

УДК 004.021

## АЛГОРИТМ НАХОЖДЕНИЯ ПРИБЛИЖЕННОЙ К ОПТИМУМУ НАЧАЛЬНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ОТБОРА ДЛЯ ОПТИМИЗАЦИИ РАСПИСАНИЯ

**Димитриев А.П.**

*ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова»,  
Чебоксары, e-mail: dimitrie1@yandex.ru*

Объект исследования – задача составления оптимального расписания регулирования электрической мощности, коммутируемой группе потребителей с применением широтно-импульсного метода. Предмет исследования – алгоритм оптимизации последовательности отбора, основанный на методе имитации отжига, который является эффективным алгоритмом для решения указанной задачи. Разрабатывается дополнение к указанному алгоритму, направленное на повышение скорости сходимости к решениям, близким к оптимальным, и ставящее своей целью получение удачной начальной последовательности отбора объектов для размещения в расписании. Для этого разработан алгоритм нахождения начальной последовательности, основанный на предварительном построении множества расписаний. Предлагаемый алгоритм реализован на языке программирования Object Pascal. Для изучения работы данного алгоритма проведено несколько серий вычислительных экспериментов с тестовыми наборами исходных данных, имеющими различные особенности. Определены два оптимальных параметра этого алгоритма. Дополнение алгоритма оптимизации последовательности отбора алгоритмом нахождения начальной последовательности приводит к улучшению результатов в среднем на 29% относительно теоретически не превышаемого минимума целевой функции при неизменном времени вычислений. Предлагаемый алгоритм позволяет находить начальные последовательности, приводящие затем к решениям, близким к оптимальным, быстрее, чем ранее разработанные алгоритмы.

**Ключевые слова:** метод имитации отжига, широтно-импульсный метод, оптимизация, расписание, параметры алгоритма

## ALGORITHM FOR FINDING CLOSE TO OPTIMUM AN INITIAL SELECTION SEQUENCE FOR SCHEDULE OPTIMIZATION

**Dimitriev A.P.**

*Federal State Budget Educational Institution of Higher Education «Chuvash State University  
named after I.N. Ulyanov», Cheboksary, e-mail: dimitrie1@yandex.ru*

The object of the research is the task of drawing up an optimal schedule for the regulation of electrical power, which is switched to a group of consumers using the pulse-width method. The subject of research is selection sequence optimization algorithm based on simulated annealing method, which is an effective algorithm for solving the specified problem. An addition to this algorithm is being developed, aimed at increasing the rate of convergence to solutions close to optimal and aiming at obtaining a successful initial sequence for selecting objects for placement in the schedule. For this, an algorithm for finding the initial sequence has been developed, based on the preliminary construction of a large number of schedules. The proposed algorithm is implemented in the Object Pascal programming language. To study the operation of this algorithm, several series of computational experiments were carried out with test sets of initial data having various features. Two optimal parameters of this algorithm are determined. Supplementing the selection sequence optimization algorithm with the algorithm for finding the initial sequence leads to an improvement in the results by an average of 29% relative to the theoretically unsurpassable minimum of the objective function with constant computation time. The proposed algorithm makes it possible to find initial sequences, which then lead to solutions close to optimal, faster than previously developed algorithms.

**Keywords:** simulated annealing method, pulse-width method, optimization, schedule, algorithm parameters

В данной работе объектом исследования является задача составления оптимального расписания регулирования электрической мощности, коммутируемой группе потребителей с применением широтно-импульсного метода (ШИМ) [1; 2]. Здесь так же, как это часто происходит при решении различных других оптимизационных задач, приходится принимать компромиссные решения – необходимо делать выбор между высокой скоростью вычислений и их точностью по отношению к оптимальности [3]. Поэтому актуальной является задача разработки алгоритма, обладающего обоими качествами – как высокой скоростью сходи-

мости, так и получением решений, близких к оптимальным. Для решения указанной задачи в [4] рассмотрены модификации алгоритма оптимизации последовательности отбора (АОПО), позволяющие получать близкие к оптимальным расписания, так как этот алгоритм для конкретно рассматриваемой задачи показывает наилучшие результаты в отношении качества оптимизации по сравнению с рядом метаэвристических алгоритмов, включая эволюционные алгоритмы и др. [4, с. 214]. Однако время получения результатов с помощью данного алгоритма при нескольких десятках потребителей составляет около 20 минут.

