

МЕТОД НАЙШВИДШОГО СПУСКУ: АНАЛІЗ ОБМЕЖЕНЬ ТА ЕВОЛЮЦІЯ ЗАСТОСУВАННЯ В СУЧАСНИХ ЕНЕРГЕТИЧНИХ СИСТЕМАХ

Отримано 18 бер. 2026 р.; рекомендовано до публікації 26 чер. 2026 р.
Доступно онлайн 30 чер. 2026 р.

Коровушкін В. О.¹, Бойченко С. В.²,
Кузнецов М. П.³, Данілін О. В.⁴

Автор для кореспонденції: Коровушкін Віталій,
e-mail: vitalijkorovuskin@gmail.com

Анотація. У статті здійснено системний критичний аналіз методу найшвидшого спуску (SD) в контексті оптимізації сучасних енергетичних систем. Попри зазначення фундаментальної ролі в теорії ітераційних алгоритмів, доведено суттєві обмеження МНШ при розв'язанні великомасштабних задач, як-от економічний розподіл навантаження (ELD) та оптимальний потік потужності (AC-OPF). У роботі детально досліджено математичні причини цього явища: показано, що фізична гетерогенність компонентів енергосистеми (різні масштаби імпедансів ліній та характеристик генерації) призводить до поганої обумовленості гесіана цільової функції. Високе число обумовленості (κ) спричиняє «зигзагоподібну» траєкторію збіжності, що робить метод обмежено придатним для оперативного управління режимами в реальному часі. На основі порівняльного аналізу з методами другого порядку (метод Ньютона) та квазіньютонівськими алгоритмами (L-BFGS) обґрунтовано перехід до методів внутрішньої точки як галузевого стандарту для задач з обмеженнями. Ключова новизна роботи полягає в дослідженні розширення сфери застосування градієнтних методів в еру штучного інтелекту. Автори демонструють парадигмальний зсув: характеристики, що були обмежувальними факторами в детермінованій оптимізації (шумна траєкторія та спрощена обробка градієнта), трансформувалися у вирішальні переваги стохастичного градієнтного спуску (SGD). Доведено, що саме ця стохастична природа дає змогу ефективно навчати нейронні мережі для задач прогнозування генерації ВДЕ та управління попитом, перетворюючи традиційний метод на фундамент сучасних Data-driven рішень в енергетиці.

Ключові слова: оптимізація енергосистеми; оптимальний розподіл потужності в змінному струмі (AC-OPF); метод найкрутішого спуску; стохастичний градієнтний спуск (SGD); погано обумовлені задачі; аналіз збіжності.

Перелік використаних позначень та скорочень

ВДЕ – відновлювані джерела енергії
ККТ – умови Каруша – Куна – Такера
ЛЕП – лінія електропередачі
МНШ – метод найшвидшого спуску
AC-OPF – Optimal Power Flow змінного струму
BFGS – Broyden–Fletcher–Goldfarb–Shanno

ELD – Economic Load Dispatch,
IPM – Interior Point Methods
L-BFGS – Limited-memory BFGS
ML – Machine Learning
NLP – Nonlinear Programming,
SGD – Stochastic Gradient Descent
SQP – Sequential Quadratic Programming

Вступ. Сучасні енергетичні системи є одними з найскладніших інженерних та кіберфізичних систем, створених людством. Їх функціонування, від щосекундного балансування генерації та попиту до довгострокового планування розвитку, пронизане фундаментальними оптимізаційними викликами. Необхідність оптимізації в енергетиці диктується трьома основними факторами:

економічною ефективністю, надійністю системи та викидами декарбонізації [1–3].

З економічного погляду навіть незначне (наприклад, 0.1%) покращення ефективності роботи об'єднаної енергосистеми трансформується в мільйонні заощадження на вартості палива [4]. Задачі, такі як економічний розподіл навантаження (Economic Load Dispatch,

¹ аспірант
<https://orcid.org/0000-0002-7571-7124>

² д-р. техн. наук
<https://orcid.org/0000-0002-2489-4980>

³ д-р. техн. наук
<https://orcid.org/0000-0002-0497-7439>

⁴ канд. техн. наук
<https://orcid.org/0000-0003-3207-1156>

^{1,2,4} Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського", м. Київ, Україна

³ Інститут відновлюваної енергетики НАН України, м. Київ, Україна

ELD) та оптимальний потік потужності (Optimal Power Flow, OPF), безпосередньо мінімізують загальні витрати на генерацію [5].

З погляду надійності система повинна працювати в межах жорстких фізичних обмежень: теплові обмеження ліній електропередачі, межі напруги у вузлах, резерви генерації та обмеження стійкості. Оптимізація (зокрема, OPF) є інструментом, який знаходить найдешевший робочий режим, не порушуючи жодного з цих обмежень безпеки [6].

Нарешті, інтеграція стохастичних відновлюваних джерел енергії (ВДЕ), як-от вітрові та сонячні станції, вносить високий ступінь невизначеності. Це вимагає нових оптимізаційних підходів, що здатні оперувати в умовах стохастички, а також управляти попитом (Demand-Response) та системами зберігання енергії [7]. Особливої важливості набувають задачі оптимізації в ситуаціях з обмеженим доступом до централізованого постачання енергії та переходом до розподіленої генерації (утворення системи мікромереж, спільного використання засобів зберігання енергії, вибору точки приєднання ВДЕ для зменшення трафіку електроенергії та її втрат, питань взаємного обміну енергією тощо).

Більшість цих проблем (зокрема, AC-OPF – Optimal Power Flow змінного струму) математично формуються як великомасштабні, нелінійні, не опуклі та сильно обмежені задачі нелінійного програмування (Nonlinear Programming, NLP). Застосування розподіленої генерації з ВДЕ потребує врахування модульної побудови та переходу до цілочисельних задач. Розмірність таких задач для реальних систем може сягати десятків або сотень тисяч змінних та обмежень, а вимога до їх вирішення в реальному часі (або близькому до нього) ставить жорсткі обмеження щодо швидкодії до обчислювальних методів [8].

Мета роботи – кількісна оцінка обмежень методу найшвидшого спуску при розв'язанні класичних задач енергетики (ELD, AC-OPF) через проблему високої обумовленості (κ) та гетерогенності енергосистем, а також порівняльний аналіз його ефективності з сучасними методами оптимізації. Водночас дослідження спрямоване на переосмислення ролі методу в сучасних умовах: демонстрацію його переходу від традиційного інструменту детермінованої оптимізації до основи стохастичного градієнтного спуску (SGD), який є критично важливим для навчання нейромереж в енергетичних задачах (прогнозування ВДЕ, попиту)

Для розуміння обмежень методу спершу необхідно чітко встановити його математичні основи. МНШ є методом для вирішення задачі *безумовної* оптимізації.

Постановка задачі безумовної оптимізації. Розглянемо задачу знаходження локального мінімуму диференційовної функції $f(x)$ [5] (1):

$$\min_{x \in \mathbb{R}^n} f(x) \quad (1)$$

де $f: \mathbb{R}^n \rightarrow \mathbb{R}$ є гладкою функцією, принаймні класу C^1 (тобто має неперервні перші часткові похідні).

Ключовими інструментами аналізу є:

1. Градієнт: Вектор перших часткових похідних, $\nabla f(x) \in \mathbb{R}^n$, який вказує в напрямку *найшвидшого зростання* функції в точці x .
2. Гесіан: Матриця других часткових похідних, $\nabla^2 f(x) \in \mathbb{R}^{n \times n}$, яка описує *кривизну* функції в точці x .

Необхідною умовою першого порядку для того, щоб точка x^* була локальним мінімумом, є $\nabla f(x^*) = 0$. Такі точки називаються *стаціонарними*. МНШ, як і більшість ітеративних методів, намагається знайти саме таку стаціонарну точку.

Ідея МНШ полягає в тому, щоб на кожному кроці рухатися з поточної точки x_k у напрямку p_k , який забезпечує *найбільше локальне зменшення* значення функції f .

Для формалізації цього розглянемо розклад функції f у ряд Тейлора першого порядку навколо точки x [5] (2):

$$f(x + \alpha p) \approx f(x) + \alpha \nabla f(x)^T p \quad (2)$$

де p – це деякий одиничний напрямок ($\|p\|_2 = 1$), а $\alpha > 0$ – мала довжина кроку (передбачається наявність ортонормованого базису в області визначення). Необхідно знайти такий напрямок p , який мінімізує зміну $f(x + \alpha p) - f(x)$, що еквівалентно мінімізації добутку $\nabla f(x)^T p$.

Отже, вирішується задача (3):

$$\min_p \nabla f(x)^T p \quad \text{за умови} \quad \|p\|_2 = 1 \quad (3)$$

Згідно з нерівністю Коші – Шварца (4):

$$|\nabla f(x)^T p| \leq \|\nabla f(x)\|_2 \cdot \|p\|_2 \quad (4)$$

Мінімум (тобто найбільш негативне значення) досягається, коли p є колінеарним і протилежно спрямованим до $\nabla f(x)$. Тобто (5):

$$p = -\frac{\nabla f(x)}{\|\nabla f(x)\|_2} \quad (5)$$

Цей напрямок p називається напрямком найшвидшого спуску. Він локально гарантує найстрімкіше зменшення цільової функції.

Алгоритмічна структура. На основі цього виведення ітеративний процес МНШ будується в такий спосіб: для заданої початкової точки x_0 [5]:

Алгоритм 1: Метод найшвидшого спуску (МНШ)

1. Для $k = 0, 1, 2, \dots$
2. Якщо $\nabla f(x_k) = 0$, то Стоп
3. Обчислити напрямок спуску: $p_k = -\nabla f(x_k)$
4. Обчислити довжину кроку: $\alpha_k > 0$ (за допомогою процедури лінійного пошуку)
5. Оновити точку: $x_{k+1} = x_k + \alpha_k p_k$
6. Кінець

Алгоритм має дві ключові обчислювальні компоненти на кожній ітерації:

1. Обчислення напрямку p_k (вимагає обчислення градієнта $\nabla f(x_k)$).

2. Обчислення довжини кроку α_k (процедура, відома як linesearch, або одновимірна мінімізація).

Саме вибір α_k є критично важливим для ефективності методу.

Точний лінійний пошук (Line Search). Теоретично оптимальний підхід полягає у виборі кроку α_k , який мінімізує функцію f вздовж напрямку p_k [5] (6):

$$\alpha_k = \arg \min_{\alpha > 0} f(x_k + \alpha_k p_k) = \arg \min_{\alpha > 0} \varphi(\alpha) \quad (6)$$

Це, по суті, одновимірна задача оптимізації. Умова оптимальності для α_k дає $\varphi'(\alpha_k) = 0$. Застосовуючи ланцюгове правило, отримуємо (7):

$$\varphi'(\alpha_k) = \nabla f(x_k + \alpha_k p_k)^T \cdot p_k = 0 \quad (7)$$

Це означає, що при точному лінійному пошуку новий градієнт $\nabla f(x_{k+1})$ є ортогональним до попереднього напрямку p_k . Оскільки $p_k = -\nabla f(x_k)$, це також означає, що $\nabla f(x_{k+1})^T \nabla f(x_k) = 0$. Послідовні градієнти є ортогональними.

Саме ця властивість зумовлює характерну «зигзагоподібну» (zigzag) поведінку МНШ [8]. Уявімо вузьку, витягнуту «долину» цільової функції. На «схилі» долини градієнт вказує переважно поперек долини (в напрямку найкрутішого схилу), а не вздовж неї до мінімуму. МНШ з точним пошуком робить крок p_k і переходить у точку на протилежному «схилі» в точці x_{k+1} , де новий градієнт $\nabla f(x_{k+1})$ є ортогональним до p_k . Оскільки p_k був майже перпендикулярним до оптимального шляху (вздовж «дна» долини), новий градієнт $\nabla f(x_{k+1})$ буде майже в протилежному напрямку до $\nabla f(x_k)$. Алгоритм здійснюватиме ітераційні переходи (зигзагом) між схилами долини, роблячи дуже повільний прогрес у напрямку справжнього мінімуму.

Таким чином, «оптимальність» (точний пошук кроку) на кожному кроці призводить до глобальної субоптимальної траєкторії.

Єдиний випадок, де точний пошук є обчислювально виправданим, – це додатно визначена квадратична функція [9]. У цьому разі α_k має замкнену форму (8):

$$\alpha_k = \frac{\nabla f_k^T \cdot \nabla f_k p_k}{\nabla f_k^T \cdot Q \nabla f_k p_k} \quad (8)$$

$$\text{Де } \nabla f_k = Qx_k - b$$

Неточний лінійний пошук (Умови Вольфа). У загальному випадку, для нелінійних функцій, знаходити точний мінімум α_k є недоцільним з огляду на обчислювальні витрати (витрати ресурсів можуть перевищувати вартість самої ітерації МНШ).

На практиці використовуються процедури неточного лінійного пошуку, які гарантують «достатньо хороший» крок. Стандартом де-факто є Умови Вольфа (Wolfe Conditions), які вимагають, щоб α_k задовольняв дві умови [10]:

Умова Армїхо (Armijo) (достатнє зменшення) (9):

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^T \cdot p_k \quad (9)$$

де $c_1 \in (0, 1)$, типowo $c_1 = 10^{-4}$. Це гарантує, що крок α_k не «надто великий» і дає реальне зменшення функції.

