

ИСПОЛЬЗОВАНИЕ АЛГОРИТМА ПЧЕЛИНОЙ КОЛОНИИ ДЛЯ РЕШЕНИЯ ЗАДАЧИ ОПТИМАЛЬНОГО СОЕДИНЕНИЯ ПОТРЕБИТЕЛЕЙ И ИСТОЧНИКА ЭЛЕКТРОЭНЕРГИИ

¹Лозовский В.В., ²Богдан А.В., ²Грищенко Д.Н.

¹МИРЭА - Российский технологический университет, 119454, Россия, г. Москва, проспект Вернадского, 78
e-mail: lozovskiy_v_v@mail.ru

²ФГБОУ ВО «КубГАУ им. И.Т. Трубилина», 350044, г. Краснодар, ул. Калинина, 13
e-mail: alina48av@mail.ru, professs0r@yandex.ru

Для задачи оптимального соединения потребителей и источника электроэнергии, при достижении определённого количества входных данных, время на поиск решения становится неприемлемо большое. Стохастические алгоритмы дают хорошие результаты при решении оптимизационных задач с большим набором входных данных. Одним из стохастических алгоритмов является алгоритм пчелиной колонии, отличительной особенностью данного алгоритма является «улей», в котором «пчёлы» обмениваются информацией о результатах поиска экстремума целевой функции.

Ключевые слова: алгоритм пчелиной колонии, оптимизация, распределительная сеть, минимум технологических потерь.

THE USE OF ALGORITHM BEE COLONY TO SOLVE THE PROBLEM OF OPTIMAL CONNECTION LOAD AND POWER SOURCE

¹Lozovskiy V.V., ²Bogdan A.V., ²Grishchenko D.N.

¹MIREA - Russian Technological University, 119454, Moscow, 78 Vernadskogo Avenue, Russia,
e-mail: lozovskiy_v_v@mail.ru

²Federal State Budget Educational Institution of Higher Education «Kuban State agrarian University named after I.T. Trubilin», 350044, Krasnodar, Kalinina str., 13
e-mail: alina48av@mail.ru, professs0r@yandex.ru

For the problem of optimal connection load and power source, when reaching definite quantity input data, time of solving becomes unacceptable long. Stochastic algorithms give acceptable results for solving optimal problems with many quantity input data. One of the stochastic algorithms is algorithm bee colony, a distinctive feature of this algorithm is «hive», in which «bees» trade information about results of research extremum of the objective function.

Keywords: algorithm bee colony, optimization, distribution network, minimum technological losses.

Постановка задачи и описание алгоритма решения

Возьмём в качестве примера схему, изображённую на Рисунке 1, для данной схемы, состоящей из 17 узлов и 25 рёбер, имеем следующие параметры: $U_{\text{ист}} = 1000$ В; $R_{\text{ветви}} = 0,5$ Ом; $R_{\text{н}} = 200$ Ом. Как уже неоднократно упоминалось авторами в недавних работах, у рассматриваемой схемы, которая представляет собой неориентированный взвешенный граф, имеется 100352 остовных дерева. Даже для такого небольшого графа получается достаточно большое количество остовных деревьев, на поиск которых требуется значительное время и вычислительные мощности. Однако, найдя все остовные деревья, знаем величину наименьших технологических потерь в линиях электропередач (рёбрах графа): 2548 Вт, то есть потери, соответствующие самой оптимальной схеме. Следует отметить, что кроме минимизации потерь в распределительной сети, учитываются потери напряжения у конечных потребителей, тем самым вводится дополнительное ограничение в задачу оптимизации (потери напряжения у конечных потребителей не должны превышать 10 %), к основному ограничению: граф должен быть связным. Оптимальная конфигурация рёбер (остовное дерево) выглядит следующим образом: (0, 1), (1, 2), (1, 5), (2, 3), (2, 6), (3, 4), (4, 8), (5, 9), (6, 7), (7, 11), (8, 12), (9, 10), (9, 13), (13, 14), (14, 15), (15, 16).

