

Информационные процессы / Information processes

Оригинальная статья / Original article

УДК 004.021

[https://doi.org/ 10.31432/1994-2443-2024-19-2-84-95](https://doi.org/10.31432/1994-2443-2024-19-2-84-95)

Алгоритм построения маршрутизации для обеспечения отказоустойчивости сети связи

Г. О. Сорокин ✉, А. В. Сеницын

Федеральное государственное бюджетное образовательное учреждение высшего образования «МИРЭА — Российский технологический университет», г. Москва,
Российская Федерация,
✉ sorokinglebw@yandex.ru

Аннотация. *Введение.* Эффективность, надежность и масштабируемость сети связи во многом зависят от того, какой выбор будет сделан алгоритмом маршрутизации. Это особенно актуально сейчас, из-за частых эскалаций военных конфликтов, последствия которых могут влиять на стабильность и надежность сетевых коммуникаций.

Цель. Формализация алгоритма, обеспечивающего отказоустойчивость сети связи за счет построения оптимальной маршрутизации.

Методы. Основным методом формализации проблемы выступает математическое моделирование, а также экспериментальная проверка представленной модели на основе различных геоданных за счет практической реализации формализованного алгоритма.

Результаты. Предложена математическая модель алгоритма, в основе которой лежат алгоритмы построения минимального остовного дерева, а также дан пример его реализации.

Выводы. Разработанный алгоритм позволяет построить маршрутизацию, обеспечивающую отказоустойчивость сети связи в условиях теоретического уничтожения узлов в сетях связи за счет определения вероятности существования смежных узлов. Алгоритм может быть использован для создания сетей SDN, а также как виртуализированная сетевая функция (VNF).

Ключевые слова: маршрутизация; минимальное остовное дерево; отказоустойчивость; сети связи; теория графов

Финансирование. Финансирование отсутствовало.

Для цитирования: Сорокин Г. О., Сеницын А. В. Алгоритм построения маршрутизации для обеспечения отказоустойчивости сети связи. *Информация и инновации.* 2024;19(2):84-95. <https://doi.org/10.31432/1994-2443-2024-19-2-84-95>

Algorithm for constructing routing to ensure fault tolerance of the communication network

G. O. Sorokin ✉, A. V. Sinitsyn

MIREA — Russian Technological University, Moscow, Russian Federation

✉ sorokinglebw@yandex.ru

Abstract. *Introduction.* The efficiency, reliability and scalability of a communication network largely depend on the choices made by the routing algorithm. This is especially true now, due to frequent escalations of military conflicts, the consequences of which can affect the stability and reliability of network communications.

Purpose. Formalization of an algorithm that ensures fault tolerance of a communication network by constructing optimal routing.

Methods. Mathematical modeling, as well as experimental testing of the presented model based on various geodata through the practical implementation of a formalized algorithm.

Results. A mathematical model of the algorithm is proposed, which is based on algorithms for constructing a minimum spanning tree, as well as an example of its implementation.

Conclusions. The presented algorithm allows you to build routing that ensures fault tolerance of the communication network, in conditions of theoretical destruction of nodes in communication networks by determining the probability of the existence of adjacent nodes.

Keywords: routing; minimum spanning tree; fault tolerance; communication networks; graph theory

Funding. No funding.

For citation: Sorokin G. O., Sinitsyn A. V. Algorithm for constructing routing to ensure fault tolerance of the communication network. *Information and Innovations*. 2024;19(2):84-95. (In Russ.). <https://doi.org/10.31432/1994-2443-2024-19-2-84-95>

1. Введение

Для сети связи общего пользования критически важным является поддержание непрерывной работы даже в таких ситуациях, как отказ или выход из строя узла. Целью статьи является формализация алгоритма, обеспечивающего отказоустойчивость сети связи за счет построения оптимальной маршрутизации.

Отказоустойчивость, в основе которой лежит избыточность связности, означает возможность для любого узла оставаться на связи в случае сбоя других ее компонентов. Качество обслуживания QoS (Quality of Service) улучшается за счет построения оптимальной маршрутизации между узлами.

В статье предложен алгоритм построения маршрутизации на основе построения отказоустойчивого графа, представляющего объединение всевозможных минимальных остовных деревьев для функционирующих узлов, в котором приоритеты маршрутизации зависят от вероятности существования ребра между узлами сети. В работе [1] показано, что при использовании модели Барабаши-Альберта [2, 3] стратегия вероятностной маршрутизации в объединенных сетях не только достигает высокой пропускной способности сетевого трафика, но и обеспечивает превосходную производительность.

