

УДК 519.854.2

АЛГОРИТМИ РОЙОВОГО ІНТЕЛЕКТУ ДЛЯ ЗАДАЧІ ДИНАМІЧНОГО РОЗПОДІЛУ НАВАНТАЖЕННЯ

Бабич С. О.

Національний технічний університет України “Київський політехнічний інститут імені Ігоря Сікорського”, Україна, Київ

доктор технічних наук, старший науковий співробітник, Гуляницький Л. Ф.

Інститут кібернетики імені В. М. Глушкова Національної академії наук України, Україна, Київ

Розглянуто підходи до розв'язування задачі динамічного розподілу навантаження для замкненої енергетичної системи, що складається із електростанцій та споживачів. Особливістю задачі є зміна попиту на електроенергію протягом дня. Описано перехід від задачі із обмеженнями, що характеризують фізичні та експлуатаційні характеристики електростанцій, до задачі без обмежень із використанням функцій штрафів. Використано два алгоритми ройового інтелекту, а саме оптимізація роєм частинок та алгоритм вовчої зграї. Проведено ряд експериментів із використанням розроблених алгоритмів та аналіз отриманих результатів.

Ключові слова: динамічний розподіл навантаження, алгоритм вовчої зграї, метод рою часток, ройовий інтелект, електроенергетика.

Бабич С. А., доктор технических наук, старший научный сотрудник, Гуляницкий Л. Ф. Алгоритмы роевого интеллекта для задачи динамического распределения нагрузки / Национальный технический университет Украины «Киевский политехнический

институт имени Игоря Сикорского», Институт кибернетики имени В. М. Глушкова Национальной академии наук Украины, Украина, Киев

Рассмотрены подходы к решению задачи динамического распределения нагрузки для замкнутой энергетической системы, состоящей из электростанций и потребителей. Особенностью задачи является изменение спроса на электроэнергию в течение дня. Описаны переход от задачи с ограничениями, характеризующие физические и эксплуатационные свойства электростанций, в задачи без ограничений с использованием функций штрафов. Используются два алгоритма роевого интеллекта, а именно оптимизация роем частиц и алгоритм волчьей стаи. Проведен ряд экспериментов по использованию разработанных алгоритмов и анализ полученных результатов.

Ключевые слова: динамическое распределение нагрузки, алгоритм волчьей стаи, метод роя частиц, роевой интеллект, электроэнергетика.

S. Babych, Dr.Sc. (Technology), Senior Researcher, L. Hulianytskyi Swarm intelligence algorithms for solving dynamic load dispatch problem / National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", V. M. Glushkov Institute of Cybernetics of National Academy of Sciences of Ukraine, Ukraine, Kyiv

Approaches to solving the dynamic load dispatch problem for an isolated power system consisting of power stations and consumers are considered. The distinction of this task is moving during day power demand. The transition from a problem with constraints characterizing the physical and operational properties of power generators to a problem without ones but with the use of fines functions is described. Two swarm intelligence algorithms were used: particle swarm optimization algorithm

and grey wolf optimizer algorithm. A set of experiments are conducted on the use of developed algorithms and analysis of the obtained results is given.

Keywords: dynamic load dispatch, grey wolf optimizer, particle swarm optimization, swarm intelligence, electric power industry.

Вступ. У перспективній конкуруючій галузі електроенергетики задачі економічного розподілу навантаження (ЕРН) та динамічного розподілу навантаження (ДРН) описують процес розподілу робочої потужності між електростанціями, враховуючи всі фізичні та експлуатаційні обмеження. Особливістю електроенергетичної промисловості є відсутність можливості зберігати виготовлену продукцію – вся невикористана електроенергія буде втрачена. Через це та високу вартість палива, необхідного для роботи електростанції, ефективний розподіл виробництва може призвести до суттєвої економії коштів [1].

В останні кілька десятиліть для розв'язування задачі ЕРН було застосовано ряд підходів, таких як математичне програмування (градієнтний метод) [2], лінійне програмування [3], динамічне програмування [4], метод невизначених множників Лагранжа [5], метод гілок та меж [6] та інших. Але, з огляду на особливості форми області значень цільової функції оптимізаційної задачі, зазвичай вони затримуються в локальних оптимумах цільової функції. Крім того, вхідні та вихідні характеристики сучасних генераторів за своєю природою є нелінійними з різних причин, серед яких ефект різномірного палива, нерівномірного навантаження вузлів (клапанів), обмеження швидкості зміни потужності та інші. Ці характеристики і призводять до виникнення цілого ряду локальних оптимумів, які створюють проблеми із знаходженням глобального оптимуму.

