

Отчёт об алгоритмах оптимизации для проектирования гибридной энергосистемы

Израилева Арина

29.11.2022

Содержание

Abstract	1
1 Постановка задачи	2
2 Обзор имеющихся методов	2
2.1 Общий алгоритм многокритериальной оптимизации	2
2.2 Выбор моделей оптимизации	3
2.2.1 Метод роя частиц (Particle Swarm Optimization)	4
2.2.2 Алгоритм Джая с модификацией алгоритма поиска вектора изменения положения частиц	4
2.2.3 Алгоритм Рао	6
2.2.4 Алгоритм многокритериальной оптимизации Multi-verse	6
3 Приложение выбранных моделей оптимизации к задаче проектирование гибридной энергосистемы	7
Список литературы	8

Abstract

Далее будут рассмотрены модели оптимизации для повышения эффективности при проектировании гибридной энергосистемы. В разделе 1 описана постановка задачи оптимизации, а также обоснован выбор класса генетических оптимизационных алгоритмов; математические модели оптимизации для решения данной задачи описаны в разделе 2; в разделе 3 представлено описание выбранных методов с наилучшими показателями скорости работы и обоснование целесообразности их применения для проектирования гибридной энергосистемы.

1 Постановка задачи

Спроектированная энергосистема должна быть надежной (иметь высокий вероятностный показатель среднего времени безотказной работы), полностью обеспечивать потребности энергопотребления, снижая при этом использование дизельных генераторов, а также должна требовать адекватных первоначальных вложений. Поэтому при проектировании гибридной энергосистемы с использованием возобновляемых источников энергии (ВИЭ), то есть при выборе её компонентов из ряда существующих и определения их количества, требуется применение алгоритмов оптимизации, целевыми параметрами которой являются: средняя расчётная себестоимость производства электроэнергии на протяжении (Levelised Cost of Energy, далее LCOE), уровень выброса CO_2 , вероятность перебоя и потери электроснабжения (Loss of Power Supply Probability, далее LPSP). Выбранные параметры составляют недискретную (разрывную), многоэкстремальную целевую вектор-функцию, оптимизация которой является задачей выбираемого алгоритма. Данная задача относится к классу стохастических алгоритмов, в частности к мульти-целевым (многокритериальным) моделям оптимизации с ограничивающими параметрами с вероятностью нахождения нескольких локальных экстремумов (constrained multi-object multi-modal optimisation) [5]. Существующие предметные исследования [1] показывают, что наибольшую эффективность среди данных моделей имеют эволюционные (генетические) алгоритмы.

2 Обзор имеющихся методов

2.1 Общий алгоритм многокритериальной оптимизации

Генетические алгоритмы оптимизируют целевую функцию, взаимодействуя с нормированной формой критериев задачи. Решение принимается на основании одновременного анализа нескольких точек пространства целей. Задача мульти-целевой оптимизации определяется целевой функцией ($f(x)$), для которой необходимо найти экстремум:

$$f(x) = (f_1(x), f_2(x), \dots, f_N(x))^T \in F \quad (1)$$

$$x = (x_1, x_2, \dots, x_M)^T \in X \quad (2)$$

$$X = \begin{cases} a_i(x) \leq 0, i \in [1, k] \\ b_j(x) = 0, i \in [k + 1, L] \end{cases} \quad (3)$$

где $f(x)$ является вектор-функцией набора из N функций, зависящих от M переменных из пространства решений X размерности M . Переменные принадлежат пространству, задаваемому системой L неравенств (равенств), ограничивающий M переменных. Множество значений F целевых функций содержится в критериальном пространстве (пространстве целей) [6].

Каждый нормированный элемент критериального пространства представлен точкой (частицей); частицы объединены в популяции (вселенные). Размеры и количество популяций варьируются в зависимости от выбранного для реализации генетического алгоритма.

Однако, подход к поиску оптимального решения, представленный на рис. 1 является общим для любых генетических алгоритмов.

Критериями остановки алгоритма могут являться:

1. Постоянство (повторение) глобального лучшего решения;
2. Достижение заданного значения целевой функции (это возможно только если лучшее решение известно заранее);
3. Достижение определенного количества итераций;
4. Схождение всех частиц всех роев в одну точку.
5. Схождение всех частиц всех роев в одну точку.