Умова кривизни (Curvature) (10):

$$\nabla f(x_k + \alpha_k p_k)^T \cdot p_k \geq c_2 \nabla f(x_k)^T \cdot p_k \quad (10)$$

де $c_2 \in (c_1, 1)$, типowo $c_2 = 0.9$.

Існування такого α_k , що задовольняє обидві умови, гарантоване для гладких функцій, обмежених знизу. Використання умов Вольфа є ключовим для доведення глобальної збіжності методу.

Теоретичні гарантії збіжності. Глобальна збіжність МНШ (тобто збіжність до стаціонарної точки x^* з будь-якої x_0) гарантується за досить м'яких умов. Ключовою є умова L-гладкості.

Означення (L-гладкість): Функція f називається умовно L-гладкою, якщо її градієнт ∇f є Липшицевим з константою $L > 0$ [11], тобто (11):

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L \|x - y\|_2 \quad \forall x, y \in \mathbb{R}^n \quad (11)$$

Якщо $f \in C^2$, це еквівалентно тому, що найбільше власне значення Гессіана $\nabla^2 f(x)$ є обмеженим L , тобто $\nabla^2 f(x) \leq L I$.

Теорема 1 (Збіжність для L-гладких функцій) [12]:

Якщо f є L-гладкою та обмеженою знизу, то МНШ з неточним пошуком (що задовольняє умови Вольфа) або зі сталим кроком $\alpha_k = \alpha \in (0, 2/L)$ гарантує, що $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\|_2 = 0$.

Ця теорема лише каже, що зрештою зійдемося до точки, де градієнт дорівнює нулю. Однак, вона нічого не говорить про те, як швидко це станеться.

Аналіз швидкості збіжності. Для аналізу швидкості нам потрібна сильніша умова – сильна опуклість.

Означення (m-сильна опуклість) [13]: Функція f називається m-сильно опуклою (з $m > 0$), якщо (12):

$$f(y) \geq f(x) + \nabla f(x_k)^T (y - x) + \frac{m}{2} \|y - x\|_2^2 \quad \forall x, y \in \mathbb{R}^n \quad (12)$$

Якщо $f \in C^2$, це еквівалентно тому, що найменше власне значення гесіана $\nabla^2 f(x)$ є обмеженим знизу m , тобто $\nabla^2 f(x) \geq m I$.

Теорема 2 (Лінійна збіжність для сильно опуклих функцій) [14]:

Якщо f є L-гладкою та m-сильно опуклою, то МНШ з точним лінійним пошуком або з оптимально вибраним сталим кроком збігається до єдиного глобального мінімуму x^ з лінійною (або геометричною) швидкістю (13, 14):*

$$f(x_k) \geq f(x^*) \leq (1 - C)^k f((x_0) - f(x^*)) \quad (13)$$

$$\|x_k - x^*\|_2^2 \leq (1 - C)^k \|x_0 - x^*\|_2^2 \quad (14)$$

де $C > 0$ – константа збіжності.

На перший погляд «лінійна збіжність» виглядає задовільно. Однак існує залежність від значення константи C . Для квадратичної функції (15):

$$f(x) = \frac{1}{2}x^T Q x - b^T x, Q > 0, \quad (15)$$

константи L та m є просто найбільшим (λ_{\max}) та найменшим (λ_{\min}) власними значеннями матриці Q . У цьому разі можна показати, що (16):

$$f(x_k) - f(x^*) \leq \left(\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \right)^2 (f(x_0) - f(x^*)) \quad (16)$$

Специфіка збіжності за умов поганої обумовленості: як міра «складності». Введемо ключовий параметр будь-якої оптимізаційної задачі: число обумовленості k (17).

$$k = \frac{L}{m} = \frac{\lambda_{\max}(Q)}{\lambda_{\min}(Q)} \quad (17)$$

k вимірює відношення найбільшої кривизни до найменшої; геометрично це міра того, наскільки «витагнутою» є цільова функція.

Перепишемо вираз для швидкості збіжності через k (18):

$$f(x_k) - f(x^*) \leq \left(\frac{k-1}{k+1} \right)^{2k} (f(x_0) - f(x^*)) \quad (18)$$

Проаналізуємо формулу (18):

Випадок 1 (Ідеальний): $k = 1$.

Це означає $L = m$, $\lambda_{\max} = \lambda_{\min}$. Гесіан $Q = \lambda I$. Рівні значення функції f є ідеальними концентричними колами (або сферами в \mathbb{R}^n).

Фактор збіжності (19)

$$\left(\frac{1-1}{1+1} \right)^2 = 0 \quad (19)$$

Метод збігається за **один** крок з будь-якої початкової точки. Градієнт $\nabla f(x_0) = Qx_0 - b$ завжди вказує точно на центр $x^* = Q^{-1}b$.

Випадок 2 (Реалістичний, «погано обумовлений»): $k \gg 1$.

Це означає $L \gg m$. Рівні функції f є дуже витягнутими еліпсами (еліпсоїдами).

Фактор збіжності (20):

$$\frac{k-1}{k+1} = \frac{\frac{L}{m}-1}{\frac{L}{m}+1} = \frac{L-m}{L+m} \approx \frac{L}{L} = 1 \quad (20)$$

Більш точно, використовуючи наближення $\frac{1}{1+x} \approx 1 - x$, маємо (21):

$$\frac{k-1}{k+1} \approx 1 - \frac{2}{k} \quad (21)$$

Швидкість збіжності (22):

$$\left(1 - \frac{2}{k} \right)^{2k} \quad (22)$$

Наприклад, нехай $k = 1000$. Фактор збіжності $\approx (1 - 0.002)^{2k} \approx (0.998)^{2k} \approx (0.996)^{2k}$. Це означає, що на кожній ітерації помилка $f(x_k) - f(x^*)$ зменшується лише на 0.4 %.

Щоб зменшити помилку в 1000 разів (тобто $0.996^{k} < 0.001$), потрібно $k > \frac{\ln(0.001)}{\ln(0.996)} \approx -\frac{6.9}{-0.004} \approx 1725$ ітерацій.

Якщо $k = 10\,000$, потрібно буде понад 17 000 ітерацій.

Це математичне спостереження є прямим поясненням, чому на сьогодні МНШ обмежено придатний для реальних енергосистем. Енергосистема за своєю природою є гетерогенною. Вона складається з елементів з дуже різними фізичними властивостями, наприклад:

1. Лінії: магістральна ЛЕП 750 кВ має дуже низький активний опір ($R \ll X$), тоді як локальна розподільча лінія 10 кВ може мати $R \approx X$.
2. Генератори: криві витрат $C(P)$ мають різну кривизну (другу похідну a_i в $a_i P^2 + \dots$). Базова АЕС має майже плоску криву ($a_i \approx 0$), тоді як пікова газова турбіна має дуже круту криву ($a_i \gg 0$). Для вітрової турбіни крива потужності наближено описується логістичною кривою зі змінною кривизною, залежною від швидкості вітру, а для сонячної електростанції стохастичний характер генерації обумовлює криву потужності, теоретично розривну в кожній точці.

Ця *фізична гетерогенність* (різні масштаби імпедансів, витрат, чутливостей) *математично* транслюється в гесіан цільової функції (або гесіан Лагранжіана в обмежених задачах) з великим розкидом власних значень. λ_{\max} (що відповідає «жорстким» зв'язкам / обмеженням) буде великим, а λ_{\min} (що відповідає «м'яким» зв'язкам) буде малим.

Отже, будь-яка реалістична великомасштабна енергосистема гарантовано породжує оптимізаційну задачу з доволі великим числом обумовленості k . Відповідно до Теорема 2, МНШ гарантовано матиме неприйнятно повільну збіжність на *будь-якій* реальній задачі OPF чи ELD. Його неефективність – це прямий наслідок фізики енергосистеми.

Тепер застосуємо отримані теоретичні висновки до двох ключових задач оптимізації в енергетиці: економічного розподілу навантаження (ELD) та оптимального потоку потужності (OPF) [15].

Випадок 1: Економічний розподіл навантаження (ELD).

Постановка задачі. *Класична ELD (без втрат)* [16] (23):

$$\min_{P_{Gi}} \sum_{i=1}^N C_i P_{Gi} = \sum_{i=1}^N a_i P_{Gi}^2 + b_i P_{Gi} + c_i \quad (23)$$

за умови балансу потужності (24):

$$\sum_{i=1}^N P_{Gi} = P_{Demand} \quad (24)$$

та обмежень генерації (25):

$$P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max} \quad \forall i. \quad (25)$$

ELD з урахуванням втрат (B-матриця) [17]:

Задача ускладнюється, оскільки баланс потужності стає нелінійним (26):

$$\sum_{i=1}^N P_{Gi} = P_{Demand} + P_{Loss}(P_G) \quad (26)$$

де втрати в мережі P_{Loss} апроксимуються квадратичною формою (формула Крона, В-матриця) [17] (27):

$$P_{Loss}(P_G) = \sum_{i=1}^N \sum_{j=1}^N P_{Gi} B_{ij} P_{Gj} + \sum_{i=1}^N B_{ij} P_{Gi} + B_{00} \quad (27)$$

Цільова функція та обмеження тепер нелінійно пов'язані.

Адаптація МНШ для вирішення ELD. МНШ – це метод безумовної оптимізації. ELD – обмежена задача. Щоб застосувати МНШ, обмеження потрібно якось врахувати. Існує два поширених підходи [18]:

1. *Метод штрафних функцій:* задача перетворюється на безумовну, додаючи обмеження до цільової функції як «штраф» за їх порушення (28):

$$f(P_G) = \sum_{i=1}^N C_i(P_{Gi}) + \mu(\sum_{i=1}^N P_{Gi} - P_D - P_{Loss})^2 + \sum_{i=1}^N v_i \cdot [\max(0, P_{Gi} - P_{Gi}^{max})]^2 + \sum_{i=1}^N \omega_i \cdot [\max(0, P_{Gi}^{min} - P_{Gi})]^2, \quad (28)$$

де $\mu > 0$ – штрафний параметр за порушення балансу потужності; $v_i, \omega_i > 0$ – штрафні параметри за обмеження нерівностей; $P_{Gi}^{min}, P_{Gi}^{max}$ – нижня та верхня межі генерації; $\max(0, \cdot)$ – забезпечує штраф тільки при порушенні обмежень. Після цього до $f(P_G)$ застосовується МНШ.

2. *Метод проекції градієнта:* цей метод можна використати для простих «коробкових» обмежень ($P_{Gi}^{min}, P_{Gi}^{max}$). Ідея полягає в тому, щоб зробити крок МНШ, і якщо нова точка x_{k+1} виходить за межі, «спроєкувати» її назад на допустиму область.

Аналіз ефективності МНШ для ELD. Обидва підходи є вкрай неефективними для ELD.

Проблема 1 (Погана обумовленість). Навіть у найпростішому випадку ELD без втрат гесіан цільової функції $\sum C_i(P_{Gi})$ є діагональною матрицею $H = \text{diag}(2a_1, \dots, 2a_N)$. Число обумовленості $\kappa = \frac{\max(a_i)}{\min(a_i)}$. Як зазначалося раніше, у реальній системі з дешевими базовими (малий a_i) та «дорогими» піковими (великий a_i) генераторами, κ буде дуже великим. МНШ демонструватиме класичну повільну, зигзагоподібну поведінку.

Коли ми додаємо В-матрицю, гесіан стає $H = \text{diag}(2a_i) + 2\mu B$. Це ще більше погіршує ситуацію.

Проблема 2 (Погіршення обумовленості через застосування штрафних функцій). Інтеграція методу штрафних функцій критично знижує обчислювальну ефективність МНШ. Оскільки для точного задоволення обмежень задачі штрафний параметр μ повинен прямувати до нескінченності ($\mu \rightarrow \infty$).

Можна показати, що гесіан штрафної функції $f(P_G)$ стає все більш погано обумовленим, коли μ зростає. Фактично, $\kappa \rightarrow \infty$ коли $\mu \rightarrow \infty$.

МНШ, швидкість якого критично залежить від κ , стає практично непридатним для досягнення збіжності саме тоді, коли ми вимагаємо від нього точності.

Тож для ELD МНШ є вкрай неефективним підходом для «простої» ELD, методи, що явно враховують обмеження (наприклад, методи на основі вирішення системи умов Каруша – Куна – Таккера (ККТ) або спеціалізовані методи, що враховують структуру задачі), є на порядки ефективнішими. Існують дослідження, що використовують МНШ у поєднанні з множниками Лагранжа для вирішення екологічного економічного розподілу, але це вже модифіковані, а не «чисті» версії методу [19].

Випадок 2: Оптимальний потік потужності (OPF), складність задачі AC-OPF. Це основна задача операційного управління. Задача AC-OPF є великомасштабною нелінійною (через рівняння потоку потужності) та не опуклою (має багато локальних мінімумів) [20].

