УДК 621.31
***А.В. Егоров**, д.т.н., профессор; **Г.Н. Малиновская**, к.т.н.; **А.А. Трифонов**, к.т.н., доцент, Российский государственный университет нефти и газа имени И.М. Губкина*

Алгоритмы решения некоторых задач диспетчерского управления электротехническими системами промышленных предприятий

Развитие электротехнических систем предприятий нефтегазовой отрасли приводит к усложнению их схем, широкому применению собственных источников электроснабжения. Такие системы требуют более взвешенных подходов к управлению ими, в особой степени это относится к электротехническим системам, содержащим в своем составе генераторы или электростанции собственных нужд. В связи с все более возрастающими требованиями к степени утилизации попутного нефтяного газа доля таких систем в нефтяной и газовой промышленности постоянно растет.

Ключевые слова: диспетчерское управление электротехническими системами, формализация состояния системы, граф состояний, классификация состояний электротехнической системы, план оперативных переключений, алгоритм Форда, алгоритм Дейкстры, алгоритм Флойда.

Отдельную проблему для заметного числа нефтяных и газовых объектов представляет проблема устойчивости как технологического процесса, так и самих электротехнических систем при кратковременных нарушениях электроснабжения [1]. Практика эксплуатации подобных систем показывает необходимость организации централизованных диспетчерских служб, особенно если речь идет о распределенных объектах, например предприятиях добычи нефти и газа. Вполне понятно, что с развитием системы, ростом ее размерности задачи диспетчера становятся все более сложными, а его труд – все более напряженным и требующим все большей квалификации. Заметную помощь в решении задач оптимального диспетчерского управления такими электротехническими системами могут оказать компьютерные системы поддержки принятия оперативных решений.

Некоторые вопросы, связанные с созданием таких систем, были рассмотрены в [2]. В частности, была предложена формализация текущего состояния электротехнической системы посредством направленного графа состояний системы, предложены критерии оценки ситуаций, возникающих в электротехнической системе, показаны ограничения на ситуации, подлежащие решению путем диспетчерского управления. В целях дальнейшего рассмотрения обсуждаемых задач приведем основные положения работы [2].

Текущее состояние электротехнической системы формализуется посредством графа состояний. Вершины графа однозначно описывают состояние всех элементов системы генерации и распределения электрической энергии. Переходы из одного состояния в другое связаны с изменением состояния только одного из элементов. Предложено объединять в один формальный элемент два и более физических элементов, действие которых на систему идентично. Переход описывается характерным временем перехода, так задаются длины ребер графа. Предполагается, что любой элемент может иметь только два состояния, условно назовем их: включен или выключен. Длины ребер в предложенной формализации переменны, время перехода определяется состоянием элемента, например, штатное, отключен защитой, аварийное состояние, ремонтное состояние. Характерные времена переходов могут быть более или менее жестко связаны с текущим состоянием элемента, отслеживаемым посредством соответствующих компонентов АСУ электроснабжением [3]. Могут быть учтены и иные факторы, влияющие на данный показатель, вплоть до вполне субъективных.

Была предложена [2] следующая классификация состояний электротехнической системы, основанная на последствиях того или иного состояния для технологического процесса объекта. Все состояния делятся на следующие группы:

1) нормальное рабочее состояние – как правило, одно штатное состояние системы электроснабжения;

2) длительно допустимые состояния – состояния, отличающиеся от нормального рабочего, но не оказывающие

влияния на технологический процесс. В этих состояниях может возникать необходимость проведения ремонтных работ, снижаться надежность электроснабжения. Длительность пребывания электротехнической системы в состояниях, относящихся к данной группе, практически ничем не ограничена;

3) допустимые состояния – те состояния, которые слабо влияют на режим технологического процесса. В этих состояниях возможно отключение неответственных электроприемников, существенное снижение надежности электроснабжения, незначительная перегрузка элементов системы электроснабжения. Допустимая длительность пребывания электротехнической системы в состояниях, относящихся к данной группе, ограничена, но достаточно велика – порядка нескольких часов;

4) кратковременно допустимые состояния – такие состояния системы электроснабжения, которые существенно влияют на технологический процесс. Допустимое время пребывания электротехнической системы в состояниях данной группы составляет секунды или минуты. При превышении допустимой длительности необходима полная или частичная остановка технологического процесса;

5) критичные состояния – те состояния, допустимое время пребывания в которых составляет доли секунд – секунды. Превышение допустимой длительности требует немедленной аварийной остановки технологического процесса. Восстановление нормального режима технологического процесса возможно путем самозапуска электроприводов или по программе их короткого пуска;

6) к группе недопустимых состояний относятся те, из которых восстановление нормального технологического режима по программам самозапуска или короткого пуска невозможно.

