Математическое обеспечение расчета маршрутов по графу реперов на местности генетическим алгоритмом

Руденко Эдуард Михайлович

к.т.н., доцент филиала Военной академии Ракетных войск стратегического назначения им. Петра Великого, г. Серпухов, Россия, eduard5529@yandex.ru

Семикина Елена Викторовна

преподаватель филиала Военной академии Ракетных войск стратегического назначения им. Петра Великого, г. Серпухов, Россия, labinfo_serp@inbox.ru

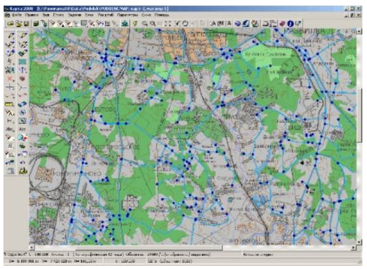
АННОТАЦИЯ

Введение: в работе рассматривается математическое обеспечение задачи маршрутизации по графу реперов на местности генетическим алгоритмом. Объектом исследования является математический граф, изоморфный графу реперов на местности. **Постановка задачи**: задача нахождения множества замкнутых маршрутов на эйлеровом, гамильтоновым и эйлеро-гамильтоновом графе формулируется как оптимизационная на условный экстремум. Критерием является величина, равная числу ребер графа, которая равна полу сумме кратностей вершин графа. При этом кратные ребра учитывается столько раз в маршрутах, какова их кратность в графе. Результатом решения является подмножество эйлеровых маршрутов. Задача маршрутизации относится к задаче целочисленного программирования и сводится с помощью кодирования ребер математическими величинами с однозначным разложением в сумму или в произведение в задачу безусловной оптимизации. **Методы**: методом решения задачи маршрутизации является генетический алгоритм, с помощью которого минимизируется до нуля целевая функция, построенная на основе структурных теорем 1, 2 и 3. **Результаты:** с помощью моделирования на ЭВМ в цикле изменения фактора кроссовера от 0,05 до 0,95 с шагом 0,05 получено решение задачи маршрутизации на графах *v*8*e*16, *v*15*e*28 и *v*25*e*50. Вероятность сходимости генетического алгоритма для графов *v*8*e*16, *v*15*e*28 составляет 100%, для графа *v*25*e*50 – 70%. Расчет гамильтоновых маршрутов на графе v8e14 составляет 100%. **Практическая значимость**: математический аппарат построения векторных целевых функций и информационно-программное обеспечение на основе генетического алгоритма может быть использовано в автоматизированной интеллектуальной системе, позволяющей решать задачи маршрутизации на графах реперов в системах управления БПЛА.

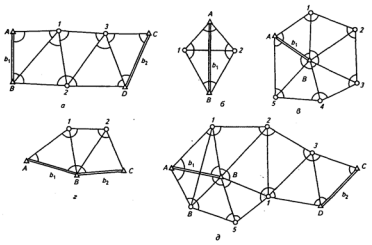
КЛЮЧЕВЫЕ СЛОВА: маршрутизация; графы; теоретико-числовые алгебраические, трансцендентные и векторные целевые функции; генетический алгоритм.

Введение

В настоящее время активно используются беспилотные летательные аппараты (БПЛА) для проведения обследования территории и расположенных на ней объектов. При облете территории БПЛА необходимо выполнить условия полноты обследования территории и безопасности полета, что может быть обеспечено построением соответствующего маршрута по реперам на местности (рис. 1).



а) карта местности



б) графы реперных точек

Рис. 1. Карта местности и графы реперных точек на ней

Для построения маршрутов на графе[1] реперов с помощью генетического алгоритма необходимо провести математическую формализацию задачи [1, 2]:

– каждая вершина графа нумеруется;

– определяется длина набора — хромосомы, соответствующая длине маршрута;

– строится целевая функция (ЦФ), аргументом которой является хромосома, состоящая из номеров вершин графа;

– начальная и конечная вершина хромосомы одинаковы, если строится замкнутый маршрут, или различны, если маршрут разомкнут;

– в хромосоме номера вершин повторяются в количестве, равном половине кратностей вершин для эйлеровых маршрутов, или повторяются по одному разу для гамильтоновых маршрутов.

Число ребер, входящих в замкнутый маршрут, содержащий все ребра эйлерова графа m = (e_1, e_2, …, e_d), определяется формулой [1]:

$$d(\mathbf{m}) = \frac{1}{2}\sum_{i=1}^{n}\rho(\upsilon_i), \tag{1}$$

где ρ(v_i) — кратность вершины $\boldsymbol{v_i}$, т.е. число ребер, сходящихся (инцидентных) в вершине v_i, n — число вершин графа.

Математическая постановка задачи нахождения множества замкнутых маршрутов на эйлеровом графе формулируется как оптимизационная [3]. Если $M = \{m\}$ множество маршрутов на эйлеровом графе $G = (V, E)$, где $V = \{v\}$ – множество вершин, $E = \{e\}$ – множество ребер, то имеем

[1] *Омельченко А.В.* Теория графов. М.: МЦНМО, 2018. 416 с.

$$M = \operatorname{argmin} d(\mathbf{m}),\ M = \{\mathbf{m}\},\ \mathbf{m} = (e_1, e_2, \ldots, e_k) \quad (2)$$
$$E \subseteq \{e_1, e_2, \ldots, e_k\},\ e_1 = (v_1, v_2), \ldots, e_k = (v_k, v_{k+1}),\ v_1 = v_{k+1},\ \{e_1, e_2, \ldots, e_k\} \backslash E \cap \{e_{i1}, e_{i2}, \ldots, e_{is}\} = \emptyset.$$

Здесь M — множество замкнутых маршрутов на графе, так как выполняется условие $v_1 = v_{k+1}$, и маршрут $\mathbf{m} = (e_1, e_2, \ldots, e_k)$ содержит минимальное число ребер, в число которых входят все различные ребра графа G, то есть решается задача минимизации числа ребер $d(\mathbf{m})$.

Концепции построения целевой функции

Множество эйлеровых маршрутов на графе может быть весьма большим, и получено в виде решения оптимизационной задачи (2).

Для определения программными средствами, что ГА построил замкнутый эйлеров маршрут, ЦФ можно представить, как сумму неотрицательных слагаемых, каждое из которых становится равным нулю на маршруте. В такой постановке задача маршрутизации формулировалась как оптимизационная, могла решаться с помощью ГА и каждый эйлеров маршрут становился оптимальным замкнутым маршрутом (ОЗМ), на котором ЦФ достигала ***глобального минимума, равного нулю***. В оптимизационной задаче (2) неизвестными становятся не величина минимума ЦФ, которая известна и равна нулю, а аргументы ЦФ, которые являются маршрутами [4, 5]. Структура генетического алгоритма[2] представлена на рис. 2.

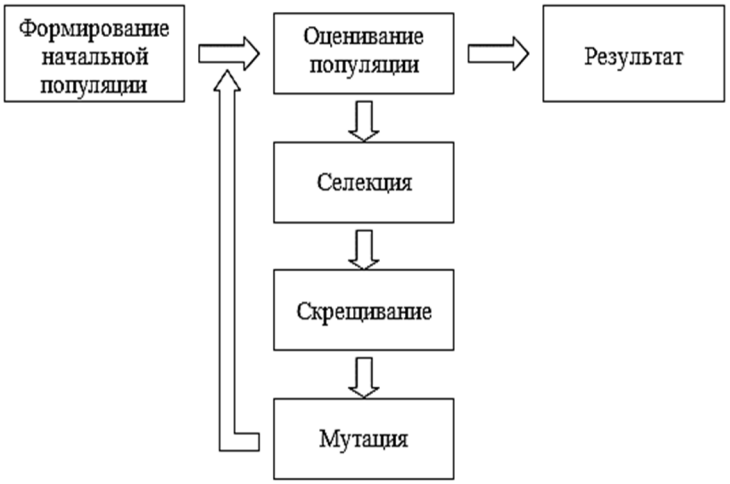


Рис. 2. Структурная схема генетического алгоритма

В ходе анализа построения ЦФ были рассмотрены три концепции: алгебраического, теоретико-числового и трансцендентного подходов (рис. 3) [6, 7, 8].

[2] *Спицин В.Г., Цой Ю.Р.* Применение генетического алгоритма для решения задач оптимизации. Учебно-методическое пособие. Томский политехнический университет, 2007. 27 с.

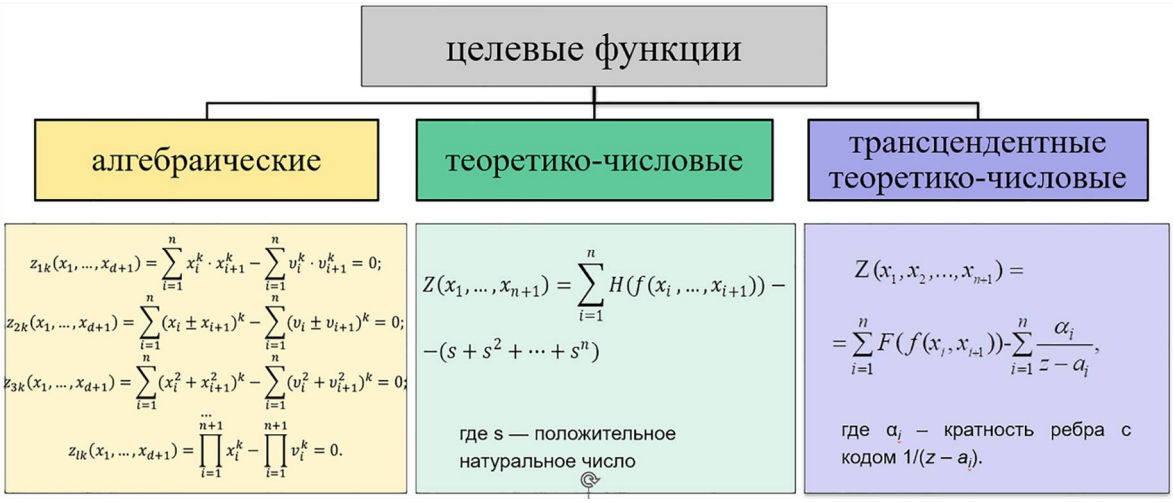
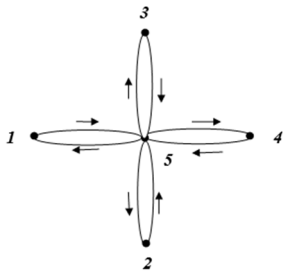


Рис. 3. Концепции построения ЦФ

Сначала рассмотрим алгебраический подход. При расчете замкнутых (эйлеровых) маршрутов на графе с малым числом реперов-вершин, целевые функции (ЦФ) строились в виде неотрицательной величины из алгебраических функций. При этом учитывалась информация о повторяемости номеров вершин графа (2) в замкнутом маршруте [2].

Следует отметить, что начальная популяция в ГА формируется случайно и представляет собой наборы (перестановки) номеров вершин графа известной длины. На графе дорог она определяется числом перекрестков плюс начальная и конечная вершина. На графе перекресток (рис. 4, *а*) численное моделирование приводило к неопределенности построения маршрута, и неопределенность становится тем выше, чем больше длина маршрута.



а) граф ***v*5*e*8**

№	Хромосома	Маршрут
1	(101 100 101 011 101 010 101 001 101)	(5,4,5,3,5,2,5,1,5)
2	(101 010 101 011 101 100 101 001 101)	(5,2,5,3,5,4,5,1,5)
3	(011 101 001 101 010 101 100 101 011)	(3,5,1,5,2,5,4,5,3)
4	(100 101 010 101 001 101 011 101 100)	(4,5,2,5,1,5,3,5,4)
5	(010 101 011 101 100 101 011 101 010)	(2,5,3,5,4,5,3,5,2)

б) набор ОЗМ на графе «перекресток»

Рис. 4. Граф *v*5*e*8 — перекресток и набор ОЗМ

На графе перекресток с пятью вершинами и восемью ребрами ***v*5*e*8** ОЗМ состоят из 8 ребер, где *v* 1, 2, 3 и 4 в маршруте встречаются по одному разу, а вершина 5 — то четыре, то пять раз. Так, **m** = (5,4,5,3,5,2,5,1,5) и (3,5,1,5,2,5,4,5,3) — ОЗМ. Первая и конечная вершина в хромосомах, набираемые в ГА случайным образом, постоянно изменяются. Неопределенность их выбора для ГА приводит к неопределенности последующих номеров вершин, которые связаны ребрами с ними и затрудняет сходимость.

