

МЕТОД ПОИСКА РАЗРЕЗОВ ГРАФА ДЛЯ ЗАДАЧИ УПРАВЛЕНИЯ ИНЖЕНЕРНОЙ ИНФРАСТРУКТУРОЙ

Вандиловская П. А.¹, Крыгин А. А.², Лукинова О. В.³,
Рощин А. А.⁴

(ФГБУН Институт проблем управления
им. В.А. Трапезникова РАН, Москва)

Целью функционирования инженерных сетей является обеспечение поставок того или иного ресурса потребителю, при этом, в идеальном случае, подача должна быть непрерывной, что напрямую зависит от целостности инфраструктуры сети. Однако различные факторы: атаки злоумышленников, природные катаклизмы, наконец, естественные технологические причины (различные аварии), приводят к отключению некоторых участков сети, что влечет нарушение режима поставки ресурса. Тогда возникает задача поиска наиболее уязвимых (критических), с точки зрения возможного ущерба, участков инженерной сети. Ее решение позволяет принять соответствующие меры по защите сети от влияния негативных факторов и обеспечить максимально бесперебойную подачу ресурсов. Инженерную сеть принято моделировать графовыми структурами, поэтому одним из методов решения данной задачи является нахождение разрезов графа сети. Такие методы существуют, но все они обладают рядом ограничений. В данной работе предлагается новый метод нахождения всех разрезов графа инженерной сети, вообще говоря, произвольной размерности; описывается алгоритм метода, а также его теоретическое обоснование. Концепция метода основана на формировании на каждой итерации особых конструкций графа (мультиразрезов) таким образом, что в результате отработки алгоритма метода осуществляется поиск всех разрезов. Примерами инженерных сетей, где данный метод может быть использован в качестве одного из инструментов принятия рациональных решений при эксплуатации сетевых объектов, являются электросети, сети водоснабжения и канализации, а также сети связи и телекоммуникаций.

Ключевые слова: граф, разрез графа, мультиразрез, инженерная сеть, свободный путь графа.

¹ Полина Александровна Вандиловская, инженер (polinavandi@yandex.ru).

² Андрей Александрович Крыгин, к.т.н. (andreyakr14@gmail.com).

³ Ольга Васильевна Лукинова, д.т.н., доцент (lobars@mail.ru).

⁴ Александр Алексеевич Рощин, к.т.н., доцент (rochinaa@ipu.ru).

1. Введение

Под инженерными инфраструктурами (ИИ) понимается комплекс технических средств, устройств, сооружений, предназначенных для жизнеобеспечения зданий, включающий водо-, электро-, тепло-, газоснабжение, канализацию, слаботочные сети и т.п. Целью функционирования таких инфраструктур является обеспечение бесперебойной подачи того или иного ресурса от источников его производства потребителям, что напрямую зависит от целостности инфраструктуры сети. Однако различные факторы: атаки злоумышленников, природные катаклизмы, наконец, естественные технологические причины (различные аварии), приводят к отключению некоторых участков сети. Тогда возникает задача поиска уязвимых участков инженерной сети, выход из строя которых нарушит режим поставки ресурса.

Для формального описания ИИ будем использовать граф, включающий вершины-источники (производители ресурса), вершины-стоки (потребители ресурса), внутренние вершины (различные технические средства инженерной сети: распределители ресурса, подстанции, накопители и т.д.). Ребрам графа соответствуют протяженные участки сети, например, линии электропередач для электросетей, трубопроводы для тепловых сетей и т.п. При этом граф представляет собой модель сети со всеми возможными резервными линиями, даже если они не задействованы в нормальном режиме функционирования.

Очевидно, что наибольшее количество аварий и повреждений происходит именно на участках сетей, что нарушает режим поставки ресурса потребителям, при этом в самом худшем случае возникает отключение всех потребителей полностью, так как переконфигурировать сеть для возобновления снабжения не представляется возможным, поскольку в модель включаются все резервные линии. Такие ситуации являются критическими и именно они являются целью моделирования в данной задаче. Например, злоумышленник ставит целью отключить некоторый фрагмент железной дороги, для чего ему необходимо обесточить все тяговые подстанции данного фрагмента. Тогда необходимо выявить все линии электросети, которые он может атаковать.

На уровне модели (графа) поврежденные участки сети ассоциируются с разрезами графа, поэтому решение исходной задачи предполагает поиск всех разрезов графа, которые прерывают все возможные пути от источников к потребителям (изолируют источники от потребителей), а также не содержат «избыточные» ребра, т.е. ребра, не лежащие на путях, ведущих от источников к потребителям.

На сегодняшний день существует достаточное количество методов построения разрезов графа [1–16], которые используют в своей основе различные формализмы. В частности, в [3, 10] описаны методы, которые основаны на алгоритмах полного перебора. Такой подход гарантированно определяет необходимые объекты, однако время вычисления растет экспоненциально. В [7, 12, 13, 15] предлагается подход, основанный на генетических алгоритмах, который сокращает время вычислений, но, в отличие от комбинаторных алгоритмов, в силу случайности поиска не дает гарантий, что хотя бы один разрез будет найден. В [2] показан способ нахождения ограниченного количества разрезов: только одно-, двух- и трехэлементных разрезов графа, что существенно сужает область применения описанного алгоритма.