З огляду на зазначені особливості, для розв'язування задачі використовуються стохастичні метаевристичні алгоритми, такі як диференціальна еволюція [7] та генетичний алгоритм [8, 9]. Серед згаданих вище популяційних алгоритмів можна виділити диференційну еволюцію, оскільки вона не потребує багато обчислюваної пам'яті, але в той же час вимагає адекватного підбору параметрів для знаходження ефективного розв'язку. Також використовують інші алгоритми стохастичного пошуку, такі як метод нейронних мереж Гопфілда та імітації відпалу. Однак ці методи вимагають суттєвого часу виконання. До того ж, неправильний вибір вхідних параметрів генетичного алгоритму знижує його продуктивність і зменшує його ефективність [10].

Алгоритм оптимізації роєм частинок (ОРЧ), вперше запропонований Кеннеді та Еберхартом [11], є ще одним сучасним прикладним методом оптимізації. В основу даного методу закладено соціально-психологічна поведінкова модель натовпу. За останні два десятиліття було розроблено багато модифікацій даного алгоритму, серед них гібридний алгоритм ОРЧ та хаотичний алгоритм ОРЧ [12].

В 2014 році з'явився новий метаевристичний алгоритм – алгоритм вовчої зграї (АВЗ) [13], який ґрунтується на моделюванні соціальної ієрархії та мисливської поведінки зграї сірих вовків. Його застосовують для розв'язання задачі ЕРН, при цьому він показав прийнятні результати на рівні із ОРЧ [14].

Стаття присвячена розв'язуванню задачі ДРН із використанням ОРЧ та АВЗ як представників алгоритмів ройового інтелекту, що успішно застосовувалися до розв'язування задач різних типів.

2. Мета та завдання. Метою дослідження є розробка зазначених алгоритмів ройового інтелекту та аналіз їх показників при розв'язуванні задачі ДРН.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- зробити аналіз задачі ДРН;
- розробити алгоритм ОРЧ та АВЗ для задачі ДРН;
- виконати аналіз результатів обчислювального експерименту.

3. Формулювання задачі динамічного розподілу навантаження. На відмінну від задачі ЕРН, задача ДРН враховує зміну попиту на електроенергію протягом дня [15]. Таким чином, задача з цільовою функцією, яка характеризуватиме вартість палива в у.о., виглядатиме наступним чином:

$$f = \sum_{t=1}^T \sum_{i=1}^N F_{ii}(P_{ii}) \rightarrow \min$$

(1)

де T – кількість запланованих періодів, N – кількість електростанцій, а $F_{ii}(P_{ii})$ – вартість палива, яке використовують для генерації робочої потужності P_i в момент часу t . Беручи до уваги особливості роботи клапанів генераторів, функцію витрат палива i -ї електростанції у (1) можна подати як суму квадратичної та синусоїдальної функції у такій формі:

$$F_{ii}(P_{ii}) = a_i P_{ii}^2 + b_i P_{ii} + c_i + |e_i \sin(y_i \cdot (P_i^{\min} - P_{ii}))|,$$

де a_i, b_i, c_i – коефіцієнти витрат i -ї електростанції, а e_i та y_i – константи, що показують вплив клапана i -ї електростанції, P_i – це робоча потужність i -ї електростанції в МВт $i = \overline{1, N}$.

Наведемо обмеження, що накладаються на замкнену систему електромережі.

1. Рівняння балансу:

$$\sum_{i=1}^N P_{ii} - P_{iD} - P_{iL} = 0,$$

де P_{td} – загальний попит на електроенергію в момент часу t ; P_{iL} – втрати робочої потужності в момент часу t . Величина P_{iL} розраховується наступним чином:

$$P_{iL} = \sum_{i=1}^N \sum_{j=1}^N P_{ii} B_{ij} P_{ij} + \sum_{i=1}^n B_{i0} P_{ii} + B_{00},$$

де B_{ij} – i, j -й елемент квадратичної матриці коефіцієнтів втрат, що має розмірність $N \times N$, B_{i0} – i -й елемент вектора коефіцієнтів втрат, що має розмірність $1 \times N$, B_{00} – константа втрат.

