оператор мутации. Он заключается в изменении значения случайно выбранного гена на случайное целое число из отрезка $\big[[med]; [sr]\big]$, если $[med] < [sr]$, и из отрезка $\big[[sr]; [med]\big]$ в противном случае, где med – медиана набора всех чисел, фигурирующих в уравнении, sr – среднее арифметическое чисел в этом наборе, $[x]$ – целая часть числа x. Если и после второй мутации значение целевой функции для хромосомы не стало равным нулю, то над хромосомой выполняется третья мутация. Она заключается в изменении значения случайно выбранного гена на целое число из отрезка $[-5; 5]$. Далее хромосомы первого поколения с нулевым значением целевой функции добавляются во множество решений ДУ.

Хромосомы второго поколения получаются из хромосом первого поколения путём попарного скрещивания последних. Таким образом, во втором поколении генерируется $\frac{n1*(n1-1)}{2}$ хромосом. При скрещивании применяется одноточечный оператор кроссинговера. После генерации хромосом второго поколения они подвергаются операторам мутации по тому же правилу, что и хромосомы первого поколения. Хромосомы с нулевым значением целевой функции попадают во множество решений ДУ.

Каждое поколение получается путём попарного скрещивания первых $n1$ хромосом предыдущего поколения. При скрещивании применяется одноточечный оператор кроссинговера. После генерации над каждой хромосомой очередного поколения выполняются мутации по тому же правилу, что и в первом поколении. Хромосомы с нулевым значением целевой функции попадают во множество решений уравнения. Генерируется всего max_gen поколений. Параметр max_gen задаётся пользователем. Алгоритм может завершиться досрочно, если на очередной итерации не нашлось новых решений, но до этого нашлось хотя бы одно решение.

Способ упрощения решения ДУ. Недостаток разработанного генетического алгоритма в том, что он пытается решать любые диофантовы уравнения, в том числе и те, которые заведомо не имеют корней. В связи с этим был реализован способ упрощения задачи решения ДУ, основанный на свойствах степеней целых чисел.

Целое число a называется квадратичным вычетом по модулю m, если существует целое число x такое, что число x^2 при делении на m даёт в остатке a [16, с. 302, 6, 18, 19]. В противном случае число a называется квадратичным невычетом по модулю m [16, с. 302]. Аналогично можно определить вычет степени n по модулю m и невычет степени n по модулю m. Из этого определения следует, что квадратичными вычетами по модулю 9 являются числа 0, 1, 4 и 7, вычетами степени 3 по модулю 9 являются числа 0, 1 и 8, вычетами степени 4 по модулю 9 являются числа 0, 1, 4 и 7, вычетами степени 5 по модулю 9 являются числа 0, 1, 2, 4, 5, 7 и 8, вычетами степени 6 по модулю 9 являются числа 0 и 1, вычетами степени 7 по модулю 9 являются числа 0, 1, 2, 4, 5, 7 и 8. Видно, что вычетов степени 2, 3, 4 и 6 по модулю 9 меньше, чем невычетов этих степеней по модулю 9. В связи с вышеизложенным, реализованный способ упрощения решения ДУ вида (1) состоит в том, что перед запуском генетического алгоритма программа определяет, какие остатки при делении на 9 может давать полином $D(x_1, \dots, x_n)$. Определение всех возможных остатков при делении на 9 у полинома $D(x_1, \dots, x_n)$ выполняется методом полного перебора всех возможных вариантов остатков при делении на 9 у переменных $x_1, \dots, x_n$. Таким образом, программа перебирает всего 9^n вариантов для уравнения вида (1). Этот перебор выполняется только в том случае, если в уравнении не более шести неизвестных. В противном случае перебор не выполняется, и программа сразу запускает генетический алгоритм.

Программная реализация алгоритма решения диофантовых уравнений и их систем. Программа разработана в среде PyCharm и представляет собой сценарий на языке Python версии 3.7. При запуске сценарий создаёт окно с пользовательским интерфейсом. Вид окна представлен на рис. 1.