Критерий остановки по повторению лучшего решения (первый критерий) является оптимальным в контексте задачи проектирования энергосистемы [2] потому что позволяет за оптимальное количество итераций находить гарантированно лучшее решение.



Рис. 1: Блок-схема, реализующая общий алгоритм модели генетической оптимизации

2.2 Выбор моделей оптимизации

Были рассмотрены следующие модели оптимизации: метод роя частиц (Particle Swarm Optimization), алгоритм Jaya, алгоритм Rao, TOPSIS и multi-verse. Алгоритмы имеют схожий принцип работы, отличаются правилами взаимодействия частиц и методы образования популяций.

2.2.1 Метод роя частиц (Particle Swarm Optimization)

Метод роя частиц (SI PSO) был выбран как высоко-эффективная и в то же время относительно простая модель генетической оптимизации. Поиск экстремума функции ведется посредством организованного взаимодействия частиц (агентов) и их популяций. Каждая частица, проходя по различным значениям функции (решениям задачи оптимизации), запоминает свое лучшее положение. Из всех лучших решений частиц выбирается лучшее решение для всей популяции (роя). Затем для всех роев (мультироя) выбирается лучшее решение. Во время работы SI PSO каждая частица изменяет свое положение каждую итерацию алгоритма. Изменение положения частицы определяется ее скоростью. Скорость изменения положения для каждой частицы определяется по формуле:

$$\vartheta_{(k+1)} = \alpha \cdot \vartheta_k + r \cdot \beta \cdot (p_{particle} - p_{current}) + r \cdot \gamma \cdot (p_{swarm} - p_{current}) + r \cdot \delta \cdot (p_{multiswarm} - p_{current}) \quad (4)$$

где

ϑ	- предыдущая скорость частицы	α	- параметр инерции
$p_{current}$	- текущее положение частицы	β	- когнитивный вес
$p_{particle}$	- лучшее положение данной частицы	γ	- социальный вес
p_{swarm}	- лучшее положение популяции данной частицы (роя)	δ	- глобальный вес
$p_{multiswarm}$	- лучшее положение среди всех популяций (мультироя)	k	- номер итерации

Коэффициент инерции α предотвращает резкий рост модуля скорости частицы и регулирует влияние скорости частицы на предыдущем шаге на новую скорость. Малая величина коэффициента инерции α будет препятствовать исследованию окрестностей частицы в поисках лучшего решения. Однако, большой коэффициент α не позволит частице прекратить поиски нового решения и перейти к улучшению в окрестностях найденного лучшего решения. Поэтому лучшим решением будет постепенное уменьшение коэффициента инерции α с ростом числа итераций алгоритма [7]. В таком случае коэффициент инерции определяется формулой:

$$\alpha_k = (\alpha_{min} - \alpha_{max}) \cdot \frac{k}{N_{it}} + \alpha_{max} \quad (5)$$

где

α_{min}	- минимальная граница, принятая за 0.4
α_{max}	- максимальная граница, принятая за 0.9
N_{it}	- количество итераций, необходимое для поиска оптимального решения

Коэффициенты β (когнитивный вес), γ (социальный вес) и δ (глобальный вес) определяют склонение частицы к своему лучшему решению, лучшему решению своей популяции и мультироя, и равны 1.49618, 1.49618 и 0.3645 соответственно [8].

2.2.2 Алгоритм Джая с модификацией алгоритма поиска вектора изменения положения частиц

Алгоритм Джая не имеет параметров управления (весов), зависящих от конкретной прикладной задачи. Изменеию подлежат только общие для всех задач параметры, такие как количество частиц

в популяции и количество повторений до момента остановки. Так же алгоритм Джая принимает решение на основании как лучшего за итеррацию значения целевой функции, так и худшего.

