

МЕТОД РЕКОНФІГУРАЦІЇ РОЗПОДІЛЕНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ, ЩО ЗАЗНАЛА ЧАСТКОВОГО ФІЗИЧНОГО РУЙНУВАННЯ

Державний університет «Київський авіаційний інститут»

Вступ

Розподілена інформаційна система (РІС) – це складний апаратно-програмний комплекс, технічною базою якого є центри обробки даних (вузли), поєднані між собою каналами передачі даних [1]. Як і будь-які матеріальні об'єкти, РІС схильні до руйнування внаслідок аварій, техногенних катастроф, воєнних дій агресора тощо. Фізичному руйнуванню піддаються апаратні компоненти центрів обробки даних (ЦОД) та дротові канали зв'язку, тоді як бездротові канали можуть бути виведені з ладу засобами радіоелектронної боротьби (РЕБ) противника.

Часткове фізичне руйнування РІС викликає необхідність оперативного прийняття комплексу управлінських, проектних і технічних рішень, спрямованих на відновлення функціональних можливостей системи у максимальному обсязі та у мінімальний термін [2]. Зокрема при знищенні або блокуванні засобами РЕБ каналів безпосереднього зв'язку між вцілілими центрами обробки даних виникає проблема встановлення обхідних маршрутів передачі інформації, що проходять через інші (проміжні) ЦОД, з подальшим перерозподілом інформаційних потоків.

При такому перерозподілі необхідно враховувати інтенсивність інформаційних потоків та показники наявних ресурсів, зокрема продуктивність ЦОД, пропускну спроможність обладнання, що забезпечує передачу даних тощо [3]. Може так статися, що наявних ресурсів недостатньо, щоб підтримувати функціонування

вцілілої частини РІС у належному режимі. Тому при встановленні обхідних маршрутів доцільно передбачити декілька варіантів з'єднання пар вузлів РІС, щоб (за необхідності) забезпечити розпаралелювання інформаційних потоків [4].

З огляду на те, що в Україні зберігається висока ймовірність часткового фізичного руйнування розподілених інформаційних систем внаслідок воєнних дій, особливо актуальною є задача розробки методу реконфігурації та збереження працездатності таких систем.

Аналіз останніх досліджень та публікацій

Існуючі методи побудови маршрутів на графах [5-8] орієнтовані переважно на пошук одиночних (оптимальних) маршрутів на зважених графах, ребрам яких зіставлені вагові коефіцієнти. Критерієм оптимальності маршруту є сума вагових коефіцієнтів ребер, що входять до цього маршруту.

Зокрема алгоритм Краскала [5] полягає в побудові мінімального кістякового дерева (*minimum spanning tree* – *MST*) шляхом послідовного додавання найменших за вагою ребер, поки всі вузли не будуть з'єднані. На першому етапі виконання алгоритму створюється порожня множина *MST*. Далі всі ребра графа сортуються за зростанням, після чого ребра з мінімальною вагою додаються до множини *MST*. Під час додавання ребер перевіряється наявність циклів у кістяковому дереві.

Принцип роботи алгоритму Лі [6] полягає в поширенні так званої хвилі, яка

проходить від початкової точки через усі сусідні вершини, досягаючи зрештою кінцевої точки. Найкоротший шлях визначається за допомогою зворотного проходу через вершини, відстань до яких є найменшою. Такий шлях вважається оптимальним та найкоротшим маршрутом. Якщо на етапі початкового проходу графа хвиля не досягає кінцевої точки, шляху не існує.

Класичним алгоритмом пошуку найкоротшого маршруту в графі є відомий алгоритм Дейкстри (*Dijkstra's Algorithm*) [7]. Принцип його роботи полягає у виборі початкової вершини та поступовому розширенні множини відвіданих вершин. На кожному кроці обирається вершина з найменшою відстанню до поточної. Даний алгоритм використовується в протоколі динамічної маршрутизації *OSPF*.

Принцип роботи алгоритму Беллмана-Форда [8] полягає в проходженні через всі ребра графа та поступовому покращенні оцінки найкоротших шляхів за допомогою методу релаксації. Під час ініціалізації алгоритму припускається, що відстані до всіх точок прямують до нескінченності. Далі відбувається перевірка можливості зменшення довжини маршруту шляхом використання іншої точки як проміжної. Процес відбувається доти, поки всі відстані не стануть оптимальними. Даний алгоритм використовується в маршрутизації мереж, зокрема в протоколі *RIP*.

Сучасні дослідження пропонують різноманітні підходи до відновлення зв'язності, адаптивності та відмовостійкості розподілених інформаційних систем шляхом реконфігурації.

Зокрема в статті *Distributed Computation and Reconfiguration in Actively Dynamic Networks* [9] розглянуто модель розподіленої реконфігурації в динамічних мережах, де кожен з вузлів може змінювати власну локальну топологію. Наведені алгоритми локальної реконфігурації, які дозволяють досягти зменшення діаметра мережі та побудови кластеризованих чи ієрархічних децентралізованих структур. В дослідженні мережа моделюється у

вигляді графа, вершинами якого є вузли системи, а ребрами – комунікаційні з'єднання.

У статті *An Efficient Mathematical Model for Distribution System Reconfiguration Using AMPL* [10] представлено математичну модель для реконфігурації енергорозподільчих мереж з метою мінімізації втрат потужності. Система моделюється у вигляді графа, в якому вузлами є підстанції чи точки розподілу, а ребра – це лінії електропередач. Обмеження моделі задаються системою лінійних нерівностей, які враховують обмеження потужності ліній, закони Кірхгофа та необхідність збереження зв'язності. Задача вирішується шляхом комбінаторної оптимізації. Хоча дана модель і орієнтована на електромережі, її можна адаптувати і до ІТ-систем, в яких важливим фактором є можливість реконфігурації графа за збереження зв'язності та мінімізації витрат.

Авторами статті *Reasoning about Reconfigurations of Distributed Systems* [11] було розроблено формальну логіку для верифікації правильності програм реконфігурації розподілених систем. В рамках дослідження було впроваджено ресурсну логіку, яка дозволяє описати, які елементи (вузли або зв'язки між ними) використовуються та змінюються в процесі реконфігурації. Для опису допустимих переходів між станами системи було розроблено спеціальну алгебру конфігурацій. Окрім того, в дослідженні було застосовано з теорії формальних доведень (Логіка Гоара). Отримані результати є важливими для реконфігурації критичних розподілених інформаційних систем, оскільки використаний підхід дозволяє забезпечити коректність подальшої роботи системи.