Формально (спрощено) (29):

$$\min_{v, \theta, P_G, Q_G} \sum C_i(P_{Gj}) \quad (29)$$

За умови (обмеження-рівності – рівняння потоку потужності) (30):

Для кожного вузла i (30), за умови, де

$x = [P_G, Q_G, V, \theta]$ – вектор оптимізації;

μ_P, μ_Q – штрафні параметри за рівняння балансу потужності;

v_i^{P+}, v_i^{P-} – штрафні параметри за обмеження P_G ;

$\omega_i^{Q+}, \omega_i^{Q-}$ – штрафні параметри за обмеження Q_G ;

$\kappa_i^{V+}, \kappa_i^{V-}$ – штрафні параметри за обмеження напруги;

λ_{ij} – штрафні параметри за обмеження ліній;

N_G – кількість генераторів, N – кількість вузлів, L – множина ліній:

$$\begin{aligned} \min f(x) = & \sum_{i=0}^{N_G} C_i(P_{Gi}) + \mu_P \sum_{i=0}^N [P_i(V, \theta) - (P_{Gi} - P_{Di})]^2 + \mu_Q \sum_{i=0}^N [Q_i(V, \theta) - (Q_{Gi} - Q_{Di})]^2 + \\ & \sum_{i=0}^{N_G} v_i^{P+} [\max(0, P_{Gi} - P_{Gi}^{max})]^2 + \\ & \sum_{i=0}^{N_G} v_i^{P-} [\max(0, P_{Gi}^{min} - P_{Gi})]^2 + \sum_{i=0}^{N_G} \omega_i^{Q+} [\max(0, Q_{Gi} - Q_{Gi}^{max})]^2 + \sum_{i=0}^{N_G} \omega_i^{Q-} [\max(0, Q_{Gi}^{min} - Q_{Gi})]^2 + \\ & \sum_{i=0}^{N_G} \kappa_i^{V+} [\max(0, |V_i| - V_i^{max})]^2 + \\ & \sum_{i=0}^{N_G} \kappa_i^{V-} [\max(0, V_i^{min} - |V_i|)]^2 + \\ & \sum_{i,j \in L} \lambda_{ij} [\max(0, S_{ij}(V, \theta) - S_{ij}^{max})]^2 \quad (30) \end{aligned}$$

Примітка: Усі штрафні параметри $\mu, v, \omega, \kappa, \lambda > 0$ є додатними та досить великими величинами [21].

За умови (обмеження-нерівності):

- Межі генерації: $P_{Gi}^{min} \leq P_{Gi} \leq P_{Gi}^{max}, Q_{Gi}^{min} \leq Q_{Gi} \leq Q_{Gi}^{max}$

- Межі напруги: $V_i^{min} \leq |V_i| \leq V_i^{max}$
- Теплові обмеження ліній: $S_{ij}(V, \theta) \leq S_{ij}^{max}$

Застосування МНШ до OPF. Пряме застосування МНШ до AC-OPF є неможливим. Це задача безумовної оптимізації, тоді як OPF визначається тисячами складних нелінійних обмежень – рівностей та нерівностей.

Як і у випадку з ELD, можна спробувати «вбудувати» всі ці обмеження в цільову функцію через штрафи та бар'єри. Це призведе до створення багатовимірної цільової функції зі складним градієнтним рельєфом, цільової функції, гесіан якої буде надзвичайно погано обумовленим, роблячи МНШ недоцільним.

Історично, ранні спроби вирішення OPF (наприклад, «Reduced Gradient Method» Карпонт'є) використовували градієнтну інформацію. Але ці методи були набагато складнішими, ніж чистий МНШ. Вони використовували рівняння потоку потужності для зменшення розмірності задачі (виражаючи залежні змінні через незалежні) і намагалися «рухатися» вздовж межі допустимої області, що визначається обмеженнями.

Сучасний підхід. Сучасні, надійні вирішувачі AC-OPF ніколи не використовують МНШ. Вони базуються на методах, що явно та ефективно обробляють обмеження, використовуючи інформацію другого порядку (або її апроксимацію):

1. Методи внутрішньої точки (Interior Point Methods, IPM). Домінуючий підхід. Вони перетворюють задачу з нерівностями на послідовність задач з рівностями за допомогою *бар'єрних* функцій. Потім вони застосовують *метод Ньютона* до системи умов ККТ для цієї бар'єрної задачі. Вони працюють з гесіаном Лагранжіана і є надзвичайно ефективними для великомасштабних опуклих та (часто) не опуклих задач.
2. Послідовне квадратичне програмування (Sequential Quadratic Programming, SQP). Інший потужний метод, який на кожній ітерації апроксимує задачу OPF *квадратичною* задачею (апроксимуючи лагранжіан) та вирішує її.

Метод змінних напрямків множників (Alternating Direction Method of Multipliers, ADMM), що набуває все більшої популярності для децентралізованої оптимізації, зокрема при управлінні мікромережами та розподіленою генерацією з ВДЕ. Він дає змогу розбити складну великомасштабну задачу на локальні підзадачі, які розв'язуються паралельно, що ідеально відповідає архітектурі сучасних інтелектуальних мереж.

Табл. 1 чітко показує, що для *кожної* з основних детермінованих оптимізаційних задач в енергетиці МНШ є або непридатним, або вкрай неефективним вибором.

Таблиця 1. Аналіз застосовності МНШ до задач в енергетиці

Задача	Тип задачі	Розмірність	Обмеження	Придатність МНШ	Чому? (Ключова проблема)
ELD (без втрат)	Опукла, квадратична	Середня	1 рівність, N нерівностей	Низька	Повільна збіжність (к визначається різницею a_i). Потребує штрафів
ELD (з B-матрицею)	Не опукла, квадратична	Середня	1 нелінійна рівність, N нерівностей	Дуже низька	Погана обумовленість (к велике). Штрафні методи руйнують к
AC OPF	Не опукла, NLP	Висока	Складні нелінійні рівності та нерівності	Непридатний	МНШ – метод <i>безумовної</i> оптимізації. Не може обробити обмеження
DC OPF	Опукла, LP (лінійне програмування)	Висока	Лінійні рівності / нерівності	Непридатний	Це задача лінійного програмування. Потрібні симплекс-метод або IPM
Оцінка стану (State Estimation, SE)	Нелінійні МНК	Висока	-	Низька	Зазвичай вирішується як нелінійні МНК (Метод найменших квадратів, Гаус – Ньютон), що набагато швидше

Обмежена ефективність МНШ мотивувала розробку досконаліших алгоритмів. Щоб зрозуміти, чому вони кращі, проаналізуємо їх у порівнянні з МНШ.

Особливість «найшвидшого» спуску. Як було продемонстровано, назва «найшвидший» є такою, що не повною мірою відображає реальні характеристики збіжності: метод є найшвидшим лише локально (тобто, з погляду

миттєвої зміни f), але призводить до однієї з найповільніших глобальних траєкторій збіжності.

Фундаментальна проблема МНШ полягає в строгій локальності [5]: на кожному кроці k не відбувається накопичення історії попередніх напрямків p_0, \dots, p_{k-1} . Він використовує лише інформацію першого порядку (∇f_k) і робить це неефективно, оскільки напрямок $p_k = -\nabla f_k$ (де

I – одинична матриця) ігнорує будь-яку інформацію про кривизну задачі.

Методи другого порядку: Метод Ньютона. Метод Ньютона використовує інформацію другого порядку (есіан $(\nabla^2 f(x))$) для побудови повної квадратичної моделі функції в точці x_k і здійснює ітераційний перехід безпосередньо в мінімум цієї моделі [22]. Ітерація методу Ньютона (31):

$$x_{k+1} = x_k + \alpha_k p_k^{Newton} \quad (31)$$

де $0 \leq \alpha_k \leq 1$ – довжина кроку;

$$p_k^{Newton} = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$$

Тобто з явним записом напрямку (32):

$$x_{k+1} = x_k + \alpha_k p_k^{Newton} = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \quad (32)$$

МНШ робить крок $p_k = -I \cdot \nabla f_k$. Метод Ньютона використовує «ідеальну» метрику: $p_k = -H^{-1} \nabla f_k$. Множення на H^{-1} (обернений гесіан) діє як *перетворення простору* (масштабування), яке перетворює витягнуті «еліпси» рівнів f на ідеальні «кола». У цьому трансформованому просторі напрямом $-\nabla f$ вказує точно на мінімум.

Переваги [23]:

- **Квадратична збіжність** ($\|x_{k+1} - x^*\| \approx C \|x_k - x^*\|^2$). Це означає, що кількість *правильних* значущих цифр подвоюється на кожній ітерації. Збіжність надзвичайно швидка.
- **Афінна інваріантність**, тобто метод нечутливий до числа обумовленості κ . Він знайде мінімум квадратичної функції з $\kappa = 1000$ так само швидко (за 1 крок), як і для $\kappa = 1$.

Недоліки (в контексті енергетичних систем) [24]:

- Метод Ньютона є непрактичним для великомасштабних задач, таких як OPF, у своєму чистому вигляді:
- Обчислення гесіана: $\nabla^2 f(x)$ має n^2 елементів. Для системи з $n = 50\,000$ вузлів це неможливо.
- Зберігання гесіана: Потрібна пам'ять $O(n^2)$.
- Розв'язання системи: Розв'язання $Hp = -g$ вимагає $O(n^3)$ операцій (для щільного H) або $O(n^{1.5}) - O(n^2)$ (для розрідженого H , як у OPF). Це все одно надто дорого для кожної ітерації.

«Розумні» методи першого порядку: Метод спряжених градієнтів (CG). Метод спряжених градієнтів (Conjugate Gradients, CG) – це ефективна модифікація вирішення проблеми відсутності накопичення передісторії та «зигзагів» МНШ, яке не вимагає гесіана [25].

Ідея полягає в тому, щоб оптимізувати вибір напрямку p_k . Напрямок p_k будується як лінійна комбінація поточного негативного градієнта та попереднього напрямку (33):

$$p_{k+1} = -\nabla f(x_{k+1}) + \beta_k p_k \quad (33)$$

Коефіцієнт β_k (наприклад, за формулою Флетчера – Рівза або Полака – Ріб'єра) вибирають так, щоб

напрямки p_k були Q-спряженими (для квадратичної задачі $f(x) = \frac{1}{2} x^T Q x + \dots$), тобто $p_i^T Q p_j = 0$ для $i \neq j$.

Переваги:

1. **N-крокова збіжність** для квадратичної задачі розмірності $n \times n$, CG гарантовано знаходить точний мінімум за не більше ніж n ітерацій.
2. **Ефективність.** Він уникає «зигзагів» МНШ, оскільки напрямки p_k враховують передісторію кроків і залишаються спряженими.
3. **Вартість**, тобто складність ітерації ($O(n)$) та пам'ять ($O(n)$) – такі самі, як у МНШ.

Отже, для великомасштабних квадратичних задач (або нелінійних, де він застосовується ітеративно), CG у всіх відношеннях перевершує МНШ.

Квазіньютонівські методи (BFGS / L-BFGS) намагаються досягти швидкої збіжності Ньютона, не уникаючи значних обчислювальних витрат та обернення гесіана [26].

Ідея полягає в тому, щоб не обчислювати $H_k = \nabla^2 f(x_k)$. Замість цього будувати *апроксимацію* оберненого гесіана, $B_k \approx [\nabla^2 f(x_k)]^{-1}$, на кожному кроці.

Ітерація: $x_{k+1} = x_k - \alpha_k B_k \nabla f_k$.

Оновлення: B_{k+1} отримують з B_k додаванням поправки низького рангу, яка використовує *лише* інформацію з градієнтів:

$$1. \quad s_k = x_{k+1} - x_k$$

$$2. \quad y_k = \nabla f_{k+1} - \nabla f_k$$

B_{k+1} = (формула оновлення BFGS, що залежить від B_k, s_k, y_k)

Найуспішнішою формулою оновлення є BFGS (Бройден – Флетчер – Гольдфарб – Шанно) [27].

Переваги (BFGS): Суперлінійна збіжність (майже така сама швидка, як у Ньютона).

- Проблема (BFGS): Потрібно зберігати та оновлювати матрицю B_k , що вимагає $O(n^2)$ пам'яті. Це знову неможливо для $n = 50\,000$.
- Рішення: L-BFGS (Limited-memory BFGS)

Це високоефективне алгоритмічне рішення.

Ідея тут полягає в тому, щоб не зберігати повну матрицю B_k . Зберігати лише m останніх пар векторів $\{s_i, y_i\}$ (де m мале, наприклад, $m = 10$ або $m = 20$).

Алгоритм: Напрямок $p_k = -B_k \nabla f_k$ не обчислюється *явно*. Замість цього він *відтворюється* динамічно (в процесі обчислень) за допомогою двохпрохідного рекурсивного алгоритму, який використовує лише ці m пар векторів.

Переваги (L-BFGS) [28]:

- **Збіжність:** Суперлінійна (хоч і повільніша, ніж у повного BFGS, але значно перевищує швидкість збіжності МНШ).

- Пам'ять: Вимоги до пам'яті лінійні: $O(m \cdot n)$.
- Обчислення: Вартість ітерації лінійна: $O(m \cdot n)$.

L-BFGS став де-факто стандартом для великомасштабної безумовної нелінійної оптимізації в багатьох галузях. Табл. 2 візуалізує аналіз оптимізаційних алгоритмів. Вона чітко показує, чому МНШ поступається в ефективності: CG пропонує ту саму вартість ($O(n)$), але набагато кращу збіжність. L-BFGS пропонує майже таку саму низьку (лінійну) вартість, але майже Ньютонівську збіжність і низьку чутливість до k .