Рис. 1. Интерфейс программы

Для запуска генетического алгоритма пользователю необходимо ввести диофантово уравнение или систему в первое поле сверху в окне на рис. 1. Во второе поле сверху необходимо ввести число хромосом в первом поколении $n1$ – натуральное число. В третье поле сверху необходимо ввести максимальное число поколений max_gen – натуральное число. Требуется, чтобы каждое вводимое диофантово уравнение имело вид (2), а уравнения в системе разделялись запятой.

$$f(x_1, x_2, \dots, x_n) = g(x_1, x_2, \dots, x_n), \quad (2)$$

где $f(x_1, x_2, \dots, x_n)$ – полином произвольной степени с целыми коэффициентами; $g(x_1, x_2, \dots, x_n)$ – полином произвольной степени с целыми коэффициентами.

Программа сама приводит каждое введённое диофантово уравнение к виду (1). При вводе уравнения или системы допускается, чтобы неизвестные состояли из латинских букв (строчных или прописных), цифр и знаков подчёркивания и содержали любое конечное число символов. При этом требуется, чтобы первым символом каждого неизвестного была латинская буква или знак подчёркивания. Также требуется, чтобы общее число неизвестных в уравнении или системе было не меньше двух. Числовые константы в уравнении должны быть целыми. В уравнениях также допускается использовать круглые скобки.

После ввода уравнения или системы пользователю нужно нажать кнопку «Готово», чтобы запустить генетический алгоритм. Перед решением уравнения или системы уравнений с помощью генетического алгоритма программа проверяет правильность ввода. Для этого проводится лексический анализ каждого введённого уравнения. Если он прошёл успешно, то программа проверяет наличие ровно одного знака равенства в каждом уравнении, иначе сообщает об ошибке. Если в каждом уравнении есть ровно один знак равенства, то программа проверяет наличие непустой левой и непустой правой частей в каждом уравнении, иначе сообщает об ошибке. Если в каждом уравнении есть непустая левая и непустая правая части, то программа преобразует каждое уравнение к виду (1). После этого программа составляет сумму квадратов левых частей полученных уравнений вида (1) и приравнивает её к нулю.

Далее полученное таким образом диофантово уравнение решается генетическим алгоритмом. Однако ГА запускается не во всех случаях. Если общее число неизвестных в уравнении или системе не больше шести, то программа переводит в обратную польскую нотацию левые части полученных уравнений вида (1) и по полученной обратной польской записи вычисляет множества остатков при делении на 9 значений левых частей уравнений. Если хотя бы в одном множестве нет значения 0, то вся система не имеет решений, о чём программа сообщает, даже не запуская генетический алгоритм. В противном случае программа запускает генетический алгоритм, предварительно проверив правильность ввода его параметров. Если же общее число неизвестных в уравнении или системе больше шести, то программа сразу запускает генетический алгоритм, предварительно проверив правильность ввода его параметров.

Результат работы программы (список решений уравнения или системы, либо утверждение, что решений нет, либо утверждение, что не удалось найти решения) выводится на экран в поле под кнопкой «Готово» (рис. 1).

Экспериментальные исследования. Были проведены экспериментальные исследования времени и качества работы генетического алгоритма. Под качеством работы алгоритма понимается статистическая вероятность нахождения алгоритмом хотя бы одного решения заданного диофантова уравнения или системы диофантовых уравнений. На рис. 2 представлен график зависимости качества работы алгоритма от числа хромосом в первом поколении при max_gen=400, где

max_gen – максимальное число генерируемых поколений. В качестве линейного уравнения использовалось уравнение $32 * x + 45 * y = 779$, квадратного – $x^2 + y^2 = z * (x + y)$, кубического – $x^3 = y^2 + z$, уравнения четвёртой степени – $9 * x^4 + 7 * y = 16$, пятой степени – $x^5 + 13 * y = 14$, шестой степени – $x^6 + y = 65$.