Постановка задачі

Наведені алгоритми побудови маршрутів на графах мають свої переваги та недоліки, але всі вони є малоприслужними для розв'язання проблеми відновлення функціональних можливостей РІС, що зазнала часткового руйнування. Причиною цього є те, що фізичне знищення частки вузлів та каналів зв'язку РІС вимагає, як правило,

оперативного перерозподілу функцій та інформаційних потоків між вцілілими центрами обробки даних. Це багатоваріантна задача, яка розв'язується з урахуванням наявних обчислювальних ресурсів ЦОД та пропускної спроможності ланок інформаційних маршрутів. Тому для її розв'язання необхідно мати у полі розгляду декілька варіантів з'єднання ЦОД, між якими втрачений зв'язок, тоді як довжина інформаційних маршрутів (у сенсі суми вагових коефіцієнтів їх ланок) у критичних ситуаціях набуває другорядного значення.

Метою статті є викладення методу побудови багатоланкових маршрутів обміну інформації між центрами обробки даних, які не пов'язані каналами зв'язку безпосередньо.

Виклад основного матеріалу

У формальній постановці задача визначення багатоланкових маршрутів передачі інформації між ЦОД зводиться до побудови простих скінченних ланцюгів на графі $G = (V, E)$, що відбиває топологію вцілілої частини РІС.

Елементи множини вершин графа V зіставлені центрам обробки даних РІС, а елементи множини ребер E – каналам безпосередніх зв'язків між ЦОД.

Оскільки канали зв'язку РІС використовуються, як правило, для двостороннього обміну інформації, граф G можна вважати неорієнтованим.

Передбачається, що вершини графа пронумеровані числами натурального ряду від 1 до m ($i = \overline{1, m}$), а всі ребра – від 1 до n ($j = \overline{1, n}$).

Формально граф G описується сукупністю вузлових підмножин ребер:

$$S = \{J_i; i = \overline{1, m}\},$$

де J_i – підмножина номерів ребер, інцидентних вершині $v_i \in V$.

На основі сукупності S визначаються множини (пари) вершин I_j , яким інцидентне j -е ребро графа:

$$I_j = \{i: (1 \leq i \leq m) \& (j \in J_i)\}; j = \overline{1, n}.$$

Маршрут називають скінченним, якщо для нього задані початкова і кінцева вершини, і односторонньо-нескінченним, якщо задана лише одна з кінцевих вершин.

Довжина маршруту вимірюється кількістю ребер, які до нього входять.

Простим ланцюгом на графі вважається маршрут, в який кожна вершина та кожне ребро входить не більше одного разу.

Нехай i_1 та i_2 – вершини графа G , що відповідають вузлам РІС, між якими внаслідок тих чи інших причин знищений канал безпосереднього зв'язку.

Необхідно побудувати множину простих скінченних ланцюгів між вершинами i_1 та i_2 довжиною k ребер для всіх значень параметру k від 1 до r ($k = \overline{1, r}$), де r – задана максимальна довжина маршруту.

Ідея методу

Пропонований метод передбачає побудову простих скінченних ланцюгів між заданими вершинами графа у два етапи.

На першому з них послідовно обчислюються множини $J_k(i_1, i_2)$ номерів ребер, які можуть займати k -у позицію у шуканих скінченних ланцюгах, що пов'язують вершини i_1 та i_2 :

$$J_k(i_1, i_2) = J_k(i_1) \cap J_{r-k+1}(i_2); k = \overline{1, r}, \quad (1)$$

де $J_k(i_1)$ – множина номерів ребер, які можуть займати k -у позицію в односторонньо-нескінченних простих ланцюгах, початковою вершиною яких є вершина i_1 ; $J_{r-k+1}(i_2)$ – множина номерів ребер, які можуть займати $(r - k + 1)$ -у позицію в односторонньо-нескінченних простих ланцюгах, початковою вершиною яких є вершина i_2 .

Для обчислення множин $J_k(i_1)$ та $J_{r-k+1}(i_2)$, $k = \overline{1, r}$ використовується єдиний покроковий алгоритм, тому для спрощення його опису введемо узагальнене позначення v початкових вершин i_1 та i_2 односторонньо-нескінченних простих ланцюгів: $v \in \{i_1, i_2\}$.

Нехай $I_k(v)$ – множина вершин графа, які у шуканих ланцюгах є

початковими для ребер, що можуть займати k -у позицію; $k = \overline{1, r}$.

Пропонований метод базується на двох твердженнях, які є очевидними і не вимагають доведення.

Твердження 1. Якщо дві вершини графа x та z з'єднані між собою ребром у

$$(I_y = \{x, z\}) \& [x \in I_{k-1}(v)] \& [z \in I_k(v)] \rightarrow (\{z\} = I_y \setminus \{x\}).$$

Твердження 2. Якщо вершини графа x та z з'єднані між собою ребром у та у частково побудованому ланцюгу вершина x безпосередньо передуює кінцевій вершині z , то множина ребер Y_z , що можуть бути

і при цьому x та u відомі, то вершина z може бути встановлена згідно з формулою:

$$\{z\} = I_y \setminus \{x\}. \quad (2)$$

Формальний вираз цього твердження має такий вигляд:

$$Y_z = J_z \setminus \{y\}. \quad (3)$$

Це твердження описується таким виразом:

$$(I_y = \{x, z\}) \& [x \in I_{k-1}(v)] \& [z \in I_k(v)] \rightarrow (Y_z = J_z \setminus \{y\}).$$

Очевидно, що для обчислення множини номерів ребер $J_k(v)$ необхідно знати склад множини ідентифікаторів вершин $I_k(v)$. З іншого боку, множина ідентифікаторів вершин $I_k(v)$ обчислюється, виходячи із складу множини номерів ребер $J_{k-1}(v)$.

$$I_1(v) \rightarrow J_1(v) \rightarrow I_2(v) \rightarrow J_2(v) \rightarrow \dots \rightarrow I_r(v) \rightarrow J_r(v).$$

У цій послідовності множина $I_k(v)$ є множиною вершин, які у шуканих ланцюгах є кінцевими для ребер $j \in J_{k-1}(v)$, а множина $J_k(v)$ є множиною номерів ребер, що передують вершинам $I_{k+1}(v)$.

Однак такий підхід може призвести до втрати інформації про частку шуканих ланцюгів, що йдуть від початкової вершини v . Пояснюється це тим, що у загальному випадку до будь-якої вершини $i \in I_k(v)$ можна прийти різними шляхами, отже, цій вершині можуть передувати різні ребра. Тому формування множини $I_k(v)$ шляхом виконання математичної операції об'єднання вершин, що є кінцевими для ребер $j \in J_{k-1}(v)$, може призвести до виведення з поля подальшого розгляду деяких із шуканих ланцюгів.