Решения, найденные каждой частицей изменяются по формуле [2]:

$$\vartheta_{(k+1)} = f_{current} + r_1 \cdot (f_{best} - |f_{current}|) - r_2 \cdot (f_{worst} - |f_{current}|) \quad (6)$$

где

ϑ	- новое частицы	$f_{current}$	- текущее решение одной частицы
f_{best}	- положение лучшей частицы на шаге k	f_{worst}	- положение худшей частицы на шаге k
r_1, r_2	- случайные величины в промежутке $[0, 1]$	k	- номер итерации

Алгоритм Джая с модификацией алгоритма поиска вектора изменения положения частиц был выбран в качестве усовершенствования SI PSO, так как вместо роя частиц, объединяющего несколько популяций, использует мульти-командный подход. При таком подходе все команды являются дубликатом исходной популяции, начинают свое движение из одной точки, но внутри команды частицы приходят к лучшему решению и изменяют свое положение по шести различным формулам. Такой способ изменения решений делает невозможной остановку на локальном минимуме, то есть предотвращает необходимость в увеличении количества итерраций для поиска оптимального решения.

Первым уравнением является оригинальное уравнение изменения положения частиц алгоритма Джая 6. Второе уравнение так же использует уравнение 6, вместо случайных чисел r_1, r_2 используются хаотические случайные коэффициенты, изменяющиеся по формуле:

$$c_{k+1} = 4 \cdot c_k \cdot (1 - c_k) \quad (7)$$

$$\vartheta_{(k+1)} = f_{current} + c_k \cdot (f_{best} - |f_{current}|) - c_k \cdot (f_{worst} - |f_{current}|) \quad (8)$$

Третье уравнение, используя подход опозиционного обучения нейронных сетей, позволяет выбирать коэффициент в промежутке среднего между верхней и нижней границами ϑ (устанавливаются простыми численными методами) и отраженным значением ϑ :

$$\vartheta_{opposition-based} = rand(a, b) \quad (9)$$

$$a = (lb_{\vartheta} + ub_{\vartheta})/2 \quad (10)$$

$$b = lb_{\vartheta} + ub_{\vartheta} - \vartheta \quad (11)$$

Четвертое 12 уравнение основано на 6, но вместо лучшего положения частицы f_{best} использует одно из трех лучших ($f_{randombest}$) положений частиц команды на шаге k :

$$\vartheta_{(k+1)} = f_{current} + r_1 \cdot (f_{randombest} - |f_{current}|) - r_2 \cdot (f_{worst} - |f_{current}|) \quad (12)$$

Пятое и шестое уравнения движения определяются сдвигом лучшего положения частицы на коэффициент, полученный из c_k или случайной величины от 0 до 1 (r_3, r_4, r_5):

$$\vartheta_{(k+1)} = f_{best} + r_3 \cdot (2 \cdot c_k - 1) \quad (13)$$

$$\vartheta_{(k+1)} = f_{best} + r_4 \cdot (2 \cdot r_5 - 1) \quad (14)$$

После каждой итеррации уравнение для команды с худшим показателем лучшей позиции меняется случайным образом.

2.2.3 Алгоритм Рао

Алгоритм рао был выбран, так как снижает количество вычислений, путем повышения случайности траектории частиц в рамках одной популяции, без необходимости использования команд или популяций. Уравнение изменения положения частицы задается формулой:

$$\vartheta_{k+1} = f_{current} + r_1 \cdot (f_{best} - |f_{worst}| + r_2 \cdot (|f_{current} \text{ or } f_{selected}| - (f_{current} \text{ or } f_{selected}))) \quad (15)$$

где $f_{selected}$ - значение, выбранное из всех значений частиц популяции на текущем шаге.

2.2.4 Алгоритм многокритериальной оптимизации Multi-verse

Алгоритм multi-verse лучше аналогов разделяет поиск по всему целевому пространству и улучшение вблизи экстремума решения целевой функции. Каждое решение отождествляется со вселенной, а каждая переменная с объектом внутри вселенной. В отличие от предыдущих алгоритмов, частицы, объединенные одной вселенной (популяцией), могут перемещаться с помощью концепций белых и черных дыр, возникающих при скоплении или рассредоточении частиц одной вселенной относительно лучшего решения.

$$x_i^j = \begin{cases} \begin{cases} X_j + TDR \cdot ((ub - lb) \cdot r + lb), r_2 < 0.5 \\ X_j - TDR \cdot ((ub - lb) \cdot r + lb), r_2 \geq 0.5 \end{cases} & , r_1 < WEP \\ x_i^j, r_1 \geq WEP \end{cases} \quad (16)$$