Выполненный в [2] анализ характерных значений допустимых времен пребывания системы в низких состояниях показал, что задача вывода системы из состояний пятой и шестой групп не может решаться путем диспетчерского управления системой. Эта функция должна быть возложена на систему противоаварийных защит и автоматики, запуск которой должен блокировать работу системы диспетчерского управления.

Работа системы поддержки принятия решений ориентирована на выработку оптимальных по времени реализации планов оперативных переключений для вывода электротехнической системы из низких групп состояний. Могут быть поставлены, например, следующие задачи:

• выработка плана переключений, переводящего систему в режим первой группы за минимальное время;

• выработка плана переключений, обеспечивающего перевод системы в режим первой группы с минимальным числом операций;

• выработка плана переключений, обеспечивающего перевод системы в любой из режимов группы не ниже заданной за минимальное время или путем осуществления минимального числа операций;

• выработка плана переключений, обеспечивающего перевод системы в лучшую из возможных групп состояний за время, не превышающее заданное.

Возможны и иные критерии выработки оптимального плана оперативных переключений.

На выработку оптимального по заданному критерию плана переключений могут быть наложены определенные ограничения. В частности, могут быть рассмотрены следующие:

• запрет на перевод системы в группу ниже стартовой, за исключением случаев, когда такой план переключений является единственным;

• ограничение числа операций при минимизации времени реализации плана;

• ограничение времени при минимизации числа операций;

• ограничение числа операций по синхронизации генераторов собственных нужд при реализации плана оперативных переключений, за исключением случаев, когда иных решений не существует.

Перечень ограничений может быть расширен. Собственно к алгоритмам также должен быть предъявлен ряд требований. Основными из них следует считать универсальность, устойчивость, минимально возможное время реализации.

Реализация первых двух задач аналогична поиску кратчайшего пути в графе состояний электротехнической системы. Различие состоит лишь в том, что в первом случае в качестве длины ребра графа принимается характерное время перехода, а во втором длины (веса) всех ребер считаются одинаковыми, например равными единице. Такие алгоритмы давно известны [4], поэтому задача для данного случая сводится к их анализу и выбору наиболее приемлемого для рассматриваемых задач диспетчерского управления.

Для поиска кратчайшего пути чаще всего используются алгоритмы Форда и Дейкстры. Рассмотрим их несколько подробнее и оценим применимость для решения поставленной задачи. Алгоритм Дейкстры применим для поиска кратчайшего пути в ориентированном нагруженном графе от заданной начальной вершины до всех остальных вершин графа. Отличительной особенностью данного алгоритма является то, что он работает только с неотрицательной матрицей весов. Алгоритм Форда также применим для поиска кратчайшего пути в ориентированном нагруженном (то есть имеющем ребра различной длины) графе от заданной начальной вершины до всех остальных вершин графа. Отличительной особенностью данного алгоритма является то, что он работает с произвольной матрицей весов, но обязательно требует соблюдения условия отсутствия в графе цикла отрицательного веса. Таким образом, оба названных алгоритма пригодны для решения поставленных задач.

В нашем случае задача может быть сформулирована как поиск кратчайших путей между всеми парами вершин в графе. В этом случае очевидный способ ее решения заключается в n-кратном применении рассмотренных ранее алгоритмов Дейкстры или Форда. Если граф имеет большую размерность, то время на ее решение таким способом становится весьма заметным. Для этих целей можно использовать другой алгоритм, реализующий совершенно иной подход к поиску кратчайших путей между всеми парами вершин в графе, заданном произвольной матрицей весов. Это алгоритм Флойда [4], разработанный достаточно давно, но используемый относительно редко, в связи с тем что гораздо чаще приходит-