Наиболее известными способами нахождения разрезов графа являются алгоритмы Каргера – Штейна [11] и Штор – Вагнера [14, 16]. Хотя оба алгоритма имеют достаточно приемлемые характеристики эффективности, у них есть свои ограничения. Алгоритм Каргера – Штейна – это рандомизированный алгоритм с достаточно высокой, но не абсолютной вероятностью нахождения разреза графа. Алгоритм Штор – Вагнера – это детерминированный алгоритм, который выполняется за $O(n^3)$ времени. К тому же алгоритм Каргера – Штейна неприменим для решения задач, где имеется источник и приемник, а именно это условие является существенным для поиска уязвимых фрагментов инженерных сетей; также оба этих метода нацелены на нахождение всего лишь одного разреза, причем минимального.

Кроме того, исследования литературы выявили следующую особенность: многие источники предлагают способы поиска разрезов графа, но для решения задач в конкретных прикладных областях, и этот факт диктует те или иные ограничивающие условия на постановку, алгоритм поиска и результат [5, 6]. Например,

в [5] при логическом проектировании дискретных устройств ставится задача поиска максимального разреза взвешенного графа. Алгоритм решения задачи строится с учетом некоторой функции, оценивающей качество разделения (как правило, на основе суммарного минимального/максимального веса ребер разреза). Для решения рассматриваемой в данной статье задачи строить алгоритм с учетом оптимизации по весам ребер просто неактуально.

В данной работе представлен метод нахождения всех таких разрезов графа, моделирующего инженерную инфраструктуру, которые изолируют источники ресурса от потребителей при наступлении аварийной ситуации. Метод основан на конструировании разрезов таким образом, что на каждом шаге генерируется новый набор ребер, который всегда является разрезом. В этом заключается существенное отличие от полного перебора, характеризующегося тем, что найденный набор является разрезом далеко не всегда, за счет чего, собственно, и возникает необходимость в полном переборе, ведущая к росту времени вычислений.

2. Описание метода

Инженерная сеть традиционно моделируется в виде ориентированного графа $G = \{V, E\}$, где V – множество вершин графа, E – множество ребер графа, $|V| = n$, $|E| = m$.

В литературе элементы множества E ориентированного графа называются ребрами, или дугами [18], мы будем пользоваться термином «ребро».

В решаемой задаче вершины графа соответствуют точечным объектам сети (подстанциям, потребителям, электростанциям и т.п.), а ребра – протяженным (линиям электропередач, трубопроводам и т.п.). Ориентация ребра выбирается согласно направлению передачи ресурса.

Обозначим ребро $e_E = (v_i, v_j)$, где $v_i, v_j \in V$ (здесь и далее у символа « e_E » нижний индекс в виде заглавной буквы обозначает принадлежность объекта, в частности, ребра, к тому или иному множеству).

Особенностью данной задачи является тот факт, что множество вершин V описывается тройкой $\{S, T, U\}$, где S – множество

источников, T – множество приемников и U – множество остальных вершин: $V = S \cup T \cup U$. Ребра, инцидентные источникам, ориентированы от вершин-источников, а ребра, инцидентные приемникам, – к вершинам-приемникам. Сделаем также допущение о том, что не существует ребер, соединяющих пары источников или пары потребителей.

Поскольку граф G имеет несколько источников $S = (s_1, s_2, \dots, s_r)$ и стоков (приемников) $T = (t_1, t_2, \dots, t_l)$, построим их фиктивными вершинами S^* и T^* (следует заметить, что введение фиктивных вершин является важным фактом с точки зрения алгоритмической реализации предлагаемого метода, который начинает свою работу именно с фиктивной начальной вершины S^* и заканчивает фиктивной конечной вершиной T^*). Также будем считать, что граф G связан.

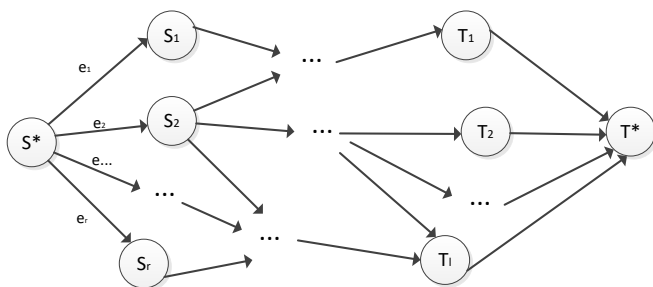


Рис. 1. Исходный граф $G = \{V, E\}$ с фиктивными вершинами S^*, T^*

Определение 1. Под разрезом связного графа с вершинами $V = V_1 \cup V_2$, где V_1 и V_2 – два непересекающихся подмножества V , принято понимать множество P всех тех ребер, которые имеют одну вершину в V_1 , а другую – в V_2 [6].

Оrientированный разрез в ориентированном графе – это разрез $\langle V_1, V_2 \rangle$, все ребра которого или исходят из V_1 , или заходят в V_2 [6].

Специфика данной задачи требует, чтобы определение разреза учитывало следующее условие.

Условие 1. V_1 должно обязательно содержать все источники S , а V_2 – все стоки T . Поэтому задача заключается в нахождении только таких разрезов графа G , что $S \subseteq V_1$ и $T \subseteq V_2$ (в дальнейшем под термином «все разрезы графа» понимается подмножество всех разрезов, обладающих указанным свойством).

Введем понятие «мультиразрез».

Определение 2. Мультиразрез – это подмножество ребер, содержащих один или несколько разрезов.

Для дальнейших рассуждений удобнее пользоваться определениями разреза и мультиразреза, сформулированными через понятие «путь».

Определение 3. Ориентированным маршрутом в ориентированном графе называется конечная последовательность вершин $v_0, v_1, \dots, v_{k-1}, v_k$ такая, что (v_{i-1}, v_i) , $1 \leq i \leq k$, является ребром графа. При этом вершины v_0, v_k называются концевыми [6].