Таблиця 2. Порівняльний аналіз збіжності методів

Метод	Швидкість збіжності	Вартість ітерації (Обчислення)	Вимоги до пам'яті	Чутливість до k
Найшвидший спуск (SD)	Лінійна (повільна)	$O(n)$	$O(n)$	Дуже висока
Спряжені градієнти (CG)	Суперлінійна (n-крокова для квадр.)	$O(n)$	$O(n)$	Середня
L-BFGS ($m=10$)	Суперлінійна (швидка)	$O(m \cdot n)$	$O(m \cdot n)$	Низька
Метод Ньютона	Квадратична (найшвидша)	$O(n^3)$	$O(n^2)$	Відсутня

Таким чином, МНШ не має практичних переваг у застосуванні до детермінованих задач.

Теоретичне та прикладне значення МНШ. Аналіз, проведений у цій роботі, приводить до однозначного висновку: як детермінований інструмент для вирішення класичних, великомасштабних, погано обумовлених задач оптимізації в енергетиці (таких як OPF, ELD, SE), метод найшвидшого спуску є дещо обмежено придатним для сучасних задач високої розмірності та практично непридатним.

Його повільна лінійна збіжність, яка суттєво погіршується зі зростанням числа обумовленості k , робить його неконкурентоспроможним порівняно з методами спряжених градієнтів, L-BFGS, і тим більше з методами внутрішньої точки, які домінують у сучасних промислових вирішувачах OPF.

Однак цінність МНШ сьогодні є педагогічною. Це базовий (архетипний) алгоритм нелінійної оптимізації. Розуміння його недоліків (критична залежність від k , зигзагоподібна поведінка) є необхідною умовою для розуміння, чому були винайдені і чому так добре працюють L-BFGS, CG та IPM.

Концептуальна трансформація градієнтних методів: Стохастичний градієнтний спуск (SGD). З появою нової парадигми оптимізації, керованої даними – машинного навчання (Machine Learning, ML) – МНШ набув нового етапу актуалізації [29].

Сучасна енергетика дедалі більше покладається на ML для вирішення нових задач [30]:

- Прогнозування генерації ВДЕ (вітер, сонце).
- Прогнозування навантаження (коротко- та довгострокове).
- Навчання моделей попиту (Demand Response).
- Оцінка стану на основі PMU (Phasor Measurement Units).

Математично більшість задач навчання (наприклад, тренування нейронної мережі) зводяться до задачі мінімізації емпіричного ризику (34) [31]:

$$\min_{w \in \mathbb{R}^n} f(w) = \frac{1}{N} \sum_{i=1}^N L(w, data_i) \quad (34)$$

де w – параметри моделі (ваги мережі), L – функція втрат, а N – кількість точок даних, яка може бути величезною (мільйони або мільярди).

Проблема в тому, щоб зробити один крок МНШ, L-BFGS або Ньютона, потрібно обчислити повний градієнт $\nabla f(w)$ (35) [32]:

$$\nabla f(w) = \frac{1}{N} \sum_{i=1}^N \nabla_w L(w, data_i) \quad (35)$$

Це вимагає N обчислень, тобто «перебору» всієї бази даних. Це надто дорого. Рішення в застосуванні стохастичного градієнтного спуску (SGD) [33], де на кожному кроці k замість повного градієнта, ми беремо лише одну точку (або невеликий «пакет») даних i та робимо крок у напрямку її градієнта (стохастичного градієнта) (36):

$$w_{k+1} = w_k - \alpha_k \nabla_w L(w_k, data_i) \quad (36)$$

Напрямок $\nabla_w L(w_k, data_i)$ є «шумною», але незміщеною ($E[\nabla_w L] = \nabla f(w)$) оцінкою повного градієнта. По суті, SGD – це МНШ з дуже «шумним» градієнтом.

Для наочного та кількісного підтвердження аналітичних висновків щодо впливу гетерогенності системи на ефективність оптимізації, було проведено чисельне моделювання. Як тестове середовище використано класичну задачу мінімізації додатно визначеної квадратичної функції вигляду (15), що дає змогу ізольовано керувати числом обумовленості k через власні значення матриці Гессе Q , імітуючи різницю масштабів параметрів реальних енергосистем.

Для обчислення кроку МНШ застосовувався точний лінійний пошук згідно з (8). З метою забезпечення математичної строгості експерименту та відтворення найгіршого сценарію збіжності (за нерівністю Канторовича), початкова точка x_0 генерувалася з урахуванням напрямків головних осей еліпсоїда. Комплексні результати порівняльного моделювання наведено на рисунку.

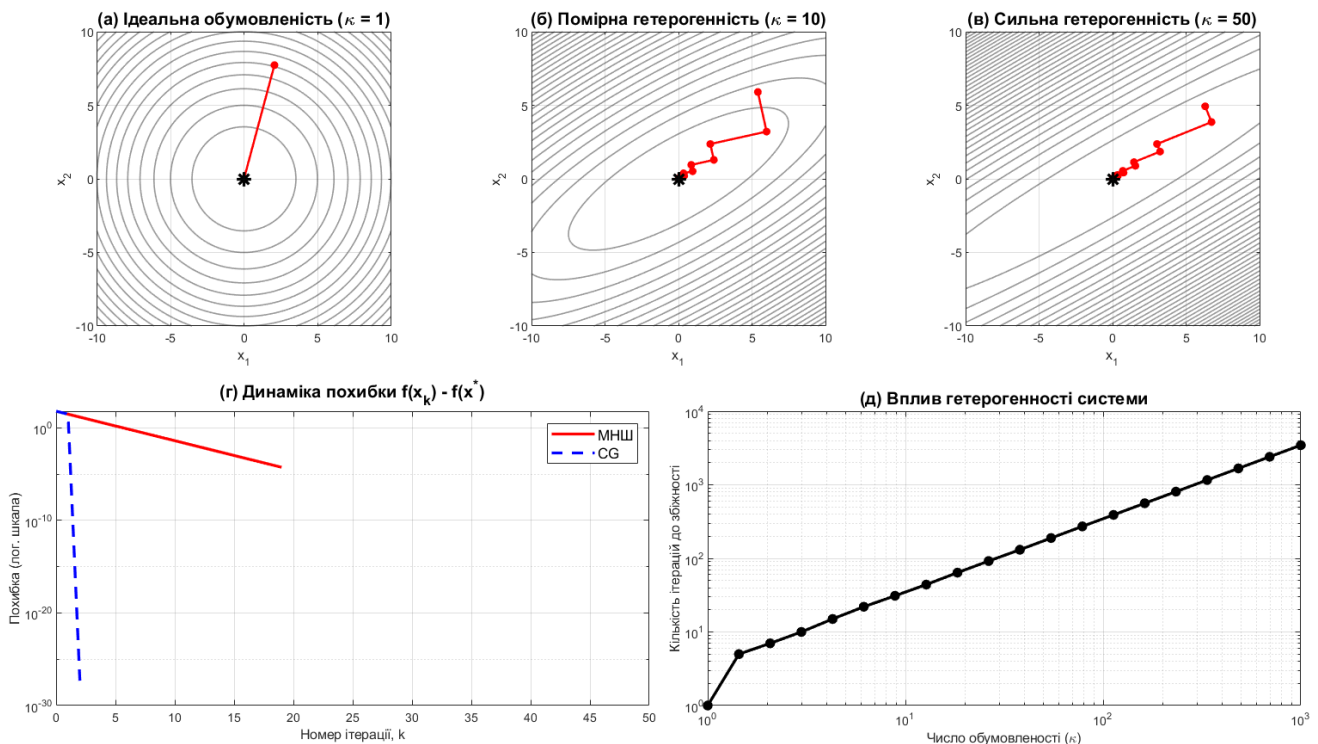


Рисунок. Чисельне дослідження ефективності МНШ на тестовій квадратичній задачі

Еволюція траєкторії збіжності за умов: (а) ідеальної обумовленості, $\kappa = 1$; (б) помірної гетерогенності, $\kappa = 10$; (в) сильної гетерогенності, $\kappa = 50$ (демонстрація зигзагоподібної поведінки). Графік (г) ілюструє повільну лінійну збіжність МНШ порівняно з N-кроковою збіжністю методу спряжених градієнтів (CG). Залежність (д) підтверджує теоретичні висновки: при значеннях $\kappa > 10^3$, характерних для енергосистем, необхідна кількість ітерацій зростає експоненційно, роблячи метод обчислювально неефективним.

Важливо зазначити, що хоча SGD і класичний МНШ мають спільне математичне коріння (обчислення градієнта), вони представляють різні парадигми оптимізації. SGD є ефективним алгоритмом для навчання нейронних мереж не завдяки властивостям класичного детермінованого спуску, а, імовірно, завдяки специфічним ефектам стохастичності. Напрямок спуску, що є «шумною» оцінкою повного градієнта, діє як неявна регуляризація, допомагаючи моделі уникати локальних мінімумів та сідлових точок. Таким чином, спільна математична база адаптувалася під принципово нові обчислювальні потреби сучасних Data-driven рішень в енергетиці.

Висновки

Метод найшвидшого спуску є класичним базовим алгоритмом, чия роль в оптимізації енергетичних систем кардинально змінилася.

Метод найшвидшого спуску, незважаючи на свою фундаментальну теоретичну та педагогічну важливість, демонструє низьку швидкість збіжності та обчислювальну неефективність при застосуванні до реальних, погано

обумовлених оптимізаційних задач в енергетиці. Його аналіз є критично важливим не через його пряме сучасне застосування, а як відправна точка для розуміння необхідності та еволюції до більш досконалих методів (наприклад, спряжених градієнтів, квазіньютонівських методів та методів внутрішньої точки), які сьогодні домінують у галузі.

Метод найшвидшого спуску, незважаючи на свою фундаментальну теоретичну та педагогічну важливість, демонструє суттєві обмеження при застосуванні до реальних погано обумовлених оптимізаційних задач в енергетиці. Згідно з математичним аналізом та чисельним моделюванням, швидкість збіжності методу знижується пропорційно κ^2 . Для типових задач AC-OPF, де $\kappa \sim 10^3$ через різницю масштабів імпедансів ліній та характеристик генерації, це вимагає на 2–3 порядки більше ітерацій порівняно з методами внутрішньої точки (IPM) або методами спряжених градієнтів.

Теоретичне та прикладне значення МНШ в енергетиці полягає не в його поточному використанні, а в тому, як його обчислювальні обмеження на реальних задачах стимулювали розвиток альтернативних підходів. Проблеми, які МНШ не міг вирішити (наприклад, AC-OPF), змусили дослідників не стільки покращувати МНШ, скільки спростувати проблему, що, наприклад, привело до широкого розповсюдження лінеаризованих моделей (наприклад, DC-OPF (Optimal Power Flow постійного струму)). Таким чином, МНШ – це частково і поява спрощених моделей, які досі використовуються, оскільки повноцінні задачі були нерозв'язними для алгоритмів того часу.

Подяка. Частина досліджень виконано авторами С. В. Бойченком і В. О. Коровушкіним у межах науково-дослідного проекту, що фінансується за рахунок державного бюджету України (за Наказом Міністерства освіти і науки України від 09.01.2026 № 23), а також ініціативних наукових тематик кафедри автоматизації електротехнічних і мехатронних комплексів КПІ ім. Ігоря Сікорського.