По рис. 2 видно, что все уравнения, кроме линейного, решаются в среднем во всех случаях при $n1 = 25, 30, 35, 40, 45, 50$. Это может быть связано с тем, что в линейном уравнении фигурируют самые большие по модулю числа. В пользу этого может говорить тот факт, что, например, линейное уравнение $10 * x + 7 * y = 19$, а также уравнения $10 * x^2 + 7 * y = 19$, $10 * x * y * z + 7 * t = 17$ решаются в среднем в 100 процентах случаев при $n1 = 25, 30, 35, 40, 45, 50$. Шаг изменения параметра $n1$ был выбран равным пяти, так как этого было достаточно для исследования. Чем меньше значение $n1$, тем меньше хромосом алгоритм перебирает и тем меньше времени он работает. Поэтому оптимальное значение $n1$ равно 45 – наименьшему из исследуемых значений, при котором все рассматриваемые уравнения решались в среднем во всех случаях.

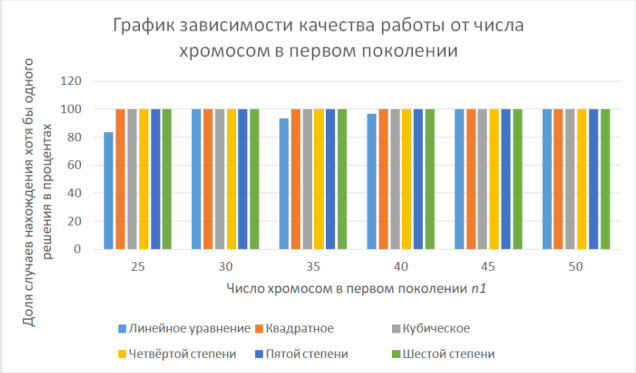

Рис. 2. График зависимости качества работы алгоритма от числа хромосом в первом поколении

Оптимальное значение параметра max_gen (максимально возможное число поколений) по результатам экспериментальных исследований составило 300. В этих исследованиях значение параметра $n1$ было принято равным 45 и в качестве линейного уравнения использовалось уравнение $10 * x + 7 * y = 19$, квадратного – $10 * x^2 + 7 * y = 19$, кубического – $10 * x * y * z + 7 * t = 17$, уравнения четвёртой степени – $9 * x^4 + 7 * y = 16$, пятой степени – $x^5 + 13 * y = 14$, шестой степени – $x^6 + y = 65$. Все уравнения, кроме линейного, решались в среднем в 100 процентах случаев при max_gen=250, 300, 350, 400. Шаг изменения значения параметра max_gen был выбран равным 50, так как этого было достаточно для исследования. Время работы алгоритма прямо пропорционально максимально возможному числу поколений, а 300 оказалось наименьшим из исследуемых значений, при котором все рассматриваемые уравнения решались в среднем во всех случаях. Время выполнения алгоритма при $n1 = 45$, max_gen=300 при решении вышеуказанных шести уравнений было не более 1 секунды. Было также экспериментально установлено, что имеется квадратичная зависимость между временной сложностью алгоритма и общим числом генерируемых хромосом.

Кроме того, было экспериментально установлено, что предложенный способ упрощения решения диофантовых уравнений позволяет в ряде случаев сократить время работы программы. Например, на рис. 3 представлен результат работы про-

граммы для уравнения $x^3 + y^3 + z^3 = 4$. Это уравнение содержит три неизвестных, поэтому программа сначала определит, какие остатки при делении на 9 может давать значение полинома $x^3 + y^3 + z^3 - 4$. Также это уравнение не имеет целочисленных решений, так как сумма трёх кубов не может при делении на 9 давать в остатке 4 или 5. На рис. 4 представлен результат работы программы для системы уравнений, состоящей из ДУ $x^3 + y^3 + z^3 = 4$, $t^3 + u^3 + v^3 = 5$ и $v^2 + w^2 = 3$. В этой системе в общей сложности больше шести неизвестных, поэтому программа сразу запустит генетический алгоритм. Также эта система не имеет решений, так как не имеют решений первое и второе уравнение этой системы.