Определение 4. Ориентированный маршрут называется ориентированной цепью, если все его ребра различны [6].

Определение 5. Открытая ориентированная цепь называется ориентированным путем, если различны все ее вершины [6].

Определение 6. Мультиразрез – такое подмножество ребер, что любой путь из S^* в T^* содержит хотя бы одно ребро из этого подмножества.

Определение 7. Разрез – это мультиразрез, не содержащий в себе другого мультиразреза.

Справедливо следующее утверждение, которое позволит в дальнейшем рассуждать в терминах путей в графе.

Утверждение 1. Определения 1 и 7 эквивалентны.

Доказательство. Пусть имеется разрез P , разделяющий вершины на два подмножества V_1 и V_2 , $S^* \in V_1, T^* \in V_2$. Покажем, что 1) любой путь из S^* в T^* содержит хотя бы одно ребро из P , и 2) не существует подмножества $\tilde{P}$, для которого выполняется первый пункт и такого, что $\tilde{P} \subset P$. Предположим, что существует путь из s^* в t^* , не содержащий ребер из P . Так как S^* и T^* принадлежат разным подмножествам, то на этом пути встретится ребро (v_{i-1}, v_i) такое, что $v_{i-1} \in V_1, v_i \in V_2$, но такое ребро должно принадлежать P . Пусть подмножество $\tilde{P}$ существует и ребро (v_{i-1}, v_i) принадлежит P , но не принадлежит $\tilde{P}$. Так как

граф связный, то существуют пути $s^* \rightarrow v_{i-1}$ и $v_i \rightarrow t^*$, и тогда для $\tilde{P}$ не выполняется первый пункт.

Пусть имеется множество ребер P такое, что любой путь из S^* в T^* содержит хотя бы одно ребро из P и любое подмножество $\tilde{P} \subset P$ этим свойством не обладает. Покажем, что P – разрез. Очевидно, что P – это несокращаемое множество ребер, при удалении которого из графа граф $\tilde{G} = G(V, E \setminus P)$ перестает быть связным, так как вершины S^* и T^* относятся к разным компонентам связности. То есть по определению P – разрезающее множество. Если бы компонент связности было больше двух, то была бы возможность построить подмножество $\tilde{P}$, исключив из P ребро, связывающее компоненту, содержащую s^* и компоненту, не содержащую ни S^* , ни T^* . По теореме о связи разрезающего множества с разрезом [6], P – разрез.

Утверждение доказано.

Постановка задачи. Найти все разрезы исходного графа G , удовлетворяющие условию 1.

Будем обозначать полное множество таких разрезов D .

Рассматриваемые далее подмножества ребер будем обозначать Q (с индексом или без), если подмножество является мультиразрезом, и P , если оно является разрезом.

Для решения исходной задачи в работе предлагается метод, позволяющий конструировать множество D . Ниже описана его идея.

Рассмотрим следующую операцию, которую, по аналогии с методом поиска в ширину, назовем *операцией ветвления*. Пусть имеются некоторый разрез P и ребро этого разреза e_0 с конечной вершиной v_0 . Обозначим подмножество ребер, для которых вершина v_0 является начальной N . Определим подмножество ребер Q следующим образом: $Q = N \cup P \setminus e_0$.

Ниже (в теореме п.3) будет показано, что возможны два варианта: полученное после операции ветвления над разрезом подмножество ребер Q обязательно будет являться: а) либо мультиразрезом, содержащим единственный разрез, б) либо собственно разрезом. Несложно показать, что мультиразрез Q является разрезом тогда и только тогда, когда $\forall e_k \in Q$ подмножество ребер $Q \setminus e_k$ не является разрезом. Соответственно (по утверждению 1),

если Q – разрез, то для любого подмножества $Q \setminus e_k$ найдется путь из S^* в T^* , не содержащий ребер этого подмножества.

Ниже представлена *процедура анализа* мультиразреза Q , в ходе которой определяется тот или иной вариант:

1. Из множества ребер E исключаются элементы, входящие в мультиразрез Q .

2. Фиксируется ребро $e_k \in Q$, где $k = 1, 2, \dots, |Q|$; оно возвращается в граф, после чего проверяется достижимость вершины T^* из S^* с учетом данного ребра (проверка на достижимость осуществляется с помощью любого стандартного метода, например, обхода в ширину).

3. Если из S^* в T^* не образуется пути(-ей), то это означает, что найденная конструкция сама по себе не является разрезом, но содержит его внутри. Для его нахождения ребро e_k исключается из Q , так как является избыточным. Назовем такое действие *операцией очистки*.

4. Если из S^* в T^* пути образовались, ребро не является избыточным и принадлежит к разрезу.

5. Пункты 2–4 повторяются для всех ребер мультиразреза Q .

После отработки процедуры анализа и операции очистки окончательный набор ребер устраняет достижимость T^* из S^* и не содержит избыточных ребер, т.е. является разрезом. Таким образом, если в процессе выполнения процедуры анализа хотя бы один раз была выполнена операция очистки, то имеем вариант 1, в противном случае – вариант 2.

Для выявления всех разрезов операции ветвления и очистки проводятся для каждого ребра текущего разреза, т.е. количество новых разрезов (без учета повторов) будет равно количеству ребер в разрезе P .

Для пояснения процедуры анализа рассмотрим пример графа на рис. 2.

