

Робастное, адаптивное и сетевое управление

© 2021 г. Г.Г. ГРЕБЕНЮК, д-р техн. наук (gggrebenuk@gmail.com),
А.А. КРЫГИН, канд. техн. наук (andreyagr@yandex.ru)
(Институт проблем управления им. В.А. Трапезникова РАН, Москва)

МЕТОДЫ ПОИСКА КОНФИГУРАЦИЙ РАСПРЕДЕЛИТЕЛЬНЫХ СЕТЕЙ

Рассматривается задача нахождения полного множества допустимых конфигураций распределительной сети. При ее решении используется аппарат теории графов для нахождения предельных графов. Предложен новый, по сравнению с известными в литературе, метод нахождения полного множества предельных графов, доказаны ряд свойств этого метода и его корректность. На качественном уровне выполнено сравнение эффективности различных методов и показано, что предложенный метод отличается существенно более высокой скоростью вычислений.

Ключевые слова: распределительные сети, поиск конфигурации сети, графовые модели сетей, предельные графы.

DOI: 10.31857/S0005231021050032

1. Введение

Поиск конфигураций физических сетей распределения энергии и ресурсов от их источников к потребителям осуществляется при решении различных задач, среди которых выделим две: структурная оптимизация режимов распределительных сетей (в частности, электрических) и определение степени уязвимости потребителей из-за прерывания их снабжения энергией и ресурсами при негативных воздействиях на сеть поставки.

К задаче структурной оптимизации диспетчерские службы обращаются в режимах балансировки нагрузки, минимизации потери электроэнергии, анализа устойчивости, противоаварийного управления и в других случаях [1–9]. Оптимизация сети выполняется на взвешенном графе, вершины которого соответствуют узлам коммутации, а ребра — линиям связи. Решение задачи заключается в поиске и выборе оптимальной конфигурации посредством подключения узлами коммутации тех или иных связей к сети передачи и распределения энергии и ресурсов. Для рассматриваемых сетей поиск оптимальной конфигурации представляется как многоэтапная и многокритериальная задача:

- Поиск возможных конфигураций;
- Выбор оптимальной конфигурации.

Далее рассмотрен первый этап — поиск возможных конфигураций посредством подключения узлами коммутации тех или иных связей к сети распределения энергии и ресурсов.

Вторая задача, также рассматриваемая далее, это определение степени уязвимости потребителей из-за прерывания снабжения энергией и ресурсов при негативных воздействиях на сеть. Решение этой задачи рассматривается в разрезе нахождения вариантов присоединения потребителей к распределительной сети и представляет интерес при выработке мер противодействия негативным воздействиям на элементы сетевой инфраструктуры.

Значительная избыточность, вводимая в распределительные сети для повышения надежности, создает определенные трудности в поиске конфигураций сети. На их разрешение направлен метод построения предельных графов, изложенный в [8]. Также в [8] доказаны корректность метода и ряд свойств предельных графов, в [9] показано применение метода для структурной оптимизации электрических сетей.

В данной статье указаны недостатки метода [8], расширено понятие предельных графов, предложен более эффективный метод их построения, доказана его корректность и проведена качественная сравнительная оценка двух методов.

2. Графы распределительных сетей

Рассматриваются распределительные сети, которые обладают следующим свойством: каждый потребитель снабжается от одного источника. Как правило, данное свойство выполняется для большинства распределительных сетей. Такие сети характеризуются разомкнутой структурой электроснабжения, формируемой узлами коммутации. Пример графа распределительной некоммутируемой сети приведен на рис. 1,а, вершины $S1$ и $P1$ соответствуют источнику и потребителю. Обозначим множество всех вершин графа, исключая вершины источники и вершины потребители через U , через $|U|$ обозначим мощность множества U .

В задаче структурной оптимизации вершины $U1 - U3$ соответствуют элементам сети, таким как подстанции, тепловые пункты и другие сооружения, в составе которых находятся узлы коммутации. Обозначим множество коммутаторов через C и расположим эти коммутаторы на тех связях, которыми они управляют. Граф, соответствующей коммутируемой сети, изображен на рис. 1,б.