2. Обмеження генерації робочої потужності:

$$P_i^{\min} \leq P_{ii} \leq P_i^{\max},$$

(2)

де P_i^{\min}, P_i^{\max} – це мінімальна та максимальна межі реальних робочих потужностей для i -ї електростанції.

3. Заборонені зони.

Через дію клапанів та вібрації в підшипниках розглядають діапазони значень робочої потужності, при яких робота генератора неможлива або завдає великих збитків. Ці діапазони називають забороненими зонами експлуатації. Практично найкраща економія досягається шляхом уникнення експлуатації в таких місцях протягом всієї операції. Робочі зони i -ї електростанції описуються так:

$$\begin{aligned} P_i^{\min} &\leq P_{ii} \leq P_{i,1}^u, \\ P_{i,j-1}^l &\leq P_{ii} \leq P_{i,j}^u, \\ P_{i,k}^l &\leq P_{ii} \leq P_i^{\max}, \\ j &= \overline{2, k} \end{aligned}$$

де l, u – нижня та верхня межі для j -ї робочої зони, а k – кількість робочих зон.

4. Обмеження швидкості зміни робочої потужності:

$$P_{ii} - P_{(t-1)i} \leq UR_i,$$

$$P_{(t-1)i} - P_{ii} \leq DR_i,$$

(3)

де UR_i та DR_i – це верхня та нижня межа різниці між потужностями в попередній та поточний періоди. Використовуючи обмеження генерації робочої потужності (2), обмеження (3) можна переписати наступним чином:

$$\max(P_i^{\min}, P_{(t-1)i} - DR_i) \leq P_{ii} \leq \min(P_i^{\max}, P_{(t-1)i} + UR_i).$$

4. Алгоритм вовчої зграї. У математичній моделі соціальної ієрархії у популяції сірих вовків альфою вважають найкращий розв'язок задачі. Відповідно, другий найкращий розв'язок називають бетою, а третій – дельтою. Розв'язки, які залишилися, приймають як омега. У алгоритмі вовчої зграї оптимізація – це моделювання процесу полювання, яким займаються альфа, бета та дельта [13]. Омега-вовки лише допомагають у пошуку здобичі.

Сірі вовки оточують здобич під час полювання. Поведінка навколишнього середовища моделюється наступним чином:

$$\bar{X}(t+1) = \bar{X}_p(t) - \bar{A}|\bar{C}\bar{X}_p(t) - \bar{X}(t)|,$$

$$\bar{A} = 2\bar{a}r_1 - \bar{a},$$

$$\bar{C} = 2\bar{r}_2,$$

де \bar{X} – вектор координат розв'язку (омега вовк) на поточній ітерації t , \bar{X}_p – вектор координат одного із найкращих на поточній ітерації розв'язків $p \in \{\alpha, \beta, \delta\}$, \bar{A} – вектор соціальних коефіцієнтів, \bar{C} – вектор когнітивних коефіцієнтів, \bar{r}_1 та \bar{r}_2 – вектори випадкових величин рівномірно розподілений на проміжку $[0,1]$, \bar{a} – вектор, компоненти якого лінійно зменшуються від 2 до 0 протягом виконання алгоритму.

Наведемо схему роботи алгоритму вовчої зграї.

Крок 1. Генерація популяції. Для популяції розміру M компоненти розв'язку (робочі потужності) генеруються випадковим чином. Для N генераторів i -й розв'язок X_i для задачі ЕРН виглядатиме наступним чином – $X_i = (X_{i1}, X_{i2}, \dots, X_{iN})$ та для задачі ДРН – $X_i = (X_{i1}, X_{i2}, \dots, X_{iN^*})$, $N^* = 24N$ при умові, що генератор працює 24 години.

Крок 2. Для кожного розв'язку X_i^k скорегувати його положення

$$X_{ij}^k = \begin{cases} X_{ij}^k, & \text{якщо } X_{ij}^k \in [P_i^{\min} \dots P_i^{\max}], \\ \text{rand}(P_i^{\min} \dots P_i^{\max}), & \text{якщо } X_{ij}^k \notin [P_i^{\min} \dots P_i^{\max}]. \end{cases}$$

Крок 3. Для кожного розв'язку X_i^k визначити значення цільової функції.