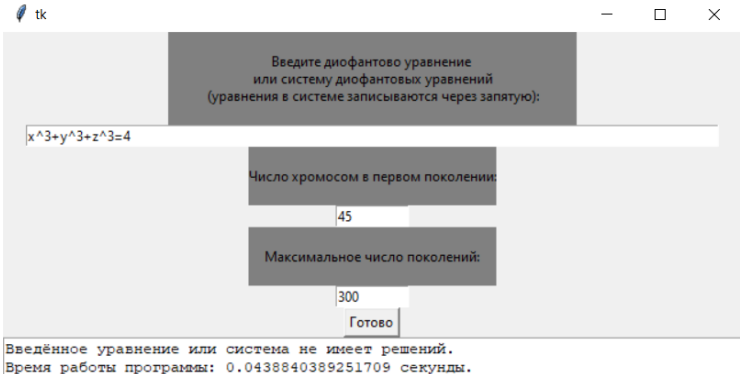

Рис. 3. Результат работы программы для уравнения

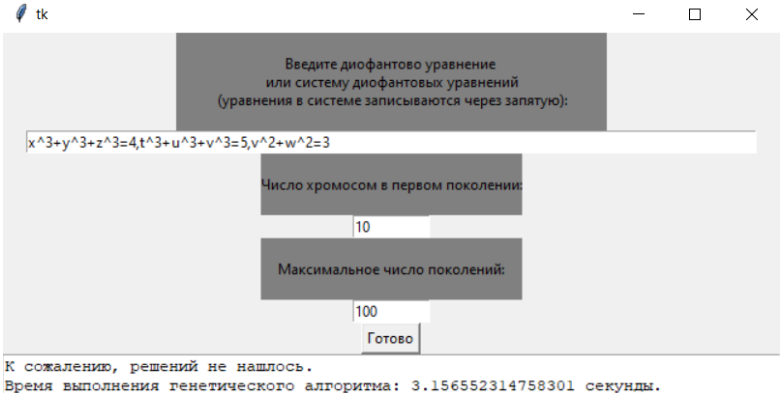

Рис. 4. Результат работы программы для системы уравнений

По рис. 3 и 4 видно, что в случае предварительного определения возможных остатков при делении на 9 у значений полинома $x^3 + y^3 + z^3 - 4$ программа работает быстрее приблизительно на 3.11 секунды, чем без такого определения даже с меньшими значениями параметров генетического алгоритма, которые прямо пропорциональны времени работы алгоритма.

На рис. 5 представлен график зависимости времени работы алгоритма от числа хромосом. Можно сделать вывод, что разработанный ГА обладает квадратичной временной сложностью.

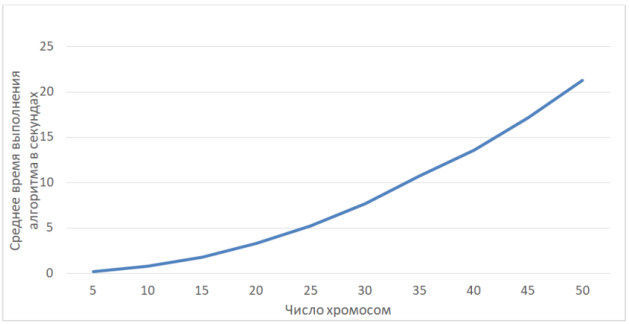

Рис 5. График зависимости времени работы алгоритма от числа хромосом в первом поколении

Проведённые экспериментальные исследования показали, что время и качество работы генетического алгоритма зависят от числа хромосом в первом поколении, максимального числа генерируемых поколений, а также от корней уравнения.