Рассмотрим следующие графовые модели распределительных сетей.

Некоммутируемую распределительную сеть представим в виде графа \mathfrak{J} со свойствами:

— множество вершин этого графа V состоит из трех непересекающихся подмножеств $V = S \cup P \cup U$. Множество S будем называть множеством вершин источников, множество P — множеством вершин потребителей, U — определено выше;

— для каждой вершины $p \in P$ существует путь $s \rightarrow p$ хотя бы из одной вершины $s \in S$.

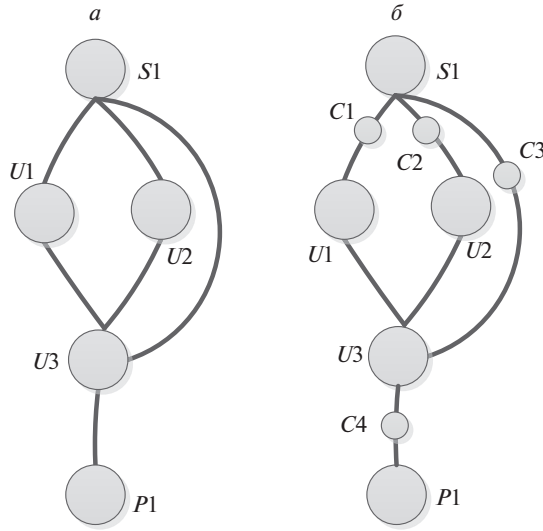


Рис. 1. Примеры некоммутируемого (а) и коммутируемого (б) графов распределительных сетей.

Коммутируемую распределительную сеть будем представлять в виде графа \mathfrak{J} со свойствами:

— множество вершин \mathfrak{J} описывается четверкой $V = S \cup P \cup C \cup U$, где множество C составляют вершины коммутаторы, а множество U — все остальные вершины, исключая вершины множеств S , P и C ;

— маршруты передачи определяются состоянием вершин коммутации: $C = C^+ \cup C^-$, где C^+ — подмножество вершин коммутаторов в состоянии “открыто”, C^- — подмножество вершин коммутаторов в состоянии “закрыто”;

— свойство существования пути аналогично графам некоммутируемых сетей;

— степень каждой вершины $c \in C$ равна двум; степень каждой вершины $p \in P$ равна единице;

— каждое ребро, выходящее из вершины источника, соединено с вершиной коммутатором.

При решении описанных выше задач удобно использовать общие (расширенные по сравнению [8]) определения графа конфигурации и предельного графа.

Обозначим через Z подмножество вершин графа, $Z \subseteq U \cup C$. Для коммутируемых сетей $Z = C$, для некоммутируемых сетей $Z = U$. Для любого подмножества вершин W , $W \subseteq Z$ определим следующую операцию:

- удалим из графа подмножество W ;
- после этого также удалим из графа все вершины, не лежащие ни на одном пути $s \rightarrow p$, где $s \in S$, $p \in P$. Удаляемой вершиной может быть и вершина источник $\hat{s} \in S$, если для любого $p \in P$ нет ни одного пути $\hat{s} \rightarrow p$.

Назовем описанную операцию “построением W подграфа”.

Получившийся в результате описанной операции W подграф будем называть графом конфигурации, если для каждой вершины $p \in P$ существует путь $s \rightarrow p$ хотя бы из одной вершины $s \in S$.

Граф конфигурации будем называть предельным графом, если при удалении из него любой вершины $z \in Z \setminus W$ найдется вершина $p \in P$, для которой не будет ни одного пути $s \rightarrow p$, где $s \in S$.

Таким образом, нахождение всех графов конфигурации и всех предельных графов сводится к нахождению соответствующих им подмножеств W или подмножеств $Z \setminus W$.

3. Описание метода поиска конфигураций в графах распределительных сетей

Так как граф конфигурации, согласно определению, содержит пути от источников к потребителям, то в комбинациях этих путей могут содержаться циклы и присутствовать пути с избыточным количеством вершин множества U , при удалении которых сохраняется достижимость вершин множества P из S .