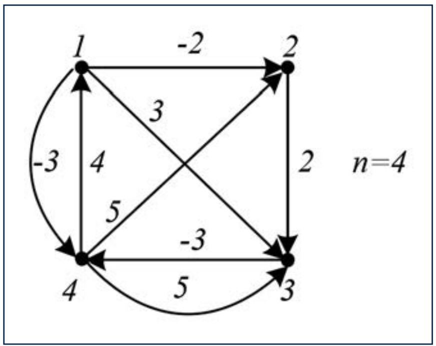

Рисунок. Пример графа состояний

ся искать путь из заданной начальной вершины во множество конечных. Мы имеем дело с обратной задачей, поиском путей из множества начальных вершин в ограниченное множество конечных вершин. Названный алгоритм базируется на использовании последовательности из n-преобразований начальной матрицы весов, где n – это число вершин в графе. Если этот алгоритм применить к графам, заданным неотрицательной матрицей весов, то он сэкономит почти 50% времени по сравнению с n-кратным применением алгоритмов Дейкстры и Форда. Таким образом, мы имеем граф, вершины которого соответствуют возможным состояниям структуры электротехнической системы. Иными словами, каждая вершина однозначно описывается вектором положения коммутационных аппаратов. Ребра графа соответствуют возможным переходам из одного состояния в другое. По-прежнему [2] предполагаем, что в фиксированный момент времени возможно изменение положения только одного коммутационного аппарата. Длина (вес) ребра соответствует характерному времени перехода из одного состояния в другое, как это было отмечено выше. Перейдем к изложению сути алгоритма.

Для начала работы алгоритма подготовим матрицу, описывающую структуру имеющегося графа G – так называемую матрицу весов – $C=||C_{ij}||$, в которой изначально C_{ij} = весу дуги (i, j) и $C_{ij} = \infty$ – в случае отсутствия такой дуги. Кроме того, полагаем все диагональные элементы этой матрицы равными нулю, т.е. $C_{ii} = 0$ для всех $i = 1, n$, где n – число вершин в графе.

Алгоритм основан на следующих соображениях. Пусть i, j, k – три любые вершины графа G, и мы хотим получить кратчайший путь из i в j, не содержащий внутренних вершин, кроме k. Очевидно, что для этого достаточно выбрать меньшую из двух величин: вес прямого пути из i в j (это элемент C_{ij}) или же вес пути из i в j через вершину k (это величина $C_{ik} + C_{kj}$). Если зафиксировать k и проделать эту трехместную операцию для всех пар (i, j), то получим матрицу $C^k = ||C^k_{ij}||$, у которой $C^k_{ij} = \min\{C^{k-1}_{ij}, C^{k-1}_{ik} + C^{k-1}_{kj}\}$. Затем фиксируется другое значение k и трехместная операция применяется к матрице C^k и т.д.

В результате строится последовательность матриц $C^0, C^1, \ldots, C^n$, последняя из которых C^n содержит длины кратчайших путей между всеми парами вершин графа.

Алгоритм поиска перечня вершин кратчайших путей строится следующим образом.

Чтобы найти сами пути вместе с матрицей $C^к$, на каждой итерации строится матрица $P^к=||P^к_{ij}||$. Элемент P_{ij} указывает вершину, непосредственно предшествующую вершине j в кратчайшем пути от i к j. Матрице $P^0=||P^0_{ij}||$ присваиваются начальные значения, а именно $P^0_{ij} = i$ для всех значений i и j.

По диагонали этой матрицы всегда стоят нули, так как нет предшествующей вершины в кратчайшем пути из самой себя в саму себя же.

На k-м шаге алгоритма осуществляется обновление матрицы $P^к$, каждый элемент которой обновляется в соответствии со следующим правилом:

$$P^к_{ij}=\begin{cases} P^{к-1}_{kj}, \text{ если } C^{к-1}_{ij}>(C^{к-1}_{ik}+C^{к-1}_{kj}) \\ \text{не изменяется, т.е.} = P^{к-1}_{ij}, \\ \text{если } C^{к-1}_{ij}\leqslant(C^{к-1}_{ik}+C^{к-1}_{kj}) \end{cases},$$

то есть каждый элемент этой матрицы содержит номер непосредственно предшествующей вершины перед j (в пути от i к j).