Здесь вершиной-источником S^* является вершина под номером 0, а вершиной-приемником T^* – 6. Рассмотрим разрезы $P_1 = \{e_1, e_6\}$ и $P_2 = \{e_3, e_6\}$ и применим к каждому из них операцию ветвления по отношению к ребру e_6 . В результате получим множества $Q_1 = \{e_1, e_7\}$ и $Q_2 = \{e_3, e_7\}$ (в данном случае индексы 1 и 2 служат для понимания того, что перед нами разные наборы

ребер Q , которые были сконструированы от соответствующих разрезов P_1 и P_2). Далее подвергнем их операции очистки, вследствие которой получим разрезы $P_3 = \{e_7\}$, $P_4 = \{e_7\}$, являющиеся дубликатами по отношению друг к другу. Очевидно, что все последующие преобразования P_3 будут совершенно справедливыми и для P_4 , поэтому будет достаточно продолжить работу только с одним из них.

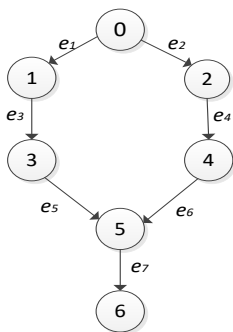


Рис.2. Ситуация дублирования разреза

Отсюда можно сформулировать правило проверки на дубликаты: если новый разрез P_j совпадает с уже содержащимся во множестве искомым разрезом P_i ($P_j = P_i$ ($i \neq j$)), то P_j является дубликатом и далее не рассматривается; если дубликата не нашлось, то P_j добавляется в D .

Алгоритм представленного в работе метода реализуется двумя этапами. Первый этап, начальный, предназначен для обработки мультиразреза, построенного на ребрах, выходящих из фиктивной вершины S^* . Второй – основной этап работает для всех остальных, «внутренних» мультиразрезов.

Начальный этап:

Шаг 1. Из вершины S^* конструируется начальный набор ребер Q^* , который, очевидно, является мультиразрезом. Например, для графа, изображенного на рис. 1, начальным набором будет являться $Q^* = \{e_1, e_2, \dots, e_r\}$.

Шаг 2. Осуществляется процедура анализа, т.е. определяется, что найденный мультиразрез Q^* является разрезом ($Q^* = P_1$) или

содержит таковой ($Q^* \supset P_1$). Во втором случае для Q^* проводится операция очистки.

Шаг 3. Найденный P_1 включается во множество D .

Основной этап:

Шаг 1. Для каждого из ребер очередного добавленного в D разреза P_j с помощью операции ветвления строится новый мультиразрез Q .

Шаг 2. Для сконструированного Q выполняется процедура анализа. Выявляется разрез P_{j+1} .

Шаг 3. Для P_{j+1} осуществляется проверка на дубликат. Если P_{j+1} уникален, то он добавляется во множество D и для него начинают выполняться шаги 1–3.

Рассмотрим работу метода на примере графа, изображенного на рис. 3. Для удобства вершины обозначены номерами.

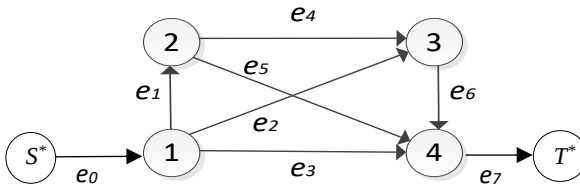


Рис. 3. Граф для демонстрации работы метода

Согласно шагу 1 начального этапа конструируем начальный набор ребер $Q^* = \{e_0\}$. На шаге 2 осуществляется процедура анализа, которая показывает, что набор Q^* является разрезом, т.е. $Q^* = P_1$, который на шаге 3 включается в искомое множество D .

Переходим к основному этапу. Так как P_1 включает одно ребро, то должен быть создан один новый мультиразрез. Применим операцию ветвления для e_0 , получим $Q = \{e_1, e_2, e_3\}$. На шаге 2 совершаем процедуру анализа, которая показывает, что Q является разрезом $P_2 = \{e_1, e_2, e_3\}$. На шаге 3 осуществляется проверка на дубликат, которая разрешает включить P_2 в D .

Так как P_2 включает три ребра, то должно быть создано три новых мультиразреза. Применим операцию ветвления для e_1 , по-

лучим $Q = \{e_4, e_5, e_2, e_3\}$. На шаге 2 совершаем процедуру анализа, которая показывает, что Q является разрезом $P_3 = \{e_4, e_5, e_2, e_3\}$. На шаге 3 осуществляется проверка на дубликат, которая разрешает включить P_3 в D .

Для найденного P_3 сконструируем новые мультиразрезы (к мультиразрезам из P_2 вернемся, когда будут найдены все разрезы в результате операции ветвления ребра e_1). Применим операцию ветвления к ребру e_4 и получим набор ребер $Q = \{e_6, e_5, e_2, e_3\}$. Применим процедуру анализа: ребро e_2 является избыточным, исключаем его и в результате очистки получаем набор ребер $P_4 = \{e_6, e_5, e_3\}$, который становится разрезом и после проверки на дубликат включается в D .

В найденном разрезе P_4 три ребра, поэтому должно быть сконструировано три новых мультиразреза. Применим операцию ветвления к ребру e_6 и получим набор $Q = \{e_7\}$, который является разрезом P_5 и после проверки на дубликат включается в D . Поскольку ребро e_7 входит в приемник T^* , дальнейшая работа с P_5 заканчивается.

Дальнейшая работа по конструированию разрезов проводится аналогичным образом.

Таким образом, результатом работы метода для графа, изображенного на рис. 3, будет следующее множество разрезов: $D = \{P_1 = \{e_0\}, P_2 = \{e_1, e_2, e_3\}, P_3 = \{e_4, e_5, e_2, e_3\}, P_4 = \{e_5, e_6, e_3\}, P_5 = \{e_7\}, P_6 = \{e_1, e_6, e_3\}\}$.