Далее предлагается метод построения графов конфигурации сети, которые после операции проверки на поглощение (также описанной ниже) преобразуются в полное множество предельных графов.

Предлагаемый метод формирования графов конфигурации (далее компонентный метод) использует поиск в ширину (*BFS*) при движении от вершин из множества P против направления потока к вершинам из множества S . В процессе движения формируются пути, образующие компоненту. При пересечении одного пути с другим происходит слияние путей и формируются деревья. Как и в *BFS*, у каждого дерева есть активная вершина, из которой происходит движение (развитие), но в отличие от *BFS* используется двойная очередь: очередь компонент и очередь активных вершин компоненты.

Опишем метод. Предварительно введем ряд определений.

Компонента — структура, состоящая из деревьев и очереди активных вершин. Очередь активных вершин соответствует вершинам каждого дерева, из которых происходит движение, при этом у активной вершины хранится ссылка на дерево, к которому она принадлежит.

Выходной массив компонент — конечный результат, представленный в виде массива.

Операция клонирования заданной компоненты — создание новой компоненты и копирование в нее заданной компоненты, т.е. всех деревьев и очереди активных вершин.

Начальная инициализация — формирование начальной компоненты, состоящей из $|P|$ вершин потребителей и очереди активных вершин, в которую заносятся все вершины потребителей, а также создание очереди компонент, в которую заносится начальная компонента, и создание выходного массива компонент (нулевого для начальной инициализации).

Алгоритм.

Шаг 0. Начальная инициализация.

Шаг 1. Из очереди компонент отбирается компонента, которую обозначим через I .

Шаг 2. Из очереди активных вершин компоненты I отбирается вершина i , $i \in K$, где K — дерево:

2.1. Для i определяются ее вершины потомки;

2.2. Для каждой вершины потомка j выполняется клонирование компоненты I с добавлением к дереву K вершины j . После клонирования компонента I удаляется;

2.3. Обработка клонированной компоненты:

2.3.1. Клонированная компонента удаляется, если дерево K содержало вершину j еще до его клонирования в п. 2.2. Переход к шагу 3;

2.3.2. В очередь активных вершин клонированной компоненты добавляется вершина j , если она не является вершиной источником и не является вершиной другого дерева;

2.3.3. Если вершина j является вершиной другого дерева M , то все дерево K присоединяется к дереву M .

2.4. Если в клонированной компоненте нет активных вершин, то она помещается в выходной массив, если активные вершины есть, то она помещается в очередь компонент.

Шаг 3. Если очередь активных вершин компоненты I не пуста, то переход к шагу 2.

Шаг 4. Если очередь компонент не пуста, то переход к шагу 1.

Конец.

4. Свойства компонентного метода

Перечислим необходимые для дальнейших доказательств результаты публикации [8].

В [8] приведен алгоритм получения полного множества предельных графов, использующий построение комбинаций всех путей от вершин источников до вершин потребителей. Этот алгоритм включал следующие этапы:

1. Формирование списков всех возможных путей до каждой вершины множества P от вершин множества S . Общее число списков равно $|P|$.

2. Составление всех комбинаций путей из списков. Комбинация представляет собой объединение вершин из $|P|$ путей, l -й путь выбирается из l -го списка. Каждой j -й комбинации соответствуют граф конфигурации τ_j и подмножества C_j^+ и C_j^- , C_j^+ — вершины коммутаторы, принадлежащие τ_j , C_j^- — остальные вершины коммутаторы.

3. Построение предельных графов. Выполняется последовательное сравнение между собой комбинаций и удаление тех из них, в которые осуществляется вложение вершин множества C^+ . Так, если для m -й и k -й комбинаций выполняется условие $C_k^+ \subseteq C_m^+$, то удаляется m -я комбинация. Графы, соответствующие оставшимся комбинациям, являются предельными графами τ_j^{lim} . Будем называть такую операцию “проверкой на поглощение”, а весь алгоритм — “методом простых комбинаций”.