Цель исследования: усовершенствование АОПО, направленное на повышение скорости сходимости к решениям, близким к оптимальным.

### Материалы и методы исследования

В рассматриваемой оптимизационной задаче минимизируется целевая функция, обозначаемая  $C$ , которая рассмотрена в [5]. Значение  $C$  зависит от параметров объектов, представляющих  $n$  потребителей, и от составленного расписания. Имеются две группы параметров:  $T = \{t_i\}$ ,  $i = 1, \dots, n$ , которые представляют величины силы тока, и  $P = \{p_i\}$ ,  $i = 1, \dots, n$ , которые представляют временные продолжительности коммутации, выражаемые целыми числами в диапазоне от 0 до  $m$ , где  $m$  – число дискретных интервалов времени в расписании.

Для исследования использованы файлы исходных данных  $f1 - f5$ , представленные в [5], и некоторые другие файлы исходных данных, обладающие характерными особенностями. Во всех этих файлах описывается по  $n = 40$  объектов, за исключением файла  $in28$ , для которого  $n = 28$ . Файл  $f7$  имеет равномерное распределение обоих параметров по одинаковым областям значений. Еще в одном файле все значения  $t_i = 30$ ,  $i = 1, \dots, n$ .

В вышеуказанной работе также предложены алгоритмы с номерами от 1 до 34 (с некоторыми пропусками), предназначенные для минимизации  $C$ . Впоследствии номенклатура алгоритмов была расширена до 39. Некоторые из этих алгоритмов основаны на сортировке объектов по критериям с соответствующими номерами. Предложена также первоначальная версия АОПО, которая затем модифицировалась в [4].

АОПО основан на методе имитации отжига [6] и является одним из алгоритмов стохастической оптимизации [3], свойством которых является нестрогая детерминированность времени работы. Одним из важных факторов для существенного приближения к глобальному минимуму при использовании метода имитации отжига является выбор удачного начального приближения [7, с. 12]. Соответственно, в данном случае на скорость сходимости АОПО среди прочих факторов положительно влияет удачный выбор начальной последовательности (НП). НП – это получаемая каким-либо образом за небольшое время последовательность объектов, согласно которой эти объекты размещаются в расписании. АОПО использует в качестве отправной точки данную последовательность и в своей работе оптимизирует её.

НП ранее создавалась с использованием минимального из результатов минимизации с помощью алгоритмов с номерами из диапазона с 1 по 39, из которых отобраны 19 наиболее эффективных. При выборе какого-либо из указанных результатов производится их сравнение. Например, при использовании в качестве исходных данных файла  $f1$  алгоритм 33 приводит к расписанию со значением целевой функции  $C_{33} = 2879435$ , а алгоритм 24 – соответственно с  $C_{24} = 2877487$ , т.е. алгоритм 33 приводит к решениям, на  $C_{33}/C_{24} - 1 \approx 0,07\%$  худшим, чем при использовании алгоритма 24.

Сравнение возможно и с другой точки зрения. Для этого в [4] для данной целевой функции рассмотрено понятие теоретического минимума для входных данных, который обозначим  $T_M$ . Значение глобального минимума при большой размерности входных данных определить обычно невозможно, тогда как  $T_M$  можно оценить. Если сопоставлять достигаемые значения  $C$  со значением  $T_M$ , то  $(C_{33} - C_{24}) / (C_{33} - T_M) \approx 0,0048$ , т.е. алгоритм 24 приводит к значениям на 0,48% ближе к  $T_M$ , чем алгоритм 33.

Проведены вычисления, показавшие, что если при НП значение  $C$  намного больше, чем значение  $C$  для расписания, получаемого алгоритмом 2, обозначаемого здесь  $C_2$ , то либо образуется значительная задержка для дальнейшей оптимизации, либо при одинаковом времени выполнения достигаются худшие результаты. Так, если использовать для создания НП алгоритм 10, то  $C_{10} = 2894266$ , и далее при работе АОПО в течение минуты при частоте процессора 1,5 ГГц достигается значение  $C$  в среднем на 0,006% большее, чем  $C_2 = 2877679$ .