Как правило, в литературе оценка вычислительной сложности алгоритма на графах зависит от количества вершин (n) и ребер (m) графа. Однако для данной задачи следует учитывать характеристики мультиразрезов: зафиксируем некоторое ребро, тогда мультиразрез, содержащий ребро (и, в частности, зафиксированное ребро), будет проходить процедуру анализа $p_1 \cdot p_2 \cdot \dots \cdot p_k$ раз, где k – количество ребер в мультиразрезе, а p_i – количество путей от вершин источников к i -му ребру мультиразреза. Перейдем к усредненным величинам; пусть p – среднее количество путей от вершин источников к ребру, k – среднее количество ребер в мультиразрезе, полученном при ветвлении разреза, r – среднее количество таких мультиразрезов, в которых присутствует одно и то же ребро. Тогда количество операций проверки для одного

ребра можно оценить как $r \cdot p^k$. При этом не удалось найти работ, посвященных оценке величин p, k, r через n и m (а также максимальных значений количества путей, количества ребер в мультиразрезе и количества мультиразрезов, содержащих одно и то же ребро). Таким образом, теоретическую оценку вычислительной сложности провести можно, но она будет неинформативной.

3. Доказательство корректности метода

Вопрос теоретического обоснования корректности предложенного метода заключается в доказательстве следующих фактов:

1. Каждый новый мультиразрез Q , полученный с помощью алгоритма метода, содержит только один разрез, т.е. в Q не содержится других разрезов, не выявляемых процедурой анализа.

2. Множество D является полным.

Обоснование этих фактов следует из приведенной ниже теоремы о свободных путях, а также следствия из нее. Чтобы сформулировать теорему, введем следующее определение.

Определение 8. Пусть заданы некоторый мультиразрез и ребро этого мультиразреза. Пути, проходящие от источников до потребителей и содержащие это ребро, будем называть *свободными* (для заданного мультиразреза и ребра), если они не содержат других ребер мультиразреза.

Интерпретация данного определения заключается в следующем. Очевидно, что у каждого ребра e_P разреза P имеется хотя бы один свободный путь. В противном случае (когда на любом пути, проходящем через e_P , также имеется другое ребро разреза P) подмножество $P \setminus e_P$ является мультиразрезом, т.е. P – не разрез. Если мультиразрез Q не является разрезом, то любое ребро e_Q , имеющее в Q свободный путь, будет входить в любой разрез $P \subset Q$, так как в противном случае P не будет являться разрезом.

Теорема (о свободных путях). Пусть R_{old} – разрез, над которым проводится ветвление; e_0 – ребро, из которого производилось ветвление; v_0 – конечная вершина, инцидентная e_0 , из которой осуществляется ветвление; N – множество добавляемых

ребер; $R_{new} = N \cup R_{old} \setminus e_0$ – множество ребер, получившееся в результате операции ветвления; $\tilde{R} \subset R_{new}$ – произвольный разрез, содержащийся в множестве R_{new} .

Тогда набор ребер R_{new} , построенный из разреза R_{old} с помощью операции ветвления над ребром e_0 , является мультиразрезом и содержит в себе единственный разрез.

Доказательство. Логика доказательства теоремы основана на том, что будут обоснованы следующие тезисы:

1. Любая конструкция, полученная с помощью операции ветвления, является мультиразрезом Q .
2. Мультиразрез Q содержит хотя бы один разрез P .
3. Разрез $P \in Q$ является единственным.

Доказательство *первого тезиса* заключается в том, что любой путь, проходящий через e_0 , также проходит через одно из ребер N . Любой путь, не проходящий через e_0 , также проходит через одно из ребер $R_{old} \setminus e_0$, поэтому R_{new} – мультиразрез.

Второй тезис, т.е. тот факт, что любой мультиразрез Q содержит в себе хотя бы один разрез, следует из процедуры анализа (см. п. 2).

Доказательство единственности существования.

Логика доказательства единственности, т.е. того, что в мультиразрезе R_{new} содержится единственный разрез, следующая. Будет показано, что R_{new} однозначно разбивается на непересекающиеся подмножества ребер. И для каждого подмножества будет показано, что либо все его ребра должны принадлежать любому разрезу, содержащемуся в R_{new} , либо ни одно ребро подмножества не принадлежит ни одному разрезу. Из такой однозначности будет следовать, что разрез единственный.

Ниже будет показано, что $N \cap R_{old} \setminus e_0 = \emptyset$, так как R_{old} – разрез. Поэтому любое ребро из R_{new} принадлежит либо N , либо $R_{old} \setminus e_0$. Для разреза R_{old} у ребра e_0 есть свободный путь. По построению этот путь проходит через ребро из N (обозначим его e_{n1}). Тогда в мультиразрезе R_{new} у ребра $e_{n1} \in N$ есть свободный путь. Поэтому в любом разрезе $\tilde{R} \subset R_{new}$, присутствует e_{n1} . В общем случае в мультиразрезе R_{old} у ребра e_0 есть несколько свободных путей; обозначим $E_N^+ \subset N$ – подмножество ребер N , через

которое проходит хотя бы один свободный путь для мультиразреза R_{old} и ребра e_0 . Как было показано, все подмножество E_N^+ входит в любой разрез $\tilde{R} \subset R_{new}$ и это подмножество не пусто. Обозначим $E_N^- = N \setminus E_N^+$. Также из существования свободного пути для разреза R_{old} у ребра e_0 следует, что существует путь от вершины источника до e_0 , не содержащий ребер из R_{old} .