ПОСИЛАННЯ

- Status of Power System Transformation: Leading Topics of 2024. National Renewable Energy Laboratory, 2024.
URL: <https://docs.nrel.gov/docs/fy24osti/91357.pdf> (дата звернення: 07.01.2026)
- Коровушкін В. О., Босак А. В. Підвищення надійності електропостачання від нетрадиційних та відновлюваних джерел енергії. *Vidnovluvana energetika*. 2025. № 2(81). С. 88–96. URL: [https://doi.org/10.36296/1819-8058.2025.2\(81\).88-96](https://doi.org/10.36296/1819-8058.2025.2(81).88-96) (дата звернення: 12.01.2026).
- Modern optimization technologies in hybrid renewable energy systems: a systematic review of research gaps and prospects for decisions / V. Korovushkin et al. *Energies*. 2025. Vol. 18, no. 17. P. 4727.
URL: <https://doi.org/10.3390/en18174727> (date of access: 12.01.2026).
- Targeting energy savings? Better on primary than final energy and less on intensity metrics / M. Rodríguez та ін. *Energy economics*. 2023. С. 106797. URL: <https://doi.org/10.1016/j.eneco.2023.106797> (дата звернення: 07.01.2026).
- Wood A. J., Wollenberg B. F., Sheblé G. B. *Power Generation, Operation, and Control*. 3rd ed. Hoboken, NJ : Wiley, 2013. 656 p.
- Smart grid, demand response and optimization: a critical review of computational methods / U. Assad та ін. *Energies*. 2022. Т. 15, № 6. С. 2003. URL: <https://doi.org/10.3390/en15062003> (дата звернення: 07.01.2026).
- Castillo A., O'Neill R. P. Survey of Approaches to Solving the ACOF. Optimal Power Flow Paper 4. Federal Energy Regulatory Commission, 2013.
URL: <https://www.ferc.gov/sites/default/files/2020-05/acopf-4-solution-techniques-survey.pdf> (дата звернення: 07.01.2026).
- Using gradient descent: cost decreases, then increases. Mathematics Stack Exchange.
URL: <https://math.stackexchange.com/questions/975180/using-gradient-descent-cost-decreases-then-increases> (дата звернення: 07.01.2026).
- International Energy Agency. *World Energy Outlook 2023*. Paris: IEA, 2023.
- Wood A. J., Wollenberg B. F., Sheblé G. B. *Power Generation, Operation, and Control*. 3rd ed. New York: Wiley, 2013.
- Kundur P. *Power System Stability and Control*. New York: McGraw-Hill, 1994.
- Blaabjerg F., Yang Y., Yang D., Wang X. Distributed Power-Generation Systems and Protection. *Proceedings of the IEEE*. 2017. Vol. 105. No. 7. P. 1311–1331.
- Low S. H. Convex Relaxation of Optimal Power Flow. *Proceedings of the IEEE*. 2014. Vol. 102. No. 1. P. 1–20.
- Dommel H. W., Tinney W. F. Optimal Power Flow Solutions. *IEEE Transactions on Power Apparatus and Systems*. 1968. Vol. PAS-87. No. 10. P. 1866–1876.
- Duffin R. J. *Nonlinear Networks: II* // *Proceedings of the IRE*. 1959. Vol. 4., No. 3. P. 388–394.
- Nocedal J., Wright S. J. *Numerical Optimization*. 2nd ed. New York: Springer, 2006.
- Bertsekas D. P. *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1999.
- Fletcher R. *Practical Methods of Optimization*. 2nd ed. New York: Wiley, 1987.
- Wolfe P. Convergence Conditions for Ascent Methods // *SIAM Review*. 1969. Vol. 11. No. 2. P. 226–235.
- Hestenes M. R., Stiefel E. Methods of Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards*. 1952. Vol. 49. No. 6. P. 409–436.
- Dennis J. E., Schnabel R. B. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs: Prentice-Hall, 1983.
- Nesterov Y. *Introductory Lectures on Convex Optimization*. Boston: Springer, 2004.
- Nesterov Y. A Method for Solving the Convex Programming Problem of Unconditional Minimization // *Soviet Mathematics Doklady*. 1983. Vol. 27. P. 372–376.
- Park J. B., Park K. S., Won J. R., Lee K. Y. Economic Load Dispatch Considering Transmission Losses // *IEEE Transactions on Power Systems*. 2005. Vol. 20. No. 1. P. 343–352.
- Kron A. *Tensor Analysis of Networks*. New York: Wiley, 1939.
- Coelho L. S. Constrained Optimization in Power Systems // *Electric Power Systems Research*. 2009. Vol. 79. P. 615–623.
- Bukhsh W. A., Grothey A., McKinnon K. I. M., Trodden P. A. Local Solutions of Optimal Power Flow. *IEEE Transactions on Power Systems*. 2013. Vol. 28. No. 4. P. 4780–4788.

28. Carpentier J. Optimal Power Flows // International Journal of Electrical Power & Energy Systems. 1989. Vol. 11. No. 1. P. 3–15.
29. Research on photovoltaic power prediction using an LSTM recurrent neural network / S. Boichenko et al. Studies in systems, decision and control. Cham, 2024. P. 421–443. URL: https://doi.org/10.1007/978-3-031-67091-6_19 (дата звернення: 07.01.2026).
30. Boyd S., Vandenberghe L. Convex Optimization. Cambridge: Cambridge University Press, 2004.
31. Spall J. C. Introduction to Stochastic Search and Optimization. Hoboken: Wiley, 2003.
32. Castillo A. A Review of AC Optimal Power Flow Methods // Electric Power Systems Research. 2013. Vol. 95. P. 223–232.
33. Bottou L. Stochastic Gradient Descent Tricks // In: Neural Networks: Tricks of the Trade. Springer, 2012. P. 421–436.

STEEPEST DESCENT METHOD: LIMITATIONS ANALYSIS AND EVOLVING APPLICABILITY IN MODERN POWER SYSTEMS

Received Mar. 07, 2026; accepted Jun. 26, 2026
Available online June. 30, 2026

Korovushkin V.¹, Boichenko S.²,
Kuznietsov M.³, Danilin O.⁴

Author for correspondence: Korovushkin Vitalii,
e-mail: vitalijkorovuskin@gmail.com

¹ PhD student
<https://orcid.org/0000-0002-7571-7124>

² Dr. of Sciences (Tech.)
<https://orcid.org/0000-0002-2489-4980>

³ Dr. of Sciences (Tech.)
<https://orcid.org/0000-0002-0497-7439>

⁴ Cand. of Sciences (Tech.)
<https://orcid.org/0000-0003-3207-1156>

^{1, 2, 4} National Technical University of
Ukraine “Igor Sikorsky Kyiv Polytechnic In-
stitute”, Kyiv, Ukraine

³ Institute of renewable energy, NAS of
Ukraine, Kyiv, Ukraine

Abstract. *The article provides a systematic critical analysis of the steepest descent (SD) method in the context of optimizing modern energy systems. Despite its historical significance as the archetype of iterative algorithms, SD has been proven to be significantly limited in solving large-scale problems such as economic load dispatch (ELD) and optimal power flow (AC-OPF). The paper investigates the mathematical reasons for this phenomenon in detail: it shows that the physical heterogeneity of power system components (different scales of line impedances and generation characteristics) leads to a poor condition of the objective function Hessian. A high condition number (κ) causes a “zigzag” convergence trajectory, which makes the method unsuitable for real-time operational control. Based on a comparative analysis with second-order methods (Newton's method) and quasi-Newton algorithms (L-BFGS), the transition to interior point methods as an industry standard for constrained problems is justified. The key novelty of the work lies in the study of the relevance of gradient methods in modern computational paradigms in the era of artificial intelligence. The authors demonstrate a paradigm shift: characteristics that were limiting factors in deterministic optimization (noisy trajectory and simplified gradient processing) have been transformed into decisive advantages of stochastic gradient descent (SGD). It has been proven that it is precisely this stochastic nature that allows neural networks to be effectively trained for the tasks of forecasting RES generation and demand management, transforming the traditional method into the foundation of modern data-driven solutions in the energy sector.*

Keywords: *Power System Optimization; AC Optimal Power Flow (AC-OPF); Steepest Descent Method; Stochastic Gradient Descent (SGD); Ill-conditioned problems; Convergence analysis.*

List of Symbols and Abbreviations

AC-OPF – Alternating Current Optimal Power Flow
BFGS – Broyden–Fletcher–Goldfarb–Shanno
ELD – Economic Load Dispatch
IPM – Interior Point Methods
KKT – Karush-Kuhn-Tucker conditions

L-BFGS – Limited-memory BFGS
ML – Machine Learning
NLP – Nonlinear Programming
SD – Steepest Descent
SGD – Stochastic Gradient Descent
SQP – Sequential Quadratic Programming

Introduction. Modern energy systems are among the most complex engineering and cyber-physical systems ever created by humankind. Their operation—from the second-by-second balancing of generation and demand to long-term development planning—is permeated by fundamental optimization challenges. The need for optimization in the energy sector is driven by three main factors: economic efficiency, system reliability, and the challenges of decarbonization [1–3].

From an economic standpoint, even a slight (e.g., 0.1%) improvement in the efficiency of the integrated power system

translates into millions in fuel cost savings [4]. Tasks such as Economic Load Dispatch (ELD) and Optimal Power Flow (OPF) directly minimize total generation costs [5].

From a reliability perspective, the system must operate within strict physical constraints: thermal limits of transmission lines, voltage limits at nodes, generation reserves, and stability constraints. Optimization (in particular, OPF) is a tool that finds the cheapest operating mode without violating any of these safety constraints [6].

Finally, the integration of stochastic renewable energy sources (RES), such as wind and solar power plants,

introduces a high degree of uncertainty. This requires new optimization approaches capable of operating under stochastic conditions, as well as managing demand response and energy storage systems [7]. Optimization problems become particularly important in situations with limited access to centralized energy supply and the transition to distributed generation (creation of microgrid systems, shared use of energy storage facilities, selection of RES connection points to reduce electricity traffic and losses, issues of mutual energy exchange, etc.).

Mathematically, most of these problems (in particular, AC-OPF—Alternating Current Optimal Power Flow) are formulated as large-scale, nonlinear, non-convex, and strongly constrained nonlinear programming (NLP) problems. The application of distributed generation from RES requires consideration of modular structure and a transition to integer programming problems. The dimensionality of such problems for real-world systems can reach tens or hundreds of thousands of variables and constraints, and the requirement to solve them in real time (or close to it) imposes strict performance constraints on computational methods [8].

The aim of this work is to quantitatively assess the limitations of the steepest descent method in solving classical power system optimization problems (ELD, AC-OPF) due to the high conditionality (κ) and heterogeneity of power systems, as well as to conduct a comparative analysis of its effectiveness against modern optimization methods. At the same time, the study aims to rethink the role of the method in modern conditions: demonstrating its transition from a traditional tool for deterministic optimization to the basis of stochastic gradient descent (SGD), which is critically important for training neural networks in power system problems (forecasting RES, demand).

To understand the limitations of the method, it is first necessary to clearly establish its mathematical foundations. SD is a method for solving the unconstrained optimization problem.

Formulation of the unconstrained optimization problem. Consider the problem of finding a local minimum of the differentiable function $f(x)$ [5](1):

$$\min_{x \in \mathbb{R}^n} f(x) \quad (1)$$

where $f: \lambda_{\max} \rightarrow \mathbb{R}$ is a smooth function of at least class C^1 (i.e., it has continuous first-order partial derivatives).

The key tools of the analysis are:

1. Gradient: The vector of first partial derivatives, $\nabla f(x) \in \mathbb{R}^n$, which points in the direction of the steepest rise of the function at the point x .
2. Hessian: Matrix of second partial derivatives, $\nabla^2 f(x) \in \mathbb{R}^{n \times n}$, which describes the curvature of the function at a point x .

A first-order necessary condition for a point x^* to be a local minimum is $\nabla f(x^*) = 0$. Such points are called stationary points. The Gradient Descent method, like most iterative methods, attempts to find precisely such a stationary point.

The idea behind the SD is to move at each step from the current point x_k in the direction p_k , which ensures the greatest local decrease in the value of the function f .

To formalize this, let us consider the first-order Taylor series expansion of the function f around the point x [5] (2):

$$f(x + \alpha p) \approx f(x) + \alpha \nabla f(x)^T p \quad (2)$$

where p — is a unit direction vector ($\|p\|_2 = 1$), and $\alpha > 0$ — small step size (assuming the presence of an orthonormal basis in the domain). We need to find a direction p that minimizes the change $f(x + \alpha p) - f(x)$, which is equivalent to minimizing the product $\nabla f(x)^T p$.

So, the problem is solved (3):

$$\min_p \nabla f(x)^T p \text{ provided that } \|p\|_2 = 1 \quad (3)$$

According to the Cauchy–Schwarz inequality (4):

$$|\nabla f(x)^T p| \leq \|\nabla f(x)\|_2 \cdot \|p\|_2 \quad (4)$$

The minimum (i.e., the most negative value) is achieved when p is collinear with and opposite in direction to $\nabla f(x)$. That is, (5):

$$p = -\frac{\nabla f(x)}{\|\nabla f(x)\|_2} \quad (5)$$

This direction p is called the direction of the steepest descent. It locally guarantees the steepest decrease in the objective function.

Based on this conclusion, the iterative process of the gradient descent method is constructed as follows: for a given initial point x_0 [5]:

Algorithm 1: The Steepest Descent Method (SD)

1. For $k = 0, 1, 2, \dots$
2. If $\nabla f(x_k) = 0$, then Stop
3. Calculate the descent direction: $p_k = -\nabla f(x_k)$
4. Calculate the step size: $\alpha_k > 0$ (using the linear search procedure)
5. Update the point: $x_{k+1} = x_k + \alpha_k p_k$
6. End

The algorithm has two key computational components in each iteration:

3. Calculating the direction p_k (requires calculating the gradient $\nabla f(x_k)$).

1. Calculation of the step size α_k (a procedure known as line search or one-dimensional minimization).
2. The choice of α_k is critical to the method's effectiveness.

Line Search. The theoretically optimal approach involves selecting a step size α_k , that minimizes the function f along the direction p_k [5] (6):

$$\alpha_k = \arg \min_{\alpha > 0} f(x_k + \alpha_k p_k) = \arg \min_{\alpha > 0} \varphi(\alpha) \quad (6)$$

This is, in essence, a one-dimensional optimization problem. The optimality condition for α_k gives $\varphi'(\alpha_k) = 0$. Applying the chain rule, we get (7):

$$\varphi'(\alpha_k) = \nabla f(x_k + \alpha_k p_k)^T \cdot p_k = 0 \quad (7)$$

This means that in an exact linear search, the new gradient $\nabla f(x_{k+1})$ is orthogonal to the previous direction p_k . Since $p_k = -\nabla f(x_k)$, also implies that $\nabla f(x_{k+1})^T \nabla f(x_k) = 0$. Consecutive gradients are orthogonal.

It is precisely this property that determines the characteristic “zigzag” behavior of the SD [8]. Let us imagine a narrow, elongated “valley” of the objective function. On the “slope” of the valley, the gradient points predominantly across the valley (in the direction of the steepest slope), rather than along it toward the minimum. The exact search gradient descent takes a step and moves to a point on the opposite “slope” at x_{k+1} , where the new gradient $\nabla f(x_{k+1})$ is orthogonal to p_k . Since p_k was nearly perpendicular to the optimal path (along the “bottom” of the valley), the new gradient $\nabla f(x_{k+1})$ will be nearly in the opposite direction to $\nabla f(x_k)$. The algorithm will perform iterative transitions (zig-zagging) between the valley slopes, making very slow progress toward the true minimum.

