

ВПРОВАДЖЕННЯ ГЕНЕТИЧНИХ АЛГОРИТМІВ У ФЕС

Макогон В.К., студентка, viktoriia.makohon.mext@donntu.edu.ua

Колларов О.Ю., к.т.н, доцент, зав.каф. ЕлІн,

oleksandr.kollarov@donntu.edu.ua

Остренко Д.О., dmytro.ostrenko@gmail.com

ДВНЗ «Донецький національний технічний університет»,

м. Луцьк, Україна

Метою даної наукової роботи є вивчення та розробка алгоритмів оптимізації ефективності фотоелектричної станції (ФЕС) за допомогою генетичних алгоритмів (ГА). Основний акцент буде зроблено на побудові мат. моделі ФЕС та розробці паралельних генетичних алгоритмів для підвищення вихідної енергії та оптимізації точки максимальної потужності (MPPT).

Об'єктом дослідження є фотоелектрична станція (ФЕС), в тому числі фізичні та математичні аспекти її функціонування та оптимізації. Дослідження передбачає використання ГА для покращення управління потужністю ФЕС і забезпечення точки максимальної потужності за різних умов експлуатації.

Ця робота покликана визначити ефективність та можливості використання ГА для оптимізації роботи фотоелектричних станцій, розробити математичну модель та реалізувати алгоритми в середовищі програмування Python.

Аналіз стану питань: на сучасному етапі розвитку відновлювальних джерел енергії, ФЕС займають ключове місце в стратегіях зменшення залежності від традиційних джерел. Використання генетичних алгоритмів (ГА) для оптимізації роботи ФЕС є перспективним напрямом, що відкриває нові можливості в підвищенні ефективності та покращення енергетичної складової станцій.

Так у [3] вимірювання проводяться за двома умовами: у лабораторії під штучним освітленням та безпосередньо природних умовах сонячного світла. Внутрішні параметри двох фотоелектричних елементів, SC_1 і SC_2 , а також однієї ФП, визначаються двома методами: генетичним алгоритмом - стохастичним методом та п'ятьма параметрами - аналітичним методом. Порівняння результатів, отриманих двома методами, за допомогою графіків, абсолютних похибок та середньо квадратичної помилки, показує, що генетичний алгоритм точніше визначає результати для пар (V, I) . Таким чином, генетичний алгоритм може бути використаний для точного та ефективного визначення параметрів фотоелектричних елементів та панелей.

Отже, на міжнародному рівні існує зростаючий інтерес до використання ГА в області фотоелектричних систем. Великі енергетичні компанії та наукові групи активно вивчають можливості оптимізації роботи ФЕС з ме-

тою підвищення ефективності та екологічної стійкості. Спостерігається зацікавленість до використання паралельних обчислень для швидкого впровадження оптимальних рішень в реальному часі.

В енергетиці України також активно розвивається ВДЕ і зростає інтерес до використання ШІ для підвищення продуктивності ФЕС. Однак, наразі ця галузь лишається менш дослідженою, існуючі практичні проекти переважно фокусуються на традиційних методах управління.

Актуальність проблеми: зростання кількості ФЕС по всьому світу ставить перед галуззю завдання підвищення ефективності та надійності функціонування. Актуалізація питання використання ГА полягає в тому, що традиційні методи оптимізації можуть бути обмежені в умовах змінних параметрів та оточуючого середовища.

Впровадження ГА в управління ФЕС може визначити новий етап в розвитку відновлювальної енергетики, забезпечивши оптимальне використання сонячної енергії. Потенційні переваги включають збільшення виходу енергії, зменшення витрат на утримання та підвищення стабільності систем.

Виклики та рішення при впровадженні ГА для оптимізації ФЕС:

Виклик 1: моделювання реального середовища ФЕС. Реальні умови експлуатації ФЕС містять багато змінних, таких як змінна інтенсивність світла, температурні коливання, атмосферні умови, інші зовнішні впливи. Рішення: розробка точних математичних моделей, які враховують реальні умови, та їх регулярне оновлення з урахуванням нових даних.

Виклик 2: потреба в великій кількості обчислень. Використання ГА може вимагати значних обчислювальних ресурсів, особливо при оптимізації в реальному часі. Рішення: використання паралельних обчислень, оптимізація алгоритмів для швидкої збіжності та використання високоефективних обчислювальних ресурсів.

Виклик 3: потреба в постійному моніторингу та оновленні алгоритмів. Умови роботи ФЕС можуть змінюватися з часом, і алгоритми ГА повинні адаптуватися до нових умов для збереження ефективності. Рішення: розробка систем постійного моніторингу та автоматичного оновлення алгоритмів, можливо, з використанням машинного навчання.