При расчете на ЭВМ ГА чаще всего не сходился к маршруту. Причина оказалась в том, что первая и конечная вершина в генерируемых случайным образом наборах постоянно изменялась. Эта проблема была решена ***закреплением начальной и конечной вершин*** (рис. 4, *б*), что снижает с одной стороны длину особей, а с другой устраняет неопределенность в выборе для ГА первой и последней вершины. На последующие места ГА ставит вершину случайным образом, и, если она не связана ребром с предыдущей, то ЦФ получает большой штраф и ГА в ходе минимизации эту вершину заменит на ту, которая связана ребром.

Известно, что граф может быть представлен матрицей инцидентности или списком инцидентности. Из соображений простоты был выбран способ представления информации о ребрах в виде многочленной (разделяющей) функции двух переменных, которая при подстановке двух номеров вершин ребра в цикле по хромосоме выдает значения весов ребер. Если пара номеров вершин (x, y) является ребром на графе, то значение $f(x, y)$ попадает во множество A, в противном случае — значение попадает во множество B. Множества A и B не пересекаются: $A \cap B = \varnothing$. Ниже приведены разделяющие многочлены для графов ***v4e6*** (рис. 5), ***v8e16*** (рис. 6), ***v25e50*** (рис. 9):

$$f_{v4e6}(x, y) = x^2 + y^2; f_{v8e16}(x, y) = 2x \cdot y - 3(x + y) + 11;$$
$$f_{v25e50}(x, y) = 2(x \cdot y)^3 - 7(x - y)^4 - 13(x \cdot y)^2 - 5(x - y)^2 - 3(x + y) - 11x \cdot y - 600000;$$

Объем файла программы разделяющего многочлена для графа ***v25e50*** в пять раз меньше занимает места в байтах, чем файл с матрицей весов ребер (листинг 1).

Листинг 1

```
g = 30000000000;
G=[g      -600079 -600240 -600751      g        g        g   . . .    g;
-600079     g     -600129     g    -600043      g        g   . . .    g;
-600240  -600129    g     -598581  -596496      g        g   . . .    g;
-600751     g     -598581     g    -589459      g        g   . . .    g;
   g     -600043 -596496 -589459      g     -558075      g   . . .    g;
   g        g       g        g    -558075      g     -475269 . . .    g;
   g        g       g        g        g     -475269      g   . . .    g;
   g        g       g        g        g     -409470  -290209 . . .    g;
   g        g       g        g        g     -324231      g   . . .    g;
   g        g       g        g        g        g       20867 . . .    g;
-1327696 -631609    g        g        g        g        g   . . . 52025817;
   g        g       g        g        g        g        g   . . .    g;
   g        g       g        g        g        g        g   . . .    g;
   g        g       g        g        g        g      1139339. . .    g;
   g        g       g        g        g        g        g   . . .    g;
   g        g       g     -275596     g        g        g   . . .    g;
   g        g       g        g        g        g        g   . . .    g;
   g        g       g        g        g        g        g   . . .    g;
-1327696    g       g        g        g        g        g   . . .    g;
   g        g       g        g        g        g        g   . . .    g;
   g        g       g        g        g        g        g   . . . 328210697;
   g        g       g        g        g        g        g   . . . 375314024;
   g        g       g        g        g        g        g   . . . 26713241;
   g        g       g        g        g        g        g   . . .    g];
```

Моделирование показало, что быстродействие определения весов ребер ***v*25*e*50** по хромосоме длиной 51 разделяющим многочленом не намного быстрее, чем с помощью матрицы. Для сравнения эта разность равна: Δ = 0,10167 – 0,09633 = 0,00533 секунды. Простое математическое вычисление весов ребер многочленом является компактным в отличие от матрицы инцидентности размерности VxV = 25x25, где V равно числу ребер графа. Для графа ***v*25*e*50** это матрица 25x25 (листинг 1).

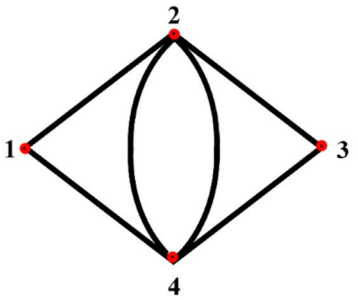
Разделяющий многочлен должен быть построен заранее и только потом используется в генетическом алгоритме. Для больших графов разделяющие многочлены могут быть построены в виде суммы симметричных многочленов, умноженных на простые числа.

Для графа с малым числом ребер ***v*4*e*6** (рис. 5) все ОЗМ могут быть получены с помощью ЦФ, учитывающей повторяемость вершин в замкнутом маршруте.

Для графа с четырьмя вершинами и шестью ребрами ***v*4*e*6**: **ρ(v_1) = ρ(v_3) = 2, ρ(v_2) = ρ(v_4) = 4, *d*(m) = 6.** Вершина с номером один закреплена и считается за одну, так как «наложив» их друг на друга получаем замкнутый маршрут.

$$z_{1k}(x_1,..,x_7)=\sum_{i=1}^{7}x_i^k-(1\cdot 1^k+2\cdot 3^k+2\cdot 4^k+1\cdot 3^k)-1^k=0.$$

Так как каждое ребро графа представляет собой пару вершин (v_i, v_{i+1}), то маршрут можно записать в виде последовательности вершин **m** = (v_1, v_2, …, v_{d+1}). Так, набор **m** = (1, 2, 4, 2, 3, 4, 1) является ОЗМ.



а) граф ***v*4*e*6**

№ п/п	ОЗМ
1	(1, 2, 3, 4, 2, 4, 1)
2	(1, 2, 4, 2, 3, 4, 1)
3	(1, 2, 4, 3, 2, 4, 1)
4	(1, 4, 2, 3, 4, 2, 1)
5	(1, 4, 2, 4, 3, 2, 1)
6	(1, 4, 3, 2, 4, 2, 1)

б) набор ОЗМ на графе «ромб»

Рис. 5. Граф *v*4*e*6 — ромб и набор ОЗМ

Решение задачи маршрутизации с увеличением числа вершин в графе приводит к возникновению противоречий, которые требовалось разрешить именно в структуре ЦФ. При увеличении числа вершин в графе на первое место выступила ***в построении ЦФ информация о ребрах***, которые кодировались математическими величинами, независимыми в аддитивных ЦФ или неприводимыми величинами в мультипликативных ЦФ [2]. На рис. 6 пунктиром показано появление ложных ребер при расчёте по алгебраической ЦФ для графа ***v*8*e*16.**

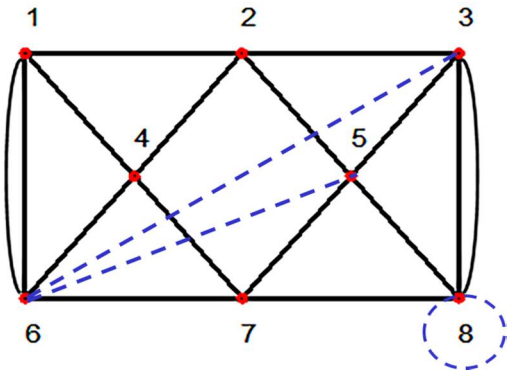


Рис. 6. Граф *v*8*e*16

Таким образом, построение ЦФ графа (обращающейся в нуль только на ОЗМ) с помощью функций, которые построены на основе информации замкнутости маршрута и частоты повторяемости номеров вершин графа, недостаточна, так как не используется информация о ребрах графа.

Избавится от этого недостатка, позволяет теоретико-числовой подход, т.е. изменить ЦФ так, чтобы всем ребрам были присвоены индивидуальные коды, сумма которых содержала бы информацию о каждом ребре, и эта информация не «смешивалась» бы с информацией о других ребрах.

Такой структурой для аддитивного случая обладают *s*-ические числа в виде суммы:

$$A = a_0 + a_1 * s + a_2 * s^2 + \ldots + a_n * s^n,$$

где s — целое число, a_i — целое число, удовлетворяющее условию $0 \le a_i < s$, A — аддитивный код графа [1, 9, 10].

Для мультипликативного случая однозначностью представления обладает разложение по степени простых чисел:

$$M = \prod_{i=1}^{n} p_i^{a_i}$$

M — мультипликативный код графа [2, 11].

Генетический алгоритм как интеллектуальная система принятия решений по построению и выбору маршрутов на графе во многом опирается на структуру ЦФ, которая должна базироваться на теоремах алгебры об однозначном разложении в сумму независимых или в произведение неприводимых величин.

Теоретико-числовой подход на основе однозначного разложения чисел для графа ***v*4*e*6** «ромб» приводит к ***целой мультипликативной ЦФ:***

$$Z(x_1, \ldots, x_7) = \prod_{i=1}^{6} F(f(x_i, x_{i+1})) - 2 \cdot 3 \cdot 5 \cdot 7^2 \cdot 11.$$

В формуле число 7 стоит во второй степени, так как ребро (2, 4) имеет кратность два.

Однозначное разложение рационального числа по степеням простых чисел приводит к ***дробной мультипликативной функции***.

$$Z(x_1,...,x_7)=\prod_{i=1}^{6}G(f(x_i,x_{i+1}))-2^{-1}\cdot 3\cdot 5^{-1}\cdot 7^{2}\cdot 11^{-1}.$$

Опираясь на единственность представления чисел в виде сумм, получим ЦФ графа ***в аддитивной форме:***

$$Z(x_1,...,x_7)=\sum_{i=1}^{6}H(f(x_i,x_{i+1}))-(s^{-2}+s^{-1}+s^{0}+s^{1}+s^{2})$$

Коды ребер могут принимать значения в алгебраическом множестве, где выполняется теорема об однозначном разложении в сумму или произведение. Такими множествами могут быть:

– натуральные целые, рациональные числа с разложением на простые числа, $2\cdot 3\cdot 5\cdot 7^2\cdot 11$ или $2\cdot 3^{-1}\cdot 5\cdot 7^{-2}\cdot 11$; $2+2^2+2^3+2^4+\ldots+2^n$;

– многочлены с разложением в сумму одночленов или на неприводимые множители: $1+x+x^2+x^3+x^4+\ldots+x^n$; $(z-a_1)^{\alpha 1}\cdot(z-a_2)^{\alpha 2}\cdot\ldots\cdot(z-a_n)^{\alpha n}$;

– с разложением многочленов в сумму по степеням переменных *x, y, z* или в произведение неприводимых множителей т.д.;

– рациональные функции с разложением на множители или в сумму элементарных (простейших) дробей;

– конечномерные векторные пространства E_n с однозначным разложением по базису;

– бесконечномерные функциональные пространства с базисными функциями рядов Фурье, многочленов Лагерра, Якоби, Бернулли и т.д.

Таким образом, имеет место теорема о построении ЦФ графа.

Теорема 1. Целевая функция оптимальных замкнутых маршрутов эйлерова графа может быть построена на основе теоретико-числовых свойств однозначного представления числа в виде произведения или суммы кодов ребер [2].

Теорема 1 об однозначном разложении, лежащая в основе построения ЦФ, позволяет использовать коды ребер из функциональных систем. Это приводит к рассмотрению трансцендентных ЦФ [3].

Для построения трансцендентных ЦФ необходимо использовать теоремы алгебры об однозначном разложении многочленов, рациональных функций, базисов в рядах Фурье и в гильбертовых пространствах.

При этом появляется противоречие для применения функциональных кодов ребер (фенотип) в ГА, т.к. наборы хромосом из кодов ребер должна быть числовыми (генотип). Для преодоления этого противоречия между фенотипом и генотипом, используются свойства трансцендентных чисел. Переводу из нечислового функционального фенотипа в числовой генотип удается разрешить с помощью ***использования в функциях-кодах вместо переменных x, y, z и т.д. алгебраически независимых трансцендентных чисел***, которые для рациональных коэффициентов ***Q*** воспринимаются как неизвестные величины или переменные, и

позволяют сохранить однозначность разложения согласно теореме 1. Это вытекает из теоремы теории чисел о том, что трансцендентные числа не могут быть корнями никакого многочлена с коэффициентами из рациональных или алгебраических чисел[3].

Эйлеровы графы имеют простую геометрическую структуру, которая заключается в четности кратностей всех вершин. Теорема 1 о целевых функциях эйлеровых графов позволяет сформулировать условие эйлеровости графа в терминах теоретико-числовых свойств кодов ребер.

Теорема 2. Граф является эйлеровым тогда и только тогда, когда существует такой набор вершин графа $n = (v_1, v_2, \ldots, v_{\#E}, v_1)$, содержащий $\#E + 1$ вершин, который обращает в нуль функцию в форме либо (3), либо (4):

$$Z_V(\mathbf{n}) = \sum_{i=1}^{\#d} cod_E(v_i, v_{i+1}) - A_E = 0; \tag{3}$$

$$Z_V(\mathbf{n}) = \prod_{i=1}^{\#V} cod_E(v_i, v_{i+1}) - M_E = 0, \tag{4}$$

где A_E — аддитивный код всех ребер графа, а M_E — мультипликативный код всех ребер графа. A_E и M_E можно назвать аддитивным и мультипликативным кодами графа. В этом случае функция (3) является аддитивной, а (4) — мультипликативной целевой функцией графа [2].