где

- x_i^j - j -ая переменная i -ой вселенной
- X_j - j -ая переменная вселенной с лучшим значением целевой функции
- TDR - коэффициент расстояния перемещения между вселенными (travelling distance rate)
- WEP - коэффициент вероятности перехода решения между вселенными (warmhole's existence probability)
- ub - верхнее значение j -ой переменной из множества X
- lb - нижнее значение j -ой переменной из множества X
- r_1, r_2 - случайные величины от 0 до 1

Адаптивные коэффициенты TDR и WEP изменяются по следующим формулам:

$$TDR = 1 - \frac{k^{1 \setminus p}}{K^{1 \setminus p}} \quad (17)$$

$$WEP = min + k \cdot \left(\frac{max - min}{K} \right) \quad (18)$$

где коэффициент p влияет на скорость выполнения алгоритма (в контексте решаемой задачи, коэффициент p должен быть максимальным),

- max - минимум WEP , равный 0.2 min - максимум WEP , равный 1
- K - конечное число итераций k - текущая итерация

3 Приложение выбранных моделей оптимизации к задаче проектирование гибридной энергосистемы

Для оптимизации построения гибридной электростанции были выбраны следующие параметры:

1. LCOE),
2. уровень выброса CO_2 ,
3. LPSP

Для выбора лучшего оптимизирующего алгоритма была составлена математическая модель, позволяющая измерять показатели оптимизируемых параметров на основании почасовых данных генерации и потребления:

$$LCOE = \frac{c \cdot CRF + m}{\sum_{h=1}^{8760} p(h)} \quad (19)$$

c	- цена топлива	CRF	- взнос на амортизацию единицы
m	- стоимость эксплуатации и обслуживания	$p(h)$	- почасовая пропорция работы системы

Выброс CO_2 оптимизируется как коэффициент отношения энергии, выработанной дизельными генераторами к энергии, сгенерированной ВИЭ:

$$CO_2emission = (1 - \frac{\sum_{disel}}{\sum}) \cdot 100 \quad (20)$$

Коэффициент надежности системы рассчитывается по вероятностной формуле для выхода из строя каждого компонента системы:

$$LPSP = \frac{\sum(P_{load} - P_{wind} - P_{solar} + P_{bat})}{\sum P_{load}} \quad (21)$$

Переменными для алгоритма оптимизации являются количество и мощность ВИЭ. Входными данными алгоритма являются параметры ВИЭ, из которых может быть спроектирована гибридная энергосистемы. Стратегия работы энергосистемы заключается в использовании и накоплении всей электроэнергии, полученной от ВИЭ. Дизельная генерация используется только при недостатке энергии ВИЭ для обеспечения потребления. Алгоритм применения модели оптимизации представлен на блок-схеме Рис 2.