Thus, “optimality” (exact step search) at each step leads to a globally suboptimal trajectory.

The only case where an exact search is computationally justified is a positive-definite quadratic function [9]. In this case, α_k has the closed-form expression (8):

$$\alpha_k = \frac{\nabla f_k^T \cdot \nabla f_k p_k}{\nabla f_k^T \cdot Q \nabla f_k p_k} \quad (8)$$

where $\nabla f_k = Qx_k - b$

Inexact line search (Wolfe Conditions). In general, for non-linear functions, finding the exact minimum α_k is impractical due to computational costs (resource costs may exceed the cost of the SD iteration itself).

In practice, inexact line search procedures are used, which guarantee a “sufficiently good” step. The de facto standard is the Wolfe Conditions, which require that α_k satisfy two conditions [10]:

The Armijo Condition (sufficient decrease) (9):

$$f(x_k + \alpha_k p_k) \leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^T \cdot p_k \quad (9)$$

where $c_1 \in (0, 1)$, typically $c_1 = 10^{-4}$. This ensures that the step size α_k is not “too large” and results in a real decrease in the function.

The curvature condition (10):

$$\nabla f(x_k + \alpha_k p_k)^T \cdot p_k \geq c_2 \nabla f(x_k)^T \cdot p_k \quad (10)$$

where $c_2 \in (c_1, 1)$, typically $c_2 = 0.9$.

The existence of an α_k , that satisfies both conditions is guaranteed for smooth functions that are bounded from below. The use of Wolf’s conditions is key to proving the global convergence of the method.

Theoretical convergence guarantees. The global convergence of the SD (i.e., convergence to a stationary point x^* from any x_0) is guaranteed under fairly mild conditions. The L-smoothness condition is key.

Definition (L-smoothness): A function f is said to be conditionally L-smooth if its gradient ∇f is Lipschitz with a constant $L > 0$ [11], i.e., (11):

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L \|x - y\|_2 \quad \forall x, y \in \mathbb{R}^n \quad (11)$$

If $f \in C^2$, this is equivalent to the largest eigenvalue of the Hessian $\nabla^2 f(x)$ being bounded by L , i.e. $\nabla^2 f(x) \leq LI$.

Theorem 1 (Convergence for L-smooth functions) [12]: If f is L-smooth and bounded below, then the SD with an inexact search (satisfying Wolfe’s conditions) or with a constant step size $\alpha_k = \alpha \in (0, 2/L)$ guarantees that $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\|_2 = 0$.

This theorem states that we will eventually converge to a point where the gradient is zero. However, it says nothing about how quickly this will happen.

Analysis of the rate of convergence. To analyze the rate, we need a stronger condition—strong convexity. *Definition (m-strong convexity) [13]:* A function f is called m-strongly convex (with $m > 0$) if (12):

$$f(y) \geq f(x) + \nabla f(x_k)^T (y - x) + \frac{m}{2} \|y - x\|_2^2 \quad \forall x, y \in \mathbb{R}^n \quad (12)$$

If $f \in C^2$, this is equivalent to the smallest eigenvalue of the Hessian $\nabla^2 f(x)$ being bounded from below by m , $\nabla^2 f(x) \geq mI$.

Theorem 2 (Linear convergence for strongly convex functions) [14]:

If f is L-smooth and m-strongly convex, then the SD with exact linear search or with an optimally chosen constant step converges to a unique global minimum x^* at a linear (or geometric) rate (13, 14):

$$f(x_k) \geq f(x^*) \leq (1 - C)^k (f(x_0) - f(x^*)) \quad (13)$$

$$\|x_k - x^*\|_2^2 \leq (1 - C)^k \|x_0 - x^*\|_2^2 \quad (14)$$

where $C > 0$ — is the convergence constant.

At first glance, “linear convergence” seems satisfactory. However, there is a dependence on the value of the constant C . For the quadratic function (15):

$$f(x) = \frac{1}{2} x^T Q x - b^T x, \quad Q > 0, \quad (15)$$

the constants L and m are simply the largest (λ_{\max}) and smallest (λ_{\min}) eigenvalues of the matrix Q . In this case, it can be shown that (16):

$$f(x_k) - f(x^*) \leq \left(\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \right)^2 (f(x_0) - f(x^*)) \quad (16)$$

The Specifics of Convergence Under Ill-Conditioned Scenarios: κ as a Measure of “Complexity”. Let us introduce a key parameter of any optimization problem: the condition number κ (17).

$$\kappa = \frac{L}{m} = \frac{\lambda_{\max}(Q)}{\lambda_{\min}(Q)} \quad (17)$$

κ measures the ratio of the largest to the smallest curvature; geometrically, it is a measure of how “stretched” the objective function is.

Let us rewrite the expression for the convergence rate in terms of κ (18):

$$f(x_k) - f(x^*) \leq \left(\frac{\kappa - 1}{\kappa + 1} \right)^{2k} (f(x_0) - f(x^*)) \quad (18)$$

Let’s analyze formula (18):

Case 1 (Ideal): $\kappa = 1$.

This means $L = m$, $\lambda_{\max} = \lambda_{\min}$. The Hessian $Q = \lambda I$. The level sets of the function f are ideal concentric circles (or spheres in \mathbb{R}^n).

Convergence factor (19):

$$\left(\frac{1-1}{1+1}\right)^2 = 0 \quad (19)$$

The method converges in **one** step from any initial point. The gradient $\nabla f(x_0) = Qx_0 - b$ always points exactly to the center $x^* = Q^{-1}b$.

Case 2 (Realistic, «bad-conditioned»): $\kappa \gg 1$.

This implies $L \gg m$. The levels of the function f are highly elongated ellipses (ellipsoids).

Convergence factor (20):

$$\frac{\kappa-1}{\kappa+1} = \frac{\frac{L}{m}-1}{\frac{L}{m}+1} = \frac{L-m}{L+m} \approx \frac{L}{L} = 1 \quad (20)$$

More precisely, using the approximation $\frac{1}{1+x} \approx 1 - x$, we obtain (21):

$$\frac{\kappa-1}{\kappa+1} \approx 1 - \frac{2}{\kappa} \quad (21)$$

The rate of convergence (22):

$$\left(1 - \frac{2}{\kappa}\right)^{2k} \quad (22)$$

For example, let $\kappa = 1000$. The convergence factor $\approx (1 - 0.002)^{2k} \approx (0.998)^{2k} \approx (0.996)^{2k}$. This means that at each iteration, the error $f(x_k) - f(x^*)$ decreases by only 0.4%.

To reduce the error by a factor of 1000 (i.e. $0.996^k < 0.001$), we need $k > \frac{\ln(0.001)}{\ln(0.996)} \approx -\frac{6.9}{-0.004} \approx 1725$ iterations.

If $\kappa = 10\,000$, more than 17,000 iterations will be required.

This mathematical observation directly explains why, as of today, the SD is of limited applicability to real power systems. A power system is heterogeneous by nature. It consists of elements with very different physical properties, for example:

1. Lines: a 750 kV transmission line has very low active resistance ($R \ll X$), whereas a local 10 kV distribution line may have $R \approx X$.
2. Generators: The power-cost curves $C(P)$ have different curvatures (second derivative a_i in $a_i P^2 + \dots$).

A base-load nuclear power plant has an almost flat curve ($a_i \approx 0$), whereas a peaking gas turbine has a very steep curve ($a_i \gg 0$). For a wind turbine, the power curve is approximately described by a logistic curve with variable curvature depending on wind speed, while for a solar power plant, the stochastic nature of generation results in a power curve that is theoretically discontinuous at every point.

This physical heterogeneity (different scales of impedances, losses, sensitivities) is mathematically reflected in the Hessian of the objective function (or the Hessian of the Lagrangian in constrained problems) with a wide spread of eigenvalues. λ_{\max} (corresponding to “hard” constraints) will

be large, while λ_{\min} (corresponding to “soft” constraints) will be small.

Thus, any realistic, large-scale power system is guaranteed to generate an optimization problem with a fairly large condition number κ . According to Theorem 2, the SD is guaranteed to have unacceptably slow convergence on any real OPF or ELD problem. Its inefficiency is a direct consequence of the physics of the power system.

Now, let us apply the obtained theoretical conclusions to two key optimization problems in power engineering: economic load dispatch (ELD) and optimal power flow (OPF) [15].

Case 1: Economic Load Dispatch (ELD).

Problem formulation. Classical ELD (lossless) [16] (23):

$$\min_{P_{Gi}} \sum_{i=1}^N C_i P_{Gi} = \sum_{i=1}^N a_i P_{Gi}^2 + b_i P_{Gi} + c_i \quad (23)$$

subject to power balance (24):

$$\sum_{i=1}^N P_{Gi} = P_{Demand} \quad (24)$$

and generation constraints (25):

$$P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max} \quad \forall i. \quad (25)$$

ELD accounting for losses (B-matrix) [17]:

The problem becomes more complex because the power balance becomes nonlinear (26):

$$\sum_{i=1}^N P_{Gi} = P_{Demand} + P_{Loss}(P_G) \quad (26)$$

where the network losses P_{Loss} are approximated by a quadratic form (Kron's formula, B-matrix) [17] (27):

$$P_{Loss}(P_G) = \sum_{i=1}^N \sum_{j=1}^N P_{Gi} B_{ij} P_{Gj} + \sum_{i=1}^N B_{ij} P_{Gi} + B_{00} \quad (27)$$

The objective function and constraints are now nonlinearly related.

Adapting SD to solve ELD. SD — is an unconstrained optimization method. ELD — is a constrained problem. To apply SD, the constraints must be accounted for in some way. There are two common approaches [18]:

1. *Penalty function method: the problem is transformed into an unconstrained one by adding the constraints to the objective function as a “penalty” for violating them (28):*

$$f(P_G) = \sum_{i=1}^N C_i(P_{Gi}) + \mu (\sum_{i=1}^N P_{Gi} - P_D - P_{Loss})^2 + \sum_{i=1}^N v_i \cdot [\max(0, P_{Gi} - P_{Gi}^{\max})]^2 + \sum_{i=1}^N \omega_i \cdot [\max(0, P_{Gi}^{\min} - P_{Gi})]^2, \quad (28)$$

where $\mu > 0$ - is the penalty parameter for violating the power balance; $v_i, \omega_i > 0$ - are penalty parameters for inequality constraints; $P_{Gi}^{\min}, P_{Gi}^{\max}$ - are the lower and upper generation limits; $\max(0, \cdot)$ - imposes a penalty only when constraints are violated. After that, the SD is applied to $f(P_G)$.

2. *Gradient projection method can be used for simple "box" constraints* ($P_{G_i}^{\min}, P_{G_i}^{\max}$). The idea is to take an SD step, and if the new point x_{k+1} goes out of bounds, "project" it back into the feasible region.

Analysis of SD efficiency for ELD. Both approaches are highly inefficient for ELD.

Problem 1 (Bad Conditioning). Even in the simplest case of lossless ELD, the Hessian of the objective function $\sum C_i(P_{G_i})$ is a diagonal matrix $H = \text{diag}(2a_1, \dots, 2a_N)$. The condition number $\kappa = \frac{\max(a_i)}{\min(a_i)}$. As noted earlier, in a real system with cheap base (small a_i) and "expensive" peak (large a_i) generators, κ will be very large. SD will exhibit classic slow, zigzag behavior.

When we add the B-matrix, the Hessian becomes $H = \text{diag}(2a_i) + 2\mu B$. This further worsens the situation.

Problem 2 (Deterioration of condition number due to the use of penalty functions). The integration of the penalty function method critically reduces the computational efficiency of the SD. Since, to precisely satisfy the constraints of the problem, the penalty parameter μ must tend to infinity ($\mu \rightarrow \infty$).

It can be shown that the Hessian of the penalty function $f(P_G)$ becomes increasingly ill-conditioned as μ increases. In fact, $\kappa \rightarrow \infty$ as $\mu \rightarrow \infty$. SD, whose speed critically depends on κ , becomes practically unusable for achieving convergence precisely when we require it to be accurate.

Thus, for ELD, the SD is an extremely inefficient approach for "simple" ELD; methods that explicitly account for constraints (e.g., methods based on solving the Karush-Kuhn-Tucker (KKT) conditions or specialized methods that account for the structure of the problem) are orders of magnitude more efficient. There are studies that use the SD in combination with Lagrange multipliers to solve environmental-economic allocation problems, but these are modified, rather than "pure," versions of the method [19].

Case 2: Optimal Power Flow (OPF), the complexity of the AC-OPF problem. This is the main operational control problem. The AC-OPF problem is large-scale, nonlinear (due to the power flow equations), and non-convex (it has many local minima) [20].