Доказательство проведем для аддитивной функции (3).

Необходимость. Если граф эйлеров, функции (3) обращается в нуль и является аддитивной ЦФ.

Достаточность. Пусть функция (3) равна нулю на некотором наборе вершин графа $n = (v_1, v_2, \ldots, v_{\#E}, v_1)$, которые содержат $\#E + 1$ вершин. Тогда n – замкнутый маршрут на графе. Число A_E раскладывается в сумму уникальных кодов и, следовательно, содержит коды всех ребер графа. Поэтому в $n = (v_1, v_2, \ldots, v_{\#E}, v_1)$ входят все ребра графа, и это — эйлеров маршрут. Значит, для любого такого набора получим, что он эйлеров маршрут. Отсюда граф - эйлеров, а функции (3) является аддитивной ЦФ эйлерова графа.

Методика построения векторных целевых функций на графах

Информация о ребрах, которые кодируются различными числами, например, степенями целого положительного числа s: s^u, s^{u+1}, …, s^{t-1}, s^t для графов ***v*15*e*28, *v*18*e*36, *v*25*e*50** (рис. 7, 8, 9), где степени u и t меняются от 1 до 50, хотя и позволяет различать их, но возникает другая проблема, а именно, в большом диапазоне изменения величин s^u и s^t.

ГА быстро расставляет в маршруте ребра с большими кодами: …, s^{t-1}, s^t и слабо чувствует ребра с малыми кодами: s^{-u}, s^{-u+1}, …, что приводит к преждевременному вырождению популяции и не построению маршрута.

Широкое многообразие множеств с однозначным разложением позволяет выбрать такое, которое снижает большой диапазон различия величин между кодами ребер. Примером такой системы являются векторные пространства E_n.

Рассмотрим граф ***v*15*e*28**, состоящий из 15 вершин и 28 ребер (рис. 7).

[3] *Гельфонд А.О.* Трансцендентные и алгебраические числа. М.: URSS, 2019. 224 с.

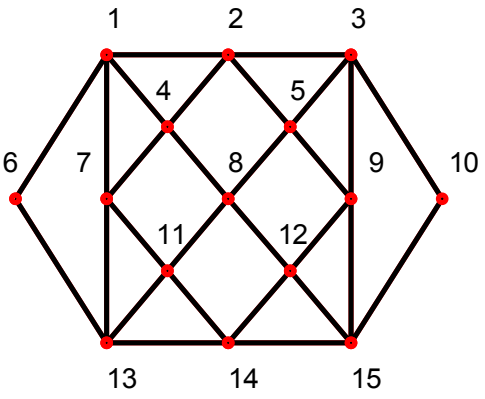


Рис. 7. Эйлеров граф *v*15*e*28

Граф является эйлеровым, так как все его вершины имеют четную кратность. Для сохранения в построении ЦФ однозначности разложения суммы в качестве кодов ребер графа используется *s*-ические одночлены s^n с разными основаниями *s* и степенями *n*, помещенные в базисные вектора a_i в качестве координат [7, 12]. Размерность векторов равна числу ребер графа ***v*15*e*28**, то есть 28: $a_i \in E_{28}$. Для предотвращения потери точности при сложении больших степеней s^n в компьютере вектора кодов ребер разбиты на четыре подпространства размерности семь. Первые семь ребер (1; 2), (1; 4), (1; 6), (1; 7), (2; 3), (2; 4) и (2; 5) имеют коды 2^7, 2^8, 2^9, 2^{10}, 2^{11}, 2^{12} и 2^{13}, которые помещены соответственно на место первой, второй и т.д. седьмой координаты в 28-мерные векторы. Остальные координаты равны нулю. Векторные коды ребер имеют вид:

$$
\begin{aligned}
&a_1 = (2^7;\ 0_2;\ 0_3;\ \ldots;\ 0_{28}),\\
&a_2 = (0_1;\ 2^8;\ 0_3;\ 0_4;\ \ldots;\ 0_{28}),\\
&a_3 = (0_1;\ 0_2;\ 2^9;\ 0_4;\ \ldots;\ 0_{28}),\ \ldots,\\
&a_7 = (0_1;\ \ldots;\ 0_6;\ 2^{13};\ 0_8;\ \ldots;\ 0_{28}).
\end{aligned}
\tag{5}
$$

Здесь нижний индекс около нулей показывает координату их местоположения в векторе размерности 28. Семь векторов из (1) образуют базис в подпространстве E_1 размерности семь.

Аналогично коды следующих семи ребер (3; 5), (3; 9), (3; 10), (4; 7), (4; 8), (5; 8), (5,9) равны: e^3, e^4, e^5, e^6, e^7, e^8, e^9, где e = 2,718281828459045… — трансцендентное число, и которые поставлены на места с восьмой по 14-той координаты в базисные вектора:

$$
\begin{aligned}
&a_8 = (0_1;\ \ldots;\ 0_7;\ e^3;\ 0_9;\ \ldots;\ 0_{28}),\\
&a_9 = (0_1;\ \ldots;\ 0_8;\ e^4;\ 0_{11};\ \ldots;\ 0_{28}),\\
&a_{10} = (0_1;\ \ldots\ 0_8;\ 0_9;\ e^5;\ 0_{11},\ \ldots;\ 0_{28}),\ \ldots,\\
&a_{14} = (0_1;\ \ldots;0_8;\ \ldots;\ 0_{13};\ e^9;\ 0_{15};\ \ldots;\ 0_{28}).
\end{aligned}
\tag{6}
$$

Векторы из (2) образуют базис в подпространстве E_2.

Следующие семь ребер (6; 13), (7; 11), (7; 13), (8; 11), (8; 12), (9; 12), (9; 15) имеют коды 3^2, 3^3, 3^4, 3^5, 3^6, 3^7 и 3^8 и векторные коды:

$$a_{15} = (0_1;\ \ldots;\ 0_{14};\ 3^2;\ 0_{16};\ \ldots;\ 0_{28}),$$
$$a_{16} = (0_1;\ \ldots;\ 0_{14};\ 0_{15};\ 3^3;\ 0_{17};\ \ldots;\ 0_{28}),$$
$$a_{17} = (0_1;\ \ldots;\ 0_{15};\ 0_{16};\ 3^4;\ 0_{18};\ \ldots;\ 0_{28}),\ \ldots,$$
$$a_{21} = (0_1;\ \ldots;\ 0_{20};\ 3^8;\ 0_{22};\ \ldots;\ 0_{28}). \quad (7)$$

Векторы из (3) образуют базис в подпространстве E_3. Последние семь ребер (10; 15), (11; 13), (11; 14), (12; 14), (12; 15), (13; 14), (14; 15) имеют коды π^2, π^3, π^4, π^5, π^6, π^7, π^8, где π = 3,141592653589793… — трансцендентное число. Векторные коды имеют значения:

$$a_{22} = (0_1;\ \ldots;\ 0_{21};\ \pi^2;\ 0_{23};\ \ldots;\ 0_{28}),$$
$$a_{23} = (0_1;\ \ldots;0_{21};\ 0_{22};\ \pi^3;\ 0_{24};\ \ldots;\ 0_{28}),$$
$$a_{24} = (0_1;\ \ldots;\ 0_{21};\ 0_{22};\ 0_{23};\ \pi^4;\ 0_{25};\ \ldots;\ 0_{28}),\ \ldots,$$
$$a_{28} = (0_1;\ \ldots;\ 0_{22};\ \ldots;\ 0_{27};\ \pi^8). \quad (8)$$

Векторы из (4) образуют базис в подпространстве E_4.

Таким образом, 28-мерное векторное пространство E распадается в прямую сумму векторных подпространств

$$E = E_1 \oplus E_2 \oplus E_3 \oplus E_4. \quad (9)$$

Представление кодов ребер графа в виде базисных векторов 28-мерного векторного пространства E распараллеливает работу ГА и снижает число слагаемых в каждом подпространстве. Для выравнивания сходимости в каждом подпространстве E_i (i = 1, ..., 4) степени оснований s = 2, e, 3, π, 5 подобраны с разными показателями так, чтобы значения 2^{13}, e^9, 3^7, π^8 были приблизительно одной величины.

Целевая функция $Z(M)$ на наборах $M = M_1 \oplus M_2 \oplus M_3 \oplus M_4$ строится как сумма четырех неотрицательных слагаемых $Z_1(M_1)$, $Z_2(M_2)$, $Z_3(M_3)$ и $Z_4(M_4)$:

$$Z(M) = Z_1(M_1) + Z_2(M_2) + Z_3(M_3) + Z_4(M_4),$$
$$Z_1(M_1) = \mathrm{abs}(\mathrm{sum}(\mathrm{sum1}(1{:}7) - a(1{:}7))),$$
$$Z_2(M_2) = \mathrm{abs}(\mathrm{sum}(\mathrm{sum1}(8{:}14) - a(8{:}14))),$$
$$Z_3(M_3) = \mathrm{abs}(\mathrm{sum}(\mathrm{sum1}(15{:}21) - a(15{:}21))),$$
$$Z_4(M_4) = \mathrm{abs}(\mathrm{sum}(\mathrm{sum1}(22{:}28) - a(22{:}28))), \quad (10)$$

где a — суммарный вектор кодов 28-мерного векторного пространства:

$$a = (2^7;\ 2^8;\ 2^9;\ 2^{10};\ 2^{11};\ 2^{12};\ 2^{13};$$
$$e^3;\ e^4;\ e^5;\ e^6;\ e^7;\ e^8;\ e^9;$$
$$3^2;\ 3^3;\ 3^4;\ 3^5;\ 3^6;\ 3^7;\ 3^8;$$
$$\pi^2;\ \pi^3;\ \pi^4;\ \pi^5;\ \pi^6;\ \pi^7;\ \pi^8). \quad (11)$$

Величины $a(1{:}7)$, $a(8{:}14)$, $a(15{:}21)$, $a(22{:}28)$ равны частным суммам координат вектора (6):

$$a(1{:}7) = 2^7 + 2^8 + 2^9 + 2^{10} + 2^{11} + 2^{12} + 2^{13},$$
$$a(8{:}14) = e^3 + e^4 + e^5 + e^6 + e^7 + e^8 + e^9,$$
$$a(15{:}21) = 3^2 + 3^3 + 3^4 + 3^5 + 3^6 + 3^7 + 3^8,$$
$$a(22{:}28) = \pi^2 + \pi^3 + \pi^4 + \pi^5 + \pi^6 + \pi^7 + \pi^8. \quad (12)$$

Маршрут на графе ***v*15*e*28** с помощью ГА получается при одновременном обнулении четырех слагаемых целевой функции (10). Левые части слагаемых $Z_i(M_i)$ ЦФ (10) на наборах M_i суммируют векторные коды ребер графа в переменной sum1 с помощью цикла (листинг 2):

Листинг 2

```
sum1=0;
for i=1:28
sum1=sum1 + pasv15e28vector(m(i), m(i+1));
end
```

Цикл из *листинга* 2 перебирает инцидентные между собой ребра из набора M: $(m_i, m_{i+1}) = (m(i), m(i+1))$, и с помощью пользовательской подпрограммы pasv15e28vector (разработанной авторами) ставит им в соответствие векторные коды.

При появлении в наборе M пар (m_i, m_{i+1}), которые не являются ребрами на графе (ложные ребра), но могут быть сгенерированы ГА, функция pasv15e28vector всем таким ребрам ставит в соответствие один и тот же векторный код-штраф straf значительно больший, чем степени 2^{13}, e^9, 3^8, π^8:

$$\text{straf} >> \max(2^{13}, e^9, 3^8, \pi^8)\cdot\text{ones}(28), \quad (13)$$

где ones(28) — единичный вектор 28 мерного пространства E. Так как код-штраф ложных ребер в ЦФ (10) не компенсируется в правой части, то ГА, реализующий оптимизацию минимизации, от таких пар избавляется в первых поколениях.

Численное моделирование расчета маршрутов на графе ***v*15*e*28** было проведено при различных начальных условиях по числу поколений, наборов-особей и признаках останова, но впервые при настройках:

- 'Generations', 160,
- 'PopulationSize', 12000,
- 'StallGenLimit', 144,
- 'Vectorized', 'on'.