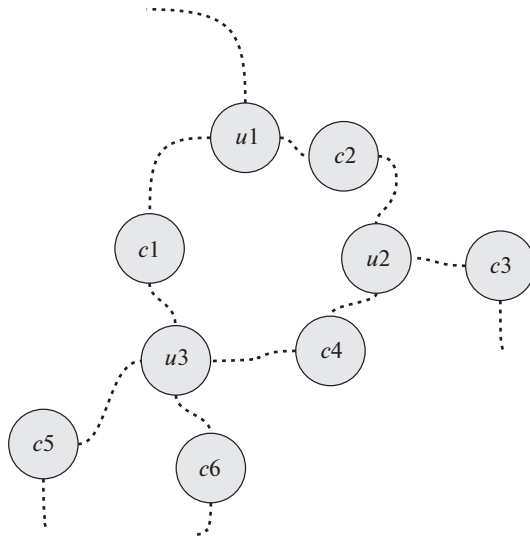


Рис. 2. Фрагмент исходного графа.

В [8] были доказаны следующие утверждения:

1. Корректность алгоритма построения комбинаций.

2. Массив, состоящий из графов конфигурации и содержащий полное множество предельных графов, после выполнения проверки на поглощение будет содержать только полное множество предельных графов.

Для каждой компоненты I выходного массива построим W подграф, выбрав $W = Z \setminus I$ и соответственно для коммутлируемых сетей — $C^+ = I \cap Z$. Справедливы три утверждения:

Утверждение 1. Каждый W подграф является графом конфигурации, а массив W подграфов содержит полное множество предельных графов.

Утверждение 2. При выполнении шага 1 и шага 2 компонентного метода порядок выбора компоненты и активной вершины может быть произвольным.

Утверждение 3. Если в исходном графе каждая вершина ветвления (вершина, степень которой больше двух) окружена вершинами коммутаторами (или в случае некоммутируемых сетей — вершинами степени 2, рис. 2), то массив графов будет содержать только полное множество предельных графов без других графов конфигурации.

Доказательства утверждений 1–3 приведены в Приложении.

5. Качественное сравнение вычислительной сложности компонентного метода и метода простых комбинаций

Результаты моделирования показали, что компонентный метод существенно превосходит по скорости метод простых комбинаций. При этом во многих случаях наиболее быстро работал следующий метод:

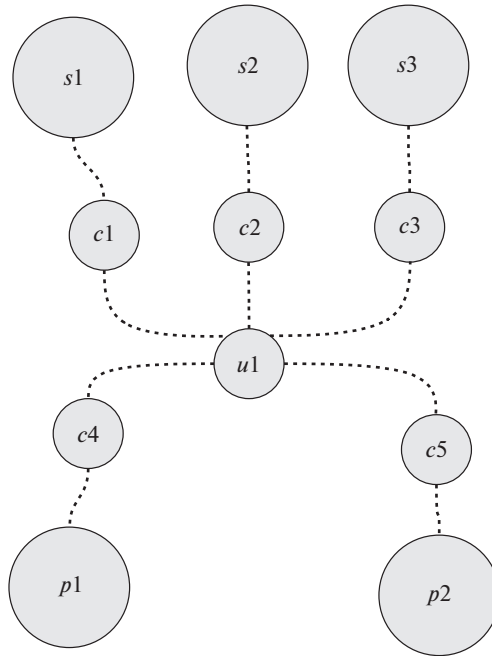


Рис. 3. Пример исходного графа.

— сначала выполнялось построение компонент от одной выбранной вершины потребителя. В этом случае каждая компонента состояла из пути от выбранной вершины потребителя до вершины источника;

— после этого продолжалось построение компонент от остальных вершин потребителей.

Корректность этого метода вытекает из утверждения 2.

Количественная оценка вычислительной сложности обоих методов выходит за рамки данной статьи. Укажем здесь на качественное преимущество компонентного метода.