Описываемый алгоритм может быть применен для создания сетей SDN и использоваться как виртуализированная сетевая функция (VNF) [4]. Данная работа продолжила исследования, начатые в [5].

2. Постановка задачи

Сеть связи будет представляться планарным неориентированным ациклическим графом без петель:

$$\begin{aligned} G(v, e, w), v \in V = \{v_0, \dots, v_n\}, \\ v \in E(n, n) = e_{ij}, w = w_{ij} \end{aligned} \quad (1)$$

Граф состоит из n узлов, вершин v и ребер e с весами w , соединяющих вершины согласно квадратной треугольной матрице смежности E (без главной диагонали) вида:

$$E = \begin{pmatrix} 0 & e_{01} & \dots & e_{0n} \\ 0 & 0 & \dots & e_{1n} \\ \dots & \dots & \dots & e_{n-1,n} \\ 0 & 0 & \dots & 0 \end{pmatrix} \quad (2)$$

$$e_{ij} = \begin{cases} e_{ij}, i < j \\ 0, i \geq j \end{cases}$$

Предполагается, что не более k -узлов могут быть выведены из строя и (1) примет вид:

$$G^*(v, e), n \in V^* \{v_1, \dots, v_{n-k}\} \in V. \quad (3)$$

2.1. Проблема построения отказоустойчивой матрицы смежности

Для (1) требуется построить такую матрицу смежности

$$E^*(n, n) = e_{ij}^*, \quad (4)$$

которая для оставшегося множества вершин (3) обеспечивала бы связность при выходе из строя любых k — вершин (3), т.е. любая пара оставшихся вершин имела бы маршрут $\forall (v_i, v_j) \in V^* \exists (e_{i1} \dots e_{jm})$, иными словами $\forall G^* \exists E^*(n-k, n-k)$, являющийся подграфом (4).

Решение такой проблемы позволит любому узлу оставаться на связи в критических ситуациях.

2.2. Функционал минимизации разности матриц смежности

Для всевозможных (4) предпочтительнее выбрать такую матрицу смежности, которая бы мало отличалась бы от заданной т.е. функционал $F(E, E^*)$ имел наименьшее значение:

$$F(E, E^*) = \sum_{i,j=0}^{n,n} |e_{ij} - e_{ij}^*| \rightarrow \min. \quad (5)$$

Минимизация (5) позволит найти способ дополнения существующей сети связи до отказоустойчивой с минимальными затратами и максимальным использованием существующих линий связи (ребер).

2.3. Проблема оптимальной маршрутизации

Для каждого узла v_i (1) в построенной отказоустойчивой сети (3–4) необходимо построить T таблицу маршрутизации согласно приоритету в множестве ее смежных ребер $\{e_{ab}$, где $a = i, b \in V$ или $a \in V, b = i\}$, где максимальный приоритет имеет ребро с максимальной вероятностью существования после уничтожения k -вершин, а минимальный — с минимальной вероятностью:

$$T: v_i \rightarrow \{v_a \text{ или } v_b \sim \{e_{ab}\}\}. \quad (6)$$

Создание подходов, позволяющих решать поставленные проблемы, пополняет инструментарий систем эксплуатационной поддержки OSS (англ. Operations Support Systems), в частности управление неисправностями (англ. Fault Management)¹ и управление качеством предоставляемых услуг (англ. SLA Management) для обеспечения заказанного QoS².

¹ Сайт: JSR-263 v1.0 — Fault Management API Specifications. URL: <https://www.tmforum.org/resources/standard/jsr-263-v1-0-fault-management-api-specifications/> (дата обращения: 20.05.2024).

² Сайт: TMF623 SLA management api rest specification — Fault Management API Specifications. URL: <https://www.tmforum.org/resources/interface/tmf623-sla-management-api-rest-specification-r14-5-0/> (дата обращения: 20.05.2024).

3. Описание алгоритма решения задачи

Для решения задачи предлагается алгоритм, основанный на построении и объединении минимальных остовных деревьев (Minimal Spanning Tree, MST) для подграфов меньшей размерности, чем исходный, например, реализацией MST могут служить алгоритмы Прима, Борувки, обратного удаления или Крускала, из которых наиболее эффективным является последний [6].