При изменении фактора кроссовера от 0,05 до 0,95 с шагом 0,1 (10 различных значений фактора) [13, 14, 15] ГА показал 100% сходимость к 10 маршрутам (рис. 7), приведенным в табл. 1.

Моделирование проводилось на двуядерном компьютере с частотой процессора 1,76 ГГц. Для более мощных компьютеров расчеты могут составить секунды или доли секунды.

Таблица 1

Результаты расчета ГА маршрутов на графе ***v*15*e*28** с помощью ЦФ (10)

Фактор кроссовера	Маршруты на графе
0,5	(1 4 2 5 8 12 14 15 10 3 9 12 15 9 5 3 2 1 6 13 14 11 7 4 8 11 13 7 1)
1,5	(1 4 2 5 9 3 10 15 12 14 15 9 12 8 5 3 2 1 6 13 14 11 13 7 11 8 4 7 1)
2,5	(1 2 3 5 9 12 15 9 3 10 15 14 11 13 14 12 8 5 2 4 1 7 11 8 4 7 13 6 1)
3,5	(1 2 5 3 9 12 15 10 3 2 4 7 11 8 12 14 13 7 1 6 13 11 14 15 9 5 8 4 1)
4,5	(1 6 13 14 15 10 3 9 5 2 1 7 11 8 12 9 15 12 14 11 13 7 4 8 5 3 2 4 1)
5,5	(1 7 11 13 14 11 8 5 2 3 5 9 3 10 15 9 12 14 15 12 8 4 7 13 6 1 2 4 1)
6.5	(1 4 7 11 8 4 2 1 6 13 14 15 9 12 8 5 23 9 5 3 10 15 12 14 11 3 7 1)
7,5	(1 7 13 11 7 4 2 3 5 9 12 14 11 8 4 1 6 13 14 15 10 3 9 15 12 8 5 2 1)
8,5	(1 6 13 11 14 12 8 5 3 9 12 15 9 5 2 1 4 8 11 7 13 14 15 10 3 2 4 7 1)
9,5	(1 6 13 14 11 13 7 11 8 4 1 7 4 2 5 9 15 14 12 8 5 3 10 15 12 9 3 2 1)

Если рассмотреть граф ***v*18*e*36** (рис. 8), то для построения векторной ЦФ векторные коды ребер будут базисными векторами в пространстве *E* размерности 36, которое разобьем в сумму четырех подпространств размерности 9.

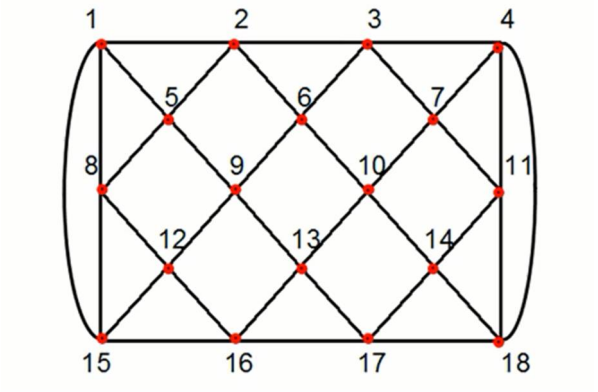


Рис. 8. Эйлеров граф *v*18*e*36

Первым девяти ребрам (1; 2), (1; 5), (1; 8), (1; 15), (2; 3), (2; 5), (2; 6), (3; 4), (3;6) поставим 2-ические коды: 2^{-1}, 2^3, 2^5, 2^6, 2^7, 2^8, 2^{10}, 2^{11}, 2^{13}, которым соответствуют векторные коды:

$$
\begin{aligned}
a_1 &= (2^{-1}; 0_2; 0_3; \ldots; 0_{36}),\\
a_2 &= (0_1; 2^3; 0_3; 0_4; \ldots; 0_{36}),\\
a_3 &= (0_1; 0_2; 2^5; 0_4; \ldots; 0_{36}), \ldots,\\
a_9 &= (0_1; \ldots; 0_8; 2^{13}; 0_{10}; \ldots; 0_{36}).
\end{aligned} \tag{14}
$$

Следующим девяти ребрам: (3; 7), (4; 7), (4; 11), (4; 18), (5; 8), (5; 9), (6; 9), (6; 10), (7; 10) поставим *e*-ические коды: e, e^2, e^3, e^4, e^5, e^6, e^7, e^8, e^9, которым соответствуют векторные коды:

$$a_{10} = (0_1; \ldots; 0_9; e; 0_{11}; \ldots; 0_{36}),$$
$$a_{11} = (0_1; \ldots; 0_{10}; e^2; 0_{12}; \ldots; 0_{36}),$$
$$a_{12} = (0_1; \ldots\ 0_{11}; e^3; 0_{13}, \ldots; 0_{36}), \ldots,$$
$$a_{18} = (0_1; \ldots; 0_{17}; e^9; 0_{19}; \ldots; 0_{36}). \quad (15)$$

Следующим девяти ребрам: (7; 11), (8; 12), (8; 15), (9; 12), (9; 13), (10; 13), (10; 14), (11; 14), (11; 18) поставим 3-ические коды: 3^{-1}, 3, 3^2, 3^3, 3^4, 3^5, 3^6, 3^7, 3^8 которым соответствуют векторные:

$$a_{19} = (0_1; \ldots; 0_{18}; 3^{-1}; 0_{20}; \ldots; 0_{36}),$$
$$a_{20} = (0_1; \ldots; 0_{19}; 3; 0_{21}; \ldots; 0_{36}),$$
$$a_{21} = (0_1; \ldots; 0_{20}; 3^2; 0_{22}; \ldots; 0_{36}), \ldots,$$
$$a_{27} = (0_1; \ldots; 0_{26}; 3^8; 0_{29}; \ldots; 0_{36}). \quad (16)$$

Последние девять ребер: (12; 5), (12; 16), (13; 16), (13; 17), (14; 17), (14; 18), (15; 16), (16; 17), (17; 18) имеют коды π^0, π, π^2, π^3, π^4, π^5, π^6, π^7, π^8. Векторные коды имеют значения:

$$a_{28} = (0_1; \ldots; 0_{27}; \pi^0; 0_{29}; \ldots; 0_{36}),$$
$$a_{29} = (0_1; \ldots; 0_{28}; \pi; 0_{30}; \ldots; 0_{36}),$$
$$a_{30} = (0_1; \ldots; 0_{28}; 0_{29}; \pi^4; 0_{31}; \ldots; 0_{36}), \ldots,$$
$$a_{36} = (0_1; \ldots; 0_{34}; 0_{35}; \pi^8). \quad (17)$$

Таким образом, 36-мерное векторное пространство E распадается в прямую сумму четырех векторных подпространств

$$E = E_1 \oplus E_2 \oplus E_3 \oplus E_4, \quad (18)$$

что приводит к разложению маршрута на графе ***v*18*e*36** в сумму векторных кодов $M = M_1 \oplus M_2 \oplus M_3 \oplus M_4$ и ЦФ $Z(M)$ на наборах M строится как сумма четырех неотрицательных слагаемых $Z_1(M_1)$, $Z_2(M_2)$, $Z_3(M_3)$ и $Z_4(M_4)$ аналогично выражению (15):

$$Z(M) = Z_1(M_1) + Z_2(M_2) + Z_3(M_3) + Z_4(M_4), \quad (19)$$
$$Z_1(M_1) = \mathrm{abs}(\mathrm{sum}(\mathrm{sum1}(1{:}9) - a(1{:}9))),$$
$$Z_2(M_2) = \mathrm{abs}(\mathrm{sum}(\mathrm{sum1}(10{:}18) - a(10{:}18))),$$
$$Z_3(M_3) = \mathrm{abs}(\mathrm{sum}(\mathrm{sum1}(19{:}27) - a(19{:}27))),$$
$$Z_4(M_4) = \mathrm{abs}(\mathrm{sum}(\mathrm{sum1}(28{:}36) - a(28{:}36))),$$

где a – суммарный вектор кодов 36-мерного векторного пространства:

$$a = (2^{-1}; 2^3; 2^5; 2^6; 2^7; 2^8; 2^{10}; 2^{11}; 2^{13};$$
$$e; e^2; e^3; e^4; e^5; e^6; e^7; e^8, e^9;$$
$$3^{-1}; 3; 3^2; 3^3; 3^4; 3^5; 3^6; 3^7; 3^8;$$
$$\pi^0; \pi; \pi^2; \pi^3; \pi^4; \pi^5; \pi^6; \pi^7; \pi^8). \quad (20)$$

Величины $a(1{:}9)$, $a(10{:}18)$, $a(19{:}27)$, $a(28{:}36)$ равны частным суммам координат вектора (20):

$$\begin{aligned} a(1{:}9) &= 2^{-1} + 2^3 + 2^5 + 2^6 + 2^7 + 2^8 + 2^{10} + 2^{11} + 2^{13}, \\ a(10{:}18) &= e + e^2 + e^3 + e^4 + e^5 + e^6 + e^7 + e^8 + e^9, \\ a(19{:}27) &= 3^{-1} + 3 + 3^2 + 3^3 + 3^4 + 3^5 + 3^6 + 3^7 + 3^8, \\ a(28{:}36) &= \pi^0 + \pi + \pi^2 + \pi^3 + \pi^4 + \pi^5 + \pi^6 + \pi^7 + \pi^8. \end{aligned} \tag{21}$$

Численное моделирование расчета маршрутов на графе при настройках: 'Generations', 120, 'PopulationSize', 36000, 'StallGenLimit', 110, 'Vectorized', 'on' и изменении фактора кроссовера от 0,5 до 0,95 с шагом 0,05 (10 различных значений фактора) ГА показал 100% сходимость к 10 маршрутам, приведенным в табл. 2.

Таблица 2

Результаты расчета ГА маршрутов на графе ***v*18*e*36** с помощью ЦФ (19)

Фактор кроссовера	Маршруты на графе
0,5	(1 2 6 3 4 18 11 14 17 18 14 10 7 11 4 7 3 2 5 9 13 17 16 12 9 6 10 13 16 15 12 8 5 1 8 15 1)
0,55	(1 8 15 12 8 5 9 12 16 13 9 6 3 4 7 3 2 5 1 15 16 17 18 4 11 18 14 10 7 11 14 17 13 10 6 2 1)
0,6	(1 2 3 7 11 14 18 4 11 18 17 13 16 17 14 10 13 9 5 1 15 12 16 15 8 12 9 6 10 7 4 3 6 2 5 8 1)
0,65	(1 8 15 16 17 18 4 7 10 14 18 11 14 17 13 16 12 15 1 5 9 12 8 5 2 6 10 13 9 6 3 4 11 7 3 2 1)
0,7	(1 8 12 15 16 13 10 14 17 18 11 14 18 7 11 4 3 7 10 6 3 2 5 1 15 8 5 9 12 16 17 13 9 6 2 1)
0,75	(1 15 16 13 10 7 3 6 10 14 17 18 11 4 18 14 11 4 3 2 6 9 12 8 15 2 16 17 13 9 5 2 1 8 5 1)
0.8	(1 5 2 3 4 11 7 10 13 17 16 15 8 12 15 1 8 5 9 12 16 13 9 6 10 14 11 18 14 17 18 4 7 3 6 2 1)
0,85	(1 15 16 12 15 8 5 2 3 6 10 14 18 4 3 7 11 14 17 18 11 4 7 10 13 17 16 13 9 5 1 2 6 9 12 8 1)
0,9	(1 5 2 1 8 5 9 6 3 7 4 3 2 6 10 7 11 4 18 11 14 10 13 16 12 9 13 17 18 14 17 16 15 8 12 15 1)
0,95	(1 15 16 17 18 11 7 10 14 11 4 18 14 17 13 10 6 2 5 9 6 3 4 7 3 2 1 8 12 9 13 16 12 15 8 5 1)

Следует отметить, что ЦФ (10) и (19) могут быть перепрофилированы под умножение кодов ребер путем замены нулей в векторах на единицы, а сложение векторов на умножение [4]. Построенные таким образом ЦФ становятся мультипликативными, а векторные пространства заменить на прямое произведение мультипликативных групп. Примерами групп по умножению могут быть группы, порождаемые простыми числами: $G_p = \{p^i \mid i \in Z\}$.