Заключение. В данной статье был предложен ГА решения ДУ, адаптированный для решения систем ДУ. Также был программно реализован способ, упрощающий в ряде случаев решение ДУ путём определения некоторых уравнений, заведомо не имеющих решений. В результате экспериментальных исследований выяснилось, что этот способ может уменьшить время работы программы. Также выяснилось, что имеется квадратичная зависимость между временной сложностью алгоритма и общим числом генерируемых хромосом. Также были определены оптимальные значения параметров генетического алгоритма.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Абдулджаббар И.А., Абдулла С.М.* Эволюционный алгоритм решения проблемы планирования расписания академических курсов. – URL: https://bsj.uobaghdad.edu.iq/index.php/BSJ/article/view/5309/3677 (дата обращения: 29.10.2021).
2. *Гладков Л.А., Курейчик В.В., Курейчик В.М.* Генетические алгоритмы. – 2-е изд. – М.: Физматлит, 2006. – 320 с.
3. *Голдберг Д.Э.* Генетические алгоритмы поиска, оптимизации и машинного обучения. – Изд. компания «Эддисон-Уэсли», 1989.
4. *Деликаташ Д., Озкан Э., Устун О., Торкуль О.* Эволюционные алгоритмы для многоцелевого гибкого планирования ячеек рабочих мест. – URL: https://reader.elsevier.com/reader/sd/pii/S1568494621008127?token=B450EEE1CF1DDA0740952A46F8D57C861AD0F25F848BC39EAAFC2CD99565128625281A46285BAEAF57055ED57155F17D&originRegion=eu-west-1&originCreation=20211029073249 (дата обращения: 29.10.2021).
5. *Дэвенпорт Х.* О проблеме Уоринга для кубов // Acta Mathematica. – 1939. – Т. 71.
6. *Кавамура С., Комано Ю., Симидзу Х., Йонемура Т.* Алгоритмы Монтгомери с использованием квадратичной остаточности. – URL: https://link.springer.com/content/pdf/10.1007/s13389-018-0195-8.pdf (дата обращения: 29.10.2021).
7. *Карпенко А.П.* Современные алгоритмы поисковой оптимизации. Алгоритмы, вдохновлённые природой: учеб. пособие. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2014. – 446 с.
8. *Крейцберг П., Серанг О.* О решении вероятностных линейных диофантовых уравнений. – URL: https://jmlr.org/papers/volume22/17-474/17-474.pdf (дата обращения: 29.10.2021).
9. *Ман Ю.К.* Перспективный подход к решению линейного диофантова уравнения. – URL: https://www.tandfonline.com/doi/pdf/10.1080/0020739X.2020.1745915 (дата обращения: 28.10.2021).
10. *Манин Ю.И.* Вычислимое и невычислимое. – М.: Советское радио, 1980. – 128 с.
11. *Саймон Д.* Алгоритмы эволюционной оптимизации. – М.: Изд-во «ДМК Пресс», 2020. – 1002 с.

12. *Матиясевич Ю.В.* Десятая проблема Гильберта. – М.: Физматлит, 1993. – 224 с.
13. *Митчелл М.* Введение в генетические алгоритмы. – Лондон: Первое издание издательства Массачусетского технологического института в мягкой обложке, 1998. – 158 с.
14. *Осипов Н.Н., Кытманов А.А.* Алгоритм решения семейства диофантовых уравнений четвертой степени, удовлетворяющих условию Рунге. – URL: https://events.rudn.ru/event/20/papers/178/files/445-Osipov-Kytmanov-CA-2019.pdf (дата обращения: 28.10.2021).
15. *Осипян В.О.* Разработка математической модели дисимметричной биграммной криптосистемы на основе параметрического решения многостепенной системы диофантовых уравнений // Инженерный вестник Дона. – 2020. – № 6.
16. *Фороузан Б.А.* Криптография и безопасность сетей: учеб. пособие: пер. с англ. / под ред. А.Н. Берлина. – М.: Интернет-Университет Информационных Технологий: БИНОМ. Лаборатория знаний, 2010. – 784 с.
17. *Холланд Д.Х.* Генетические алгоритмы. – URL: https://www2.econ.iastate.edu/tesfatsi/holland.GAIntro.htm (дата обращения: 30.10.2021).
18. *Чанг К.-С., Ли К.-Т., Чен К.* Сохранение конфиденциальности и обратимое сокрытие информации на основе арифметики квадратичных вычетов. – URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8679991 (дата обращения: 29.10.2021).
19. *Чэнь Ю.-Г., Ян Х.-Х.* Гипотеза Шаркози о квадратичных вычетах. – URL: https://reader.elsevier.com/reader/sd/pii/S0022314X21001402?token=B17EF4980D92E71F5CA05919D590CE859BA43F68FD937D66C630F36D032BCDB1A6541630B232868DE20F772B14520015&originRegion=eu-west-1&originCreation=20211029110553 (дата обращения: 29.10.2021).
20. *Шарифи М.Р., Акбарифард С., Кадери К., Мадади М.Р.* Сравнительный анализ некоторых эволюционных моделей в оптимизации работы водохранилищ плотины. – URL: https://www.nature.com/articles/s41598-021-95159-4.pdf (дата обращения: 29.10.2021).