В MST нас прежде всего интересует построение связности для имеющихся вершин с минимальным весом w_{ij} , который означает минимальную стоимость линии связи, если весом является пропускная способность линии связи, например, STM1 (155,52 Мбит/с), ..., STM256 (40 Гбит/с).

Для построения матрицы смежности (4) для любого набора $n - k$ вершин необходимо рассмотреть все наборы C_n^k выбывших узлов:

$$V^*(k) = (v_{i_1} \dots v_{i_k}), v_{i_j} \in V, j = 1 \dots k \quad (7),$$

где

$$i_1 = \{0, n - k + 1\}$$

$$i_2 = \{i_1, n - k + 2\}$$

...

$$i_k = \{i_{k-1}, n\}.$$

Все возможные наборы оставшихся вершин определяются как дополнения множества (7) до (1):

$$V^*(n - k) = V \setminus V^*(k). \quad (8)$$

Сначала построим минимальное остовное дерево для каждого конкретного набора $(v_{i_1} \dots v_{i_k})$ из (8):

$$G(V^*(n - k)) = MST(V^*(n - k)) \quad (9)$$

и рассмотрим объединение всех графов (9):

$$G^* = \bigcup_{i_1 \dots i_k} G(V^*(n-k)). \quad (10)$$

Полученный граф (10) обеспечивает полную связность для любого набора вершин (8) и решает проблему построения отказоустойчивой матрицы смежности.

Следующая проблема выбора минимально отличающего от исходного графа решается путем изменения списка вершин в исходном списке (1). Очевидно, что существует такая последовательность в списке (1), которая минимизирует (5), и такой список несложно составить, например, расположив вершины в порядке убывания их степени, т.е. количеству смежных ребер, выходящих из данной вершины.

Наконец, последняя проблема построения оптимальной таблицы маршрутизации (6) решается путем подсчета количества остовных деревьев (9), в которые входит каждое конкретное ребро:

$$n_{ij} = \|G(V^*(n-k)) \ni e_{ij}\|. \quad (11)$$

Приоритеты на множестве смежных ребер очевидно положить пропорциональными (11), а именно, равными вероятностям P_{ij} существования ребра при любых наборах:

$$P_{ij} = \frac{n_{ij}}{C_n^k}. \quad (12)$$

Таблица (6) имеет для каждой вершины список маршрутизации, равный степени вершины, в котором адреса назначения расположены на другом конце смежных ребер, расположенных в порядке убывания вероятностной характеристики существования ребра.

Таким образом, предлагаемый алгоритм успешно решает все три поставленные проблемы одновременно.

3.1. Численные эксперименты

Работа алгоритма иллюстрируется численным экспериментом для $n=10$ узлов (например, городов Ленинградской области) для числа неисправных узлов $k=1, 3$ и 5. Полученные отказоустойчивые сети и маршрутизации показаны на рис. 1.

Список приоритетных маршрутов для каждой вершины рассчитывается пропорционально вероятности существования каждого соседнего ребра графа, что показано в табл. 1 для случаев из рис. 1.

Список приоритетов можно пересчитать, если существование всех остальных узлов можно указать с некоторой вероят-

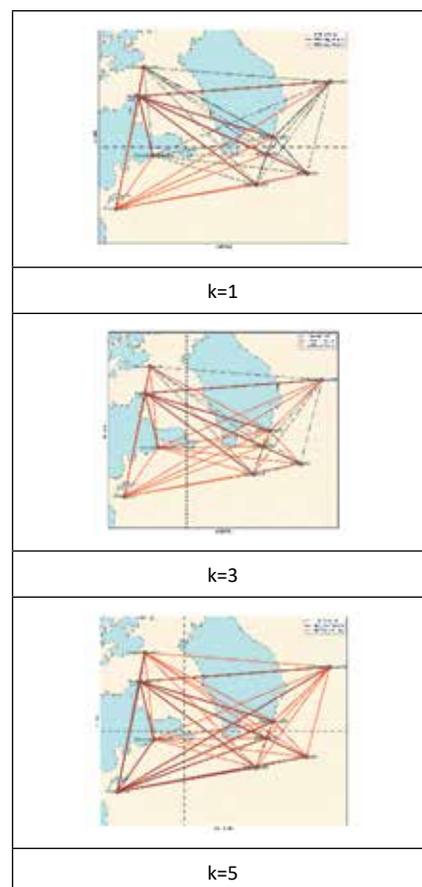


Рис. 1. Отказоустойчивая сеть для $n = 10$ при различных $k=1,3,5$
Fig. 1. Fault-tolerant network for $n = 10$ at different $k=1,3,5$

ностью от 0 до 1, в отличие от вышедших из строя узлов (см. табл. 1).