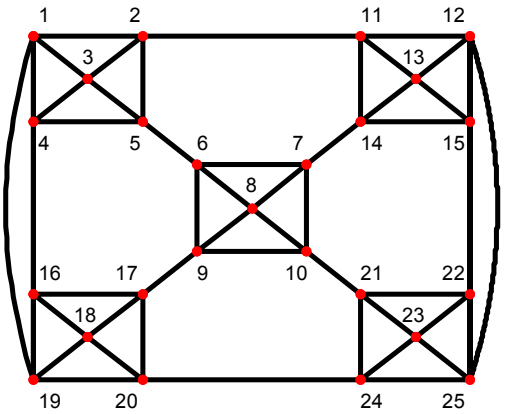


Рис. 9. Граф *v*25*e*50

Построенная ЦФ на основе векторного пространства E_n, получила название векторной ЦФ (ВЦФ). Суть метода кодирования ребер и построения ВЦФ состоит в разбиении кодов ребер по подпространствам. Так для графа с 50-ю ребрами можно разбить коды ребер по 10 подпространствам размерности пять. В каждом 5-ти мерном подпространстве можно использовать степени: s, s^2, s^3, s^4, s^5, умножая их на единичные базисные вектора для аддитивной ЦФ или простые числа: 2, 3, 5, 7, 11 для мультипликативной ВЦФ.

Ниже представлены векторные коды ребер графа ***v*25*e*50** при построении аддитивной ЦФ в 50-мерном пространстве E_{50}:

$f(1, 2) = a_1 = (s^6;\ 0_2;\ 0_3;\ 0_4;\ 0_5;\ 0_6;\ \ldots;\ 0_{50})$;
$f(1, 3) = a_2 = (0_1;\ s^{12};\ 0_3;\ 0_4;\ 0_5;\ 0_6;\ \ldots;\ 0_{50})$;
$f(1, 4) = a_3 = (0_1;\ 0_2;\ s^{17};\ 0_4;\ 0_5;\ 0_6;\ \ldots;\ 0_{50})$;
$f(1, 19) = a_4 = (0_1;\ 0_2;\ 0_3;\ s^{23};\ 0_5;\ 0_6;\ \ldots;\ 0_{50})$;
$f(2, 3) = a_5 = (0_1;\ 0_2;\ 0_3;\ 0_4;\ s^{29};\ 0_6;\ \ldots;\ 0_{50})$;

$f(2, 5) = a_6 = (0_1;\ \ldots;\ 0_5;\ 2^4;\ 0_7;\ 0_8;\ 0_9;\ 0_{10};\ 0_{11};\ \ldots;\ 0_{50})$;
$f(2, 11) = a_7 = (0_1;\ \ldots;\ 0_5;\ 0_6;\ 2^8;\ 0_8;\ 0_9;\ 0_{10};\ 0_{11};\ \ldots;\ 0_{50})$;
$f(3, 4) = a_8 = (0_1;\ \ldots;\ 0_5;\ 0_6;\ 0_7;\ 2^{12};\ 0_9;\ 0_{10};\ 0_{11};\ \ldots;\ 0_{50})$;
$f(3, 5) = a_9 = (0_1;\ \ldots;\ 0_5;\ 0_6;\ 0_7;\ 0_8;\ 2^{16};\ 0_{10};\ 0_{11};\ \ldots;\ 0_{50})$;
$f(4, 5) = a_{10} = (0_1;\ \ldots;\ 0_5;\ 0_6;\ 0_7;\ 0_8;\ 0_9;\ 2^{20};\ 0_{11};\ \ldots;\ 0_{50})$;

$f(4, 16) = a_{11} = (0_1;\ \ldots;\ 0_{10};\ e^3;\ 0_{12};\ 0_{13};\ 0_{14};\ 0_{15};\ 0_{16};\ \ldots;\ 0$
$f(5, 6) = a_{12} = (0_1;\ \ldots;\ 0_{10};\ 0_{11};\ e^5;\ 0_{13};\ 0_{14};\ 0_{15};\ 0_{16};\ \ldots;\ 0_{50}$
$f(6, 7) = a_{13} = (0_1;\ \ldots;\ 0_{10};\ 0_{11};\ 0_{12};\ e^8;\ 0_{14};\ 0_{15};\ 0_{16};\ \ldots;\ 0_{50}$
$f(6, 8) = a_{14} = (0_1;\ \ldots;\ 0_{10};\ 0_{11};\ 0_{12};\ 0_{13};\ e^{11};\ 0_{15};\ 0_{16};\ \ldots;\ 0_5$
$f(6, 9) = a_{15} = (0_1;\ \ldots;\ 0_{10};\ 0_{11};\ 0_{12};\ 0_{13};\ 0_{14};\ e^{14};\ 0_{16};\ \ldots;\ 0_5$

$f(7, 8) = a_{16} = (0_1;\ \ldots;\ 0_{15};\ 2{\cdot}3^2;\ 0_{12};\ 0_{13};\ 0_{14};\ 0_{15};\ 0_{16};\ \ldots;\ 0_{50})$;
$f(7, 10) = a_{17} = (0_1;\ \ldots;\ 0_{15};\ 0_{16};\ 3^5;\ 0_{13};\ 0_{14};\ 0_{15};\ 0_{16};\ \ldots;\ 0_{50})$;
$f(7, 14) = a_{18} = (0_1;\ \ldots;\ 0_{15};\ 0_{16};\ 0_{17};\ 2{\cdot}3^7;\ 0_{14};\ 0_{15};\ 0_{16};\ \ldots;\ 0_{50})$;
$f(8, 9) = a_{19} = (0_1;\ \ldots;\ 0_{15};\ 0_{16};\ 0_{17};\ 0_{18};\ 3^{10};\ 0_{15};\ 0_{16};\ \ldots;\ 0_{50})$;
$f(8, 10) = a_{20} = (0_1;\ \ldots;\ 0_{15};\ 0_{16};\ 0_{17};\ 0_{18};\ 0_{19};\ 3^{13};\ 0_{16};\ \ldots;\ 0_{50})$;

$f(9, 10) = a_{21} = (0_1;\ \ldots;\ 0_{20};\ 2\pi^3;\ 0_{22};\ 0_{23};\ 0_{24};\ 0_{25};\ 0_{26};\ \ldots;\ 0_5$
$f(9, 17) = a_{22} = (0_1;\ \ldots;\ 0_{20};\ 0_{21};\ 3\pi^5;\ 0_{23};\ 0_{24};\ 0_{25};\ 0_{26};\ \ldots;\ 0_5$
$f(10, 21) = a_{23} = (0_1;\ \ldots;\ 0_{20};\ 0_{21};\ 0_{22};\ 2\pi^8;\ 0_{24};\ 0_{25};\ 0_{26};\ \ldots;\ 0_5$
$f(11, 12) = a_{24} = (0_1;\ \ldots;\ 0_{20};\ 0_{21};\ 0_{22};\ 0_{23};\ 3\pi^{11};\ 0_{25};\ 0_{26};\ \ldots;\ 0$
$f(11, 13) = a_{25} = (0_1;\ \ldots;\ 0_{20};\ 0_{21};\ 0_{22};\ 0_{23};\ 0_{24};\ 3\pi^{14};\ 0_{26};\ \ldots;\ 0$

$f(11, 14) = a_{26} = (0_1;\ \ldots;\ 0_{25};\ 3{\cdot}5;\ 0_{22};\ 0_{23};\ 0_{29};\ 0_{30};\ 0_{31};\ \ldots;\ 0_{50})$;
$f(12, 13) = a_{27} = (0_1;\ \ldots;\ 0_{25};\ 0_{26};\ 2{\cdot}5^3;\ 0_{23};\ 0_{29};\ 0_{30};\ 0_{31};\ \ldots;\ 0_{50})$;
$f(12, 15) = a_{28} = (0_1;\ \ldots;\ 0_{25};\ 0_{26};\ 0_{27};\ 2{\cdot}5^5;\ 0_{29};\ 0_{30};\ 0_{31};\ \ldots;\ 0_{50})$;
$f(12, 25) = a_{29} = (0_1;\ \ldots;\ 0_{25};\ 0_{26};\ 0_{27};\ 0_{28};\ 5^7;\ 0_{30};\ 0_{31};\ \ldots;\ 0_{50})$;
$f(13, 14) = a_{30} = (0_1;\ \ldots;\ 0_{25};\ 0_{26};\ 0_{27};\ 0_{28};\ 0_{29};\ 4{\cdot}5^8;\ 0_{31};\ \ldots;\ 0_{50})$;

$f(13, 15) = a_{31} = (0_1;\ \ldots;\ 0_{30};\ 2{\cdot}7;\ 0_{32};\ 0_{33};\ 0_{34};\ 0_{35};\ 0_{36};\ \ldots;\ 0$
$f(14, 15) = a_{32} = (0_1;\ \ldots;\ 0_{30};\ 0_{31};\ 7^3;\ 0_{33};\ 0_{34};\ 0_{35};\ 0_{36};\ \ldots;\ 0$
$f(15, 22) = a_{33} = (0_1;\ \ldots;\ 0_{30};\ 0_{31};\ 0_{32};\ 2{\cdot}7^4;\ 0_{34};\ 0_{35};\ 0_{36};\ \ldots;\ 0$
$f(16, 17) = a_{34} = (0_1;\ \ldots;\ 0_{30};\ 0_{31};\ 0_{32};\ 0_{33};\ 5{\cdot}7^5;\ 0_{35};\ 0_{36};\ \ldots;\ 0$
$f(16, 18) = a_{35} = (0_1;\ \ldots;\ 0_{30};\ 0_{31};\ 0_{32};\ 0_{33};\ 0_{34};\ 2{\cdot}7^7;\ 0_{36};\ \ldots;\ 0$

$f(16, 19) = a_{36} = (0_1;\ \ldots;\ 0_{35};\ 11;\ 0_{37};\ 0_{38};\ 0_{39};\ 0_{40};\ 0_{41};\ \ldots;\ 0$
$f(17, 18) = a_{37} = (0_1;\ \ldots;\ 0_{35};\ 0_{36};\ 3{\cdot}11^2;\ 0_{38};\ 0_{39};\ 0_{40};\ 0_{41};\ \ldots;\ 0$
$f(17, 20) = a_{38} = (0_1;\ \ldots;\ 0_{35};\ 0_{36};\ 0_{37};\ 3{\cdot}11^3;\ 0_{39};\ 0_{40};\ 0_{41};\ \ldots;\ 0$
$f(18, 19) = a_{39} = (0_1;\ \ldots;\ 0_{35};\ 0_{36};\ 0_{37};\ 0_{38};\ 6{\cdot}11^4;\ 0_{40};\ 0_{41};\ \ldots;$
$f(18, 20) = a_{40} = (0_1;\ \ldots;\ 0_{35};\ 0_{36};\ 0_{37};\ 0_{38};\ 0_{39};\ 9{\cdot}11^5;\ 0_{41};\ \ldots;$

$f(19, 20) = a_{41} = (0_1;\ \ldots;\ 0_{40};\ 13;\ 0_{42};\ 0_{43};\ 0_{44};\ 0_{45};\ 0_{46};\ \ldots;\ 0$
$f(20, 24) = a_{42} = (0_1;\ \ldots;\ 0_{40};\ 0_{41};\ 2{\cdot}13^2;\ 0_{43};\ 0_{44};\ 0_{45};\ 0_{46};\ \ldots;\ 0$
$f(21, 22) = a_{43} = (0_1;\ \ldots;\ 0_{40};\ 0_{41};\ 0_{42};\ 2{\cdot}13^3;\ 0_{44};\ 0_{45};\ 0_{46};\ \ldots;\ 0$
$f(21, 23) = a_{44} = (0_1;\ \ldots;\ 0_{40};\ 0_{41};\ 0_{42};\ 0_{43};\ 3{\cdot}13^4;\ 0_{45};\ 0_{46};\ \ldots;$
$f(21, 24) = a_{45} = (0_1;\ \ldots;\ 0_{40};\ 0_{41};\ 0_{42};\ 0_{43};\ 0_{44};\ 4{\cdot}13^5;\ 0_{46};\ \ldots;$

$f(22, 23) = a_{46} = (0_1;\ \ldots;\ 0_{45};\ 17;\ 0_{47};\ 0_{48};\ 0_{49};\ 0_{50})$;
$f(22, 25) = a_{47} = (0_1;\ \ldots;\ 0_{45};\ 0_{46};\ 17^2;\ 0_{48};\ 0_{49};\ 0_{50})$;
$f(23, 24) = a_{48} = (0_1;\ \ldots;\ 0_{45};\ 0_{46};\ 0_{47};\ 17^3;\ 0_{49};\ 0_{50})$;
$f(23, 25) = a_{49} = (0_1;\ \ldots;\ 0_{45};\ 0_{46};\ 0_{47};\ 0_{48};\ 17^4;\ 0_{50})$;
$f(24, 25) = a_{50} = (0_1;\ \ldots;\ 0_{45};\ 0_{46};\ 0_{47};\ 0_{48};\ 0_{49};\ 17^5)$.