Виклик 4: порівняння із іншими методами оптимізації. Існують інші методи оптимізації, такі як метод ШНМ та методи математичного програмування (МПК). Рішення: проведення комплексного порівняльного аналізу, де розглядаються витрати на реалізацію, час оптимізації та досягнуті результати. Виділення переваг ГА у випадках великої кількості змінних.

Виклик 5: управління непередбаченими аномаліями. Рішення: впровадження систем діагностики та резервування, які дозволяють виявляти та виправляти аномалії в реальному часі.

На рис. 1 зображений обраний метод гібридизації генетичного алгоритму (ГА) із методом Тагучі, що є поширеним при роботі саме із ФЕС, бо

використовує переваги ГА для глобальної числової оптимізації та додає досвід самого методу, щоб мати вищий рівень ефективності порівняно із стандартними ГА. Якщо коротко описувати даний метод, то ГА перенавчається до тих пір поки модель не набуде властивостей ФЕС, із певною допустимою помилкою.

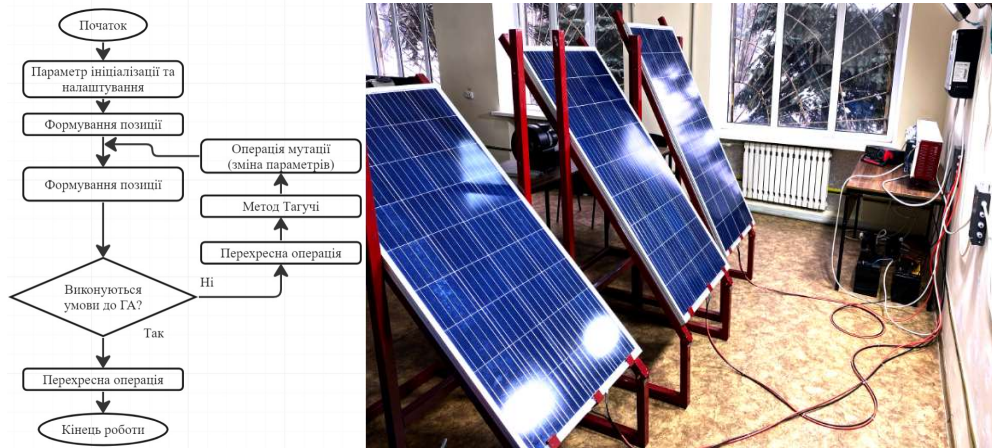


Рисунок 1 – Робота алгоритма ГА та ФЕС до якої він застосовується

На рис. 2-3 наведено код створення ГА для ФЕС побудований у Python.

```

1 import random
2 import numpy as np
3 import matplotlib.pyplot as plt
4
5 class SolarPanel:
6     def __init__(self, n, vt, is_, il):
7         self.n = n
8         self.vt = vt
9         self.is_ = is_
10        self.il = il
11    def calculate_current(self, voltage):
12        return self.il - self.is_ * ((np.exp(voltage / (self.n * self.vt)) - 1))
13    def calculate_power(self, voltage):
14        current = self.calculate_current(voltage)
15        return voltage * current
16    def calculate_volt_amp_curve(self, v_min, v_max):
17        voltages = np.linspace(v_min, v_max, 100)
18        currents = self.calculate_current(voltages)
19        powers = self.calculate_power(voltages)
20        return voltages, currents, powers
21    def print_info(self, voltages, powers):
22        max_power_idx = np.argmax(powers)
23        max_power_voltage = voltages[max_power_idx]
24        print(f"Напруга при максимальній потужності: {max_power_voltage:.2f} В")
25    def plot_volt_amp_curve(self, voltages, currents, powers, label):
26        fig, ax = plt.subplots(figsize=(8, 6))
27        ax.plot(voltages, powers, label=label)
28        ax.set_title('Вольт-амперна характеристика фотоелектричної панелі')
29        ax.set_xlabel('Напруга (В)')
30        ax.set_ylabel('Потужність (Вт)')
31        ax.legend()
32        plt.show()
33
34 class GeneticAlgorithm:
35     def __init__(self, panel_capacity, battery_capacity, num_panels, num_batteries):
36         self.panel_capacity = panel_capacity
37         self.battery_capacity = battery_capacity
38         self.num_panels = num_panels
39         self.num_batteries = num_batteries
40     def fitness_function(self, panels, batteries):
41         total_power = sum([self.panel_capacity * \
42             panels[i] for i in range(self.num_panels)])
43         total_capacity = sum([self.battery_capacity * \
44             batteries[i] for i in range(self.num_batteries)])
45         return min(total_power, total_capacity)
46     def generate_population(self, population_size):
47         return [[random.randint(0, 10) for _ in range(self.num_panels \
48             + self.num_batteries)] for _ in range(population_size)]
49     def selection(self, population, scores, num_parents):
50         selected_indices = sorted(range(len(scores)), \
51             key=lambda k: scores[k], reverse=True)[:num_parents]
52         return [population[i] for i in selected_indices]
53     def crossover(self, parent1, parent2):
54         crossover_point = random.randint(1, len(parent1) - 2)
55         child = parent1[:crossover_point] + parent2[crossover_point:]
56         return child
57     def mutate(self, child):
58         mutation_point = random.randint(0, len(child) - 1)
59         child[mutation_point] += random.randint(-1, 1)
60         return child
61     def run_genetic_algorithm(self, population_size, generations):
62         population = self.generate_population(population_size)
63         for generation in range(generations):
64             scores = [self.fitness_function(panels[:self.num_panels], \
65                 panels[self.num_panels:]) for panels in population]
66             parents = self.selection(population, scores, population_size // 2)
67             offspring = []
68             while len(offspring) < population_size:
69                 parent1, parent2 = random.choice(parents), random.choice(parents)
70                 child = self.crossover(parent1, parent2)
71                 if random.random() < 0.1: # ймовірність мутації
72                     child = self.mutate(child)
73                 offspring.append(child)