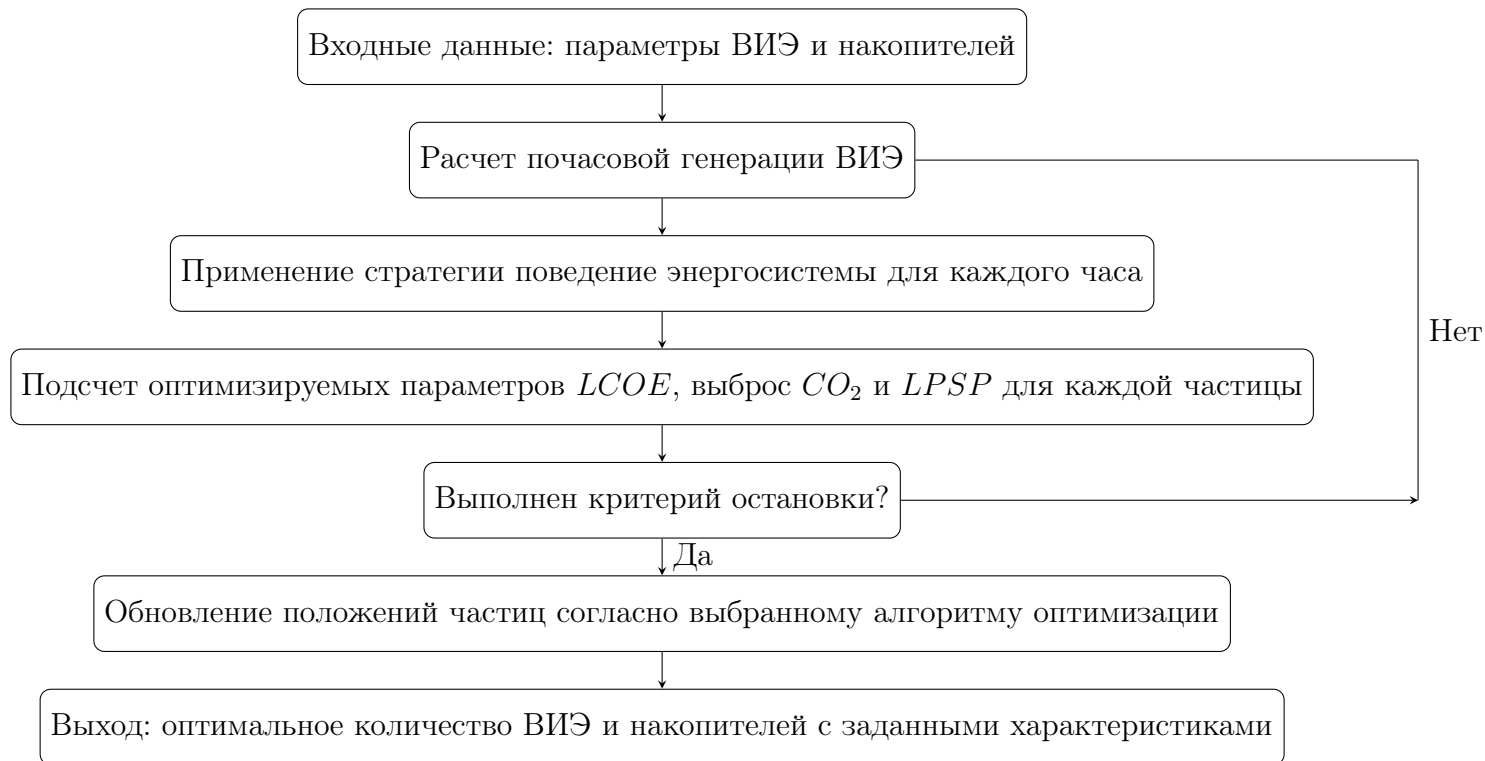


Рис. 2: Блок-схема, реализующая применение моделей оптимизации к задаче проектирования гибридной энергосистемы

Список литературы

- [1] A.M. Hemeida, M.H. El-Ahmar, A.M. El-Sayed, Hany M. Hasanien and T. Senjyu Ain Shams Eng. J. (2020): Optimum design of hybrid wind/PV energy system for remote area
- [2] V. R. Ravipudi, H. S. Keesari (2022): Design Optimization of Renewable Energy Systems Using advanced Optimization Algorithms
- [3] R. Lavikka, H. U. Rehman, F. Reda, A. S. Kazi (2022) Positive Energy Buildings Concepts, Challenges and Opportunities
- [4] И. А. Прохорова, С. С. Аверьянова (2020) Применение генетических алгоритмов при решении многокритериальных задач
- [5] D. Simon (2013): Evolutionary optimization algorithms Biologically-Inspired and Population-Based Approaches to Computer Intelligence
- [6] Е.А. Сопов, С. А. Сопов (2011): Вероятностный генетический алгоритм решения сложных задач многокритериальной оптимизации с адаптивной мутацией и прогнозом множества Парето
- [7] X. Hu, R. Eberhart (2022): Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization
- [8] А. П. Карпенко (2013): Ко-эволюционный алгоритм глобальной оптимизации на основе алгоритма роя частиц

- [9] R.V. Rao, H. S. Keesari (2018): Multi-team perturbation guiding Jaya algorithm for optimization of wind farm layout
- [10] S. Mirjalili, S. M. Mirjalili, A. Hatamlou (2015): Multi-Verse Optimizer: a nature-inspired algorithm for global optimization