Ложные ребра, которые может сгенерировать генетический алгоритм, штрафуются большим числом $3{\cdot}10^{10}$. Таким образом, 50-мерное векторное пространство E_{50} распадается в прямую сумму десяти 5-мерных векторных подпространств:

$$E_{50} = A_1 \oplus A_2 \oplus A_3 \oplus A_4 \oplus A_5 \oplus A_6 \oplus A_7 \oplus A_8 \oplus A_9 \oplus A_{10}. \qquad (22)$$

Маршрут на графе ***v*25*e*50** с помощью ГА получается при одновременном обнулении 10 слагаемых целевой функции:

$$Z(\mathbf{m}) = Z_1(\mathbf{m}_1) + Z_2(\mathbf{m}_2) + \ldots + Z_{10}(\mathbf{m}_{10}),$$
$$Z_1(\mathbf{m}_1) = abs(sum(sum1(1:5) - a(1:5))),$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$Z_{10}(\mathbf{m}_{10}) = abs(sum(sum1(46:50) - a(46:50)))$$

где a — суммарный вектор базисных кодов 50-мерного пространства:

$a = (s^6, s^{12}, s^{17}, s^{23}, s^{29}, 2^4, 2^8, 2^{12}, 2^{16}, 2^{20}, e^3, 2e^5, 2e^8, e^{11}, e^{14}, 2{\cdot}3^2, 3^5, 2{\cdot}3^7, 3^{10}, 3^{13}, 2{\cdot}3^4, 3^5, 2{\cdot}3^7, 3^{10}, 3^{13}, 2\pi^2, 3\pi^4, 2\pi^7, 3\pi^9, 3\pi^{12}, 3{\cdot}5, 2{\cdot}5^3, 2{\cdot}5^5, 5^7, 4{\cdot}5^8, 2{\cdot}7, 7^3, 2{\cdot}7^4, 5{\cdot}7^5, 2{\cdot}7^7, 11, 3{\cdot}11^2, 3{\cdot}11^3, 6{\cdot}11^4, 9{\cdot}11^5, 13, 2{\cdot}13^2, 2{\cdot}13^3, 3{\cdot}13^4, 4{\cdot}13^5, 17, 17^2, 17^3, 17^4, 17^5)$.

Здесь величины *s*-ических кодов выравнены между всеми 5-мерными подпространствами. Подсчет ОЗМ с помощью ВЦФ на графе ***v*25*e*50** произведен при настройках 'Generations', 360, 'PopulationSize', 11700, StallGenLimit', 220 и меняющимся в цикле параметра *пс* — фактора кроссовера от 0,05 до 0,85 с шагом 0,1.

Таблица 3

Маршруты на графе ***v*25*e*50**

пс	ОЗМ графа v25e50
0,05	1 3 2 1 19 20 18 16 19 18 17 9 8 10 9 6 7 10 21 24 23 21 22 15 13 11 2 5 4 3 5 6 8 7 14 13 12 11 14 15 12 25 22 23 25 24 20 17 16 4 1
0,15	1 2 11 12 15 14 13 15 22 23 24 20 19 16 18 20 17 16 4 5 6 7 14 11 13 12 25 23 21 24 25 22 21 10 7 8 10 9 8 6 9 17 18 19 1 3 2 5 3 41
0,25	1 4 3 5 4 16 18 20 17 9 8 6 9 10 21 24 25 12 13 15 22 21 23 25 22 23 24 20 19 16 17 18 19 1 2 11 12 15 14 13 11 14 7 8 10 7 6 5 2 3 1
0,45	1 4 3 1 2 11 12 25 23 21 24 25 22 23 24 20 19 18 17 9 10 8 6 7 10 21 22 15 12 13 14 15 13 11 14 7 8 9 6 5 2 3 5 4 16 17 20 18 16 19 1
0,55	1 4 16 19 18 20 17 18 16 17 9 8 6 5 4 3 2 1 3 5 2 11 13 14 15 12 25 22 23 25 24 21 22 15 13 12 11 14 7 10 8 7 6 9 10 21 23 24 20 19 1
0,65	1 2 3 4 16 17 18 20 19 18 16 19 1 3 5 2 11 12 25 24 20 17 9 6 7 14 15 12 13 11 14 13 15 22 25 23 22 21 23 24 21 10 9 8 7 10 8 6 5 4 1
0,85	1 19 16 18 19 20 17 9 8 7 14 15 13 14 11 13 12 25 23 24 25 22 15 12 11 2 1 4 5 2 3 5 6 8 10 9 6 7 10 21 22 23 21 24 20 18 17 16 4 3 1

ВЦФ, устраняя потерю точности расчета маршрутов в графах большой размерности, открывают новый подход к исследованию работы ГА при обработке длинных хромосом с учетом структуры самих графов.

Расчет гамильтоновых маршрутов

На рис.10 приведены результаты на графе ***v*8*e*14** по ЦФ с помощью ГА.

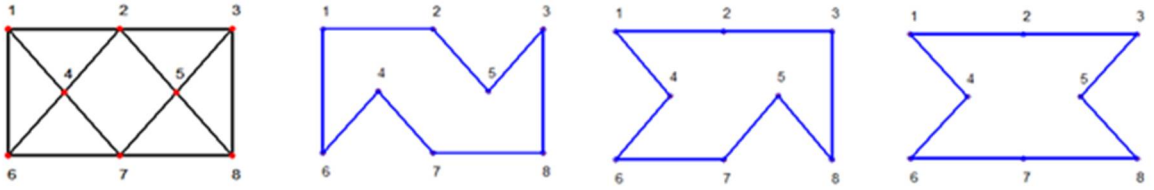


Рис. 10. Граф *v*8*e*14 и некоторые гамильтоновы маршруты на нем

Целевая функция на гамильтоновых графах построена на основании Теоремы 3:

Теорема 3. Граф является гамильтоновым тогда и только тогда, когда два уравнения аддитивной системы (23) или мультипликативной системы (24)

$$Z_E(\mathbf{n}) = \sum_{i=1}^{\#V} addweight_E(v_i, v_{i+1}) - A_E = 0;$$
$$Z_V(\mathbf{n}) = \sum_{i=1}^{\#V} cod_V(v_i) - A_V = 0 \quad (23)$$

или

$$Z_E(\mathbf{n}) = \prod_{i=1}^{\#V} multweight_E(v_i, v_{i+1}) - M_E = 0;$$
$$Z_V(\mathbf{n}) = \prod_{i=1}^{\#V} cod_V(v_i) - M_V = 0. \quad (24)$$

При тривиальных весах ребер $addweight_E(v_i, v_{i+1}) = 1$, $A_E = \#V$, либо $multweight_E(v_i, v_{i+1}) = a$, $M_E = a^{\#V}$ и уникальных кодах $cod_V(v_i)$ вершин, удовлетворяющих условиям однозначного разложения некоторой математической величины, имеют решение хотя бы на одной перестановке вершин $n = (v_1, v_2, \ldots, v_{\#V}, v_{\#V+1})$, содержащей $\#V+1$ вершин графа.

Доказательство проведем для аддитивной системы уравнений (23).

Необходимость. Пусть имеется граф $\Gamma = (V, E)$. Рассмотрим систему уравнений (23). Пусть она имеет решение на наборе вершин $n = (v_1, v_2, \ldots, v_k, v_{k+1})$ графа Γ и $k = \#V$.

Тогда первое уравнение системы (23) означает, что при выполнении $A_E = \#V$ и $addweight_E(v_i, v_{i+1}) = 1$ имеем, что $n = (v_1, v_2, \ldots, v_{\#V}, v_1)$ — замкнутый маршрут на графе, состоящий из ребер (v_i, v_{i+1}) графа Γ. Второе уравнение (23) означает, что сумма кодов вершин дает величину A_V. Из однозначности представления A_V в виде суммы следует, что набор n содержит все вершины графа. Следовательно, $n = (v_1, v_2, \ldots, v_{\#V}, v_1)$ — замкнутый маршрут на графе Γ, содержащий все вершины точно один раз. Поэтому $n = (v_1, v_2, \ldots, v_{\#V}, v_1)$ – гамильтонов маршрут, а граф Γ — гамильтонов.

Достаточность вытекает из определения гамильтонова графа. Из однозначности разложения величины A_V следует, что $n = (v_1, v_2, \ldots, v_{\#V}, v_1)$ — гамильтонов маршрут, и поэтому граф Γ — гамильтонов.

Аналогично доказательство проводится и для мультипликативной ЦФ.

Теорема 3 означает, что целевая функция гамильтонова графа Γ состоит из суммы абсолютных величин двух функций

$$Z_\Gamma(n) = \mathrm{abs}(Z_E(\mathbf{n})) + \mathrm{abs}(Z_V(\mathbf{n})), \quad (25)$$

и на гамильтоновом графе Γ для подсчета ГА замкнутых маршрутов (циклов), проходящих один раз по всем вершинам, необходимо и достаточно выполнение одновременно двух уравнений (23) и (24) или одного уравнения (25). Важным фактом в построении ЦФ (24) гамильтонова графа является то, что во втором слагаемом используются величины $cod_V(v_i)$, которые являются кодами вершин графа, а не ребер.

В табл. 4 приведены результаты расчёта полёта шести БПЛА по эйлеровым маршрутам на графе ***v8e16*** с соблюдением условий безопасности: не встречаться в одинаковых вершинах и рёбрах [16, 17, 18]. При этом предполагается, что все ребра, независимо от их протя-

женности, проходятся всеми БПЛА за одинаковое время, т.е. длинное ребро проходится с большей скоростью, а короткое — с меньшей.

Таблица 4

Безопасный полет шести БПЛА по графу реперов ***v*8*e*16**

№ столбца	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
БПЛА № 1	1	6	7	8	3	5	2	1	6	4	7	5	8	3	2	4	**1|**	6	7	8
БПЛА № 2		1	6	7	8	3	5	2	1	6	4	7	5	8	3	2	4	**1|**	6	7
БПЛА № 3			1	6	7	8	3	5	2	1	6	4	7	5	8	3	2	4	**1|**	6
БПЛА № 4				1	6	7	8	3	5	2	1	6	4	7	5	8	3	2	4	**1|**
БПЛА № 5					1	6	7	8	3	5	2	1	6	4	7	5	8	3	2	4
БПЛА № 6						1	6	7	8	3	5	2	1	6	4	7	5	8	3	2

В таблицах 5 и 6 показан безопасный полёт группы из трёх БПЛА с интервалами два и три ребра.

Таблица 5

Безопасный полет трех БПЛА по одинаковым маршрутам по графу реперов ***v*8*e*16**

№ столбца	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ОЗМ	1	6	7	8	3	5	2	1	6	4	7	5	8	3	2	4	1|	4	6	7
ОЗМ			1	6	7	8	3	5	2	1	6	4	7	5	8	3	2	4	1|	4
ОЗМ					1	6	7	8	3	5	2	1	6	4	7	5	8	3	2	4

Таблица 6

Безопасный полет трех БПЛА по различным маршрутам по графу реперов ***v*8*e*16**

№ столбца	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ОЗМ	1	6	7	8	3	5	2	1	6	4	7	5	8	3	2	4	1|	4	6	7
ОЗМ				1	4	7	6	4	2	3	8	3	5	7	8	5	2	1	6	1|
ОЗМ							1	6	4	2	5	8	4	1	2	3	8	3	8	5

В табл. 7 показана организация непредсказуемого мониторинга при изменении маршрутов у всей группы БПЛА одновременно за счет горизонтального автоморфизма графа **v6e11** (рис. 11):

$$\gamma_г(1) = 3;\ \gamma_г(2) = 4;\ \gamma_г(3) = 1$$
$$\gamma_г(4) = 2;\ \gamma_г(5) = 5;\ \gamma_г(6_6) = 6.$$

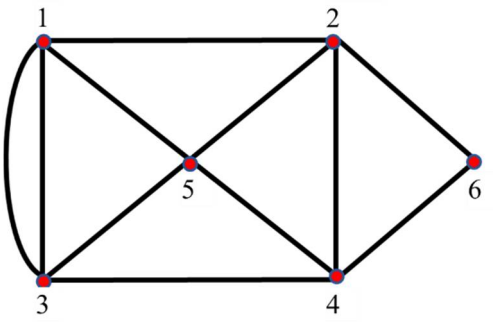


Рис. 11. Граф ***v6e*11**

Полет БПЛА по смешанным эйлеро-гамильтоновым маршрутам на графе ***v6e*11** с переходом с маршрута на маршрут в столбце № 9 (табл. 7).