Для сравнения, хотелось бы вскользь отметить результаты работы алгоритма роя частиц (particle swarm optimization), всё-таки в данной работе речь идёт о другом алгоритме. В результате работы алгоритма, имея в рое всего 10 частиц, был получен результат по потерям 2572 Вт. Да, мы не получили оптимальную схему,

однако, времени на поиск решения было затрачено существенно меньше, а полученный результат отличается от оптимального менее чем одну сотую процента.

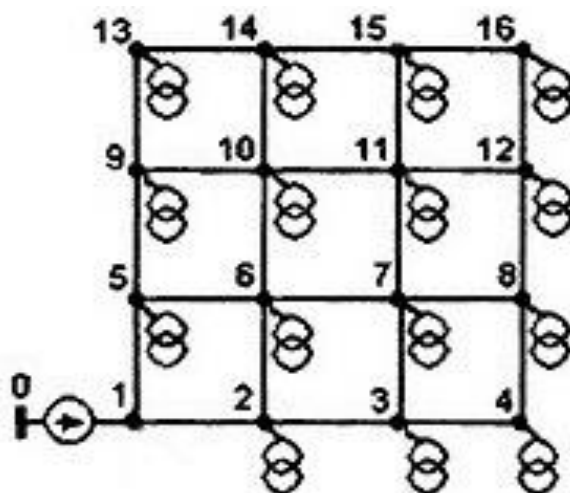


Рисунок 1 – Исходная схема распределительной схемы электроснабжения

Что касается логики алгоритма пчелиной колонии, то в целом и общем она имеет много общего с алгоритмом роя частиц. На первоначальном этапе задаются параметры, описывающие количественных и качественные характеристики «роя» и «улья». Затем для каждой «пчелы» случайным образом инициализируется её местоположение, которое задаётся перечнем рёбер, которые необходимо исключить из графа, после чего рассчитываются значения целевой функции для каждой из «пчёл», сравниваются и ранжируются. Несмотря на то, что местоположение «пчелы» задаётся случайным образом, на выбор рёбер влияет величина протекающего по этой ветви (ребру) тока — чем ниже ток, тем выше вероятность того, что именно это ребро будет выбрано. Кроме того, что логично и очевидно, ветви с наибольшими значениями токов, имеют самые малые шансы быть исключёнными из схемы (графа). Помимо величины протекающего в ветви тока, на возможность исключения ребра из графа, решающее значение оказывает часть кода, в которой проверяется необходимость исключения рёбер из графа, данная часть кода изображена на Рисунке 2.

```
def __check_graph(self):  
    """  
    функция для проверки возможности удаления ребра из графа  
    если количество ветвей равно числу узлов минус единица, то уже найдено остаточное дерево и больше удалять  
    ребра нельзя  
    :return:  
    """  
    counter_of_branches = 0  
    for edge in self.__local_graph.edges():  
        if str(self.__local_graph.edges[edge]['type_edge']) == str('Branch'):  
            counter_of_branches += 1  
    if counter_of_branches == int(self.__local_graph.number_of_nodes() - 1):  
        return False  
    return True
```

Рисунок 2 – Часть исходного кода программы — функция *check_graph*.

Далее выбираются лучшие места для исследований и «обычные» места для исследований, и в эти области отправляется, определённое на начальном этапе количество «пчёл». Очевидно, что насколько правильно выбрана конфигурация «улья», настолько будет эффективна работа алгоритма с точки зрения быстродействия и сходимости экстремума целевой функции. Хотелось бы уделить внимание ещё одному аспекту работы алгоритма в контексте его применения к задаче оптимизации распределительной сети — сравнение координат двух «пчёл». Авторами была написана функция, реализующая данное сравнение, блок исходного кода этой функции представлен на Рисунке 3.