Степень вершины графа deg показывает количество выходящих из этой вершины ребер. Таким образом, можно узнать среднюю степень узлов графа $\langle deg \rangle = \frac{1}{N} \sum_{i=1}^N deg_i$, которая может дать

понять, насколько степень связности графов, вычисляемых по представленному алгоритму, отличается от начального графа.

В качестве примера взяты два абстрактных графа (без привязки к геокоординатам), состоящих из 20 узлов, первый из

Таблица 1. Сравнение маршрутизаций при $n=10, k=1, 2, 3$
Table 1. Comparison of routings with $n=10, k=1, 2, 3$

Город (из)	Город (в)	Приоритет	Город (в)	Приоритет	Город (в)	Приоритет
n=10	k=1		k=3		k=5	
Выборг(0)	Сланцы(1)	0.80	Сланцы(1)	0.47	Сланцы(1)	0.22
	Сосновый Бор(2)	0.80	Сосновый Бор(2)	0.47	Сосновый Бор(2)	0.22
	Сясьстрой(3)	0.80	Сясьстрой(3)	0.47	Сясьстрой(3)	0.22
	Светогорск(4)	0.80	Светогорск(4)	0.47	Светогорск(4)	0.22
	Подпорожье(5)	0.80	Подпорожье(5)	0.47	Подпорожье(5)	0.22
	Тихвин(6)	0.80	Тихвин(6)	0.47	Тихвин(6)	0.22
	Сясьстрой(7)	0.80	Сясьстрой(7)	0.47	Сясьстрой(7)	0.22
	Волхов(8)	0.80	Волхов(8)	0.47	Волхов(8)	0.22
	Кириши(9)	0.80	Кириши(9)	0.47	Кириши(9)	0.22
...						
Волхов(8)	Выборг(0)	0.80	Выборг(0)	0.47	Выборг(0)	0.22
	Сланцы(1)	0.10	Сланцы(1)	0.18	Сланцы(1)	0.14
			Сосновый Бор(2)	0.05	Сосновый Бор(2)	0.08
			Сясьстрой(3)	0.01	Сясьстрой(3)	0.04
			Светогорск(4)		Светогорск(4)	0.02
		Подпорожье(5)		Подпорожье(5)	0.0	
Кириши(9)	Выборг(0)	0.80	Выборг(0)	0.47	Выборг(0)	0.22
	Сланцы(1)	0.10	Сланцы(1)	0.18	Сланцы(1)	0.14
			Сосновый Бор(2)	0.05	Сосновый Бор(2)	0.08
			Сясьстрой(3)	0.01	Сясьстрой(3)	0.04
			Светогорск(4)		Светогорск(4)	0.02
		Подпорожье(5)		Подпорожье(5)	0.0	

которых является полным графом, второй — разряженным. Значение $\langle deg \rangle$ для каждого из этих графов равны 19 и 7.6 соответственно. На рис. 2 видно, что средняя степень узла при увеличении параметра k как раз стремится к этим значениям. Это значит, что чем больше было уничтожено узлов k , тем больше конечный отказо-

стойчивый граф приближается к начальному.

Помимо этого, на рис. 2 отображено время работы алгоритма t в зависимости от k , которое показывает как долго исполнялся алгоритм для каждого значения k , при этом алгоритм исполнялся при помощи параллельных вычислений на 4 ядрах.

Таблица 2. Сравнение маршрутизации при различных вероятностях существования узлов при $n=10, k=1$
Table 2. Comparison of routing for different probabilities of node existence with $n=10, k=1$

Город (из)	Город (в)	Приоритет	Город (в)	Приоритет
$n=10, k=1$	Вероятности=[1,1,1,1,1,1,1,1,1,1]		Вероятности=[0.865, 0.8515, 0.94, 0.856, 0.9685, 0.8725, 0.919, 0.994, 0.9235, 0.892, 0.9265]	
Выборг(0)	Сланцы(1)	0.80	Сланцы(1)	0.69
	Сосновый Бор(2)	0.80	Сосновый Бор(2)	0.67
	Сясьстрой(3)	0.80	Сясьстрой(3)	0.65
	Светогорск(4)	0.80	Светогорск(4)	0.64
	Подпорожье(5)	0.80	Подпорожье(5)	0.64
	Тихвин(6)	0.80	Тихвин(6)	0.62
	Сясьстрой(7)	0.80	Сясьстрой(7)	0.60
	Волхов(8)	0.80	Волхов(8)	0.59
	Кириши(9)	0.80	Кириши(9)	0.59
...				
Кириши(9)	Выборг(0)	0.80	Выборг(0)	0.62
	Сланцы(1)	0.10	Сланцы(1)	0.08