Основной недостаток метода простых комбинаций заключается в том, что количество предельных графов составляет малую долю от количества всех комбинаций и операция проверки на поглощение требует значительных вычислительных ресурсов. Причина такого соотношения состоит в вершинах ветвления множества U . Рассмотрим это на примере, приведенном на рис. 3.

Очевидно, что у этого графа будет 3 предельных графа, в каждом из которых имеется единственная вершина источник. На 1-м этапе метода простых комбинаций определяется по 3 пути для каждой вершины потребителя. Множество всех возможных комбинаций этих путей (2-й этап) состоит из 27 комбинаций (часть из которых будут одинаковыми). Появление комбинации (на 2-м этапе), не соответствующей предельному графу, возникает всякий раз, когда объединяются два пути, проходящие через вершину ветвления u_1 , у которых фрагменты от вершины источника до вершины ветвления различны. Проведем следующую упрощенную оценку. Пусть d — степень некоторой

вершины ветвления u , P — множество вершин потребителей, у которых существуют пути из вершин источников, проходящие через u , $p = |P|$. Пусть для каждой вершины $r \in P$ существует единственный путь $u - r$. Для приведенного примера $d = 5$, $p = 2$. Тогда из всех $(d - p)^p$ возможных комбинаций на этом фрагменте только $(d - p)$ могут соответствовать предельным графам. Общее количество комбинаций, не соответствующее предельным графам, увеличится как минимум в $[(d - p)^p - (d - p)]$ раз. Аналогичная ситуация будет происходить для каждой вершины ветвления. Поэтому доля комбинаций, соответствующая предельным графам, так мала в общем количестве всех комбинаций. Компонентный метод лишен этого недостатка из-за операции присоединения деревьев (п. 2.3.3).

6. Заключение

Предложен компонентный метод поиска конфигураций распределительных сетей, который может применяться в задачах повышения безопасности, управления режимами сетей и др.

Доказана корректность метода, а также предложены его модификации и дополнительные условия на распределительные сети, при которых скорость метода существенно возрастает.

На качественном уровне показано, что компонентный метод позволяет построить варианты конфигураций с меньшим числом операций, чем метод простых комбинаций путей от вершин источников до вершин потребителей [8].

Доказательство утверждения 1. Рассмотрим компоненту I выходного массива. Она является лесом, так как связность каждого дерева обусловлена построением, а отсутствие циклов гарантируется процедурой клонирования и проверкой на циклы (п. 2.3.1). В соответствии с начальной инициализацией (шаг 0) I содержит все вершины потребители, а каждое дерево в компоненте содержит вершину источник, так как только в этом случае компонента помещается в выходной массив. Следовательно, I и построенный по ней W подграф являются графами конфигурации. Очевидно, что для каждого предельного графа можно явно указать последовательность выбора вершин потомков (п. 2.2), приводящую к появлению этого графа в выходном массиве. В соответствии с доказанным в [8] утверждением, приведенным выше (утверждение 2), после выполнения проверки на поглощение останется только полное множество предельных графов. Утверждение 1 доказано.

Доказательство утверждения 2. Докажем справедливость этого утверждения для следующего случая:

- выберем одну вершину потребителя;
- будем строить компоненты и при этом игнорировать выбранную вершину, т.е. пропускать все действия метода, связанные с ней;
- когда не останется активных вершин кроме выбранной, продолжим строить компоненты, рассматривая пути от выбранной вершины.

Покажем, что выходные массивы в этом случае и в методе построения компонент совпадают. Пусть M — максимальная длина пути к вершине источнику среди всех вершин потребителей кроме выбранной. Рассмотрим граф Γ ,

полученный из исходного графа H по следующему принципу: к выбранной вершине потребителю добавляется “хвост” из M обычных (не коммутаторов) вершин, соединенных последовательно. Метод построения компонент для Γ даст такие же результаты, что и для H , с точностью до “хвоста”. С другой стороны, последовательность выбора вершин потребителей в методе построения компонент для Γ будет такой же, как в описанном случае.