Разделим множество ребер $R_{old} \setminus e_0$ на два подмножества E_{old}^+ и E_{old}^- . Ребро $e_{old} \in E_{old}^+$, если в разрезе R_{old} у e_{old} есть свободный путь, не проходящий через вершину v_0 , в противном случае $e_{old} \in E_{old}^-$. Для $e_{old} \in E_{old}^+$ указанный свободный путь также будет свободным и в мультиразрезе R_{new} . И, следовательно, $E_{old}^+ \in \tilde{R}$ для любого разреза $\tilde{R} \subset R_{new}$.

Будем называть ребро $e_{old} \in E_{old}^-$ *лежащим выше* v_0 , если для мультиразреза R_{old} и ребра e_{old} существует свободный путь следующей последовательности вершин и ребер: $s \in S \rightarrow \dots \rightarrow v_0 \dots \rightarrow e_{old} \rightarrow \dots \rightarrow t \in T$, в противном случае ребро $e_{old} \in E_{old}^-$ *лежит ниже* v_0 . Покажем, что все ребра $e_{old} \in E_{old}^-$ лежат ниже v_0 . Для этого рассмотрим рис. 4.

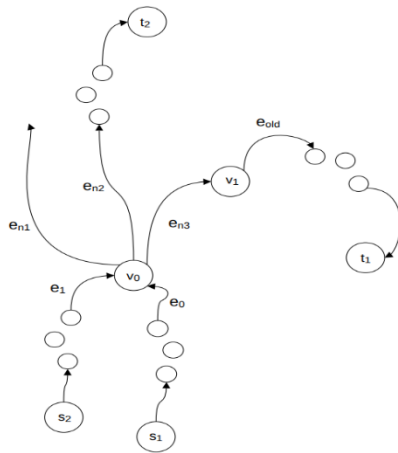


Рис. 4. Пример графа №1

В разрез R_{old} входят ребра e_{old} и e_0 , при этом у каждого из них есть свободные пути, проходящие через v_0 . Свободный путь

для e_{old} состоит из $(s_2 \rightarrow \dots e_1 \rightarrow v_0 \rightarrow e_{n3} \rightarrow v_1 \rightarrow e_{old} \rightarrow \dots \rightarrow t_1)$, свободный путь для e_0 состоит из $(s_1 \rightarrow \dots e_0 \rightarrow v_0 \rightarrow e_{n2} \rightarrow \dots \rightarrow t_2)$. На рисунке показана ситуация, когда e_{old} лежит выше v_0 и при этом есть наличие свободных путей. Тогда путь $(s_2 \rightarrow \dots e_1 \rightarrow v_0 \rightarrow e_{n2} \rightarrow \dots \rightarrow t_2)$ вообще не проходит через ребра разреза, то есть R_{old} – не разрез и не мультиразрез.

Вернемся к множеству E_N^- – подмножеству N , через которое не проходят свободные пути для разреза R_{old} и ребра e_0 . Выше было показано, что существует путь от вершины источника до e_0 , не содержащий ребер из R_{old} , обозначим его w_0 . Рассмотрим путь, состоящий из w_0 , e_0 , $e_n^- \in E_N^-$ и дальнейшего отрезка пути w_1 до вершины потребителя. Покажем, что на пути w_1 лежит ребро из E_{old}^+ . Для разреза R_{old} и ребра e_0 этот путь не свободен, поэтому на пути w_1 лежит ребро (или несколько ребер) $e_* \in R_{old}$. Предположим (от противного), что $e_* \in E_{old}^-$; если на этом пути лежит несколько ребер из E_{old}^- , то будем рассматривать последнее ребро e_* , после которого на пути w_1 нет ребер из R_{old} (рис. 5).

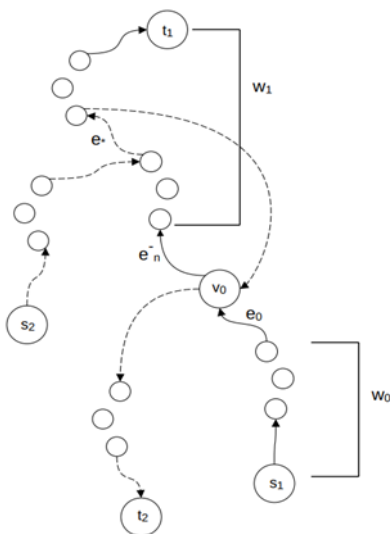


Рис. 5. Пример графа №2

На рисунке сплошной линией показан рассматриваемый путь. У ребра e_* в разрезе R_{old} есть свободный путь и, как было показано, e_* лежит ниже v_0 ; этот путь изображен пунктирной линией. Тогда у e_* есть свободный путь, не содержащий v_0 : $s_2 \rightarrow \dots e_* \rightarrow \dots \rightarrow t_1$, поэтому $e_* \in E_{old}^+$ – противоречие.

Соответственно, для мультиразреза R_{new} и любого ребра $e_N^- \in E_N^-$ любой путь, проходящий через e_N^- , также проходит через ребро $e_* \in E_{old}^+$. Все подмножество E_{old}^+ включено в каждый разрез $\tilde{R} \subset R_{new}$ и поэтому e_N^- (и все подмножество E_N^-) не включено в $\tilde{R}$, так как у него не будет свободных путей в $\tilde{R}$.

Аналогично и для любого ребра $e_{old} \in E_{old}^-$: все свободные в R_{old} пути, проходящие через e_{old} , подходят снизу к v_0 и далее проходят либо через ребро из E_N^- и ребро из E_{old}^+ , либо через ребро из E_N^+ , т.е. E_{old}^- не включено ни в один разрез $\tilde{R} \subset R_{new}$.