Це обумовлює необхідність використання в алгоритмі обчислення множин $J_k(v)$, $k = \overline{1, r}$ кортежів – впорядкованих наборів, які можуть містити декілька екземплярів одного елементу. Будемо вважати,

Тому алгоритм встановлення множин $J_k(v)$, $k = \overline{1, r}$, які можуть займати k -у позицію у шуканих скінченних ланцюгах, можна було б представити у вигляді такої послідовності обчислень:

що множині $I_k(v)$ поставлений у відповідність кортеж $Z_k(v)$, а множині $J_k(v)$ – кортеж $Y_k(v)$; $k = \overline{1, r}$.

Зміст цих кортежів такий: $Z_k(v) = \langle z_{kl} \mid l = \overline{1, p_k} \rangle$ – кортеж ідентифікаторів вершин, які одночасно є кінцевими для ребер, що у шуканих ланцюгах можуть займати $(k - 1)$ -у позицію, і початковими для ребер, що можуть займати наступну k -у позицію; $Y_k(v) = \langle y_{kl} \mid l = \overline{1, q_k} \rangle$ – кортеж номерів ребер, для яких вершини z_{kl} $l = \overline{1, p_k}$ є початковими, тобто ребер, що у шуканих ланцюгах можуть займати k -у позицію; $k = \overline{1, r}$.

У наведених формулах p_k та q_k – показники розмірності кортежів $Z_k(v)$ та $Y_k(v)$ відповідно.

Для спрощення обчислень та уникнення невизначеності доцільно ввести в алгоритм додатковий кортеж $X_k(v) = \langle x_{kl} \mid l = \overline{1, q_k} \rangle$, елементами якого є

ідентифікатори вершин, що є початковими для ребер y_{kl} , $l = \overline{1, q_k}$.

$$\{z_{kl}\} = I_{y_{k-1,l}} \setminus \{x_{k-1,l}\}; \quad k = \overline{2, r}; \quad l = \overline{1, p_k}. \quad (4)$$

Розмірність кортежу $Z_k(v)$ дорівнює розмірності кортежу $Y_{k-1}(v)$: $p_k = q_{k-1}$. Кортежі $Y_{k-1}(v)$ та $X_{k-1}(v)$ мають однакову розмірність q_{k-1} .

На першому кроці (для $k = 1$) кортеж $Z_1(v)$ складається з єдиного елементу – ідентифікатора початкової вершини v односторонньо-нескінченних простих ланцюгів, що будуються:

$$Z_1(v) = \langle z_{11} \rangle = \langle v \rangle.$$

Кортеж $Y_k(v)$ складається з елементів підмножин $J_{kl}(v)$, які містять номери ребер, інцидентних вершині z_{kl} , за виключенням ребра $y_{k-1,l}$. Елементи підмножини $J_{kl}(v)$ визначають ребра, які у шуканих ланцюгах можуть бути їх продовженнями від вершини z_{kl} . Склад цих підмножин визначається формулою:

$$J_{kl}(v) = J_{z_{kl}} \setminus \{y_{k-1,l}\}; \quad k = \overline{2, r}; \quad l = \overline{1, p_k}. \quad (5)$$

При формуванні кортежу $Y_k(v)$ дотримується черговість підмножин $J_{kl}(v)$ згідно з їх нумерацією, а номери ребер, що повторюються у різних підмножинах, не виключаються.

Розмірність кортежу $Y_k(v)$ обчислюється за формулою:

$$q_k = \sum_{l=1}^{p_k} |J_{kl}(v)| - p_k; \quad k = \overline{2, r}.$$

На першому кроці (для $k = 1$) кортеж $Y_1(v)$ складається з номерів ребер, інцидентних початковій вершині v :

$$Y_1(v) = \langle y_{1l} \mid l = \overline{1, q_1} \rangle,$$

де $y_{1l} = j \in J_v$, а q_1 – потужність множини J_v : $q_1 = |J_v|$.

Множина $J_k(v)$ номерів ребер, які в односторонньо-нескінченних простих ланцюгах, що починаються з вершини v , можуть займати k -у позицію, обчислюється шляхом об'єднання елементів кортежу $Y_k(v)$:

$$J_k(v) = \bigcup_{l=1}^{q_k} \{y_{kl}\}; \quad k = \overline{1, r}. \quad (6)$$

Кортежі $X_k(v)$ формуються лише для кроків $k = \overline{2, r-1}$, оскільки на останньому ($k = r$)-у кроці в цьому немає необхідності.

Елементи кортежу $Z_k(v)$ встановлюються, виходячи із елементів кортежів $Y_{k-1}(v)$ та $X_{k-1}(v)$, згідно з формулою:

Обчислення множин $J_k(v)$ окремо для $v = i_1$ та $v = i_2$ надає змогу за формулою (1) встановити множину $J_k(i_1, i_2)$ номерів ребер, що можуть займати k -у позицію у зустрічних скінченних ланцюгах, початковими вершинами яких є вершини i_1 та i_2 .

Встановленням множин $J_k(i_1, i_2)$ для всіх $k = \overline{1, r}$ завершується перший етап реалізації запропонованого методу.

Множини $J_k(i_1, i_2)$, $k = \overline{1, r}$ дають змогу будувати скінченні ланцюги між вершинами i_1 та i_2 будь-якої довжини r' в межах $1 \leq r' \leq r$.

На другому етапі здійснюється покрокова побудова шуканих скінченних ланцюгів, кожен з яких представляється послідовністю номерів ребер j_k , $k = \overline{1, r}$, що займають k -у позицію:

$$L = (j_k \mid k = \overline{1, r}).$$

На першому кроці (для $k = 1$) ребро j_1 обирається із множини $J_1(i_1, i_2)$.

На всіх подальших кроках (для $k = \overline{2, r}$) чергове ребро j_k обирається із множини $J^D(j_k)$ – сукупності номерів ребер, які у ланцюзі, що будується, можуть займати k -у позицію. Склад множини $J^D(j_k)$ залежить від того, яке саме ребро займає попередню ($k - 1$)-у позицію, і обчислюється за формулою:

$$J^D(j_k) = J_{i^K(j_{k-1})} \cap J_k(i_1, i_2), \quad (7)$$

де $i^K(j_{k-1})$ – ідентифікатор вершини, яка в ланцюзі, що будується, є кінцевою для попереднього ребра j_{k-1} :

$$\{i^K(j_{k-1})\} = I_{j_{k-1}} \setminus \{i^H(j_{k-1})\}, \quad (8)$$

де $i^H(j_{k-1})$ – ідентифікатор початкової вершини ребра j_{k-1} .