Таблица 7

Мониторинг группы БПЛА при одновременном изменении маршрутов

Номер столбца	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1-й ЛА	1	2	6	4	3	5	1	2	**1**	5	3	4	6	2	1|
2-й ЛА		1	2	6	4	3	5	1	**5**	3	4	6	2	1|	5
3-й ЛА			1	2	6	4	3	5	**3**	4	6	2	1|	5	3
4-й ЛА				1	2	6	4	3	**4**	6	2	1|	5	3	4
5-й ЛА					1	2	6	4	**6**	2	1|	5	3	4	6
6-й ЛА						1	2	6	**2**	1|	5	3	4	6	2

Подводя итог можно отметить, что ГА является способом обработки многовариантных задач и получения информации для устранения неопределенностей при синтезе интеллектуальных систем принятия решении [19, 20, 21].

Заключение

1. Теоремы 1, 2 и 3 являются теоретической основой для построения целевых функций на эйлеровом, на гамильтоновом, на эйлеро-гамильтоновом графе (ЭГГ).

2. Получение ОЗМ дает информацию для решения задач маршрутизации группой БПЛА.

3. Теоретико-числовые алгебраические и трансцендентные ЦФ графа показывают, что принцип кодирования ребер может быть перенесен на кодирование и вершин графа при решении задач с приоритетом информации о вершинах, например, в задаче о маршрутах на гамильтоновых графах, на которых существует маршрут, проходящий по всем вершинам один раз и т.д.

4. Векторные целевые функции позволяют распараллелить подсчет значений суммы или произведения кодов ребер в частных ЦФ Zi(Mi) на наборах предполагаемых маршрутов в ГА;

5. Векторные ЦФ удовлетворяют однозначности разложения суммы кодов ребер, так как векторные коды представляют собой базисные вектора, а числовые коды представляют s-ические полиномы по основанию систем счисления из целых чисел 2, 3, 5 и трансцендентных чисел e и π, которые удовлетворяют алгебраической независимости между собой, а по отношению к целым числам являются независимыми переменными.

6. Ограничивая число слагаемых или произведений в частных ЦФ Zi(Mi) удается избежать выхода суммы или умножения чисел за пределы точности машинного округления.

7. Оценка времени подсчета маршрута ГА с помощью векторных ЦФ показывает приемлемое время для их реализации в бортовых ЭВМ БПЛА [22, 23, 24].

8. ВЦФ позволяют разрешить противоречие о большом различии степеней su и st в аддитивных и в большом разбросе между кодами из простых чисел 2, 3, 5, 7 и 11 в мультипликативных теоретико-числовых ЦФ, а также о потере точности в сходимости к нулю ГА при кодировании ребер графа функциональными кодами с трансцендентными переменными, т.к. в каждом подпространстве сумма или произведение состоит из малого числа элементов.

9. Математический аппарат построения векторных целевых функций и информационно-программное обеспечение на основе генетического алгоритма может быть использовано в автоматизированной интеллектуальной системе, позволяющей решать задачи маршрутизации на графах реперов в системах управления БПЛА [25, 26, 27, 28].

Литература

1. *Руденко Э.М., Аллилуева Н.В., Семикина Е.В.* Маршрутизация на графе, теоретико-числовые целевые функции и генетический алгоритм // Наукоемкие технологии в космических исследованиях Земли. 2019. № 6. С. 4–14.

2. *Дараган А.Д. Руденко Э.М., Залогин В.С., Ефремов А.А., Финогенов М.А.* Системы искусственного интеллекта. Монография. Серпухов: ФВА РВСН им. Петра Великого, 2020. 205 с.

3. *Руденко Э.М., Семикина Е.В.* Маршрутизация беспилотных летательных аппаратов, трансцендентные целевые функции графа и генетический алгоритм // Наукоемкие технологии в космических исследованиях Земли. 2021. Т. 13. № 1. С. 6–16.

4. *Лебедев Г.Н., Малыгин В.Б.* Оценка возможности применения генетического алгоритма для оптимизации операций в аэропортах на основе принципов совместного принятия решений. // Научный вестник МГТУ ГА. 2019. Т. 22. №5. С. 85–93. URL: https://doi.org/10.26467/2079-0619-2019-22-5-85-93 (дата обращения 12.05.2022)

5. *Акопов А.С., Бекларян А.Л., Тхакур М., Верма Б.Д.* Разработка параллельных генетических алгоритмов вещественного кодирования для систем поддержки принятия решений социально-экономического и экологического планирования // Научный журнал «Бизнес-информатика». 2019. Т. 13. № 1. С. 33–44. URL: https://bijournal.hse.ru/data/2019/05/06/1501882974/3.pdf (дата обращения 31.06.2022)

6. *Руденко Э.М., Семикина Е.В.* Интеллектуальный мониторинг группы БПЛА на эйлеро-гамильтоновых графах реперов на местности // Научно-технический журнал «Известия ИИФ». 2021. №4 (62). С.75–80.

7. *Руденко Э.М., Семикина Е.В.* Методика построения векторной целевой функции расчёта маршрутов на графах генетическим алгоритмом // Научно-технический журнал «Известия ИИФ». 2022. №1 (63). С.93–97.

8. *Гончаренко В.И., Лебедев Г.Н., Михайлин Д.А.* Задача оперативной двумерной маршрутизации группового полета беспилотных летательных аппаратов // Известия РАН. Теория и системы управления. 2019. № 1. С. 153–165.

9. *Sabino S., Horta N., Grilo A.* Centralized unmanned aerial vehicle mesh network placement scheme: A multi-objective evolutionary algorithm approach. // Sensors. 2018. Vol. 18. No. 12. https://doi.org/10.3390/s18124387

10. Agatz N., Bouman P., Schmidt M. Optimization approaches for the traveling salesman problem with drone. // Transp. Sci. 2018. Vol. 52. No. 4. Pp. 965–981. https://doi.org/10.1287/trsc.2017.0791

11. *Shuai Zhang, Siliang Liu, Weibo Xu, Wanru Wang*. A novel multi-objective optimization model for the vehicle routing problem with drone delivery and dynamic flight endurance. // Computers & Industrial Engineering. 2022. Vol. 173. URL: https://doi.org/10.1016/j.cie.2022.108679

12. *Борсоев В.А., Лебедев Г.Н., Малыгин В.Б., Нечаев Е.Е.* Принятие решения в задачах управления воздушным движением. Методы и алгоритмы. М.: Радиотехника, 2018. 432 с.

13. *Вирсански Э.* Генетические алгоритмы на Python. Пер. с англ. А. А. Слинкина. М.: ДМК Пресс, 2020. 286 с. ISBN 978-5-97060-857-9

14. *Шабанов А.С.* Прямой генетический алгоритм на основе приоритетов для логистической сети // Интернаука: электрон. научн. журн. 2022. Т. 244. №21. URL: https://internauka.org/journal/science/internauka/244 (дата обращения: 06.09.2022).

15. *Adam J. Sobey, Przemyslaw Grudniewski*. Re-inspiring the genetic algorithm with multi-level selection theory: Multi-Level Selection Genetic Algorithm (MLSGA) // Bioinspiration & Biomimetics. 2018. Vol. 13. No. 5. DOI:10.1088/1748-3190/aad2e8 (дата обращения: 02.07.2022).

16. *Веселов Г.Е., Гладков Л.А., Ясир М.Д.* Разработка и исследование гибридного алгоритма решения задачи размещения элементов ЭВА // Материалы XI Международной научно-практической конференции «Интегрированные модели и мягкие вычисления в искусственном интеллекте» (Коломна, 16-19 мая 2022). Т. 1. С. 287–297. URL: http://www.raai.org/resurs/papers/immv/2022/imsc2022-vol1.pdf (дата обращения 10.08.2022)

17. *Harshavardhan P.K., Kousik Krishnan.* Hierarchical Genetic Algorithms with evolving objective functions // arXiv.org. 05 Aug 2019. URL: https://arxiv.org/pdf/1812.10308.pdf (дата обращения: 09.07.2022).

18. *Xin J., Zhong J., Yang F., Cu, Y., Sheng J.* An Improved Genetic Algorithm for Path-Planning of Unmanned Surface Vehicle. // Sensors 2019. Vol. 19. No. 11. https://doi.org/10.3390/s19112640

19. Карпов В. Э. От роевой робототехники к социуму роботов. // Материалы Всероссийской научно-технической конференции «Искусственный интеллект: проблемы и пути решения – 2018» (Кубинка, 14-15 марта 2018), Кубинка, 2018. С.122–130. URL: http://raai.org/library/books/Konf_II_problem-2018/book1_intellect.pdf

20. *Лебедев Г.Н., Михайлин Д.А.* Автоматизированная система планирования групповых действий беспилотников при заданном графике обслуживания мобильных наземных объектов в прогнозируемой динамической обстановке. // Сборник научных статей по материалам VI Международной научно-практической конференции «АВИАТОР» (Воронеж, 14–15 февраля 2019), Воронеж, 2019. URL: https://vva.mil.ru/upload/site21/document_file/JG8GpML43d.pdf

21. *Khanra A., Pal T., Maiti M.K., Maiti M.* Multi-objective four-dimensional imprecise TSP solved with a hybrid multi-objective ant colony optimization-genetic algorithm with diversity. // J. Intell. Fuzzy Syst. 2019. Vol. 36. No. 1. Pp. 47-65. DOI: 10.3233/JIFS-172127

22. *Bence Keresztury*. Genetic algorithms and the Traveling Salesman Problem. Budapest. ELTE, Department of Operations Research. 2017. p. 56. URL: https://web.cs.elte.hu/blobs/diplomamunkak/bsc_alkmat/2017/keresztury_bence.pdf

23. *Patchara Kitjacharoenchai, Mario Ventresca, Mohammad Moshref-Javadi, Seokcheon Lee, Jose MA Tanchoco, Patrick A Brunese.* Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. // Computers & Industrial Engineering. 2019. Vol. 129. Pp. 14–30. URL: https://doi.org/10.1016/j.cie.2019.01.020

24. *Dharm Raj Singh, Manoj Kumar Singh, Tarkeshwar Singh, Rajkishore Prasad*. Genetic Algorithm for Solving Multiple Traveling Salesmen Problem using a New Crossover and Population Generation. // Computación y Sistemas, Vol. 22, No. 2, 2018, Pp. 491–503 doi: 10.13053/CyS-22-2-2956

25. Хачумов В. М. Концепция комплексной интеллектуализации наземных станций командно-измерительных систем аэрокосмического назначения. // Материалы Всероссийской научно-технической конференции «Искусственный интеллект: проблемы и пути решения – 2018» (Кубинка, 14-15 марта 2018), Кубинка, 2018. С.44–49. URL: http://raai.org/library/books/Konf_II_problem-2018/book1_intellect.pdf

26. *Kayan Herdiana, Made Candiasa, Gede Indrawan.* Optimization of Adaptive Genetic Algorithm Parameters in Traveling Salesman Problem. // Journal of Computer Networks, Architecture and High-Performance Computing. 2022. Vol. 4. No. 2. Pp. 169–176. URL: https://doi.org/10.47709/cnahpc.v4i2.1581

27. Иванов Д. Я. Система поддержки принятия решений оператора объектов ответственного применения на основе гибридных методов интеллектуальной обработки данных. // Материалы Всероссийской научно-технической конференции «Искусственный интеллект: проблемы и пути решения – 2018» (Кубинка, 14-15 марта 2018), Кубинка, 2018. С.110–114. URL: http://raai.org/library/books/Konf_II_problem-2018/book1_intellect.pdf

28. Чернышев Ю. О., Венцов Н. Н. Разработка алгоритма адаптивного поиска оптимальных решений на основе нечетких декодеров. // Материалы Всероссийской научно-технической конференции «Искусственный интеллект: проблемы и пути решения – 2018» (Кубинка, 14-15 марта 2018), Кубинка, 2018. С. 169–172. URL: http://raai.org/library/books/Konf_II_problem-2018/book1_intellect.pdf. DOI: 10.1109/ ICALT.2015.105. 8.