Таким образом, мультиразрез R_{new} однозначно разбивается на 4 подмножества: E_{old}^- , E_{old}^+ , E_N^- , E_N^+ ; подмножества E_{old}^- , E_N^- не входят ни в один разрез $\tilde{R} \subset R_{new}$, подмножества E_{old}^+ , E_N^+ целиком входят в каждый $\tilde{R} \subset R_{new}$, откуда следует единственность разреза $\tilde{R} \subset R_{new}$.

Теорема доказана.

Следствие. Множество D , формируемое методом, содержит полное множество разрезов.

Доказательство. Процедура анализа и проверки на дубликаты позволяют сделать вывод о том, что каждый элемент множества D является уникальным разрезом. Покажем от противного, что D является полным множеством, предположив, что существует разрез $P \notin D$. Такая ситуация может возникнуть в двух случаях:

1. Когда операция проверки выполнялась над мультиразрезом, содержащим несколько разрезов, среди которых присутствует P .

2. Когда P не содержался ни в одном из мультиразрезов, над которыми проводилась операция проверки.

По теореме о свободных путях первого случая быть не может, так как операция ветвления всегда выполнялась над разрезами. Покажем, что и второй случай невозможен.

Так как P – разрез, то у каждого его ребра есть хотя бы один свободный путь. Будем рассматривать лес, состоящий из отрезков свободных путей от вершин источников до ребер P . Если свободные пути двух ребер пересекаются, то отбросим один из участков от вершины источника до первой (считая от мультиразреза) вершины пересечения (рис. 6).

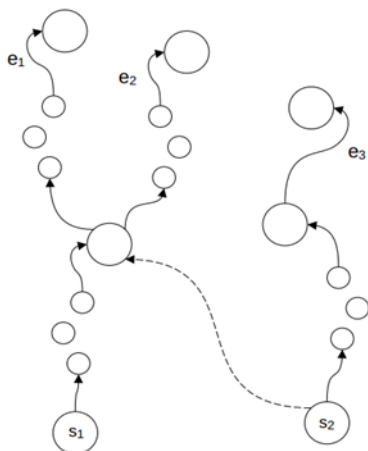


Рис. 6. Объединение свободных путей

На рис. 6 $e_1, e_2, e_3 \in P$, а пунктирной линией показан отброшенный участок.

Ввиду полноты перебора различных наборов операций {ветвления – проверки} для достижения противоречия достаточно показать, что существует последовательность этих операций, приводящая к разрезу P .

Эту последовательность будем строить следующим образом: проводить ветвление только тех ребер, которые принадлежат построенному лесу. Если после операций ветвления и проверки ребро разреза не будет принадлежать лесу, то дальнейшее ветвление из него не проводится. Дальнейшее ветвление также не проводится и из ребра, принадлежащего P . Поэтому (так как P – разрез) указанная последовательность операций приведет к P ,

и противоречие не возникнет только в том случае, когда в ходе операции проверки будет удалено ребро из леса.

Рассмотрим любой мультиразрез после операции ветвления разреза, полученный в описанной последовательности. Так как лес состоит из свободных путей, которые должны быть перекрыты мультиразрезом и по построению на этих путях есть только одно ребро мультиразреза, то никакое ребро мультиразреза, принадлежащее лесу, не может быть удалено в ходе операции проверки, так как оно принадлежит свободному пути.

Следствие доказано.

4. Заключение

В статье представлен метод поиска всех разрезов ориентированного графа с учетом специфики предметной области. Концепция метода основана на алгоритме поиска разрезов таким образом, что на каждом шаге генерируется новый набор ребер (мультиразрез), который всегда либо является разрезом, либо содержит единственный разрез. Описан алгоритм метода, сформулированы правила, положенные в его основу, представлена теорема, обосновывающая положения метода, а также следствие, доказывающее тот факт, что найденное множество разрезов полное.

Выявление таких компонент графа позволяет принимать более рациональные решения при управлении инженерными инфраструктурами, позволяющие свети к минимуму количество отключений потребителей от ресурсов при наступлении деструктивных факторов.

Вопросы вычислительной сложности алгоритма метода, в силу невозможности показать теоретические оценки, авторы планируют исследовать экспериментально на различных вычислителях и представить в будущих работах. Хотя следует заметить, что исходный граф, а также любой найденный разрез состоят из конечного числа ребер, кроме того, метод включает проверку на дубликаты, благодаря которой исключаются циклические операции. Эти факты говорят о том, что предлагаемый метод гарантирует обход графа за конечное время.