Початкова вершина для всіх елементів множини $J_1(i_1, i_2)$ збігається з початковою вершиною шуканих ланцюгів:

$$[\forall j \in J_1(i_1, i_2)][i^H(j) = i_1].$$

Якщо на деякому k -у кроці виявляється, що множина $J^D(j_k)$ містить більше одного елементу, це означає, що після вершини $i^H(j_k)$ частково побудований ланцюг розгалужується. У такому випадку надалі окремо добудовуються всі можливі продовження цього часткового маршруту.

Алгоритми

Пропонований метод передбачає реалізацію двох алгоритмів. Перший з них призначений для обчислення множин $J_k(i_1, i_2)$, $k = \overline{1, r}$, другий – для побудови ланцюгів ребер L .

Перший алгоритм реалізується окремо для початкових вершин $v = i_1$ та $v = i_2$ і передбачає послідовне виконання на кожному k -у кроці ($k = \overline{1, r}$) таких дій.

1. Встановлення кінцевих вершин ребер, номери яких є елементами кортежу $Y_{k-1}(v)$, згідно з формулою (4). На першому кроці ($k = 1$) ця операція не виконується.

2. Формування кортежу $Z_k(v)$, елементами якого стають ідентифікатори вершин, кінцевих для ребер з номерами із кортежу $Y_{k-1}(v)$. На першому кроці ($k = 1$) кортеж $Z_k(v)$ складається з єдиного елемента: $Z_1(v) = \langle z_{11} \rangle = \langle v \rangle$.

3. Обчислення підмножин номерів ребер $J_{kl}(v)$ для кожного елемента кортежу вершин $Z_k(v)$ за формулою (5).

4. Формування кортежу $Y_k(v)$, елементами якого стають номери ребер із всіх підмножин $J_{kl}(v)$.

5. Обчислення множини $J_k(v)$ номерів ребер, які в односторонньо-нескінченних простих ланцюгах, що починаються з вершини v , можуть займати k -у позицію, за формулою (6).

6. Формування кортежу $X_k(v)$, що складається з ідентифікаторів вершин, початкових для ребер, номери яких є елементами кортежу $Y_k(v)$.

Алгоритм побудови скінченних ланцюгів L , що з'єднують вершини i_1 та i_2 , реалізується окремо для кожного елемента $j \in J_1(i_1, i_2)$ та полягає у наступному.

На першому кроці ($k = 1$) обирається ребро $j_1 \in J_1(i_1, i_2)$, яке у подальшому розглядається як перша ланка ланцюга, що будується.

На всіх подальших кроках $k = \overline{2, r}$ послідовно виконуються такі дії.

1. Встановлення кінцевої вершини $i^K(j_{k-1})$ ребра $j_{k-1} \in J^D(j_{k-1})$ згідно з формулою (8).

2. Обчислення за формулою (7) множини $J^D(j_k)$ номерів ребер, які у ланцюгу, що будується, можуть займати k -у позицію.

3. Обрання ребра $j_k \in J^D(j_k)$ для заповнення k -ї позиції у ланцюгу, що будується.

Після побудови всіх ланцюгів, які починаються ребром $j^* = j_1$, його номер видаляється із складу множини $J_1(i_1, i_2)$ і формується скорочена множина номерів ребер, що у шуканих ланцюгах можуть займати першу позицію:

$$J_1^*(i_1, i_2) = J_1(i_1, i_2) \setminus \{j^*\}.$$

Побудовані ланцюги нумеруються числами натурального ряду, починаючи з одиниці.

Приклад

Заданий неорієнтований граф, який відображає структуру фрагменту вцілілої частини РІС, що зазнала часткового фізичного руйнування. Припустимо, що граф має 7 вершин ($m = 7$) та 11 ребер ($n = 11$). Для більшої наочності обчислень вершини графа позначені літерами, а ребра – натуральними числами (рис.).

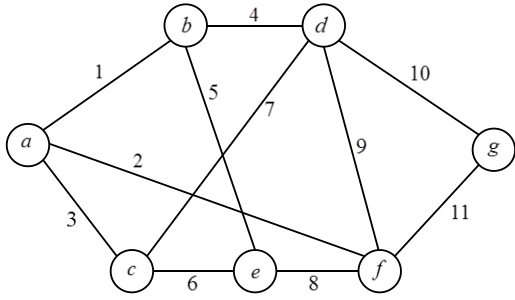


Рис. Приклад структури фрагменту РІС

Наведений граф описується сукупністю вузлових підмножин ребер:

$$S = \{J_a; J_b; J_c; J_d; J_e; J_f; J_g\},$$

де $J_a = \{1, 2, 3\}$; $J_b = \{1, 4, 5\}$; $J_c = \{3, 6, 7\}$; $J_d = \{4, 7, 9, 10\}$; $J_e = \{5, 6, 8\}$; $J_f = \{2, 8, 9, 11\}$; $J_g = \{10, 11\}$.

Множини I_j , $j = \overline{1, n}$ ідентифікаторів вершин, яким інцидентні ребра графа: $I_1 = \{a, b\}$; $I_2 = \{a, f\}$; $I_3 = \{a, c\}$; $I_4 = \{b, d\}$; $I_5 = \{b, e\}$; $I_6 = \{c, e\}$; $I_7 = \{c, d\}$; $I_8 = \{e, f\}$; $I_9 = \{d, f\}$; $I_{10} = \{d, g\}$; $I_{11} = \{f, g\}$.

Припустимо, що внаслідок часткового фізичного руйнування або дій засобів РЕБ знищений канал безпосереднього зв'язку між вузлами a та g .

Необхідно знайти множину простих ланцюгів, що пов'язують вузли a і g та

$$J_{21}(a) = J_{z_{21}} \setminus \{y_{11}\} = \{1, 4, 5\} \setminus \{1\} = \{4, 5\},$$

$$J_{22}(a) = J_{z_{22}} \setminus \{y_{12}\} = \{2, 8, 9, 11\} \setminus \{2\} = \{8, 9, 11\},$$

$$J_{23}(a) = J_{z_{23}} \setminus \{y_{13}\} = \{3, 6, 7\} \setminus \{3\} = \{6, 7\}.$$

4. З елементів множин $J_{2l}(a)$, $l = \overline{1, 3}$ формується кортеж номерів ребер, що в односторонньо-нескінченних простих

$$Y_2(a) = \langle y_{2l} | l = \overline{1, 3} \rangle = \langle 4, 5, 8, 9, 11, 6, 7 \rangle$$

5. За формулою (6) обчислюється множина номерів таких ребер:

$$J_2(a) = \{4, 5, 6, 7, 8, 9, 11\}.$$

складаються з двох або трьох ланок ($r = 3$).