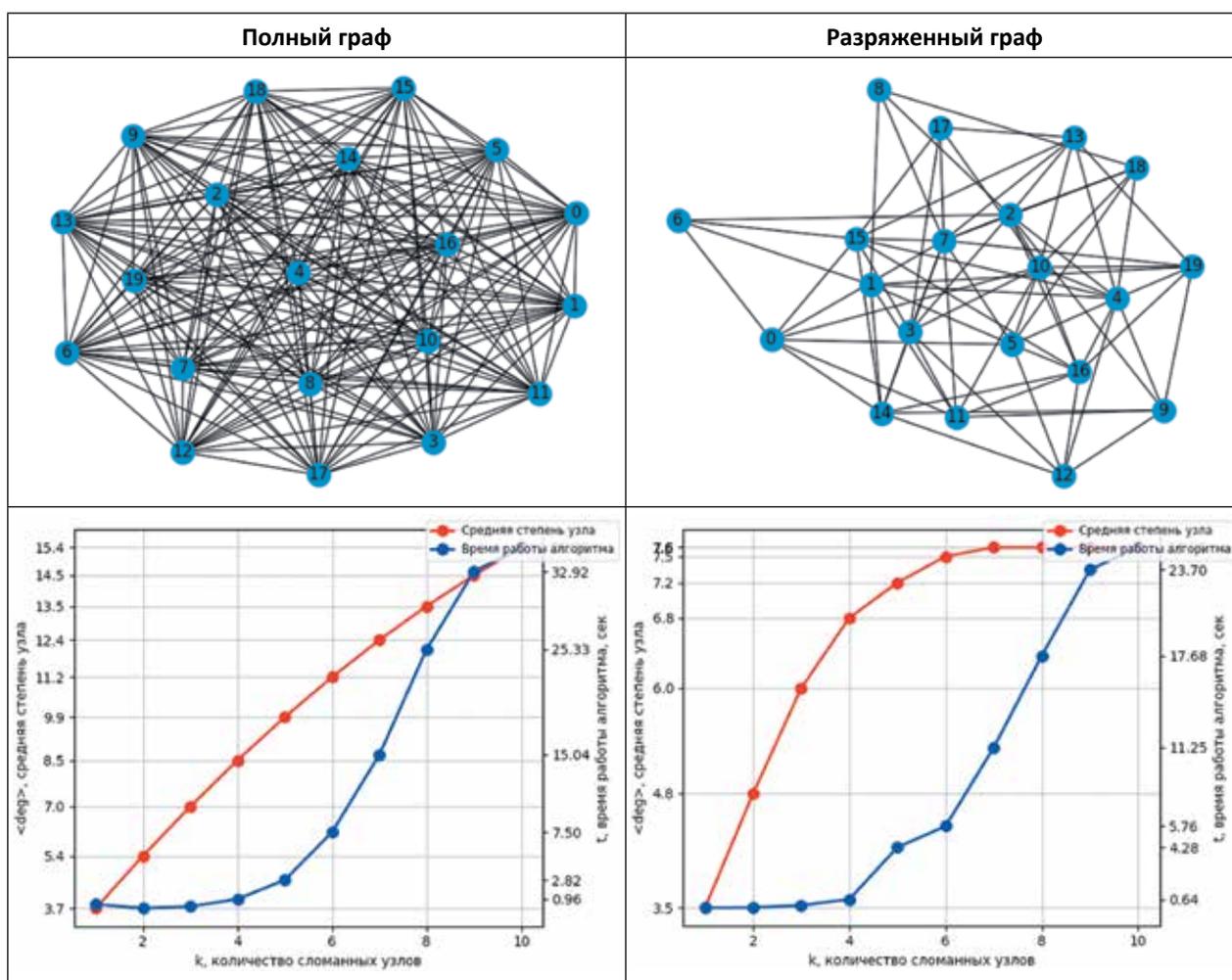


Рис. 2. Результаты работы алгоритма для полного и разреженного графа
Fig. 2. Results of the algorithm for a complete and sparse graph