Крок 4. Відсортувати розв'язки у порядку зростання їх значень фітнес-функції. Присвоїти значення: $X_\alpha = X_1^k$, $X_\beta = X_2^k$, $X_\delta = X_3^k$.

Крок 5. Якщо досягнута максимальна кількість ітерацій – перейти до кроку 8.

Крок 6. Для кожного розв'язку (омега-вовки) $X_i^k, i = 4, M$, де M – кількість вовків. Визначити значення компонентів вектора розв'язку, використовуючи формулу:

$$\begin{aligned} \bar{X}_1 &= \bar{X}_\alpha - \bar{A}_1 \left| \bar{C}_1 \bar{X}_\alpha - \bar{X}_i^k \right| \\ \bar{X}_2 &= \bar{X}_\beta - \bar{A}_2 \left| \bar{C}_2 \bar{X}_\beta - \bar{X}_i^k \right|, \\ \bar{X}_3 &= \bar{X}_\delta - \bar{A}_3 \left| \bar{C}_3 \bar{X}_\delta - \bar{X}_i^k \right| \\ \bar{X}_i^{k+1} &= \frac{\bar{X}_1 + \bar{X}_2 + \bar{X}_3}{3}. \end{aligned}$$

Крок 7. Перейти до кроку 2.

Крок 8. Повернути розв'язок альфа-вовка X_α як найкращий знайдений розв'язок.

5. Оптимізація роєм частинок. У алгоритмі ОРЧ координати кожної частинки являються собою можливий розв'язок, який пов'язаний із розташуванням та вектором швидкості інших частинок рою [12]. На швидкість частки впливають три компоненти: інерційна, когнітивна та соціальна. Інерційний компонент імітує інертну поведінку фізичних об'єктів, що продовжують рухатися деякий час по інерції в попередньому напрямку після рішення змінити його. Когнітивний компонент моделює пам'ять частинки щодо свого попереднього найкращого положення, а соціальна складова – вплив найкращого розв'язку, отриманого зграєю за весь процес роботи. Частинки рухаються у багатовимірному просторі пошуку до тих пір, поки вони не знайдуть оптимальний чи близький до нього розв'язок. Модифіковану швидкість кожного агента можна розрахувати, використовуючи поточну швидкість та відстань від $Pbest$ (найкращого розв'язку конкретного агента) та $Gbest$ (найкращого розв'язку рою):

$$V_{ij}^t = wV_{ij}^{t-1} + C_1r_1(Pbest_{ij}^{t-1} - X_{ij}^{t-1}) + C_2r_2(Gbest_i^{t-1} - X_{ij}^{t-1}), i = \overline{1, N_D}, j = \overline{1, M}, (4)$$

де t – поточна ітерація, V_{ij}^t – швидкість j -ї частки в i -ому просторі для поточної ітерації t , X_{ij}^t – позиція j -ї частки в i -ому просторі для поточної ітерації t , w – вага інерції, C_1, C_2 – коефіцієнти прискорення, $Pbest_{ij}^{t-1}$ – найкраща позиція j -ї частки в i -ому просторі до ітерації t , $Gbest_i^{t-1}$ – i -й компонент найкращого розв'язку рою до ітерації t , N_D – число змінних, M – кількість часток в рої, r_1, r_2 – випадкові величини рівномірно розподілені на проміжку $[0,1]$.

Отримана швидкість із виразу використовується для обчислення позиції агента:

$$X_{ij}^t = X_{ij}^{t-1} + V_{ij}^t$$

(5).

Наведемо схему роботи алгоритму оптимізація роєм частинок для задачі розподілу навантаження.

Крок 1. Генерація популяції. Для популяції розміру M компоненти частинок (робочі потужності) генеруються випадковим чином, їх значення рівномірно розподілені на проміжку від 0 до 1. Для N генераторів i -а частинка X_i виглядатиме для задачі ЕРН $X_i = (X_{i1}, X_{i2}, \dots, X_{iN})$ та для задачі ДРН $X_i = (X_{i1}, X_{i2}, \dots, X_{iN^*})$, $N^* = 24N$ при умові, що генератор працює 24 години, де $X_{ij} = X_j^{\min} + r(X_j^{\max} - X_j^{\min})$, $r \in [0;1]$.

Крок 2. Встановити результат для найкращого розв'язку популяції $Gbest = \infty$ та найкращого розв'язку i -ї частинки $Pbest_i = \infty$, де $i = \overline{1, N}$.