Алгоритм обчислення множин номерів ребер, які можуть займати k -у позицію ($k = \overline{1, 3}$) в односторонньо-нескінченних простих ланцюгах, початковою вершиною яких є вершина a , полягає у послідовному виконанні таких дій.

Крок 1. Формуються початкові кортежі вершин та ребер:

$$Z_1(a) = \langle z_{11} \rangle = \langle a \rangle;$$

$$Y_1(a) = \langle y_{11}, y_{12}, y_{13} \rangle = \langle 1, 2, 3 \rangle.$$

Крок 2.

1. За формулою (4) встановлюються кінцеві вершини ребер, номери яких входять до складу кортежу $Y_1(a)$:

$$\{z_{21}\} = \{a, b\} \setminus \{a\} = \{b\};$$

$$\{z_{22}\} = \{a, f\} \setminus \{a\} = \{f\};$$

$$\{z_{23}\} = \{a, c\} \setminus \{a\} = \{c\}.$$

2. Формується кортеж

$$Z_2(a) = \langle z_{21}, z_{22}, z_{23} \rangle = \langle b, f, c \rangle.$$

3. За формулою (5) для кожного елемента z_{2l} ($l = \overline{1, 3}$) кортежу $Z_2(a)$ обчислюються множини номерів ребер, які у шуканих ланцюгах можуть бути їх продовженнями від вершини z_{2l} :

ланцюгах, початковою вершиною яких є вершина a , можуть займати другу позицію:

6. Формується кортеж початкових вершин ребер, номери яких входять до складу кортежу $Y_2(a)$:

$$X_2(a) = \langle x_{2l} | l = \overline{1, 7} \rangle = \langle b, b, f, f, f, c, c \rangle.$$

Крок 3.

1. За формулою (4) встановлюються кінцеві вершини ребер, номери яких входять до складу кортежу $Y_2(a)$:

$$\begin{aligned}\{z_{31}\} &= \{b, d\} \setminus \{b\} = \{d\}, \quad \{z_{32}\} = \{b, e\} \setminus \{b\} = \{e\}, \quad \{z_{33}\} = \{f, e\} \setminus \{f\} = \{e\}, \\ \{z_{34}\} &= \{f, d\} \setminus \{f\} = \{d\}, \quad \{z_{35}\} = \{f, g\} \setminus \{f\} = \{g\}, \\ \{z_{36}\} &= \{c, e\} \setminus \{c\} = \{e\}, \quad \{z_{37}\} = \{c, d\} \setminus \{c\} = \{d\}.\end{aligned}$$

2. Формується кортеж обчислюються множини номерів ребер, які у шуканих ланцюгах можуть бути їх продовженнями від вершини z_{3l} :

$$Z_3(a) = \langle z_{3l} \mid l = \overline{1, 7} \rangle = \langle d, e, e, d, g, e, d \rangle.$$

3. За формулою (5) для кожного елементу z_{3l} ($l = \overline{1, 7}$) кортежу $Z_3(a)$

$$\begin{aligned}J_{31}(a) &= J_{z_{31}} \setminus \{y_{21}\} = \{4, 7, 9, 10\} \setminus \{4\} = \{7, 9, 10\}, \\ J_{32}(a) &= J_{z_{32}} \setminus \{y_{22}\} = \{5, 6, 8\} \setminus \{5\} = \{6, 8\}, \\ J_{33}(a) &= J_{z_{33}} \setminus \{y_{23}\} = \{5, 6, 8\} \setminus \{8\} = \{5, 6\}, \\ J_{34}(a) &= J_{z_{34}} \setminus \{y_{24}\} = \{4, 7, 9, 10\} \setminus \{9\} = \{4, 7, 10\}, \\ J_{35}(a) &= J_{z_{35}} \setminus \{y_{25}\} = \{10, 11\} \setminus \{11\} = \{10\}, \\ J_{36}(a) &= J_{z_{36}} \setminus \{y_{26}\} = \{5, 6, 8\} \setminus \{6\} = \{5, 8\}, \\ J_{37}(a) &= J_{z_{37}} \setminus \{y_{27}\} = \{4, 7, 9, 10\} \setminus \{7\} = \{4, 9, 10\}.\end{aligned}$$

4. З елементів множин $J_{3l}(a)$, $l = \overline{1, 7}$ формується кортеж номерів ребер, що в односторонньо-нескінченних простих ланцюгах, початковою вершиною яких є вершина a , можуть займати другу позицію:

$$Y_3(a) = \langle y_{3l} \mid l = \overline{1, 16} \rangle = \langle 7, 9, 10, 6, 8, 5, 6, 4, 7, 10, 10, 5, 8, 4, 9, 10 \rangle.$$

5. За формулою (6) обчислюється множина номерів таких ребер:

$$J_3(a) = \{4, 5, 6, 7, 8, 9, 10\}.$$

За наведеним алгоритмом здійснюється й обчислення множин номерів ребер, які можуть займати k -у позицію ($k = \overline{1, 3}$) в односторонньо-нескінченних простих ланцюгах, початковою вершиною яких є вершина g .

Крок 1. Формуються початкові кортежи вершин та ребер:

$$\begin{aligned}Z_1(g) &= \langle z_{11} \rangle = \langle g \rangle, \\ Y_1(g) &= \langle y_{11}, y_{12} \rangle = \langle 10, 11 \rangle.\end{aligned}$$

$$\begin{aligned}J_{21}(g) &= J_{z_{21}} \setminus \{y_{11}\} = \{4, 7, 9, 10\} \setminus \{10\} = \{4, 7, 9\}, \\ J_{22}(a) &= J_{z_{22}} \setminus \{y_{12}\} = \{2, 8, 9, 11\} \setminus \{11\} = \{2, 8, 9\}.\end{aligned}$$

Крок 2.