REFERENCES

1. *Abduldzhabbar I.A., Abdulla S.M.* Evolyutsionnyy algoritm resheniya problemy planirovaniya raspisaniya akademicheskikh kursov [An evolutionary algorithm for solving academic courses timetable scheduling problem]. Available at: https://bsj.uobaghdad.edu.iq/index.php/BSJ/article/view/5309/3677 (accessed 29 October 2021).
2. *Gladkov L.A., Kureychik V.V., Kureychik V.M.* Geneticheskie algoritmy [Genetic algorithms]. 2nd ed. Moscow: Fizmatlit, 2006, 320 p.
3. *Goldberg D.E.* Geneticheskie algoritmy poiska, optimizatsii i mashinnogo obucheniya [Goldberg Genetic algorithms in search, optimization, and machine learning]. Izd. kompaniya «Eddison-Uesli», 1989.
4. *Delikatash D., Ozkan E., Ustun O., Torkul' O.* Evolyutsionnye algoritmy dlya mnogotselevogo gibkogo planirovaniya yacheek rabochikh mest [Evolutionary algorithms for multi-objective flexible job shop cell scheduling]. Available at: https://reader.elsevier.com/reader/sd/pii/S1568494621008127?token=B450EEE1CF1DDA0740952A46F8D57C861AD0F25F848BC39EAAFC2CD99565128625281A46285BAEAF57055ED57155F17D&originRegion=eu-west-1&originCreation=20211029073249 (accessed 29 October 2021).
5. *Devenport Kh.* O probleme Uoringa dlya kubov [On Waring's problem for cubes], *Acta Mathematica*, 1939, Vol. 71.
6. *Kavamura S., Komano Yu., Simidzu Kh., Yonemura T.* Algoritmy Montgomeri s ispol'zovaniem kvadratichnoy ostatochnosti [Montgomery algorithms using quadratic residuals]. Available at: https://link.springer.com/content/pdf/10.1007/s13389-018-0195-8.pdf (accessed 29 October 2021).
7. *Karpenko A.P.* Sovremennye algoritmy poiskovoy optimizatsii. Algoritmy, vdokhnovlennye prirodoy: ucheb. posobie [Modern search engine optimization algorithms. Algorithms inspired by nature: a textbook]. Moscow: Izd-vo MGTU im. N.E. Baumana, 2014, 446 p.
8. *Kreytsberg P., Serang O.* O reshenii veroyatnostnykh lineynykh diofantovykh uravneniy [On solving probabilistic linear diophantine equations]. Available at: https://jmlr.org/papers/volume 22/17-474/17-474.pdf (accessed 29 October 2021).