Обобщим это рассуждение: очевидно, что выходной массив не изменится, если такие “хвосты” добавить к нескольким вершинам потребителям, при этом длина хвостов может быть произвольной. И таким образом каждую вершину потребителя можно “подключить” на любом шаге работы компонентного метода. Аналогичным образом можно показать, что и очередность выбора компонент на шаге 1 не важна. Утверждение 2 доказано.

Доказательство утверждения 3. Компонента I выходного массива является предельным графом, так как она является лесом. Докажем утверждение, показав, что при поставленных условиях на вершины ветвления построенный по I W подграф будет совпадать с I . Рассмотрим дерево T в I и любое ребро исходного графа, у которого одна из инцидентных вершин принадлежит T (обозначим ее t), а вторая — не принадлежит (обозначим ее q). Покажем, что вершина q — это вершина множества C . Если $t \in U$, то ее степень строго больше двух: как минимум два ребра принадлежат дереву плюс рассматриваемое ребро, т.е. t — вершина ветвления и она окружена вершинами коммутаторами. Если $t \in S$, то по свойствам рассматриваемых графов $q \in C$ t не может быть вершиной потребителем, так как единственное ее ребро принадлежит I . По условиям построения W подграфа эти вершины множества C удаляются. Отсюда следует, что W подграф, построенный по I , будет совпадать с I . Утверждение 3 доказано.

СПИСОК ЛИТЕРАТУРЫ

1. *Phyu E.E., Lin K.M., Moe T.T.* Loss Reduction and Reliability Improvement of Industrial Distribution System through Network Reconfiguration // Int. J. Energy and Power Engineer. 2018. V. 12. No. 11. P. 822–828.
2. *Sudhakara Reddy A.V., Satish Kumar Reddy M.* Network Reconfiguration of Distribution System for Loss Reduction using GWO Algorithm // Int. J. Electric. and Comp. Engineer. 2017. V. 7. No. 6. P. 3226–3234.
3. *Sedighzadeh M., Esmaili M., Mahmoodi M.* Reconfiguration of Distribution Systems to Improve Reliability and Reduce Power Losses using Imperialist Competitive Algorithm // Iranian J. Electric. & Electronic Engineer. 2017. V. 13. No. 3. P. 287–302.
4. *Sambaiah K.S.* A Review on Optimal Allocation and Sizing Techniques for DG in Distribution Systems // IJRER. 2018. V. 8. No. 3. P. 1236–1256.
5. *Landeros A., Koziel S.* Distribution Network Reconfiguration using Feasibility-Preserving Evolutionary Optimization // J. Mod. Power Syst. Clean Energy. 2019. V. 7. No. 3. P. 589–598.
6. *Kayal V., Chanda C.K.* A Simple and Fast Approach for Allocation and Size Evaluation of Distributed Generation // IJEEE. 2013. V. 4. No. 7. P. 2–10.
7. *Christine E., Doig C.* Analysis on Voltage Stability Indices. Master Thesis. Aachen Germany: RWTH Aachen University. 2012. P. 1–106.

8. *Grebenyuk G.G., Krygin A.A.* Algorithms for Optimization of the Number of Switchings in Heat Supply Networks Reconfiguration // Autom. Remote Control. 2007. V. 68. No. 12. P. 2187–2197.
Гребеннюк Г.Г., Крыгин А.А. Алгоритмы оптимизации числа переключений при реконфигурации сетей теплоснабжения // АиТ. 2007. № 12. С. 101–112.
9. *Grebenyuk G.G., Krygin A.A.* Limit Graphs in Structural Optimization of Modes in Distribution Networks // Autom. Remote Control. 2015. V. 76. No. 1. P. 120–132.
Гребеннюк Г.Г., Крыгин А.А. Предельные графы в структурной оптимизации режимов распределительных сетей // АиТ. 2015. № 1. С. 147–162.

Статья представлена к публикации членом редколлегии В.М. Вишневым.

Поступила в редакцию 25.06.2020

После доработки 31.10.2020

Принята к публикации 08.12.2020