1. За формулою (4) встановлюються кінцеві вершини ребер, номери яких входять до складу кортежу $Y_1(g)$:

$$\begin{aligned}\{z_{21}\} &= \{g, d\} \setminus \{g\} = \{d\}, \\ \{z_{22}\} &= \{g, f\} \setminus \{g\} = \{f\}.\end{aligned}$$

2. Формується кортеж $Z_2(g) = \langle z_{21}, z_{22} \rangle = \langle d, f \rangle$.

3. За формулою (5) для кожного елементу z_{2l} ($l = \overline{1, 2}$) кортежу $Z_2(g)$ обчислюються множини номерів ребер, які у шуканих ланцюгах можуть бути їх продовженнями від вершини z_{2l} :

4. З елементів множин $J_{2l}(g)$, $l = \overline{1, 2}$ формується кортеж номерів ребер, що в односторонньо-нескінченних простих ланцюгах, початковою вершиною яких є вершина g , можуть займати другу позицію:

$$Y_2(g) = \langle y_{2l} | l = \overline{1, 6} \rangle = \langle 4, 7, 9, 2, 8, 9 \rangle.$$

5. За формулою (6) обчислюється множина номерів таких ребер:

$$\begin{aligned} \{z_{31}\} &= \{b, d\} \setminus \{d\} = \{b\}, \quad \{z_{32}\} = \{c, d\} \setminus \{d\} = \{c\}, \quad \{z_{33}\} = \{f, d\} \setminus \{d\} = \{f\}, \\ \{z_{34}\} &= \{a, f\} \setminus \{f\} = \{a\}, \quad \{z_{35}\} = \{e, f\} \setminus \{f\} = \{e\}, \quad \{z_{36}\} = \{d, f\} \setminus \{f\} = \{d\}. \end{aligned}$$

2. Формується кортеж $Z_3(g) = \langle z_{3l} | l = \overline{1, 6} \rangle = \langle b, c, f, a, e, d \rangle$.

3. За формулою (5) для кожного елементу z_{3l} ($l = \overline{1, 6}$) кортежу $Z_3(g)$

$$\begin{aligned} J_{31}(g) &= J_{z_{31}} \setminus \{y_{21}\} = \{1, 4, 5\} \setminus \{4\} = \{1, 5\}, \\ J_{32}(g) &= J_{z_{32}} \setminus \{y_{22}\} = \{3, 6, 7\} \setminus \{7\} = \{3, 6\}, \\ J_{33}(g) &= J_{z_{33}} \setminus \{y_{23}\} = \{2, 8, 9, 11\} \setminus \{9\} = \{2, 8, 11\}, \\ J_{34}(g) &= J_{z_{34}} \setminus \{y_{24}\} = \{1, 2, 3\} \setminus \{2\} = \{1, 3\}, \\ J_{35}(g) &= J_{z_{35}} \setminus \{y_{25}\} = \{5, 6, 8\} \setminus \{8\} = \{5, 6\}, \\ J_{36}(g) &= J_{z_{36}} \setminus \{y_{26}\} = \{4, 7, 9, 10\} \setminus \{9\} = \{4, 7, 10\}. \end{aligned}$$

4. З елементів множин $J_{3l}(g)$, $l = \overline{1, 6}$ формується кортеж номерів ребер, що в односторонньо-нескінченних простих ланцюгах, початковою вершиною яких є вершина g , можуть займати другу позицію:

$$Y_3(g) = \langle y_{3l} | l = \overline{1, 14} \rangle = \langle 1, 5, 3, 6, 2, 8, 11, 1, 3, 5, 6, 4, 7, 10 \rangle$$

$$J_1(a, g) = J_1(a) \cap J_3(g) = \{1, 2, 3\} \cap \{1, 2, 3, 4, 5, 6, 7, 8, 10, 11\} = \{1, 2, 3\},$$

$$J_2(a, g) = J_2(a) \cap J_2(g) = \{4, 5, 6, 7, 8, 9, 11\} \cap \{2, 4, 7, 8, 9\} = \{4, 7, 8, 9\},$$

$$J_3(a, g) = J_3(a) \cap J_1(g) = \{4, 5, 6, 7, 8, 9, 10\} \cap \{10, 11\} = \{10\}.$$

Для ланцюгів довжиною $r' = 2$ склад цих множин такий:

$$J_1(a, g) = J_1(a) \cap J_2(g) = \{1, 2, 3\} \cap \{2, 4, 7, 8, 9\} = \{2\},$$

$$J_2(a, g) = J_2(a) \cap J_1(g) = \{4, 5, 6, 7, 8, 9, 11\} \cap \{10, 11\} = \{11\}.$$

Для ланцюгів довжиною $r' = 1$ склад цих множин такий:

$$J_2(g) = \{2, 4, 7, 8, 9\}.$$

6. Формується кортеж початкових вершин ребер, номери яких входять до складу кортежу $Y_2(g)$:

$$X_2(g) = \langle x_{2l} | l = \overline{1, 6} \rangle = \langle d, d, d, f, f, f \rangle.$$

Крок 3.

1. За формулою (4) встановлюються кінцеві вершини ребер, номери яких входять до складу кортежу $Y_2(g)$:

обчислюються множини номерів ребер, які у шуканих ланцюгах можуть бути їх продовженнями від вершини z_{3l} :

5. За формулою (6) обчислюється множина номерів таких ребер:

$$J_3(a) = \{1, 2, 3, 4, 5, 6, 7, 8, 10, 11\}.$$

За формулою (1) обчислюються множини номерів ребер, які можуть займати k -у позицію у шуканих скінченних ланцюгах довжиною $r = 3$, що пов'язують вершини a та g :

$$J_1(a, g) = J_1(a) \cap J_1(g) = \{1, 2, 3\} \cap \{10, 11\} = \emptyset.$$

Отримані результати свідчать про те, що між вершинами a та g існує декілька ланцюгів довжиною $r=3$ та один ланцюг довжиною $r'=2$, який складається з ребер $j_1=2$ і $j_2=11$. Безпосереднього зв'язку ($r'=1$) між вершинами a та g немає (що впливає з умов задачі), оскільки $J_1(a) \cap J_1(g) = \emptyset$.

Побудова скінченних ланцюгів, що з'єднують вершини a та g , для $r=3$ здійснюється за таким алгоритмом.

Крок 1. З множини $J_1(a, g) = \{1, 2, 3\}$ обирається один з елементів, наприклад $j_1=1$.

Крок 2.

1. За формулою (8) встановлюється кінцева вершина ребра j_1 :

$$i^K(1) = \{a, b\} \setminus \{a\} = \{b\}.$$

2. За формулою (7) обчислюється множина номерів ребер, які у ланцюгу, що будується, можуть займати другу позицію:

$$J^D(1) = \{1, 4, 5\} \cap \{4, 7, 8, 9\} = \{4\}.$$

3. Фіксація номера ребра, яке у ланцюгу, що будується, займе другу позицію: $j_2=4$.