29. *Хачумов М.В., Талалаев А.А., Хачумов В.М.* Об одном эвристическом критерии в задаче определения изоморфизма графов на основе инвариантов. // Современные наукоемкие технологии. 2022. №2. С.159-163. DOI: 10.17513/snt.39051

MATHEMATICAL SOFTWARE FOR CALCULATING ROUTES ACCORDING TO THE GRAPH OF REFERENCE POINTS ON THE GROUND BY A GENETIC ALGORITHM

EDUARD M. RUDENKO
PhD., associate professor at the Department of Mathematics of the Military Academy of Strategic Rocket Troops after Peter the Great (Serpukhov Branch).Russia, eduard5529@yandex.ru

ELENA V. SEMIKINA
lecturer at the Military Academy of Strategic Rocket Troops after Peter the Great (Serpukhov Branch).Russia, labinfo_serp@inbox.ru

ABSTRACT

Introduction: the paper considers the mathematical support of the problem of routing along the graph of benchmarks on the ground by a genetic algorithm. The object of the study is a mathematical graph isomorphic to the graph of frames on the ground. **Problem statement:** the problem of finding a set of closed routes on an Euler, Hamiltonian and Euler-Hamiltonian graph is formulated as an optimization problem for a conditional extremum. The criterion is a value equal to the number of graph edges, which is equal to half the sum of the multiplicities of the graph vertices. In this case, multiple edges are taken into account as many times in the routes as their multiplicity in the graph. The result of the solution is a subset of Euler routes. The routing problem belongs to the problem of integer programming and is reduced by coding the edges with mathematical values with a unique expansion into a sum or into a product into an unconstrained optimization problem. **Methods:** a method for solving the routing problem is a genetic algorithm, which minimizes to zero the objective function, built on the basis of structural theorems 1, 2 and 3. **Results:** using computer simulation in the cycle of changing the crossover factor from 0.05 to 0.95 with a step of 0.05, the solution of the routing problem on graphs v8e16, v15e28 and v25e50 is obtained. The probability of convergence of the genetic algorithm for graphs v8e16, v15e28 is 100%, for graph v25e50 - 70%. The calculation of Hamiltonian routes on the graph v8e14 is 100%. **Practical significance:** the mathematical apparatus for constructing vector objective functions and information software based on a genetic algorithm can be used in an automated intelligent system that allows solving routing problems on frame graphs in UAV control systems.

Keywords: routing; graphs; number-theoretic algebraic, transcendental and vector objective functions; genetic algorithm.

REFERENCES

1. Rudenko E.M., Allilueva N.V., Semikina E.V. Routing on a graph of number-theoretic objective function and genetic algorithm. Naukoemkie tehnologii v kosmicheskih issledovanijah Zemli [High tech in earth space research]. 2019. Vol. 11. No. 6. Pp. 4–14 (In Rus)
2. Daragan A.D. Rudenko E.M., Zalogin V.S., Efremov A.A., Finogenov M.A. Sistemy iskusstvennogo intellekta. Monografija. [Artificial intelligence systems. Monograph.]. Serpukhov: FVA RVSN (filial v g. Serpuhov), 2020. 205 p. (In Rus)
3. Rudenko E.M., Semikina E.V. Unmanned aircraft routing, graph transcendent target functions and genetic algorithm. Naukoemkie tehnologii v kosmicheskih issledovanijah Zemli [High tech in earth space research]. 2021. Vol. 13. No. 1. Pp. 6–16 (In Rus)
4. Lebedev G.N., Malygin V.B. Assessment of the application feasibility of the genetic algorithm for airports operations optimization based on the collaborative decision-making principles. Nauchnyj vestnik MGTU GA [Civil Aviation High TECHNOLOGIES]. 2019. Vol. 22. No. 5. Pp. 85–93.
5. Akopov A.S., Beklaryan A.L., Thakur M., Verma B.D. Developing parallel real-coded genetic algorithms for decision-making systems of socio-ecological and economic planning. Nauchnyj zhurnal «Biznes-informatika» [Modeling of social and economic systems]. 2019. Vol.13. No. 1. Pp. 33–44.
6. Rudenko E.M., Semikina E.V. Intelligent monitoring of UAV group on eyler-hamilton reference graphs on the local. Institut inzhenernoj fiziki [Institute of Engineering Physics]. 2021. Vol. 62. No. 4. Pp. 75–80.
7. Rudenko E.M., Semikina E.V. Unmanned aircraft routing, graph transcendent target functions and genetic algorithm. Institut inzhenernoj fiziki [Institute of Engineering Physics]. 2022. Vol. 63. No. 1. Pp. 93–97
8. Goncharenko V.I., Lebedev G.N., Mikhailin D.A. Online two-dimensional route planning for a group of unmanned aerial vehicles. Izvestija RAN. Teorija i sistemy upravlenija [Journal of Computer and Systems Sciences International]. Luxembourg: Pleiades Link. 2019. Vol. 58. No. 1. Pp. 153-165
9. Sabino S., Horta N., Grilo A. Centralized unmanned aerial vehicle mesh network placement scheme: A multi-objective evolutionary algorithm approach. Sensors. 2018. Vol. 18. No. 12. https://doi.org/10.3390/s18124387
10. Agatz N., Bouman P., Schmidt M. Optimization approaches for the traveling salesman problem with drone. Transp. Sci. 2018. Vol. 52. No. 4. Pp. 965–981. https://doi.org/10.1287/trsc.2017.0791
11. Shuai Zhang, Siliang Liu, Weibo Xu, Wanru Wang. A novel multi-objective optimization model for the vehicle routing problem with drone delivery and dynamic flight endurance. Computers & Industrial Engineering. 2022. Vol. 173. URL: https://doi.org/10.1016/j.cie.2022.108679
12. Borsoev, V.A., Lebedev, G.N., Malygin, V.B., Nechaev, E.E., Nikulin, A.O. and Tin, Pkhon Chzho. Prinyatie resheniya v zadachakh upravleniya vozdushnym dvizheniyem. Metody i algoritmy [Decision Making in Air Traffic Management Tasks. Methods and Algorithms]. Moscow: Radiotekhnika, 2018. 432 p. (In Rus)
13. Wirsansky E. Hands-on genetic algorithms with Python: applying genetic algorithms to solve real-world deep learning and artificial intelligence problems. Packt Publishing Ltd, 2020/ 346 p.
14. Shabanov A.S. Priority-Based Direct Genetic Algorithm for Logistics Network. Internauka [Internauka]. 2022. Vol. 244. No. 21. URL: https://internauka.org/journal/science/internauka/244
15. Adam J. Sobey, Przemyslaw Grudniewski. Re-inspiring the genetic algorithm with multi-level selection theory: Multi-Level Selection Genetic Algorithm (MLSGA). Bioinspiration & Biomimetics. 2018. Vol. 13. No. 5. DOI:10.1088/1748-3190/aad2e8 (date of access 02.07.2022).
16. Veselov G.E., Gladkov L.A., Yasir M.D. Razrabotka i issledovanie gibridnogo algoritma reshenija zadachi razmeshhenija jelementov JeVA [Development and research of the hybrid algorithm of the solution of the problem of placement of elements of computer equipment]. Materialy XI Mezhdunarodnoj nauchno-prakticheskoj konferencii «Integrirovannye modeli i mjagkie vychislenija v iskusstvennom intellekte» [Proceedings of the XI International conference «Integrated Models and Soft Compu-

ting in Artificial Intelligence (IMSC-2022) », Kolomna, May 16-19, 2022). T. 1. Pp. 287–297. URL: http://www.raai.org/resurs/papers/immv/2022/imsc2022-vol1.pdf

17. Harshavardhan P.K., Kousik Krishnan. Hierarchical Genetic Algorithms with evolving objective functions. arXiv.org. 05 Aug 2019. URL: https://arxiv.org/pdf/1812.10308.pdf (date of access: 09.07.2022).
18. Xin J., Zhong J., Yang F., Cu, Y., Sheng J. An Improved Genetic Algorithm for Path-Planning of Unmanned Surface Vehicle. Sensors. 2019. Vol. 19. No. 11. https://doi.org/10.3390/s19112640
19. Karpov V. E. Ot roevoj robototehniki k sociumu robotov. [From swarm robotics to the society of robots]. Materialy Vserossijskoj nauchno-tehnicheskoj konferencii «Iskusstvennyj intellekt: problemy i puti reshenija – 2018» [Proceedings of the All-Russian Scientific and Technical Conference "Artificial Intelligence: Problems and Solutions - 2018" (Kubinka, March 14-15, 2018)]. Pp.122–130. URL: http://raai.org/library/books/Konf_II_problem-2018/book1_intellect.pdf
20. Lebedev G.N., Mikhailin D.A. Avtomatizirovannaja sistema planirovanija gruppovyh dejstvij bespilotnikov pri zadannom grafike obsluzhivanija mobil'nyh nazemnyh ob#ektov v prognoziruemoj dinamicheskoj obstanovke [Automated system for planning group actions of drones with a given schedule for servicing mobile ground objects in a predictable dynamic environment]. Sbornik nauchnyh statej po materialam VI Mezhdunarodnoj nauchno-prakticheskoj konferencii «AVIATOR» [Collection of scientific articles based on the materials of the VI International Scientific and Practical Conference "AVIATOR" (Voronezh, February 14–15, 2019)]. 2019. Pp. 39–45. (In Rus).
21. Khanra A., Pal T., Maiti M.K., Maiti M. Multi-objective four-dimensional imprecise TSP solved with a hybrid multi-objective ant colony optimization-genetic algorithm with diversity. J. Intell. Fuzzy Syst. 2019. Vol. 36. No. 1. Pp. 47-65. DOI: 10.3233/JIFS-172127
22. Bence Keresztury. Genetic algorithms and the Traveling Salesman Problem. Budapest. ELTE, Department of Operations Research. 2017. p. 56. URL: https://web.cs.elte.hu/blobs/diplomamunkak/bsc_alkmat/2017/keresztury_bence.pdf
23. Patchara Kitjacharoenchai, Mario Ventresca, Mohammad Moshref-Javadi, Seokcheon Lee, Jose MA Tanchoco, Patrick A Brunese. Multiple traveling salesman problem with drones: Mathematical model and heuristic approach. Computers & Industrial Engineering. 2019. Vol. 129. Pp. 14–30. URL: https://doi.org/10.1016/j.cie.2019.01.020
24. Dharm Raj Singh, Manoj Kumar Singh, Tarkeshwar Singh, Rajkishore Prasad. Genetic Algorithm for Solving Multiple Traveling Salesmen Problem using a New Crossover and Population Generation. Computación y Sistemas, 2018. Vol. 22, No. 2. Pp. 491–503 doi: 10.13053/CyS-22-2-2956
25. Khachumov V. M. Koncepcija kompleksnoj intellektualizacii nazemnyh stancij komandno-izmeritel'nyh sistem ajerokosmicheskogo naznachenija [The concept of integrated intellectualization of ground stations of command and measurement systems for aerospace purposes]. Materialy Vserossijskoj nauchno-tehnicheskoj konferencii «Iskusstvennyj intellekt: problemy i puti reshenija – 2018» [Proceedings of the All-Russian Scientific and Technical Conference "Artificial Intelligence: Problems and Solutions - 2018" (Kubinka, March 14-15, 2018)]. Pp. 44–49. URL: http://raai.org/library/books/Konf_II_problem-2018/book1_intellect.pdf
26. Kayan Herdiana, Made Candiasa, Gede Indrawan. Optimization of Adaptive Genetic Algorithm Parameters in Traveling Salesman Problem. Journal of Computer Networks, Architecture and High-Performance Computing. 2022. Vol. 4. No. 2. Pp. 169–176. URL: https://doi.org/10.47709/cnahpc.v4i2.1581
27. Ivanov D. Ya. Sistema podderzhki prinjatija reshenij operatora ob#ektov otvetstvennogo primenenija na osnove gibridnyh metodov intellektual'noj obrabotki dannyh. [Decision support system for the operator of objects of responsible use based on hybrid methods of intelligent data processing]. Materialy Vserossijskoj nauchno-tehnicheskoj konferencii «Iskusstvennyj intellekt: problemy i puti reshenija – 2018» [Proceedings of the All-Russian Scientific and Technical Conference "Artificial Intelligence: Problems and Solutions - 2018" (Kubinka, March 14-15, 2018)]. Pp.110–114. URL: http://raai.org/library/books/Konf_II_problem-2018/book1_intellect.pdf

28. Chernyshev Yu. O., Ventsov N. N. Razrabotka algoritma adaptivnogo poiska optimal'nyh reshenij na osnove nechetkih dekoderov [Development of an algorithm for adaptive search for optimal solutions based on fuzzy decoders]. Materialy Vserossijskoj nauchno-tehnicheskoj konferencii «Iskusstvennyj intellekt: problemy i puti reshenija – 2018» [Proceedings of the All-Russian Scientific and Technical Conference "Artificial Intelligence: Problems and Solutions - 2018" (Kubinka, March 14-15, 2018)]. Pp. 169–172. URL: http://raai.org/library/books/Konf_II_problem-2018/book1_intellect.pdf
29. Khachumov M.V., Talalaev A.A., Khachumov V.M. On a heuristic criterion in the problem of determining the isomorphism of graphs based on invariants. Tehnicheskie nauki [Technical sciences]. 2022. No. 2. Pp.159-163. DOI: 10.17513/snt.39051