Отсюда следует предположение, что если при НП значение  $C$ , найденное алгоритмом, отличным от АОПО, будет намного меньше  $C_2$  и других значений, получаемых указанными 19 алгоритмами, то, напротив, время вычислений сократится.

### Алгоритм нахождения НП

С целью выявления факторов, которые могли бы оказывать влияние на минимизацию значений  $C$ , соответствующих НП, рассмотрены ранее разработанные алгоритмы. Применение второй группы алгоритмов (с 23 по 29) в [5, с. 54] часто приводило к наилучшим результатам. Эта группа была основана на выборе объекта каждый раз того, со значением  $t_i$  которого получится максимальная сумма произведений со значениями параметра  $T$  для объектов, оставшихся на данный момент невыбранными. По некоторой аналогии разработан

изложенный далее алгоритм нахождения НП (алгоритм ННП).

Алгоритм, так же как и алгоритмы 23–29, основан на выборе объектов и последующем их поочередном расположении в расписании в наилучшее время без последующего перемещения. Особенность алгоритма в том, что критерием выбора является составление предварительного расписания с наименьшим значением  $C$ . То есть, когда требуется принять решение, какой объект переместить вперед в последовательности, учитывается то, какое расписание может быть при этом получено. Аналогия состоит в том, что, так же как и ранее, учитывается результат последующих действий с невыбранными на данный момент объектами.

Алгоритм ННП использует следующие массивы:

–  $T, P$ , хранящие значения параметров соответственно  $t_i$  и  $p_i$  объектов,  $i = 1, \dots, n$ ;

–  $M(t)$ , хранящий средние значения  $t_i$  по всем объектам, кроме текущего:

$$M(t)_i = \sum_{k=1}^n \frac{t_{k,k \neq i}}{n-1};$$

–  $U$ , содержащий номера объектов;

–  $D$  размерности  $n \times m$ , элементы которого указывают, имеется ли в расписании тот или иной объект в некоторый момент времени;

–  $W$ , хранящий основную информацию расписания в виде номеров моментов времени начала импульсов;

–  $L, Y$ , хранящие расписания в виде копий массивов  $T, P, U, W$ .

В алгоритме используется функция  $Z$ , имеющая на входе два параметра. Она возвращает увеличенное на заданное вторым параметром, обозначаемым  $z_2$ , число процентов значение первого параметра. Алгоритм производит также обращение к внешней функции, вычисляющей значение  $C$  с временной сложностью  $O(n^2m)$ .

Алгоритм ННП следующий:

1) присвоить найденному значению целевой функции  $S \leftarrow 1000000000$ ;

2) в цикле по переменной  $i$ , принимающей значения с 1 по  $n-1$ :

2.1) присвоить  $S' \leftarrow 1000000000$ ;

2.2) в цикле по переменной  $j$ , принимающей значения с  $i+1$  по  $n$ :

2.2.1) в зависимости от заданного в начале пользователем номера алгоритма [5, с. 52] проверить соответствующее условие (критерий сортировки). Сравнение производится с использованием параметров  $i$ -го объекта и функции  $Z$  от параметров  $j$ -го объекта. Например, для 4-го критерия условием является истинность результата сравнения  $t_i \cdot M(t)_i < Z(t_j, 1) \cdot M(t)_j$ .

2.2.2) если условие по предыдущему пункту не выполнено, то перейти к концу цикла по  $j$  (шаг 2.3);

2.2.3) обмен  $i$ -го и  $j$ -го объектов (т.е. значений элементов  $i$  и  $j$  массивов  $T, P, U, M(t)$ );

2.2.4) в цикле по переменной  $k$ , принимающей значения с  $i+1$  по  $n-1$ :

2.2.4.1) в цикле по переменной  $h$ , принимающей значения с  $k+1$  по  $n$ :

2.2.4.1.1) если  $h \neq j$  и  $k \neq j$ , то в зависимости от номера алгоритма [5] проверить условие (критерий сортировки), сравнивая параметры объектов  $k$ -го и  $h$ -го, без использования функции  $Z$ . Если это условие выполняется, то выполнить обмен объектов  $k$ -го с  $h$ -м;