Formally (simplified) (29):

$$\min_{V, \theta, P_G, Q_G} \sum C_i(P_{G_j}) \quad (29)$$

Subject to (constraints—equalities—power flow equations) (30):

For each node i (30), subject to, where:

$x = [P_G, Q_G, V, \theta]$ - is the optimization vector;

μ_P, μ_Q - are penalty parameters for the power balance equation;

v_i^{P+}, v_i^{P-} - are penalty parameters for the P_G constraint;

$\omega_i^{Q+}, \omega_i^{Q-}$ - are penalty parameters for the Q_G constraint;

$\kappa_i^{V+}, \kappa_i^{V-}$ - are penalty parameters for the voltage constraint;

λ_{ij} - are penalty parameters for the line constraints;

N_G - number of generators, N - number of nodes, L - set of lines:

$$\begin{aligned} \min f(x) = & \sum_{i=0}^{N_G} C_i(P_{G_i}) + \mu_P \sum_{i=0}^{N_G} [P_i(V, \theta) - (P_{G_i} - \\ & P_{D_i})]^2 + \mu_Q \sum_{i=0}^{N_G} [Q_i(V, \theta) - (Q_{G_i} - Q_{D_i})]^2 + \\ & \sum_{i=0}^{N_G} v_i^{P+} [\max(0, P_{G_i} - P_{G_i}^{\max})]^2 + \\ & \sum_{i=0}^{N_G} v_i^{P-} [\max(0, P_{G_i}^{\min} - P_{G_i})]^2 + \sum_{i=0}^{N_G} \omega_i^{Q+} [\max(0, Q_{G_i} - \\ & Q_{G_i}^{\max})]^2 + \sum_{i=0}^{N_G} \omega_i^{Q-} [\max(0, Q_{G_i}^{\min} - Q_{G_i})]^2 + \\ & \sum_{i=0}^{N_G} \kappa_i^{V+} [\max(0, |V_i| - V_i^{\max})]^2 + \\ & \sum_{i=0}^{N_G} \kappa_i^{V-} [\max(0, V_i^{\min} - |V_i|)]^2 + \\ & \sum_{i,j \in L} \lambda_{ij} [\max(0, S_{ij}(V, \theta) - S_{ij}^{\max})]^2 \end{aligned} \quad (30)$$

Note: All penalty parameters $\mu, v, \omega, \kappa, \lambda > 0$ are positive and sufficiently large values [21].

Subject to the following constraints (inequalities):

• Generation constraints:

$$P_{G_i}^{\min} \leq P_{G_i} \leq P_{G_i}^{\max}; \quad Q_{G_i}^{\min} \leq Q_{G_i} \leq Q_{G_i}^{\max}$$

$$\text{Voltage constraints: } V_i^{\min} \leq |V_i| \leq V_i^{\max}$$

$$\text{Line thermal constraints: } S_{ij}(V, \theta) \leq S_{ij}^{\max}$$

Application of SD to OPF Direct application of SD to AC-OPF is not possible. This is an unconstrained optimization problem, whereas OPF is defined by thousands of complex nonlinear constraints—equalities and inequalities.

As with ELD, one can attempt to "embed" all these constraints into the objective function via penalties and barriers. This would result in a multidimensional objective function with a complex gradient landscape, an objective function whose Hessian would be extremely ill-conditioned, rendering the SD impractical.

Historically, early attempts to solve the OPF (e.g., Carpentier's "Reduced Gradient Method") utilized gradient information. However, these methods were far more complex than pure SD. They used power flow equations to reduce the dimensionality of the problem (expressing dependent variables in terms of independent ones) and attempted to "move" along the boundary of the feasible region defined by the constraints.

Modern approach. Modern, reliable AC-OPF solvers never use the SD. They are based on methods that explicitly and efficiently handle constraints using second-order information (or its approximation):

1. Interior Point Methods (IPM): The dominant approach. They transform the inequality problem into a sequence of equality problems using barrier functions. They then apply the Newton method to the KKT conditions for this barrier problem. They work with the Lagrangian Hessian

and are extremely efficient for large-scale convex and (often) non-convex problems.

2. Sequential Quadratic Programming (SQP): Another powerful method that, at each iteration, approximates the OPF problem with a quadratic problem (by approximating the Lagrangian) and solves it.
3. The Alternating Direction Method of Multipliers (ADMM), which is gaining increasing popularity for decentralized optimization, particularly in the manage-

ment of microgrids and distributed generation from RES. It allows a complex large-scale problem to be broken down into local subproblems that are solved in parallel, which perfectly matches the architecture of modern smart grids.

Table 1 clearly shows that for each of the main deterministic optimization problems in the power sector, the ADMM is either unsuitable or an extremely inefficient choice.

Table 1. Applicability of SGD to tasks in the energy sector analysis

Task	Problem Type	Dimensionality	Constraints	Applicability of SD	Key issue
ELD (lossless)	Convex, quadratic	Medium	1 equality, N inequalities	Low	Slow convergence (κ) is defined by the difference in a_i). Requires penalties
ELD (with B-matrix)	Non-convex, quadratic	Medium	1 nonlinear equality, N inequalities	Very low	Ill-conditioning (large κ). Penalty methods destroy κ
AC OPF	Non-convex, NLP	High	Complex nonlinear equalities and inequalities	Inapplicable	SD is an unconstrained optimization method. Cannot handle constraints
DC OPF	Convex, LP	High	Linear equalities/inequalities	Inapplicable	A Linear Programming problem. Requires Simplex or IPM
State Estimation (SE)	Nonlinear Least Squares	High	-	Low	Usually solved via nonlinear least squares (Gauss-Newton), which is much faster

The limited effectiveness of the SD motivated the development of more sophisticated algorithms. To understand why they are better, let's analyze them in comparison with the SD.

A peculiarity of the "fastest" descent. As has been demonstrated, the name "fastest" does not fully reflect the actual convergence characteristics: the method is fastest only locally (i.e., in terms of the instantaneous change in f), but leads to one of the slowest global convergence trajectories.

The fundamental problem of the SD lies in its strict locality [5]: at each step k , there is no accumulation of the history of previous directions p_0, \dots, p_{k-1} . It uses only first-order information (∇f_k) and does so inefficiently, since the direction $p_k = -I \cdot \nabla f_k$ (where I is the identity matrix) ignores any information about the curvature of the problem.

Second-order methods: Newton's method. Newton's method uses second-order information (the Hessian ($\nabla^2 f(x)$)) to construct a complete quadratic model of the function at point x_k and performs an iterative transition directly to the minimum of this model [22]. Iteration of Newton's method (31):

$$x_{k+1} = x_k + \alpha_k p_k^{Newton} \quad (31)$$

where $0 \leq \alpha_k \leq 1$ – step size; $p_k^{Newton} = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$

That is, with the explicit expression for the gradient (32):

$$x_{k+1} = x_k + \alpha_k p_k^{Newton} = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \quad (32)$$

The SD takes a step $p_k = -I \cdot \nabla f_k$. The Newton method uses an "ideal" metric: $p_k = -H^{-1} \nabla f_k$. Multiplication by H^{-1} (the inverse Hessian) acts as a space transformation (scaling) that transforms the elongated "ellipses" of the f levels into ideal "circles." In this transformed space, the direction $-\nabla f$ points exactly to the minimum.

Advantages [23]:

- *Quadratic convergence* ($\|x_{k+1} - x^*\| \approx C \|x_k - x^*\|^2$). This means that the number of correct significant digits doubles at each iteration. Convergence is extremely fast.
- *Affine invariance, i.e., the method is insensitive to the condition number κ . It will find the minimum of a quadratic function with $\kappa = 1000$ just as quickly (in 1 step) as for $\kappa = 1$.*

Disadvantages (in the context of power systems) [24]:

Newton's method is impractical for large-scale problems, such as OPF, in its pure form:

- Computing the Hessian: $\nabla^2 f(x)$ has n^2 elements. For a system with $n = 50,000$ nodes, this is impossible.
- Storing the Hessian: Requires $O(n^2)$.
- Solving the system: Solving $Hp = -g$ requires $O(n^3)$ operations (for a dense H) or $O(n^{1.5}) - O(n^2)$ (for a sparse H , as in OPF). This is still too expensive for each iteration.

“Smart” first-order methods. Conjugate Gradients (CG). The Conjugate Gradients (CG) method is an effective modification to address the lack of history accumulation and the “zigzags” of the SD, which does not require the Hessian [25].

The idea is to optimize the choice of direction p_k . The direction p_k is constructed as a linear combination of the current negative gradient and the previous direction (33):

$$p_{k+1} = -\nabla f(x_{k+1}) + \beta_k p_k \quad (33)$$

The coefficient β_k (e.g., according to the Fletcher–Reeves or Polak–Ribier formula) is chosen such that the directions p_k are Q-conjugate (for a quadratic problem $f(x) = \frac{1}{2}x^T Q x + \dots$), i.e. $p_i^T Q p_j = 0$ for $i \neq j$.

Advantages:

1. *N-step convergence for an $n \times n$ quadratic problem, CG is guaranteed to find the exact minimum in no more than n iterations.*
2. *Efficiency as it avoids the “zigzags” of the SD, since the directions p_k take into account the history of steps and remain conjugate.*
3. *Cost, i.e., the complexity of the iteration ($O(n)$) and memory ($O(n)$), is the same as that of the SD.*

Thus, for large-scale quadratic problems (or nonlinear ones, where it is applied iteratively), CG outperforms the SD in every respect.

Quasi-Newton methods (BFGS / L-BFGS) aim to achieve the fast convergence of Newton’s method without incurring significant computational costs or performing Hessian inversions [26].

The idea is to avoid computing $H_k = \nabla^2 f(x_k)$. Instead, construct an approximation of the inverse Hessian, $B_k \approx [\nabla^2 f(x_k)]^{-1}$, at each step.

- Iteration: $x_{k+1} = x_k - \alpha_k B_k \nabla f_k$.
- Update: B_{k+1} is obtained from B_k by adding a low-rank correction that uses *only* information from the gradients:
 1. $s_k = x_{k+1} - x_k$
 2. $y_k = \nabla f_{k+1} - \nabla f_k$
 3. $B_{k+1} =$
(BFGS update formula, which depends on B_k, s_k, y_k)

The most successful update formula is BFGS (Broyden–Fletcher–Goldfarb–Shanno) [27].

- Advantages (BFGS): Superlinear convergence (almost as fast as Newton’s method).
- Problem (BFGS): It is necessary to store and update the matrix B_k , which requires $O(n^2)$ memory. This is again impossible for $n=50,000$.
- Solution: L-BFGS (Limited-memory BFGS)

This is a highly efficient algorithmic solution. The idea here is not to store the full matrix B_k . Store only the m most

recent pairs of vectors $\{s_i, y_i\}$ (where m is small, e.g., $m=10$ or $m=20$).

Algorithm: The gradient $p_k = -B_k \nabla f_k$ is not computed explicitly. Instead, it is reconstructed dynamically (during computation) using a two-pass recursive algorithm that uses only these m pairs of vectors.

Advantages (L-BFGS) [28]:

- Convergence: Superlinear (though slower than full BFGS, it significantly exceeds the convergence rate of SD).
- Memory: Memory requirements are linear: $O(m \cdot n)$.
- Computational Cost: The cost per iteration is linear: $O(m \cdot n)$.

L-BFGS has become the de facto standard for large-scale unconstrained nonlinear optimization in many fields. Table 2 visualizes the analysis of optimization algorithms. It clearly shows why SD is less efficient: CG offers the same cost ($O(n)$), but much better convergence. L-BFGS offers nearly the same low (linear) cost, but near-Newtonian convergence and low sensitivity to κ .

Table 2. Comparative analysis of method convergence

Method	Convergence rate	Iteration cost (Computational)	Memory requirements	Sensitivity to κ
Steepest Descent (SD)	Linear (slow)	$O(n)$	$O(n)$	Very high
Conjugate Gradients (CG)	Superlinear (n-step for quadratic)	$O(n)$	$O(n)$	Moderate
L-BFGS (m=10)	Superlinear (fast)	$O(m \cdot n)$	$O(m \cdot n)$	Low
Newton's Method	Quadratic (fastest)	$O(n^3)$	$O(n^2)$	None

Thus, the SD offers no practical advantages when applied to deterministic problems.

Theoretical and applied significance of the SD. The analysis conducted in this paper leads to an unambiguous conclusion: as a deterministic tool for solving classical, large-scale, ill-posed optimization problems in power engineering (such as OPF, ELD, SE), the steepest descent method is of somewhat limited applicability to modern high-dimensional problems and is practically unsuitable.

Its slow linear convergence, which deteriorates significantly with increasing condition number κ , makes it uncompetitive, compared to conjugate gradient methods, L-BFGS, and even more so, compared to interior-point methods, which dominate modern industrial OPF solvers.

However, the value of the SD today is pedagogical. It is a basic (archetypal) algorithm for nonlinear optimization. Understanding its shortcomings (critical dependence on κ , zig-zag behavior) is a prerequisite for understanding why L-BFGS, CG, and IPM were invented and why they work so well.

Conceptual transformation of gradient methods: Stochastic Gradient Descent (SGD). With the emergence of a new data-driven optimization paradigm—machine learning (ML)—SGD has entered a new phase of relevance [29].

The modern energy sector is increasingly relying on ML to address new challenges [30]:

- Forecasting renewable energy generation (wind, solar).
- Load forecasting (short- and long-term).
- Training demand response models.
- Condition assessment based on PMUs (Phasor Measurement Units).

Mathematically, most learning tasks (e.g., training a neural network) boil down to the problem of minimizing empirical risk (34) [31]:

$$\min_{w \in \mathbb{R}^n} f(w) = \frac{1}{N} \sum_{i=1}^N L(w, \text{data}_i) \quad (34)$$

where w represents the model parameters (network weights), L is the loss function, and N is the number of data points, which can be enormous (millions or billions).