```

Рисунок 2 – Створення мат. моделі ФЕС та ГА у середовищі Python

Як було зазначено в одному із викликів вище, важливим є врахування усіх параметрів ФЕС, тобто це характеристики паспортні панелей, інвертору та акумуляторної станції, це як мінімум, що було зроблено на рис. 3.



Рисунок 3 – Візуалізація отриманих даних за ВАХ

Варто відзначити, що поділ коду на класи має на меті виконати налаштування спочатку роботи окремо ФЕС і тільки після цього переходити до роботи із ГА.

Висновки: Впровадження генетичних алгоритмів у оптимізацію ФЕС є перспективним, але вимагає комплексного підходу до розв'язання технічних, обчислювальних та адаптаційних викликів. Через поєднання точних моделей, великої обчислювальної потужності та систем постійного моніторингу можна досягти ефективного використання сонячної енергії. Крім того, порівняльний аналіз із конкуруючими методами дозволив визначити переваги ГА у конкретних умовах ФЕС.

Література

- [1] Колларов О. Ю., Кардаш Д. О., "Розв'язання задач оптимізації в енергетиці за допомогою генетичного алгоритму", збірник «Наукові праці ДонНТУ», серія: Електротехніка і енергетика" №1 (28) '2023, Луцьк, с. 37-41;
- [2] Колларов О. Ю., Остренко Д. О., "Проектування та розробка фотоелектричної станції ДВНЗ «Донецького національного технічного університету.»", збірник «Наукові праці ДонНТУ», серія: Електротехніка і енергетика" №2 (25) '2021, Покровськ, с. 69-75;
- [3] Колларов О. Ю., Остренко Д. О., "Аналіз впливу температури зовнішньої середовища на роботу фотоелектричних станцій.", збірник «Наукові праці ДонНТУ», серія: Електротехніка і енергетика" №1 (28) '2023, Покровськ, с. 31-36;

Анотація

В роботі було досліджено можливість використання та отримання інформації від паралельної роботи стенду фотоелектричної станції та ГА із метою забезпечення прогнозу виробленої потужності.

Було проаналізовано та визначено алгоритм функціонування ГА, так було обрано метод Тагучі. Окрім цього, в роботі визначено основні виклики, із якими зіштовхуються інженери при проектуванні та послідовному обслуговуванні даних ФЕС. Також було наведено результати математичного моделювання, у Python. Дані результати дозволять покращити досліджувану мережу.

Ключові слова: ФЕС, інвертор, штучний інтелект, електрична мережа, генетичні алгоритми, алгоритми навчання, автоматизований моніторинг.

Abstract

The paper investigated the possibility of using and receiving information from the parallel operation of the stand of the photovoltaic station and the GA in order to provide a forecast of the generated power.

The GA functioning algorithm was analyzed and defined, and the Taguchi method was chosen. In addition, the work identifies the main challenges faced by engineers during the design and subsequent data maintenance of the power plant. The results of mathematical modeling, in Python, were also given. These results will allow to improve the investigated network.

Keywords: Photoelectric station, inverter, artificial intelligence, electrical network, genetic algorithms, learning algorithms, automated monitoring.