3.2. Псевдокод

Входящими параметрами функции вычисления объединения остовных деревьев являются: *graph* — начальный граф G , k — количество вышедших из строя узлов. На основе количества узлов в графе и количества вышедших из строя узлов высчитываются все возможные комбинации, после чего строится начальное MST на основе первой комбинации из всех возможных, и это MST считается гло-

бальным в рамках функции. Далее для каждой оставшейся комбинации строится MST и осуществляется проверка того, что ребра построенного MST входят в глобальное. При совпадении ребер с глобальным MST к весам ребер глобального прибавляется 1, иначе ребрам вычисленного ставится значение 1. После этого глобальное MST обновляется путем комбинирования себя самого с вычисленным.

```
function umst(graph, k):
  n = number of nodes in graph
  combinations = all combinations of n nodes taken k at a time
  span_tree = minimum spanning tree of graph with first k nodes removed
  set all edge weights in span_tree to 1
  for each combination of nodes in combinations:
    cur_span_tree = minimum spanning tree of graph with removed nodes
    for each edge in cur_span_tree:
      if edge is in span_tree:
        increment weight of edge in span_tree by 1
      else:
        set weight of edge in cur_span_tree to 1
    span_tree = combine cur_span_tree and span_tree
  return span_tree
```

Рис. 3. Псевдокод алгоритма

Fig. 3. Algorithm pseudocode

4. Особенности прикладного применения

На примере сравнения узлов в виде городов Европы, представленных в табл. 3, можно видеть, как меняется построение маршрутизации при сохранении порядка узлов и изменение их мощностей с $[1,1,1,1,1,1]$ на $[1,2,1,3,1,4,1]$. В первом

случае у всех узлов была идентичная мощность и на результат маршрутизации влияла только вероятность существования ребра P_e , во втором же случае маршрутизация была перестроена и больший приоритет был отдан второму узлу, нежели первому, то есть ребра в маршрутизации изменились следующим образом $(1,3) \rightarrow (2,3)$ и $(1,5) \rightarrow (2,5)$.

Таблица 3. Сравнение отображений графов с различной мощностью при 7 узлах и 1 уничтоженном узле

Table 3. Comparison of graph mappings with different cardinality with 7 nodes and 1 destroyed node

Ребра начального графа	[(0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6), (5, 6)]	
Порядок узлов	Nicosia, Dublin, Oslo, Lisbon, Moscow, Rome, Berlin	
Мощности	[1,1,1,1,1,1,1]	[1,2,1,3,1,4,1]
Ребра UMST подграфа с вероятностями	(0,1): 0.71, (0,2): 0.71, (0,3): 0.71, (0,4): 0.71, (0,5): 0.71, (0,6): 0.71, (1,2): 0.14, (1,3): 0.14 , (1,4): 0.14, (1,5): 0.14 , (1,6): 0.14	(0,1): 0.71, (0,2): 0.71, (0,3): 0.71, (0,4): 0.71, (0,5): 0.71, (0,6): 0.71, (1,2): 0.14, (1,4): 0.14, (1,6): 0.14, (2,3): 0.14 , (2,5): 0.14
Отображение графов		