The problem is that to take a single step in SD, L-BFGS, or Newton, one must compute the full gradient $\nabla f(w)$ (35) [32]:

$$\nabla f(w) = \frac{1}{N} \sum_{i=1}^N \nabla_w L(w, \text{data}_i) \quad (35)$$

This requires N computations, i.e., a “scan” of the entire database. This is too expensive. The solution lies in the

application of Stochastic Gradient Descent (SGD) [33], where at each step k , instead of the full gradient, we take only one data point (or a small “batch”) i and take a step in the direction of its gradient (stochastic gradient) (36):

$$w_{k+1} = w_k - \alpha_k \nabla_w L(w_k, \text{data}_i) \quad (36)$$

The term $\nabla_w L(w_k, \text{data}_i)$ is a “noisy” but unbiased ($E[\nabla_w L] = \nabla f(w)$) estimate of the total gradient. Essentially, SGD is the SD method with a very “noisy” gradient.

To provide a visual and quantitative confirmation of the analytical conclusions regarding the impact of system heterogeneity on optimization efficiency, numerical simulations were performed. The classical problem of minimizing a positive-definite quadratic function of the form (15) was used as a test environment, which allows for isolated control of the condition number κ via the eigenvalues of the Hessian matrix Q , simulating the difference in parameter scales of real power systems.

To calculate the SD step size, an exact linear search was applied according to (8). To ensure the mathematical rigor of the experiment and to reproduce the worst-case convergence scenario (according to the Kantorovich inequality), the initial point x_0 was generated taking into account the directions of the principal axes of the ellipsoid. The comprehensive results of the comparative modelling are presented in Figure.

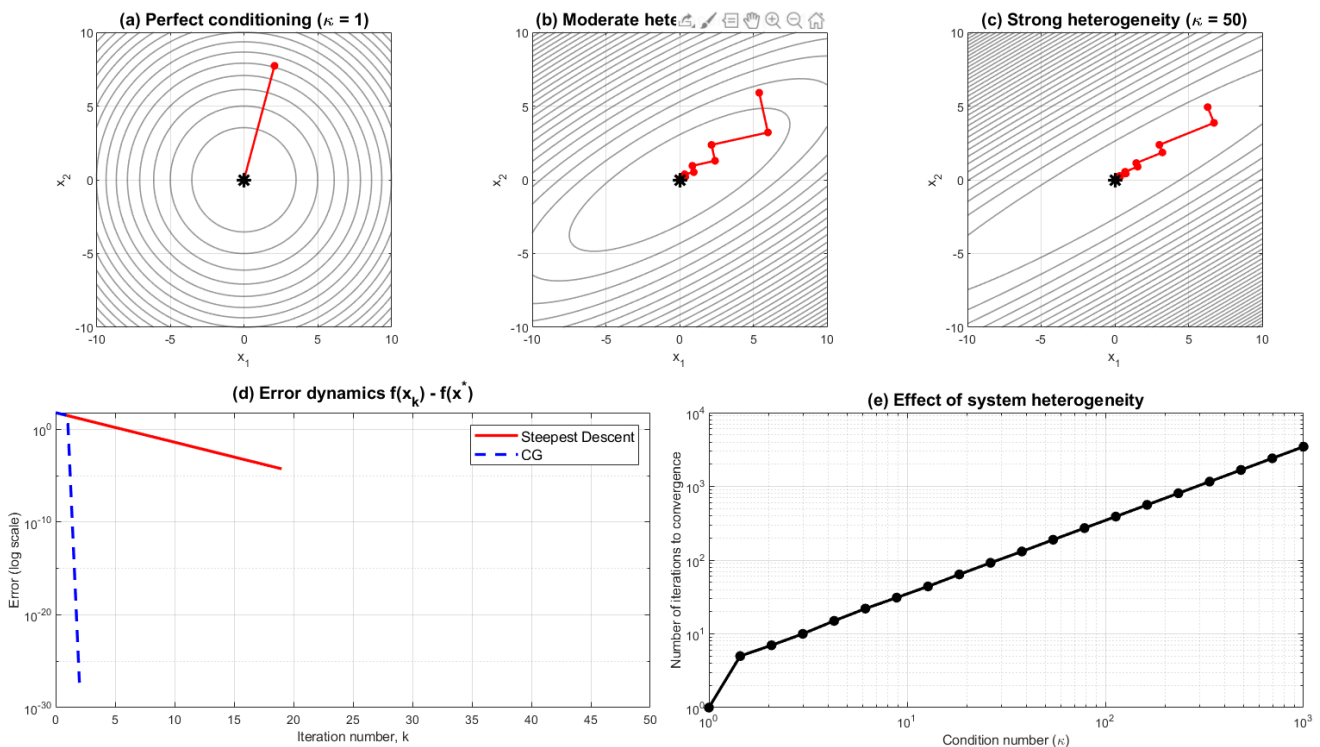


Figure. Numerical study of the effectiveness of the SD on a test quadratic problem

Evolution of the convergence trajectory under the following conditions: (a) ideal regularity, $\kappa=1$; (b) moderate heterogeneity, $\kappa=10$; (c) strong heterogeneity, $\kappa=50$ (demonstrating zigzag behavior). Graph (d) illustrates the slow

linear convergence of the SD method compared to the N -step convergence of the conjugate gradient (CG) method. The dependence (e) confirms the theoretical conclusions: for values of $\kappa > 10^3$, typical for power systems, the required

number of iterations grows exponentially, making the method computationally inefficient.

It is important to note that although SGD and classical gradient descent share a common mathematical foundation (gradient calculation), they represent different optimization paradigms. SGD is an effective algorithm for training neural networks not because of the properties of classical deterministic descent, but rather due to specific effects of stochasticity. The descent direction, which is a “noisy” estimate of the full gradient, acts as implicit regularization, helping the model avoid local minima and saddle points. Thus, the common mathematical foundation has been adapted to the fundamentally new computational needs of modern data-driven solutions in the energy sector.

Conclusions. The steepest descent method is a classic basic algorithm whose role in the optimization of power systems has changed dramatically.

Despite its fundamental theoretical and pedagogical importance, the steepest descent method exhibits slow convergence rates and computational inefficiency when applied to real-world, ill-posed optimization problems in the power industry. Its analysis is critically important not because of its direct modern application, but as a starting point for understanding the necessity and evolution toward more sophisticated methods (such as conjugate gradient, quasi-Newton, and interior-point methods) that dominate the field today.

The steepest descent method, despite its fundamental theoretical and pedagogical importance, exhibits significant limitations when applied to real-world, ill-posed optimization problems in power systems. According to mathematical analysis and numerical simulations, the convergence rate of the method decreases proportionally to κ^2 . For typical AC-OPF problems, where $\kappa \approx 10^3$ due to the difference in scales between line impedances and generation characteristics, this requires 2–3 orders of magnitude more iterations compared to interior-point methods (IPM) or conjugate gradient methods.

The theoretical and practical significance of the SD in power engineering lies not in its current use, but in how its computational limitations on real-world problems have stimulated the development of alternative approaches. Problems that the SD could not solve (e.g., AC-OPF) forced researchers not so much to improve the SD as to simplify the problem, which, for example, led to the widespread use of linearized models (e.g., DC-OPF (Direct Current Optimal Power Flow)). Thus, the theoretical and practical significance of the SD lies in part in the emergence of simplified models that are still in use today, since the full-scale problems were unsolvable for the algorithms of that time.

Acknowledgments. Part of this research was conducted by S.V. Boichenko and V.O. Korovushkin as part of a research project funded by the State Budget of Ukraine (pursuant to Order No. 23 of the Ministry of Education and Science of Ukraine dated January 9, 2026), as well as through independent research topics of the Department of Automation

of Electrical and Mechatronic Systems at Igor Sikorsky Kyiv Polytechnic Institute.

REFERENCES

1. Status of Power System Transformation: Leading Topics of 2024. National Renewable Energy Laboratory, 2024. URL: <https://docs.nrel.gov/docs/fy24osti/91357.pdf> (date of access: 07.01.2026)
2. V. Korovushkin, A. Bosak Improving the Reliability of Electricity Supply from Non-Traditional and Renewable Energy Sources (Ukr.). *Vidnovlyuvana energetika*. 2025. № 2(81). C. 88–96. URL: [https://doi.org/10.36296/1819-8058.2025.2\(81\).88-96](https://doi.org/10.36296/1819-8058.2025.2(81).88-96) (date of access: 12.01.2026).
3. Modern optimization technologies in hybrid renewable energy systems: a systematic review of research gaps and prospects for decisions / V. Korovushkin et al. *Energies*. 2025. Vol. 18, no. 17. P. 4727. URL: <https://doi.org/10.3390/en18174727> (date of access: 12.01.2026).
4. Targeting energy savings? Better on primary than final energy and less on intensity metrics / M. Rodríguez et al. *Energy economics*. 2023. C. 106797. URL: <https://doi.org/10.1016/j.eneco.2023.106797> (date of access: 07.01.2026).
5. Wood A. J., Wollenberg B. F., Sheblé G. B. *Power Generation, Operation, and Control*. 3rd ed. Hoboken, NJ : Wiley, 2013. 656 p.
6. Smart grid, demand response and optimization: a critical review of computational methods / U. Assad et al. *Energies*. 2022. T. 15, № 6. C. 2003. URL: <https://doi.org/10.3390/en15062003> (date of access: 07.01.2026).
7. Castillo A., O'Neill R. P. Survey of Approaches to Solving the ACOPF. Optimal Power Flow Paper 4. Federal Energy Regulatory Commission, 2013. URL: <https://www.ferc.gov/sites/default/files/2020-05/acopf-4-solution-techniques-survey.pdf> (date of access: 07.01.2026).
8. Using gradient descent: cost decreases, then increases. Mathematics Stack Exchange. URL: <https://math.stackexchange.com/questions/975180/using-gradient-descent-cost-decreases-then-increases> (date of access: 07.01.2026).
9. International Energy Agency. *World Energy Outlook 2023*. Paris: IEA, 2023.
10. Wood A. J., Wollenberg B. F., Sheblé G. B. *Power Generation, Operation, and Control*. 3rd ed. New York: Wiley, 2013.
11. Kundur P. *Power System Stability and Control*. New York: McGraw-Hill, 1994.

12. Blaabjerg F., Yang Y., Yang D., Wang X. Distributed Power-Generation Systems and Protection // Proceedings of the IEEE. 2017. Vol. 105, No. 7. P. 1311–1331.
13. Low S. H. Convex Relaxation of Optimal Power Flow // Proceedings of the IEEE. 2014. Vol. 102, No. 1. P. 1–20.
14. Dommel H. W., Tinney W. F. Optimal Power Flow Solutions // IEEE Transactions on Power Apparatus and Systems. 1968. Vol. PAS-87, No. 10. P. 1866–1876.
15. Duffin R. J. Nonlinear Networks: II // Proceedings of the IRE. 1959. Vol. 47, No. 3. P. 388–394.
16. Nocedal J., Wright S. J. Numerical Optimization. 2nd ed. New York: Springer, 2006.
17. Bertsekas D. P. Nonlinear Programming. Belmont, MA: Athena Scientific, 1999.
18. Fletcher R. Practical Methods of Optimization. 2nd ed. New York: Wiley, 1987.
19. Wolfe P. Convergence Conditions for Ascent Methods // SIAM Review. 1969. Vol. 11, No. 2. P. 226–235.
20. Hestenes M. R., Stiefel E. Methods of Conjugate Gradients for Solving Linear Systems // Journal of Research of the National Bureau of Standards. 1952. Vol. 49, No. 6. P. 409–436.
21. Dennis J. E., Schnabel R. B. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. Englewood Cliffs: Prentice-Hall, 1983.
22. Nesterov Y. Introductory Lectures on Convex Optimization. Boston: Springer, 2004.
23. Nesterov Y. A Method for Solving the Convex Programming Problem of Unconditional Minimization // Soviet Mathematics Doklady. 1983. Vol. 27. P. 372–376.
24. Park J. B., Park K. S., Won J. R., Lee K. Y. Economic Load Dispatch Considering Transmission Losses // IEEE Transactions on Power Systems. 2005. Vol. 20, No. 1. P. 343–352.
25. Kron A. Tensor Analysis of Networks. New York: Wiley, 1939.
26. Coelho L. S. Constrained Optimization in Power Systems // Electric Power Systems Research. 2009. Vol. 79. P. 615–623.
27. Bukhsh W. A., Grothey A., McKinnon K. I. M., Trodden P. A. Local Solutions of Optimal Power Flow // IEEE Transactions on Power Systems. 2013. Vol. 28, No. 4. P. 4780–4788.
28. Carpentier J. Optimal Power Flows // International Journal of Electrical Power & Energy Systems. 1989. Vol. 11, No. 1. P. 3–15.
29. Research on photovoltaic power prediction using an LSTM recurrent neural network / S. Boichenko et al. Studies in systems, decision and control. Cham, 2024. P. 421–443. URL: https://doi.org/10.1007/978-3-031-67091-6_19 (date of access: 07.01.2026).
30. Boyd S., Vandenberghe L. Convex Optimization. Cambridge: Cambridge University Press, 2004.
31. Spall J. C. Introduction to Stochastic Search and Optimization. Hoboken: Wiley, 2003.
32. Castillo A. A Review of AC Optimal Power Flow Methods // Electric Power Systems Research. 2013. Vol. 95. P. 223–232.
33. Bottou L. Stochastic Gradient Descent Tricks // In: Neural Networks: Tricks of the Trade. Springer, 2012. P. 421–436.