Для решения рассматриваемой задачи авторами была выбрана следующая конфигурация «улья»: 4 пчелы-разведчика, 2 лучших места, на которые поочередно отправляются по 2 пчелы из улья, 2 «обычных» места, на которые также отправляются по 2 пчелы из улья. Таким образом, в общей сложности получается 12 пчёл, 4 из которых пчелы-разведчики. В результате работы алгоритма были получены следующие результаты по величине технологических потерь (в среднем: часто получаемый результат): 2567 Вт.

```

def difference_of_lists(argument1, argument2):
    """
    функция для вычитания одного списка рёбер из другого
    :param argument1: эталон - из которого вычитают
    :param argument2: проверяемый - который вычитается
    :return:
    """
    result_list = []
    for edge in argument2:
        if edge not in argument1:
            result_list.append(edge)
    if not result_list:
        # список пуст возвращаем ничего (None)
        return None
    return result_list

```

Рисунок 3 – Часть исходного кода программы — функция *difference_of_lists*.

Данный результат также отличается от оптимального менее чем на одну сотую процента, но имеет более приемлемую асимптотику по времени. Перечень рёбер необходимых для исключения из графа: (3, 7), (11, 12), (6, 10), (10, 14), (11, 15), (7, 8), (10, 11), (5, 6), (9, 13), Результаты работы алгоритма приведены на Рисунке 4.

```

Best Position: [(3, 7), (11, 12), (6, 10), (10, 14), (11, 15), (7, 8), (10, 11), (5, 6), (9, 13)]
Best Fitness: 2567.394388

```

Рисунок 4 – Результат работы алгоритма.

Заключение

В заключении хотелось бы отметить, в данной работе авторы не уделяли должного вниманию конфигурации самого улья, как в случае работы алгоритма роя частиц, так и в случае алгоритма пчелиной колонии существенное влияние на быстродействие алгоритма и его сходимость оказывают первоначально заданные параметры, для алгоритма роя частиц это коэффициенты обучения, для алгоритма пчелиной колонии это количество пчёл и конфигурация улья.

Очевидно, что при усложнении схемы, возрастает количество входных данных (количество рёбер, количество узлов), что в свою очередь значительно усложняет поиск оптимального решения, в этом случае, воспользовавшись стохастическими алгоритмами, с оптимально подобранными параметрами (либо коэффициентами), можно получить результат близкий к оптимальному за приемлемое время.

Кроме того, в процессе множества итераций запуска алгоритма, авторами была обнаружена интересная особенность, а именно: у рассматриваемой схемы есть не одна оптимальная схема, а их несколько, то есть, при расчёте были получены одни и те же значения технологических потерь, однако, конфигурация графа (как именно соединены рёбра между собой) была различна. Какое именно конечное количество оптимальных конфигураций графа есть в рассматриваемом, сказать пока трудно, однако, можно предположить, что у других графов (схем распределительной сети) может быть не одна оптимальная схема.

Получив некоторое количество оптимальных схем, а также схем, которые были хуже оптимальных (с точки зрения величины технологических потерь), возникла идея попытаться найти такое «расстояние» между этими схемами (в ветви, которые бы давали совершенно другую более или менее оптимальную схему, установить переключатели, которые бы замыкали/размыкали ветвь), что при возникновении аварии, за минимальное количество оперативных переключений получить наиболее оптимальную схему без нарушения электроснабжения у конечных потребителей. Данная тема предмет последующих работ авторов.

Список литературы

1. Богдан А.В. Алгоритмы определения схемы сети 6 - 10 кВ с меньшими технологическими потерями / Богдан В.А., Нетребко Д.С., Грищенко Д.Н. — № 1 — Краснодар: Сельский механизатор (ФГБОУ ВО Кубанский государственный аграрный университет им. И.Т. Трубилина), 2022. — 39-41 с.

References

1. Bogdan A.V. Algorithms for determining of schemes 6 – 10 kV with minimum technological losses / Bogdan V.A., Netrebko D.S., Grishchenko D.N. - № 1 - Krasnodar: Agriculture machine operator (Federal State Budget Educational Institution of Higher Education «Kuban State agrarian University named after I.T. Trubilin»), 2022. – 39-41 p.