2.2.4.2) конец цикла по  $h$ ;

2.2.5) конец цикла по  $k$ ;

2.2.6) комментарий: на последующих шагах (до 2.2.10) производится построение расписания;

2.2.7) обнулить все элементы массива с  $D_{1,1}$  по  $D_{n,m}$ ;

2.2.8) первый объект разместить в первый момент времени следующим образом:

2.2.8.1) присвоить всем элементам массива с  $D_{1,1}$  по  $D_{1,p_1}$  значение 1;

2.2.8.2) присвоить  $W_1 \leftarrow 1$ ;

2.2.9) цикл по переменной  $k$ , принимающей значения с 2 по  $n$ . В этом цикле для  $k$ -го объекта находится наилучший начальный момент коммутации. Для этого выполнить следующее:

2.2.9.1) присвоить  $S'' \leftarrow 1000000000$ ;  $q \leftarrow 1$ ;

2.2.9.2) цикл по переменной  $g$ , принимающей значения с 1 по  $m$ :

2.2.9.2.1) присвоить  $W_k \leftarrow g$ ;

2.2.9.2.2) обнулить элементы массива с  $D_{k,1}$  по  $D_{k,m}$ ;

2.2.9.2.3) в цикле по переменной  $h$ , принимающей значения с  $W_k$  по  $W_k + p_k - 1$ , если  $h \leq m$ , то присвоить  $D_{k,h} \leftarrow 1$ , иначе присвоить  $D_{k,h-m} \leftarrow 1$ ;

2.2.9.2.4) присвоить  $S^{(3)} \leftarrow 0$ ;

2.2.9.2.5) в цикле по переменной  $h$ , принимающей значения с 1 по  $k-1$ , во вложенном цикле по переменной  $f$  с 1 по  $m$ , если  $D_{k,f} > 0$  и  $D_{h,f} > 0$ , то увеличить  $S^{(3)}$  на  $t_k \cdot t_h$ ;

2.2.9.2.6) если  $S'' > S^{(3)}$ , то присвоить  $S'' \leftarrow S^{(3)}$  и  $q \leftarrow W_k$ ;

2.2.9.3) конец цикла по  $g$ ;

2.2.9.4) помещение объекта  $k$  в наилучшую позицию следующим образом:

2.2.9.4.1) присвоить найденное значение начальному моменту:  $W_k \leftarrow q$ ;

2.2.9.4.2) обнулить элементы массива с  $D_{k,1}$  по  $D_{k,m}$ ;

2.2.9.4.3) в цикле по  $h$  с  $q$  по  $q + p_k - 1$ , если  $h \leq m$ , то присвоить  $D_{k,h} \leftarrow 1$ , иначе  $D_{k,h-m} \leftarrow 1$ ;

- 2.2.10) конец цикла по  $k$ ;
  - 2.2.11) вычислить значение  $C$ , используя внешнюю функцию;
  - 2.2.12) если  $C$  стало меньше, чем было найдено ранее для данного  $i$  (т.е.,  $C < S'$ ), то запомнить это расписание (массивы  $T, P, U, W$ ) в  $L$  и присвоить  $S' \leftarrow C$ ;
  - 2.2.13) иначе обратно обменять объекты  $i$  и  $j$ ;
  - 2.3) конец цикла по  $j$ ;
  - 2.4) если было получено более оптимальное расписание, чем раньше (т.е.,  $S' < S$ ), то запомнить его (т.е., присвоить  $Y \leftarrow L$  и  $S \leftarrow S'$ );
  - 3) конец цикла по  $i$ ;
  - 4) наилучшее полученное расписание находится в  $Y$ , значение целевой функции в  $S$ . Конец.
- Данный алгоритм в процессе построения множества расписаний выбирает наилучшее из них и запоминает соответствующую НП. Такое расписание не обязательно получается в конце работы данного алгоритма, оно может быть найдено на некотором этапе. Если использовать НП, получаемую в конце процесса упорядочения, она часто приводит к расписанию худшему, чем найденные ранее. Таким образом, сам процесс построения множества промежуточных расписаний играет роль метода статистических испытаний (Монте-Карло), однако

Указанные параметры являются оптимальными для большинства других файлов данных с некоторыми исключениями. Так, для файла  $f3$  оптимальное значение  $z_2 = 10\%$ . Если в исходных данных все  $t_i$  равны, то алгоритм работает с критерием 4 намного дольше – 70 с. Поэтому в таком случае используется критерий 2, при этом время работы 18 с с тем же результатом. Таким образом, алгоритм НП работает разное время в зависимости от значений исходных данных. При  $n = 40$  и  $z_2 \in [10\%, 20\%]$  это время составляет от 16 до 70 с.