Литература

1. ГРИШКЕВИЧ А.А., РІАТЕК Л., БУРМУТАЕВ А. *Нахождение одно-, двух- и трехэлементных разрезов графа* // Вестник ЮрГУ, серия «Математическое моделирование и программирование». – 2008. – №15(115). – Вып. 1. – С. 12–22.
2. ДОРРИ М.Х., РОЩИН А.А., СЕРЕДА Л.А. *Применение программного комплекса РДС для расчетов и визуализации последствий выхода из строя инженерных сооружений* // Автоматизация в промышленности. – 2017. – № 11. – С. 11–14.
3. РЯБИНИН И.А. *Надежность и безопасность структурно-сложных систем*. – СПб.: Изд-во СПб. гос. ун-та, 2007. – 276 с.
4. ПОТТОСИН Ю.В., ПОТТОСИНА С.А. *Поиск разреза графа в решении некоторых задач логического проектирования* // Vesci Nacyânal'naj akademiï navuk Belarusi. Seryâ fizika-matematychnyh navuk. – 2016. – №3. – С. 111–118.
5. ПИРОВА А.Ю. *Параллельные алгоритмы разделения графов: учебное пособие*. – Нижний Новгород: Нижегородский госуниверситет, 2019. – 20 с.
6. СВАМИ М., ТХУЛАСИРАМАН К. *Графы, сети и алгоритмы*. – М.: Мир, 1984. – 454 с.
7. AGUDELO L., MUNOZ N., LÓPEZ-LEZAMA J.M. *Vulnerability assessment of power systems to intentional attacks using a specialized genetic algorithm* // Dyna (Medellin, Colombia). – 2015. – Vol. 82, Iss. 192. – P. 78–84.
8. CAGNO E., GRANDE O., TRUCCO P. *Towards an integrated vulnerability and resilience analysis for underground infrastructures* // Reliability Engineering & System Safety. – 2011. – Vol. 96, Iss. 1. – P. 139–148.
9. GREBENYUK G.G., NIKISHOV S.M. *Blocking of Energy and Resource Supply of Target Objects in Network Infrastructures* // Automation and Remote Control. – 2018. – Vol. 79(3). – P. 535–544.
10. HAENNI R. *Generating Diagnoses from Conflict Sets* // Proc. of the 11th Int. Conf. FLAIRS. – 1998 – URL: www.aaai.org/Papers/FLAIRS/1998/FLAIRS98-081.pdf (дата обращения: 23.03.2023).

11. KARGER D.R. *Global Min-cuts in RNC, and Other Ramifications of a Simple Min-Cut Algorithm* // SODA: Journal. – 1993. – Vol. 93. – P. 21–30.
12. KARIMI E., MADANI S.M., EBRAHIMI A. *Power transmission system vulnerability assessment using genetic algorithm* // Intelligent Systems in Electrical Engineering Fall. – 2012. – Vol. 3, No. 3. – P. 1–10.
13. KIM T., WRIGHT S.J., BIENSTOCK D. et al. *Vulnerability Analysis of Power Systems* // IEEE Trans. on Network Science and Engineering. – 2016. – Vol. 3, Iss. 3. – P. 132–146.
14. MAY R.P. *Genetic Algorithms for Agent-Based Infrastructure Interdependency Modeling and Analysis*. – URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.455.354&rep=rep1&type=pdf> (дата обращения: 23.03.2023).
15. ROSELYNA J.P., DEVARAJB D., DASH S.S. *Multi-Objective Genetic Algorithm for voltage stability enhancement using re-scheduling and FACTS devices* // Ain Shams Engineering Journal. – 2014. – Vol. 5, Iss. 3. – P. 789–801.
16. STOER M., WAGNER F. *A simple min-cut algorithm* // Journal of the ACM. – 1997. – Vol. 44(4). – P. 585–591.
17. VALENCIA V.V., MAJ P.E. *Network Interdependency Modeling for Risk Assessment on Built Infrastructure Systems*. – 2013. – URL: <https://pdfs.semanticscholar.org/95ba/f36ae65157638a83f82084c39884b3f0fcb2.pdf?ga=2.74081425.2003714931.1570034586-1895369083.1570034586> (дата обращения: 23.03.2023).

METHOD FOR FINDING CUTS FOR ENGINEERING INFRASTRUCTURE MANAGEMENT TASKS

Polina Vandilovskaya, V.A. Trapeznikov Institute of Control Sciences of RAS, Moscow, Engineer (polinavandi@yandex.ru).

Andrey Krygin, V.A. Trapeznikov Institute of Control Sciences of RAS, Moscow, Cand.Sc., Senior Researcher (andreyakr14@gmail.com).

Olga Lukinova, V.A. Trapeznikov Institute of Control Sciences of RAS, Moscow, Doctor of Engineering Sciences, Head Scientist Researcher (lobars@mail.ru)

Alexander Roschin, V.A. Trapeznikov Institute of Control Sciences of RAS, Moscow, Cand.Sc., Senior Researcher (rochinaa@ipu.ru).

Abstract: The goal of the functioning of engineering networks is to ensure the delivery of a particular resource to a consumer, ideally with uninterrupted supply which directly depends on the integrity of the network infrastructure. However, various factors such as attacks by malicious actors, natural disasters, and technological accidents can lead to the disconnection of certain parts of the network, resulting in the disruption of resource delivery. This necessitates the task of identifying the most vulnerable parts of the engineering network in terms of potential damage, which allows appropriate measures to be taken to protect the network from negative factors and ensure maximum uninterrupted resource delivery. Engineering networks are commonly modeled as graph structures, therefore one method of solving this task is to find the cuts of the network graph. While such methods exist, they all have certain limitations. This study proposes a new method for finding all cuts of a directed graph of arbitrary dimension, which eliminates these limitations, describes the algorithm for this method, and provides theoretical justification. The concept of the method is based on constructing graph cuts (multicuts) in such a way that all cuts are identified, and can be done in a reasonable amount of time. Examples of engineering networks where this method can be used as a tool for making rational decisions in operating network objects include power grids, transportation networks, water supply and sewerage networks, as well as communication and telecommunications networks.

Keywords: graph, cut, multicuts, engineering network, free path in graph.

УДК 519.178, 658.26

ББК 22.176

DOI: 10.25728/ubs.2024.111.9

*Статья представлена к публикации
членом редакционной коллегии Н.Н. Непейводой.*

Поступила в редакцию 04.12.2023.

Опубликована 30.09.2024.