Крок 3. Для кожної частинки X_j , $j = \overline{1, M}$ із популяції.

3.1 Визначити значення цільової функції.

3.2 Якщо нове значення фітнес-функції краще за найкращий розв'язок i -ї частинки $Pbest_i$ до ітерації k , то запам'ятати нове значення в $Pbest_i$.

3.3 Запам'ятати позицію частинки.

Крок 4. Встановити значення для найкращого розв'язку популяції як $Gbest = \min\{Gbest, \min_{i=1, M} Pbest_i\}$.

Крок 5. Якщо досягнута максимальна кількість ітерацій, то перейти до кроку 11.

Крок 6. Обрахувати значення швидкості кожної частинки, використовуючи формулу 4.

Крок 7. Якщо поточна швидкість частинки більша за максимальну швидкість для j компоненти $V_{ij} > V_j^{\max}$, то $V_{ij} = V_j^{\max}$, де $V_j^{\max} = 1,5V_{ij}^{k-1}$.

Крок 8. Якщо поточна швидкість частинки менша за мінімальну швидкість для j компоненти $V_{ij} < V_j^{\min}$, то $V_{ij} = V_j^{\min}$, де $V_j^{\min} = 0,5V_{ij}^{k-1}$.

Крок 9. Змінити позицію кожної частинки, використовуючи формулу 5, при цьому якщо робоча потужність X_{ij}^{t+1} виходить за межі реальних робочих потужностей – встановити значення, що наближене до робочої потужності X_{ij}^{t+1} , що задовольняє критеріям.

Крок 10. Перейти на крок 3.

Крок 11. Повернути значення G_{best} як найкращий знайдений результат.

6. Результати експериментів. У якості вхідних даних було обрано задачу, наведену в [11]. Але оскільки там розглянуто ДРН, тому у якості значення загального попиту на електроенергію сформуємо такі модельні значення (таблиця 1), що відповідають попиту за кожну годину:

Таблиця 1.

Попит на електроенергію

Година	Попит	Година	Попит	Година	Попит	Година	Попит
1	1600	7	2700	13	1900	19	2630
2	1681	8	2690	14	2100	20	2930
3	1610	9	2000	15	2200	21	2830
4	1590	10	2100	16	2095	22	2730
5	1800	11	2010	17	2600	23	2230
6	2300	12	2100	18	2200	24	1890

Також у якості фітнес-функції, яка буде використана замість цільової функції при визначенні якості отриманого агентом розв'язку, використано таку:

$$f = \sum_{t=i}^T \sum_{i=1}^N F_{ii}(P_{ii}) + q_1 P_1 + q_2 P_2 + q_3 P_3 \rightarrow \min,$$

де $q_1 = 1$, $q_2 = 50$, $q_3 = 50$. Функція штрафу для рівняння балансу визначається наступним чином:

$$P_1 = \left(\sum_{t=1}^T \sum_{i=1}^N P_{ii} - P_{UL} - P_{UD} \right)^2.$$

Визначимо функцію штрафу для обмеження заборонених зон:

$$P_2 = \sum_{t=1}^T \sum_{j=1}^N V_{k,j},$$

$$\text{де } V_{k,j} = \begin{cases} 1, & \text{якщо } P_{k,j} \text{ знаходиться в забороненій зоні,} \\ 0 & \text{в іншому разі.} \end{cases}$$

Функція штрафу для обмеження швидкості зміни робочої потужності:

$$P_3 = \sum_{t=1}^T \sum_{j=1}^N Q_{k,j},$$

$$\text{де } Q_{k,j} = \begin{cases} 1, & \text{якщо } \Delta P \text{ порушує задані обмеження} \\ 0 & \text{в іншому разі} \end{cases}$$

Таке рішення дозволило перейти від розв'язування задачі із обмеженнями до задачі без обмежень визначеною цільовою функцією.

Проведено два обчислювальні експерименти: для дослідження тенденції зміни значення фітнес функції і цільової функції обох алгоритмів протягом процесу оптимізації та визначення математичного сподівання значень цільової функції в точці розв'язку і їх дисперсії. Зауважимо, що для кожного тестового запуску було згенеровано нову популяцію.

Для проведення першого експерименту було виконано по 20 тестових запусків кожного алгоритму. Значення фітнес-функції (рис.1) та цільової функції (рис. 2) на кожному кроці були усереднені. Необхідно відмітити, що значення цільової функції взяті за одну годину роботи генератора.