Крок 3.

1. За формулою (8) встановлюється кінцева вершина ребра j_2 :

$$i^K(4) = \{b, d\} \setminus \{b\} = \{d\}.$$

2. За формулою (7) обчислюється множина номерів ребер, які у ланцюгу, що будується, можуть займати третю позицію:

$$J^D(4) = \{4, 7, 9, 10\} \cap \{10\} = \{10\}.$$

3. Фіксація номера ребра, яке у ланцюгу, що будується, займе третю позицію: $j_3=10$.

Отже, перший із шуканих ланцюгів представляється такою послідовністю ребер: $L_1 = (1, 4, 10)$.

Далі наведений алгоритм реалізується для скороченої множини номерів ребер $J_1^*(a, g) = \{1, 2, 3\} \setminus \{1\} = \{2, 3\}$.

Крок 1. З множини $J_1^*(a, g) = \{2, 3\}$ обирається один з елементів, наприклад $j_1=2$.

Крок 2.

1. За формулою (8) встановлюється кінцева вершина ребра j_1 :

$$i^K(2) = \{a, f\} \setminus \{a\} = \{f\}.$$

2. За формулою (7) обчислюється множина номерів ребер, які у ланцюгу, що будується, можуть займати другу позицію:

$$J^D(2) = \{2, 8, 9, 11\} \cap \{4, 7, 8, 9\} = \{8, 9\}.$$

3. Фіксація номера ребра, яке у ланцюгу, що будується, займе другу позицію: $j_2=8$.

Крок 3.

1. За формулою (8) встановлюється кінцева вершина ребра j_2 :

$$i^K(8) = \{e, f\} \setminus \{f\} = \{e\}.$$

2. За формулою (7) обчислюється множина номерів ребер, які у ланцюгу, що будується, можуть займати третю позицію:

$$J^D(8) = \{5, 6, 8\} \cap \{10\} = \emptyset.$$

Це означає, що ребро $j_2=8$ не має продовження, яке привело би до кінцевої вершини g ланцюга, що будується. Тому номер цього ребра виключається із складу множини $J^D(2)$ і третій крок алгоритму реалізується повторно для ребра $j_2=9$.

1. За формулою (8) встановлюється кінцева вершина ребра j_2 :

$$i^K(9) = \{d, f\} \setminus \{f\} = \{d\}.$$

2. За формулою (7) обчислюється множина номерів ребер, які у ланцюгу, що будується, можуть займати третю позицію:

$$J^D(9) = \{4, 7, 9, 10\} \cap \{10\} = \{10\}.$$

3. Фіксація номера ребра, яке у ланцюгу, що будується, займе третю позицію: $j_3 = 10$.

Отже, другий знайдений ланцюг представляється послідовністю ребер: $L_2 = (2, 9, 10)$.

Далі наведений алгоритм реалізується для скороченої множини номерів ребер $J_1^*(a, g) = \{1, 2, 3\} \setminus \{1, 2\} = \{3\}$.

Крок 1. З множини $J_1^*(a, g) = \{3\}$ обирається останній елемент $j_1 = 3$.

Крок 2.

1. За формулою (8) встановлюється кінцева вершина ребра j_1 :

$$i^K(3) = \{a, c\} \setminus \{a\} = \{c\}.$$

2. За формулою (7) обчислюється множина номерів ребер, які у ланцюгу, що будується, можуть займати другу позицію:

$$J^D(3) = \{3, 6, 7\} \cap \{4, 7, 8, 9\} = \{7\}.$$

3. Фіксація номера ребра, яке у ланцюгу, що будується, займе другу позицію: $j_2 = 7$.

Крок 3.

1. За формулою (8) встановлюється кінцева вершина ребра j_2 :

$$i^K(7) = \{c, d\} \setminus \{c\} = \{d\}.$$

2. За формулою (7) обчислюється множина номерів ребер, які у ланцюгу, що будується, можуть займати третю позицію:

$$J^D(7) = \{4, 7, 9, 10\} \cap \{10\} = \{10\}.$$

3. Фіксація номера ребра, яке у ланцюгу, що будується, займе третю позицію: $j_3 = 10$.

Отже, третій знайдений ланцюг представляється послідовністю ребер: $L_3 = (3, 7, 10)$.

Отримані результати обчислень інтерпретуються таким чином: між заданими вершинами a і g існують три скінченних простих ланцюга $L_1 = (1, 4, 10)$, $L_2 = (2, 9, 10)$ та $L_3 = (3, 7, 10)$, що складаються з трьох ребер, а також один ланцюг $L_4 = (2, 11)$, що містить два ребра.

Висновки

Доведено, що задача реконфігурації розподіленої інформаційної системи, яка зазнала часткового фізичного руйнування, може бути зведена до задачі побудови множини скінченних простих ланцюгів на неорієнтованому графі, який моделює структуру системи.

Пропонований метод передбачає розв'язання цієї задачі для кожної пари вузлів РІС, між якими знищений зв'язок, у два етапи. На першому з них формуються множини ребер, які потенційно можуть займати кожен порядковий позицію у шуканих ланцюгах. Для цього використовується запропонований алгоритм покрокової побудови односторонньо-нескінченних простих ланцюгів, початковою вершиною яких є вершина, що зіставлена одному з пари вузлів РІС, між якими втрачений зв'язок. Формування множини ребер, які потенційно можуть займати кожен порядковий позицію у шуканих ланцюгах, здійснюється шляхом перетину підмножин ребер, що знаходяться на відповідних позиціях в односторонньо-нескінченних простих ланцюгах.

На другому етапі здійснюється безпосередня побудова шуканих скінченних простих ланцюгів. Для цього використовується запропонований алгоритм, який передбачає покрокове приєднання кожного нового ребра, що обирається із сформованої на першому етапі відповідної множини потенційних ребер, до частково побудованого скінченного ланцюга.

Попереднє формування множин ребер, які потенційно можуть займати кожен порядковий позицію у шуканих ланцюгах, фактично виділяє на графі фрагмент, якому належать всі можливі розв'язки задачі, яка розглядається. Тим самим досягається ефект суттєвого зменшення розмірності задачі і, як наслідок, скорочення витрат машинного часу на її розв'язання, що є найважливішим фактором при відновленні функціональних можливостей частково зруйнованої РІС критичного призначення.