Таким образом, номер непосредственно предшествующей вершины перед j (в пути от i к j) хранится в матрице P на последнем шаге (n – число вершин в графе) в элементе: $P^n_{ij} = j_1$.

Путь из i в j – (i, j) будет выглядеть перечислением полученных вершин в обратной последовательности их получения: $(i, j) = i, \ldots.. j_3, j_2, j_1, j$.

Проиллюстрируем работу алгоритма на очень простом графе, представленном на рисунке. Для полноты рассмотрения допустим, что веса некоторых ребер могут быть отрицательными.

Исходные матрицы имеют следующий вид.

$$C^0=C=\begin{pmatrix} 0 & -2 & 3 & -3 \\ \infty & 0 & 2 & \infty \\ \infty & \infty & 0 & -3 \\ 4 & 5 & 5 & 0 \end{pmatrix} \quad P^0=\begin{pmatrix} 0 & 1 & 1 & 1 \\ 2 & 0 & 2 & 2 \\ 3 & 3 & 0 & 3 \\ 4 & 4 & 4 & 0 \end{pmatrix}$$

Матрица C^0 – это матрица, описывающая структуру имеющегося графа, матрица P^0 – вспомогательная матрица, которая впоследствии будет использоваться для восстановления перечня вершин кратчайшего пути между любой парой вершин i и j.

$$C^1=\begin{pmatrix} 0 & -2 & 3 & -3 \\ \infty & 0 & 2 & \infty \\ \infty & \infty & 0 & -3 \\ 4 & 2 & 5 & 0 \end{pmatrix} \quad P^1=\begin{pmatrix} 0 & 1 & 1 & 1 \\ 2 & 0 & 2 & 2 \\ 3 & 3 & 0 & 3 \\ 4 & 1 & 4 & 0 \end{pmatrix}$$

Матрица C^1 – это матрица промежуточного результата, в которой хранятся минимальные веса как прямых, так и непрямых путей между всеми парами вершин через вершину 1.

Матрица P^1 – вспомогательная матрица, в которую в позицию P_{ij} записывается номер вершины, непосредственно предшествующей вершине j в кратчайшем пути между любой парой вершин i и j.

После первого шага работы алгоритма в матрице C^1 изменился только один элемент $C^1_{42} = 2$, т.е. путь из 4-й вершины во 2-ю через 1-ю вершину оказался короче, чем прямой путь $C^0_{42} = 5$. В матрице P также изменилось значение элемента $P^1_{42} = 1$, теперь в нем хранится информация о том, что кратчайший путь из вершины 4 в вершину 2 проходит через вершину 1.

$$C^2=\begin{pmatrix} 0 & -2 & 0 & -3 \\ \infty & 0 & 2 & \infty \\ \infty & \infty & 0 & -3 \\ 4 & 2 & 4 & 0 \end{pmatrix} \quad P^2=\begin{pmatrix} 0 & 1 & 2 & 1 \\ 2 & 0 & 2 & 2 \\ 3 & 3 & 0 & 3 \\ 4 & 1 & 2 & 0 \end{pmatrix}$$

После выполнения второго шага работы алгоритма будет получена матрица C^2 – это матрица промежуточного результата, в которой хранятся минимальные веса как прямых, так и непрямых путей между всеми парами вершин через вершину 1 и 2 и так далее.

$$C^3=\begin{pmatrix}0 & -2 & 0 & -3\\ \infty & 0 & 2 & -1\\ \infty & \infty & 0 & -3\\ 4 & 2 & 4 & 0\end{pmatrix}\quad P^3=\begin{pmatrix}0 & 1 & 2 & 1\\ 2 & 0 & 2 & 3\\ 3 & 3 & 0 & 3\\ 4 & 1 & 2 & 0\end{pmatrix}$$

$$C^4=\begin{pmatrix}0 & -2 & 0 & -3\\ 3 & 0 & 2 & -1\\ 1 & -1 & 0 & -3\\ 4 & 2 & 4 & 0\end{pmatrix}\quad P^4=\begin{pmatrix}0 & 1 & 2 & 1\\ 4 & 0 & 2 & 3\\ 4 & 1 & 0 & 3\\ 4 & 1 & 2 & 0\end{pmatrix}$$

На последнем шаге работы алгоритма в матрице C^4 нашего примера содержатся длины кратчайших путей между всеми парами вершин графа, причем эти пути могут содержать произвольное количество промежуточных вершин.