При применении АОПО к получаемым НП для разных входных данных использованы оптимальные параметры [4]. Проведены серии по 7 экспериментов, каждый длительностью 1 мин, результаты усреднены, округлены и представлены в таблице, где:

- $C_{\text{ННП}}$  – значение  $C$ , полученное с помощью алгоритма НП;
- $C_{\text{min } 19}$  – минимальное из значений  $C$ , полученных 19 алгоритмами;
- $C_{\text{АОПО}}$  – значение  $C$ , полученное с помощью АОПО без применения алгоритма НП;
- $C_{\text{ННП+АОПО}}$  – значение  $C$ , полученное АОПО с применением алгоритма НП;
- $B$  – улучшение  $C$  при добавлении алгоритма НП к АОПО в контексте приближения к  $T_M$ .

Итоговые результаты вычислений для различных файлов

Файл	$C_{\text{ННП}}$	$C_{\text{min } 19}$	$C_{\text{АОПО}}$	$C_{\text{ННП+АОПО}}$	$T_M$ (оценочно)	$B, \%$
$f1$	2877282	2877453	2877265	2877256	2877210	16,3
$f2$	44249611	44249630	44249465	44248775	44248300	59,2
$f3$	116133601	116148720	116132676	116132437	116128000	5,1
$f4$	33981298	33984551	33977655	33977647	33972900	0,2
$f5$	1599457	1599590	1599448	1599435	1599413	37,1
$f7$	47792884	47796166	47792533	47792335	47792060	41,9
$in28$	1247544	1248327	1247294	1247082	1246800	43,0

при суженной области возможных значений. Применение функции  $Z$  вместо использования собственно значений параметров увеличивает количество таких испытаний, увеличивая число перестановок объектов. Это увеличивает вероятность нахождения лучшего расписания.

#### Результаты исследования и их обсуждение

Предлагаемый алгоритм реализован на языке Object Pascal. Анализ проведенных с файлом данных  $f1$  вычислений показывает, что оптимальными параметрами, которые можно задать для алгоритма НП, являются: номер критерия – 4,  $z_2 = 20\%$ .

Из таблицы следует, что алгоритм НП обычно приводит к НП с меньшим значением  $C$ , чем другие алгоритмы, работающие небольшое время. Указанные 19 алгоритмов в совокупности работают около 4 с, а алгоритм НП менее 20 с. Несмотря на значительную разницу, это во много раз меньше, чем 20 мин, поэтому алгоритм НП безболезненно может быть использован.

Как следует из последнего столбца таблицы, во всех случаях имеется улучшение, следовательно, высказанное ранее предположение подтверждается. В среднем  $B$  составляет 29% со среднеквадратичным отклонением 22%. Для некоторых файлов  $B$  незначительно. Выявлена следующая

связь  $B$  с особенностями таких файлов: если диапазоны изменения значений параметров узкие относительно средних значений, то  $B$  незначительно. Например, в файле  $f4$ :  $t_i \in [40, 44]$ ,  $p_i \in [40, 59]$ ,  $i = 1, \dots, n$ . То есть значения  $t_i = 42 \pm 2$ ,  $p_i \approx 50 \pm 9$ . Обозначим как средние значения  $t_{cp}$  и  $p_{cp}$  числа до символа «±», и как отклонения  $\sigma_t$  и  $\sigma_p$  числа после этого символа, т.е.  $t_i = t_{cp} \pm \sigma_t$ ,  $p_i = p_{cp} \pm \sigma_p$ . Отношения  $\sigma_t/t_{cp}$  и  $\sigma_p/p_{cp}$  для файлов  $f3$  и  $f4$  составляют от 0,05 до 0,19, а для остальных файлов, где наблюдается значительное улучшение, эти отношения принимают значения от 0,55 до 1.