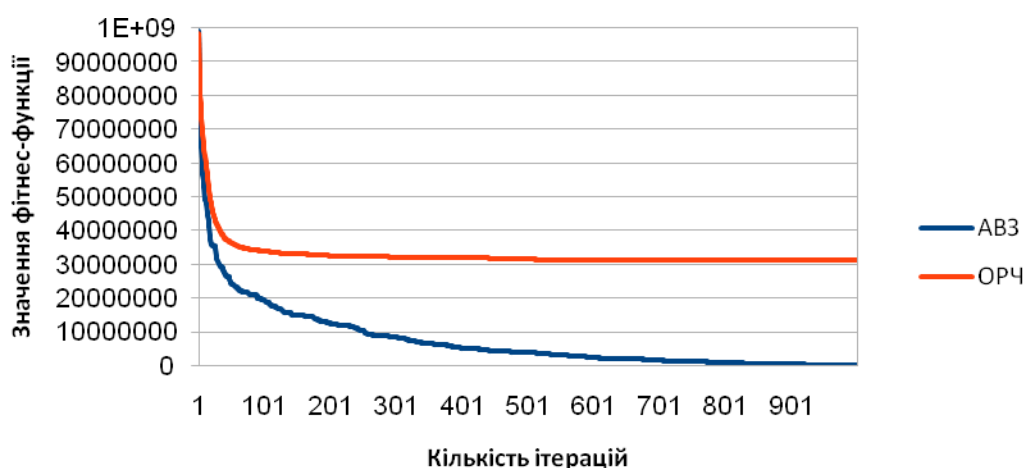


Рис. 1. Залежність значення фітнес-функції від обраної максимальної кількості ітерацій.

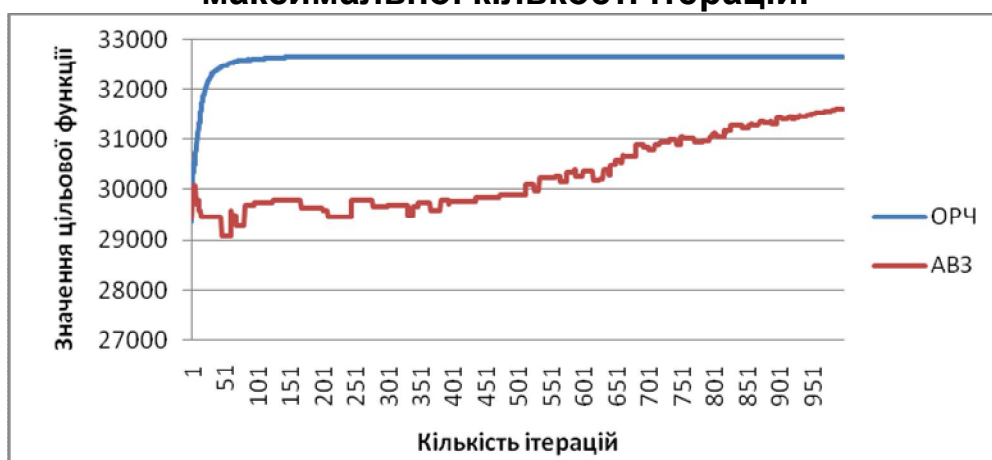


Рис. 2. Залежність значення цільової функції від обраної максимальної кількості ітерацій.

Як можна побачити, алгоритм ОРЧ досить швидко знаходить локальний оптимум і залишається в ньому (при цьому високі значення фітнес-функції демонструють присутність значного порушення обмежень задачі). Натомість АВЗ демонструє тенденцію у

поступовому спаданні значень фітнес-функції та знаходженні нових значень цільової функції на всіх етапах роботи.

Другий експеримент проводився із різними значеннями максимальної кількості ітерацій серед 100 тестових запусків. Результати наведені у таблиці 2.

Таблиця 2.

Порівняння математичного сподівання та дисперсії

К-сть ітерацій	АВЗ		ОРЧ	
	Мат. сподівання значень ц.ф.	Дисперсія значень ц.ф.	Мат. сподівання значень ц.ф.	Дисперсія значень ц.ф.
100	31767,35	313,11	38578,86	1078,75
250	30663,11	1172,15	38413,57	829,69
500	30061,90	1713,92	38158,18	1379,51
1000	29757,96	2013,4	37901,38	1305,37
2000	29607,20	2164,07	38445,99	1164,52

Як можна побачити, АВЗ отримав в середньому на 25% кращі за точністю результати, хоча і з більшою дисперсією значень цільової функції.