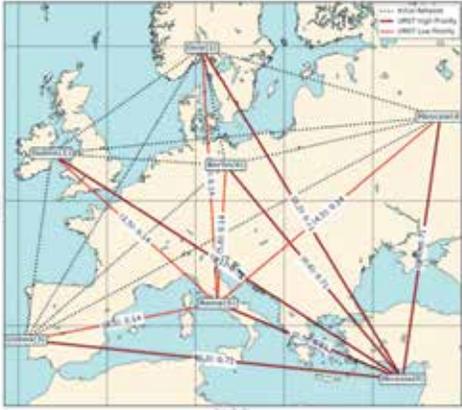
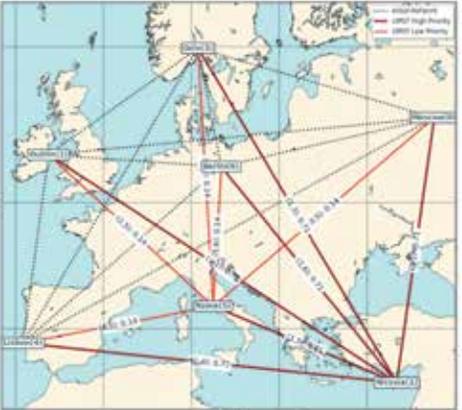
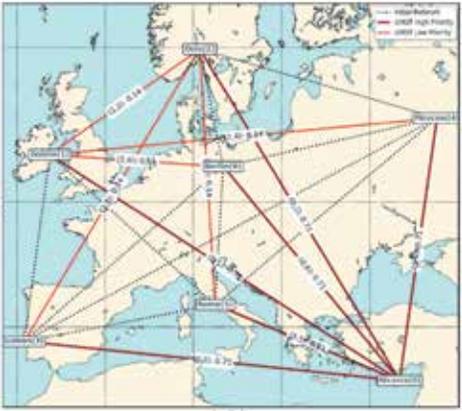
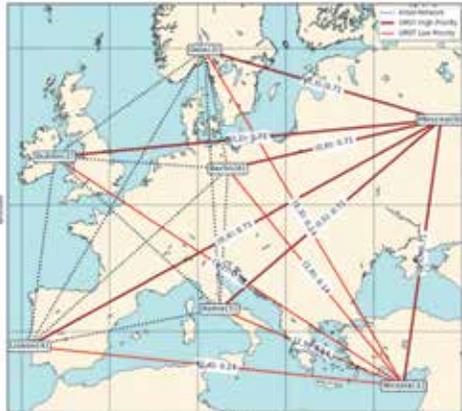
В табл. 4 приведено влияние изменений порядка узлов при одинаковых мощностях. В первом случае (при мощностях узлов 1,5,7,9,6,4,8), хотя порядок узлов и был изменен, проекция графа на карту не изменилась, в отличие от второго случая (при мощностях 1,2,1,3,1,4,1), в котором проекция изменилась. Это связано с тем, что городам были присвоены другие номера узлов, по факту конечный граф остался прежним.

5. Развитие алгоритма

Стоит рассмотреть подход поиска минимального дерева в работе [7], в которой авторы используют расширенную формулировку поиска MST для получения последовательности релаксаций QMSTP (квадратичного минимального остовного дерева), которые имеют полиномиальное количество ограничений и могут быть эффективно решены с помощью алгоритма секущей плоскости.

Таблица 4. Сравнение отображений графов с различным порядком узлов при 7 узлах и 1 уничтоженном узле

Table 4. Comparison of graph mappings with different orders of nodes with 7 nodes and 1 destroyed node

Ребра начального графа	[(0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (0, 6), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6), (5, 6)]	
Порядок узлов	Nicosia, Dublin, Oslo, Lisbon, Moscow, Rome, Berlin	Moscow, Nicosia, Dublin, Oslo, Lisbon, Rome, Berlin
Мощности	Nicosia=1, Dublin=5, Oslo=7, Lisbon=9, Moscow=6, Rome=4, Berlin=8 [1,5,7,9,6,4,8]	
Ребра УМСТ подграфа с вероятностями	(0,1): 0.71, (0,2): 0.71, (0,3): 0.71, (0,4): 0.71 , (0,5): 0.71, (0,6): 0.71 , (1,5): 0.14, (2,5): 0.14, (3,5): 0.14, (4,5): 0.14, (5,6): 0.14	(0,1): 0.71, (0,5): 0.14, (1,2): 0.71, (1,3): 0.71, (1,4): 0.71 , (1,5): 0.71, (1,6): 0.71 , (2,5): 0.14, (3,5): 0.14, (4,5): 0.14, (5,6): 0.14
Отображение графов		
Мощности	Nicosia=1, Dublin=2, Oslo=1, Lisbon=3, Moscow=1, Rome=4, Berlin=1 [1,2,1,3,1,4,1]	
Ребра УМСТ подграфа с вероятностями	(0,1): 0.71, (0,2): 0.71, (0,3): 0.71, (0,4): 0.71, (0,5): 0.71, (0,6): 0.71, (1,2): 0.14, (1,4): 0.14, (1,6): 0.14, (2,3): 0.14, (2,5): 0.14	(0,1): 0.71, (0,2): 0.71, (0,3): 0.71, (0,4): 0.71, (0,5): 0.71, (0,6): 0.71, (1,2): 0.14, (1,3): 0.14 , (1,4): 0.14, (1,5): 0.14 , (1,6): 0.14
Отображение графов		

6. Заключение

В работе сформулирована и решена задача построения маршрутизации для обеспечения отказоустойчивости сети связи. За основу алгоритма взято построение подграфа за счет объединения минимальных остовных деревьев и расчет вероятности существования ребра.