Проведены также вычисления при использовании неоптимальных параметров АОПО. Они показывают, что, несмотря на то что значение  $C$  становится несколько больше, чем при использовании оптимальных параметров, добавление алгоритма ННП позволяет при дальнейшем применении АОПО за одно и то же время получить результаты ближе к  $T_{MP}$  чем в случае без использования алгоритма ННП.

Получение за одинаковое время меньших значений  $C$  означает, что близкие к глобальному минимуму значения  $C$  достигаются при применении алгоритма ННП за значительно меньшее время, чем без его применения. Это справедливо при использовании как оптимальных параметров АОПО, так и неоптимальных, при условии равенства значений этих параметров в обоих случаях. Результаты исследований относятся к диапазону  $n \in [28, 40]$ . Таким образом, цель работы достигнута, т.е. АОПО дополнен так, чтобы скорость его сходимости была увеличена.

### Заключение

Предлагаемый алгоритм, используемый при составлении расписания коммутаций группе потребителей при ШИМ регулирования мощности, позволяет находить на-

чальные последовательности, приводящие затем к решениям, близким к оптимальным, быстрее, чем ранее разработанными алгоритмами. Временная сложность алгоритма  $O(n^5 + n^4 m^2)$ , но так как  $n$  и  $m$  одного порядка, можно считать, что временная сложность  $O(n^4 m^2)$ . Несмотря на то что это больше, чем у ранее используемых 19 алгоритмов, например у алгоритма 2 со сложностью  $O(n^2 m^2)$ , использование АОПО с предлагаемым алгоритмом за одинаковое время приводит к более оптимальным результатам.

### Список литературы

1. Ильгачев А.Н., Абрамов А.В., Васильев А.В. Способ управления группой импульсных регуляторов напряжений (варианты) // Патент РФ № 2409829. Заявитель и патентообладатель ФГОУ ВПО «Чувашский государственный университет им. И.Н. Ульянова». 2011. Бюл. № 2. [Электронный ресурс]. URL: <http://www.freepatent.ru/patents/2409829> (дата обращения: 17.09.2020).
2. Ильгачев А.Н., Абрамов А.В., Васильев А.В. Способ группового управления импульсными регуляторами напряжения (варианты) // Патент РФ № 2403612. Заявитель и патентообладатель ФГОУ ВПО «Чувашский государственный университет им. И.Н. Ульянова». 2010. Бюл. № 31. [Электронный ресурс]. URL: <http://www.freepatent.ru/patents/2403612> (дата обращения: 17.09.2020).
3. Корнеев А.М., Суханов А.В. Исследование точности и скорости сходимости алгоритмов стохастической оптимизации функций на многомерном пространстве // Вестн. Астрахан. гос. техн. ун-та. Сер. управление, вычисл. техн. и информ. 2018. № 3. С. 26–37. [Электронный ресурс]. URL: <https://www.elibrary.ru/item.asp?id=35216895> (дата обращения: 17.09.2020).
4. Димитриев А.П. Определение оптимальных параметров для модификаций алгоритма оптимизации последовательности отбора // Современные наукоемкие технологии. 2019. № 10–1. С. 213–216.
5. Димитриев А.П. Модели и алгоритмы в системах автоматизированного перевода текста // Прикладная информатика. 2013. № 6 (48). С. 045–058. [Электронный ресурс]. URL: [http://appliedinformatics.ru/r/articles/article/index.php?article\\_id\\_4=1552](http://appliedinformatics.ru/r/articles/article/index.php?article_id_4=1552) (дата обращения: 17.09.2020).
6. Kirkpatrick S., Gelatt C.D.Jr., Vecchi M.P.Jr. Optimization by Simulated Annealing. Science. 1983. Vol. 220 (4598). P. 671–680. DOI: 10.1126/science.220.4598.671.
7. Пантелеев А.В., Скавинская Д.В. Метаэвристические алгоритмы глобальной оптимизации. М.: Вузовская книга, 2019. 332 с.