9. *Man Yu.K.* Perspektivnyy podkhod k resheniyu lineynogo diofantova uravneniya [A forward approach for solving linear Diophantine equation]. Available at: https://www.tandfonline.com/doi/pdf/10.1080/0020739X.2020.1745915 (accessed 28 October 2021).
10. *Manin Yu.I.* Vychislimoe i nevychislimoe [Computable and non-computable]. Moscow: Sovetskoe radio, 1980, 128 p.
11. *Saymon D.* Algoritmy evolyutsionnoy optimizatsii [Evolutionary optimization algorithms]. Moscow: Izd-vo «DMK Press», 2020, 1002 p.
12. *Matiyasevich Yu.V.* Desyataya problema Gil'berta [Hilbert's tenth problem]. Moscow: Fizmatlit, 1993, 224 p.
13. *Mitchell M.* Vvedenie v geneticheskie algoritmy [An introduction to genetic algorithms]. London: Pervoe izdanie izdatel'stva Massachusetskogo tekhnologicheskogo instituta v myagkoy oblozhke, 1998, 158 p.
14. *Osipov N.N., Kytmanov A.A.* Algoritm resheniya semeystva diofantovykh uravneniy chetvertoy stepeni, udovletvoryayushchikh usloviyu Runge [An algorithm for solving a family of fourth-degree diophantine equations that satisfy Runge's condition]. Available at: https://events.rudn.ru/event/20/papers/178/files/445-Osipov-Kytmanov-CA-2019.pdf (accessed 28 October 2021).
15. *Osipyan V.O.* Razrabotka matematicheskoy modeli disimmetrichnoy bigrammnoy kriptosistemy na osnove parametricheskogo resheniya mnogostepennoy sistemy diofantovykh uravneniy [Development of a mathematical model of a disymmetric bigram cryptosystem based on a parametric solution of a multistep system of diophantine equations], *Inzhenernyy vestnik Dona* [Engineering Bulletin of the Don], 2020, No. 6.
16. *Forouzan B.A.* Kriptografiya i bezopasnost' setey: ucheb. posobie [Cryptography and network security: a textbook]: transl. from Engl., ed. by A.N. Berlin. Moscow: Internet-Universitet Informatsionnykh Tekhnologiy: BINOM. Laboratoriya znaniy, 2010, 784 p.
17. *Kholland D.Kh.* Geneticheskie algoritmy [Genetic algorithms]. Available at: https://www2.econ.iastate.edu/tesfatsi/holland.GAIntro.htm (accessed 30 October 2021).
18. *Chang K.-S., Li K.-T., Chen K.* Sokhranenie konfidentsial'nosti i obratimoe sokrytie informatsii na osnove arifmetiki kvadratichnykh vychetov [Privacy-preserving reversible information hiding based on arithmetic of quadratic residues]. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8679991 (accessed 29 October 2021).
19. *Chen' yu.-G., Yan Kh.-Kh.* Gipoteza Sharkozi o kvadratichnykh vychetakh [A conjecture of Sárközy on quadratic residues]. Available at: https://reader.elsevier.com/reader/sd/pii/S0022314X21001402?token=B17EF4980D92E71F5CA05919D590CE859BA43F68FD937D66C630F36D032BCDB1A6541630B232868DE20F772B14520015&originRegion=eu-west-1&originCreation=20211029110553 (accessed 29 October 2021).
20. *Sharifi M.R., Akbarifard S., Kaderi K., Madadi M.R.* Sravnitel'nyy analiz nekotorykh evolyutsionnykh modeley v optimizatsii raboty vodokhranilishch plotiny [Comparative analysis of some evolutionary-based models in optimization of dam reservoirs operation]. Available at: https://www.nature.com/articles/s41598-021-95159-4.pdf (accessed 29 October 2021).

Статью рекомендовал к опубликованию д.т.н., профессор Ф.Г. Хисамов.

Полупанова Елена Евгеньевна – Кубанский государственный университет; e-mail: jienka@mail.ru; г. Краснодар, Россия; тел.: +79284013301; кафедра вычислительных технологий; к.т.н.; доцент.

Усов Павел Евгеньевич – e-mail: lyova-pavel.usov@yandex.ru; тел.: +79186374835, кафедра вычислительных технологий; магистрант.

Polupanova Elena Evgenievna – Kuban State University; e-mail: jienka@mail.ru; Krasnodar, Russia; phone: +79284013301, the department of computational technologies; cand. of eng. sc.; associate professor.

Usov Pavel Evgenievich – e-mail: lyova-pavel.usov@yandex.ru; phone: +79186374835, the department of computational technologies; master's degree student.