Далее следует определить сам путь, представляемый в виде перечня последовательно проходимых вершин. Данная задача решается с помощью матрицы P. Например, чтобы восстановить полный перечень вершин кратчайшего пути из 2-й вершины в 1-ю, следует посмотреть, какое число хранится в элементе матрицы на последнем шаге алгоритма, для нашего примера это элемент $P^4_{21} = 4$. Это значит, что предпоследняя вершина в пути из 2 в 1 – это вершина 4 (кратчайший путь выглядит примерно так: 2, ?, …, ?, 4, 1). Дальше необходимо посмотреть, что хранится в элементе матрицы $P^4_{24} = 3$. Это значит, что предпоследняя вершина в пути из 2 в 4 – это вершина 3 (кратчайший путь выглядит примерно так: 2, ?, …, ?, 3, 4, 1). И так далее все повторяется аналогично. Для нашего примера дальше необходимо посмотреть, что хранится в элементе матрицы $P^4_{23} = 2$. Это значит, что предпоследняя вершина в пути из 2 в 3 – это вершина 2, она же и является начальной вершиной искомого пути, следовательно, кратчайший путь содержит следующий перечень вершин: 2, 3, 4, 1.

Третья и четвертая задачи имеют почти аналогичное решение. Рассмотрим подход к решению третьей задачи. Поскольку матрица C^n содержит длины кратчайших путей между всеми парами вершин графа и ее элементы остаются неизменными, до тех пор пока не произойдут изменения весов каких-либо ребер, поступим следующим образом. Выберем из матрицы C^n величины кратчайших путей между начальной вершиной и всеми вершинами, относящимися к заданной группе и к группам выше нее по приоритетам. В дальнейшем выбираем то реальное состояние электротехнической системы, которого можно достичь за минимальное время. По матрице P^n построим искомый путь, как это было описано выше. Отметим, что матрица P^n также останется неизменной, до тех пор пока не произойдут изменения весов каких-либо ребер. Восстановим по матрице P^n перечень вершин в каждом из путей и для каждого пути определим количество промежуточных вершин его составляющих. Далее, выбрав путь с минимальным числом промежуточных вершин, найдем путь с минимальным числом операций, что позволит нам выбрать план переключений, обеспечивающий перевод системы в любой из режимов группы не ниже заданной путем осуществления минимального числа операций.

Для решения четвертой из поставленных задач выделим в матрице C^n все элементы, соответствующие времени, не превышающему заданного значения, и соответствующие тем или иным конечным вершинам. Из полученного перечня вершин выберем одну или несколько, относящихся к группам с наивысшим приоритетом. Таким образом, мы определили достижимую за заданное время группу состояний. Далее по матрице P^n строим искомый путь.

Рассмотрим возможные способы соблюдения сформулированных выше ограничений на вырабатываемые планы.

Для соблюдения первого из них поступим следующим образом. Восстановим перечни вершин для всех путей, подходящих нам по любому из критериев указанных четырех задач. Для этих путей проведем анализ вершин на предмет принадлежности последующего состояния к определенной группе. Если группа оказывается ниже группы предшествующего состояния, помечаем этот путь как менее приоритетный для нас. Например, можно посчитать количество переходов системы в группы ниже стартовой. Чем их больше, тем приоритет такого пути ниже. Если находится путь, в котором нет таких переходов, то он и будет оптимальным. В противном случае можем лишь сделать вывод, что невозможно перевести систему в нужное состояние, не переводя ее промежуточное состояние в группу вершин ниже стартового уровня.

Для соблюдения второго ограничения поступим следующим образом. Выберем из матрицы C^n все элементы, соответствующие времени, не превышающему заданного значения, пересчитаем количество вершин, их составляющих, и выберем тот вариант плана, в котором число переключений будет не выше заданного. Если таких путей будет несколько, то из них выбираем путь с минимальным временем реализации плана.