Висновки. З огляду на отримані результати експериментів можна зробити висновок, що алгоритми ройового інтелекту можуть бути застосовані для розв'язування практичних задач ДРН. Але при цьому, з огляду на свою природу, отриманий розв'язок задачі може не повністю задовольняти обмеження задачі, через що з'являється необхідність у додатковому кроці «доопрацювання» по завершенні розв'язання.

АВЗ продемонстрував знаходження більш точних значень як фітнес-функції, так і цільової функції задачі у порівнянні з алгоритмом ОРЧ.

Література:

1. Sinha N. *Evolutionary Programming Techniques for Economic Load Dispatch* / N. Sinha, R. Chakrabarti, P. K. Chattopadhyay // *IEEE Transactions On Evolutionary Computation*. – 2003. – V. 7. – P. 83-94.
2. Dodu J. C. *An optimal formulation and solution of short-range operating problems for a power system with flow constraints* / J. C. Dodu, P. Martin, A. Merlin // *IEEE Proc.* – 1972. – V. 60(1). – P. 54–63.
3. Parikh J. *A multi-area linear programming approach for analysis of economic operation of the Indian power system* / J. Parikh, D. Chattopadhyay // *IEEE Trans Power Syst.* – 1996. – V. 11. – P. 2–8.
4. Liang Z. X. *A zoom feature for a dynamic programming solution to economic dispatch including transmission losses* / Z. X. Liang, J. D. Glover // *IEEE Trans Power Syst.* – 1992. – V. 7. – P. 50.
5. Nanda J. *Economic emission dispatch with line flow constraints using a classical technique* / J. Nanda, L. Hari, M. L. Kothari // *IEE ProcGener Trans Distrib.* – 1994. – V. 141. – P. 10.
6. Chen C. L. *Branch and bound scheduling for thermal generating units* / C. L. Chen, S. C. Wang // *IEEE Trans Energy Convers.* – 1993. – V. 184. – P. 9.
7. Perez-Guerrero R. E. *Economic power dispatch with non-smooth cost functions using differential evolution* / R. E. Perez-Guerrero, R. J. Cedenio-Maldonado // *Proceedings of the 37th annual North American power symposium*. – 2005. – P. 90.
8. Chiang C. L. *Genetic-based algorithm for power economic load dispatch* / C. L. Chiang // *IEE ProcGener Trans Distrib.* – 2007. – V. 1. – P. 9.

9. Bakirtzis A. *A genetic algorithm solution to the economic dispatch problem/ A. Bakirtzis, V. Petridis, S. Kazarlis // Generation, Transmission and Distribution. – 1994. – V. 141. – P. 377–382.*
10. Pradhan Moumita. *Oppositional based grey wolf optimization algorithm for economic dispatch problem of power system / Moumita Pradhan, Provas Kumar Roy, Tandra Pal // Ain Shams Engineering Journal. – 2017. – № 2. – P. 1-11.*
11. Kennedy J. *Particle Swarm Optimization / J. Kennedy // Proceedings of IEEE International Conference on Neural Networks. – 1995. – P. 1942-1948.*
12. Coelho L.S. *Solving economic load dispatch problems in power systems using chaotic and Gaussian particle swarm optimization approaches / L.S. Coelho, C. Lee // International Journal of Electrical Power. – 2008. – № 5 – P. 297-307.*
13. Mirjalili S. *Grey Wolf Optimizer / S. Mirjalili, S. M. Mirjalili, A. Lewis // Advances in Engineering Software. – 2014. – V. 69. – P. 46-61.*
14. Sharma Sudhir. *Economic Load Dispatch Using Grey Wolf Optimization / Sudhir Sharma, Shivani Mehta, Nitish Chopra // Journal of Engineering Research and Applications. – 2015. – № 4. – P. 128-132*
15. Balamurugan R. *An Improved Differential Evolution Based Dynamic Economic Dispatch with Nonsmooth Fuel Cost Function / R. Balamurugan, S. Subramanian // Electrical Systems. – 2007. – № 3.– P. 151-161.*