Пропонований метод має властивість повноти і гарантує відшукування всіх існуючих скінченних ланцюгів заданої довжини між будь-якою парою вершин неорієнтованого графа довільної структури. З мінімальними змінами даний метод може бути розповсюджений і на орієнтовані графи.

Література

1. Distributed Information Systems in Distributed System URL: <https://www.geeksforgeeks.org/distributed-information-systems-in-distributed-system> (дата звернення: 19.03.2025).
2. Building a Resilient Disaster Recovery Strategy: Best Practices for Business Continuity URL: <https://www.trigyn.com/insights/building-resilient-disaster-recovery-strategy-best-practices-business-continuity> (дата звернення: 20.03.2025).
3. Bandwidth Management URL: <https://www.geeksforgeeks.org/bandwidth-management-for-distributed-systems> (дата звернення: 24.03.2025).
4. Розподілені інформаційні системи: технології, проектування, забезпечення безпеки URL: <https://kraina.one/education/rozpodileni-informatsijni-sistemi-tehnologiji-proektuvannya-zabezpechennya-bezpeki.html> (дата звернення: 27.03.2025).
5. Kruskal's Minimum Spanning Tree (MST) Algorithm URL: <https://www.geeksforgeeks.org/kruskals-minimum-spanning-tree-algorithm-greedy-algo-2/> (дата звернення: 25.03.2025).
6. The Lee Algorithm Explained: Maze Running and Finding the Shortest Path URL: <https://www.bomberbot.com/algorithms/the-lee-algorithm-explained-maze-running-and-finding-the-shortest-path/> (дата звернення: 28.03.2025).
7. How to find Shortest Paths from Source to all Vertices using Dijkstra's Algorithm URL: <https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/> (дата звернення: 30.03.2025).
8. Bellman-Ford Algorithm URL: <https://www.geeksforgeeks.org/bellman-ford-algorithm-dp-23/> (дата звернення: 30.03.2025).
9. Michail O., Skretas G., Spirakis P. G. Distributed Computation and Reconfiguration in Actively Dynamic Networks. *Distrib. Comput.* 2022. Vol. 35. P. 185–206.
10. Mahdavi M. et al. An Efficient Mathematical Model for Distribution System Reconfiguration Using AMPL. *IEEE Access.* 2021. Vol. 9. P. 79961–79993.
11. Ahrens E. et al. Reasoning about Reconfigurations of Distributed Systems. arXiv:2107.05253. 2022. 53 p.

Савченко А.С., Литвиненко Р.В.

МЕТОД РЕКОНФІГУРАЦІЇ РОЗПОДІЛЕНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ, ЩО ЗАЗНАЛА ЧАСТКОВОГО ФІЗИЧНОГО РУЙНУВАННЯ

У статті розглянуто проблему відновлення функціональності розподілених інформаційних систем (РІС), що зазнали часткового фізичного руйнування, та запропоновано метод їх реконфігурації. В умовах фізичних ушкоджень, спричинених зовнішніми чи внутрішніми факторами, критично важливо забезпечити збереження функціональності таких систем. Втрата зв'язності між елементами може призвести до зниження ефективності обчислювальних процесів, порушення обміну даними та часткової або повної втрати керованості системою.

Запропонований метод спрямований на відновлення зв'язності та адаптацію РІС до нових умов функціонування шляхом реконфігурації її структури. Основна ідея підходу полягає у визначенні альтернативних маршрутів зв'язку між елементами системи для компенсації втрат зв'язку. Метод ґрунтується на моделюванні структури системи у вигляді неорієнтованого графа, де пошкоджені зв'язки є відсутніми ребрами, а

відновлення функціоналу відбувається шляхом побудови множини скінченних простих ланцюгів.

Актуальність дослідження зумовлена зростанням залежності від розподілених обчислювальних систем у критично важливих сферах, зокрема у військовій, енергетичній, транспортній та телекомунікаційній галузях. Ефективне відновлення їх функціональності дозволяє мінімізувати наслідки руйнувань, підвищити їх стійкість та забезпечити безперервність роботи навіть у кризових ситуаціях. Запропонований метод є універсальним і може бути адаптований для широкого спектра розподілених систем, включаючи як комп'ютерні мережі, так і інші складні кіберфізичні системи.

Наукова новизна методу полягає в його здатності забезпечувати повний пошук усіх можливих шляхів відновлення зв'язності при мінімальних обчислювальних витратах. Це досягається шляхом попереднього звуження простору можливих розв'язків, що дозволяє ефективно масштабувати метод для систем із великою кількістю вузлів. Додатковою перевагою є можливість його використання для моделювання сценаріїв відмовостійкості систем та прогнозування наслідків руйнувань, що робить метод корисним для розробки стратегій кіберзахисту та аварійного реагування.

Ключові слова: розподілена інформаційна система; структура; граф; маршрут; ланцюг; алгоритм.

Savchenko A.S., Lytvynenko R.V.

METHOD FOR RECONFIGURING A DISTRIBUTED INFORMATION SYSTEM AFTER PARTIAL PHYSICAL DESTRUCTION

The article addresses the problem of restoring the functionality of distributed information systems (DIS) that have suffered partial physical destruction and proposes a method for their reconfiguration. In cases of physical damage caused by external or internal factors, ensuring the continued operation of such systems is of critical importance. The loss of connectivity between elements can lead to decreased computational efficiency, disruption of data exchange, and partial or complete loss of system control.

The proposed method aims to restore connectivity and adapt the DIS to new operating conditions by reconfiguring its structure. The core idea of the approach is to identify alternative communication routes between system elements to compensate for lost connections. The method is based on modeling the system structure as an undirected graph, where damaged connections are represented as missing edges, and functionality is restored by constructing a set of finite simple chains.

The relevance of this research is driven by the increasing dependence on distributed computing systems in critical sectors such as military, energy, transportation, and telecommunications. Efficient recovery of their functionality minimizes the impact of destruction, enhances resilience, and ensures operational continuity even in crisis situations. The proposed method is universal and can be adapted to a wide range of distributed systems, including both computer networks and other complex cyber-physical systems.

The scientific novelty of the method lies in its ability to perform a comprehensive search for all possible connectivity restoration paths while minimizing computational costs. This is achieved by pre-narrowing the solution space, allowing the method to efficiently scale for systems with a large number of nodes. An additional advantage is its applicability for modeling system fault tolerance scenarios and predicting the consequences of destruction, making it valuable for cybersecurity strategy development and emergency response planning.

Keywords: distributed information system; structure; graph; route; path; algorithm.