Для соблюдения третьего ограничения поступим почти аналогичным образом. Выберем из матрицы C^n все элементы, соответствующие времени, не превышающему заданного значения, пересчитаем количество вершин, их составляющих, и выберем тот вариант плана, в котором число переключений будет минимальным. Если таких путей будет несколько, то из них опять выбираем путь с минимальным временем реализации плана.

Для соблюдения четвертого ограничения поступим следующим образом. Перед решением задачи пометим вершины, содержащие операции по синхронизации генераторов собственных нужд. Выберем из матрицы C^n все элементы, соответствующие времени, не превышающему заданного значения, пересчитаем количество помеченных вершин и выберем тот вариант плана, в котором число таких вершин будет минимальным. Если таких путей будет несколько, то из них выбираем путь с минимальным временем реализации плана.

Предлагаемые алгоритмы могут быть эффективно использованы при создании автоматизированных систем поддержки принятия решений в задачах диспетчерского управления электротехническими системами объектов нефтяной и газовой промышленности. В значительной мере такие задачи актуальны для распределенных объектов, имеющих в своем составе собственные генерирующие мощности. Для таких систем интуитивные решения далеко не всегда оказываются оптимальными. В связи с тем что основные матрицы могут быть сформированы заранее, а их пересчет необходим только при изменениях ожидаемого времени перехода из одного состояния в другое, предлагаемые алгоритмы отличаются высоким быстродействием. Эта особенность также важна для систем поддержки принятия решений.

Литература:

1. Ершов М.С., Егоров А.В., Трифонов А.А. Устойчивость промышленных электротехнических систем. – М.: Недра, 2010.
2. Ершов М.С., Егоров А.В., Трифонов А.А. Алгоритмизация задач диспетчерского управления системами промышленного электроснабжения объектов с электростанциями собственных нужд // Промышленная энергетика. – 2005. – № 3.
3. Егоров А.В., Малиновская Г.Н., Репина Ю.В., Головатов С.А. Функциональные задачи АСУ электроснабжением. Оценка надежности электроснабжения элементов электротехнической системы предприятия // Территория «НЕФТЕГАЗ». – 2012. – № 9.
4. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978.

UDC 621.31

A.V. Yegorov, Doctor of Sciences (Engineering), Professor; **G.N. Malinovskaya**, Candidate of Sciences (Engineering); **A.A. Trifonov**, Candidate of Sciences (Engineering), Associate Professor, Gubkin Russian State University of Oil and Gas

Algorithms for solving some tasks of electrical system dispatching control of industrial enterprises

Development of electrical systems of oil and gas industry enterprises results in complication of their schemes, extensive application of own sources of electric power supply. Such systems require more balanced approaches to their control. Especially it refers to electrical systems containing generators or auxiliary power plants. Due to increasingly more requirements to the degree of associated petroleum gas recovery, the share of such systems in the oil and gas industry constantly grows.

Key words: electrical system dispatching control, formalization of system status, state graphs, classification of the states of electrical systems, routine switching plan, Ford algorithm, Dejkstra algorithm, Floyd algorithm.

References:

1. Yershov M.S., Yegorov A.V., Trifonov A.A. Ustoichivost' promyshlennykh elektrotekhnicheskikh sistem (Stability of industrial electrical systems). – Moscow: Nedra, 2010.
2. Yershov M.S., Yegorov A.V., Trifonov A.A. Algoritmizatsiya zadach dispetcherskogo upravleniya sistemami promyshlennogo elektrosnabzheniya ob'ektov s elektrostantsiyami sobstvennykh nuzhd (Algorithmization of the tasks for dispatching control of the systems for industrial electric power supply to the facilities with auxiliary power plants) // Industrial Energy Journal. – 2005. – No. 3.
3. Yegorov A.V., Malinovskaya G.N., Repina Yu.V., Golovatov S.A. Funktsional'nye zadachi ASU elektrosnabzheniem. Otsenka nadezhnosti elektrosnabzheniya elementov elektrotekhnicheskoi sistemy predpriyatiya (Functional tasks of electric power supply automated control system. Evaluation of reliability of electric power supply to the elements of electrical system of the enterprise) // NEFTEGAS Territory. – 2012. – No. 9.
4. Kristofides N. Teoriya grafov. Algoritmicheskiy podkhod (Theory of graphs. Algorithmic approach). – Moscow: Mir, 1978.