Алгоритм реализован и протестирован на основе нескольких наборов данных,

в частности для городов Ленинградской области и Европы. Также были описаны особенности прикладного применения разработанного алгоритма.

Алгоритм может использоваться для: (1) создания сетей SDN, (2) как виртуализированная сетевая функция (VNF). Также описано возможное дальнейшее развитие предложенного алгоритма.

КОНФЛИКТ ИНТЕРЕСОВ

Авторы заявляют об отсутствии конфликта интересов.

CONFLICT OF INTERESTS

The authors declare no relevant conflict of interests.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Ma J., Ma J., Wei J. Robustness of probability routing strategy in interconnection networks. *Engineering Research Express*. 2024;6(2):025414. <https://doi.org/10.1088/2631-8695/ad4849>
2. Barabási A.-L. *Network Science*. Cambridge University Press, 2016. 475 p. ISBN-10:1107076269
3. Albert R., Barabasi A.-L. Statistical mechanics of complex networks. *Reviews of Modern Physics*. 2002;74(1):47. <https://doi.org/10.1103/RevModPhys.74.47>
4. Smelyanskiy R., Shalimov A. (2018). *Big Data and Software Defined Networks*. Institution of Engineering and Technology (United Kingdom). 2018. <https://doi.org/10.1049/PBPC015E>
5. Синицын А. В., Синицын А. А., Жаворонкова О. Г. Алгоритм построения оптимальной маршрутизации для отказоустойчивой сети связи. *Информатизация и связь*, 2023;(5):49–56. <https://doi.org/10.34219/2078-8320-2023-14-5-49-56>
Sinitsyn A. V., Sinitsyn A. A., Zhavoronkova O. G. Algorithm for building optimal routing for a reliable communication network. *Informatization and Communication*. 2023;(5):49–56. (In Russ.). <https://doi.org/10.34219/2078-8320-2023-14-5-49-56>
6. Belkacem, Salima Nait and Abbas, Moncef (2022). New Algorithm Permitting the Construction of an Effective Spanning Tree. *Foundations of Computing and Decision Sciences*, vol.48, no.3, 2023, pp. 313–329. <https://doi.org/10.2478/fcds-2023-0012>
7. Sotirov R., Verchére Z. The quadratic minimum spanning tree problem: lower bounds via extended formulations. *Vietnam Journal of Mathematics*. 2024. <https://doi.org/10.1007/s10013-024-00694-y>

ИНФОРМАЦИЯ ОБ АВТОРАХ

Александр Владимирович СИНИЦЫН, к.ф.-м.н., заведующий кафедрой, доцент кафедры информационных процессов и систем, федеральное государственное бюджетное образовательное учреждение высшего образования «МИРЭА — Российский технологический университет» (ПТУ МИРЭА), 119454, Москва, проспект Вернадского, 78, Россия; ORCID: <https://orcid.org/0000-0001-7392-1837>; website: <http://alexander.sinitsyn.info>; e-mail: alexander@initsyn.info;

Глеб Олегович СОРОКИН, федеральное государственное бюджетное образовательное учреждение высшего образования «МИРЭА — Российский технологический университет», (ПТУ МИРЭА), 119454, Москва, проспект Вернадского, 78, Россия; ORCID: <https://orcid.org/0009-0005-1755-0193>; e-mail: sorokinglebw@yandex.ru;

INFORMATION ABOUT THE AUTHORS

Alexander Vladimirovich SINITSYN, Ph.D., Head of Department, Associate Professor of the Department of Information Processes and Systems, Federal State Budgetary Educational Institution of Higher Education “MIREA — Russian Technological University” (RTU MIREA), 119454, Moscow, Vernadsky Avenue, 78, Russia; ORCID: <https://orcid.org/0000-0001-7392-1837>; website: <http://alexander.sinitsyn.info>; e-mail: alexander@initsyn.info

Gleb Olegovich SOROKIN, Federal State Budgetary Educational Institution of Higher Education “MIREA — Russian Technological University”, (RTU MIREA), 119454, Moscow, Vernadskogo Avenue, 78, Russia; ORCID: <https://orcid.org/0009-0005-1755-0193>; e-mail: sorokinglebw@yandex.ru

Поступила в редакцию / Received 31.05.2024

