

Ромасевич Ю.О., Ловейкін В.С.

Монографія

**МОДИФІКАЦІЇ МЕТОДУ
ОПТИМІЗАЦІЇ PSO В ІНЖЕНЕРІЇ**

Київ – 2025

Публікується за рішенням вченої ради Національного університету біоресурсів і природокористування України (протокол № 9 від 27 березня 2025 р.).

Рецензенти:

Капустян Олексій Володимирович, доктор фізико-математичних наук, професор, завідувач кафедри інтегральних і диференціальних рівнянь Київського національного університету імені Тараса Шевченка;

Почка Костянтин Іванович, доктор технічних наук, професор, завідувач кафедри професійної освіти Київського національного університету будівництва і архітектури;

Несвідомін Віктор Миколайович, доктор технічних наук, професор, професор кафедри нарисної геометрії, комп'ютерної графіки та дизайну Національного університету біоресурсів і природокористування України.

Ромасевич Ю.О.

Модифікації методу оптимізації PSO в інженерії. Монографія / Ромасевич Ю.О., Ловейкін В.С. – К.: ЦП „КОМПРІНТ”, 2025. – 377 с.

Монографія написана за результатами виконання українсько-ізраїльського науково-дослідного проекту. В її основі лежать дослідження з модифікацій методу оптимізації PSO та їхнього застосування у різноманітних задачах електричної та механічної інженерії. У роботі наведено опис алгоритмів модифікацій PSO, класу синтетичних цільових функцій та задач оптимального керування, нових інструментів оцінки ефективності роботи алгоритмів. Крім того, проведено аналіз застосування розроблених алгоритмів для розв'язання оптимізаційних задач та вказано найбільш перспективні в них.

Монографія призначена для наукових та інженерних працівників, а також може бути корисна аспірантам, магістрам та студентам технічних спеціальностей закладів вищої освіти.

ВСТУП

Сучасний рівень розвитку науки пов'язаний із необхідністю розв'язування різноманітних задач оптимізації. Можна навести безліч прикладів, коли оптимізація є ядром сучасних технологій [1-3]: машинне навчання, комунікації, дизайн, логістика тощо. Немає жодної сфери людських знань, де оптимізація (у широкому чи вузькому сенсі) не була б використана. Без оптимізаційних підходів неможливо уявити машинобудування та електротехніку, економіку, медицину, соціальні та комп'ютерні науки. Навіть природа «діє» таким чином, що мінімізує витрати деяких ресурсів [4].

Спостерігаючи за історією розвитку людських знань у цій галузі, можна відзначити тенденцію до зростання складності та масштабу задач оптимізації. Ця тенденція збережеться через очевидний факт – найефективніші сучасні методи оптимізації не дадуть змоги розв'язати задачі, які виникнуть у майбутньому. Тут прослідковується схема «задача-метод», коли нова задача викликає необхідність розробки нового методу її розв'язання, а розроблений метод дозволяє розв'язати нові (і більш складніші) задачі оптимізації (рис. 1). Це змушує наукове співтовариство (зокрема у прикладній математиці, інженерії та комп'ютерних науках) модифікувати відомі та розробляти нові методи оптимізації.

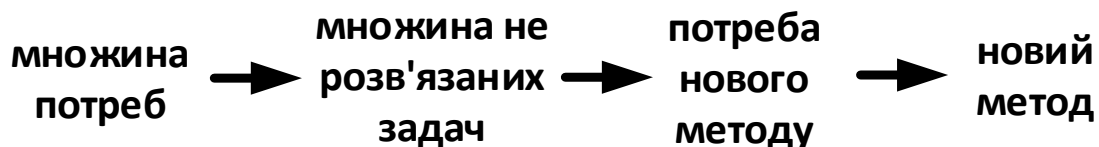


Рисунок 1 – Схема розробки нового методу для розв'язання нових задач

Складність, масштабність, а також специфіка задач оптимізації зумовили широкий спектр підходів до розробки методів їх розв'язування. Один з найперспективніших класів методів (так званий біоінспірований) включає десятки алгоритмів [5-8]. Кожен з них імітує деякий процес у живій або неживій природі. Зокрема, в живій природі можна спостерігати дуже цікаві

механізми (спілкування, полювання, розмноження, адаптація тощо), які є результатом довгої історії еволюції. Остання, як відомо, не терпить неефективності. Ось чому можемо бути до певної мірою впевнені у доцільності використання моделей цих механізмів для потреб розробки нових алгоритмів оптимізації.

Чи справді існує потреба в розробці іншого алгоритму оптимізації або в модифікації існуючого? Відповідь – так! Як і багато інших алгоритмів оптимізації, біоінспіровані мають свої переваги та недоліки. Жоден з дослідників не може заздалегідь вказати, яку ефективність матиме той чи інший метод для розв’язання тієї чи іншої задачі (ця теза відома під назвою „теорема про відсутність безкоштовного обіду”). Разом з тим, маючи широкий спектр методів, вчений може вибрати найбільш доцільний з них. Навіть послідовне застосування популярних методів оптимізації може зорієнтувати вченого на те, який саме алгоритм слід використовувати для конкретної задачі. Крім того, дослідник повинен орієнтуватися на алгоритми, які мають обширну та різноманітну історію застосування або, принаймні, використані для знаходження розв’язків значної кількості задач у різноманітних прикладних сферах. Тому важливо показати ефективність нових або оновлених алгоритмів оптимізації не тільки у рамках застосування до тестових функцій, але й для задач, які виникають у багатьох областях сучасної науки, зокрема, у інженерії.

Один із найвідоміших алгоритмів оптимізації – оптимізація роєм частинок (PSO) – був розроблений у 1995 році Кеннеді та Еберхартом [9]. З того часу було проведено колосальну кількість досліджень, які стосувалися параметрів налаштування PSO, його модифікації, гібридизації з іншими алгоритмами оптимізації, застосування до широкого кола задач оптимізації тощо. Десятиліття досліджень у цій галузі дозволили отримати «золотий стандарт» PSO алгоритмів [10-14]: їхні параметри науково-обґрунтовані, а самі ці алгоритми багаторазово застосовані для багатьох задач оптимізації. Проте, спостерігаючи за кількістю наукових праць, проіндексованих за останні роки у базах даних Scopus та Web of Science, можна констатувати

зростаючий інтерес дослідників до питань, які пов'язані з PSO. Чому цей алгоритм досі привертає таку увагу? Цьому, безумовно, сприяють його переваги (простота, відносно висока ефективність на широкому класі задач оптимізації, точна локалізація мінімуму цільової функції, висока ефективність у сенсі «об'єм обчислень – знайдений розв'язок» тощо). Однак, на наш погляд, найвагоміша причина – значні і поки що не повною мірою реалізовані перспективи подальшого вдосконалення PSO та застосування отриманих результатів у різних прикладних сферах. Навіть несуттєві зміни алгоритму PSO можуть значно підвищити ефективність пошуку розв'язку тієї чи іншої задачі оптимізації. Ця гіпотеза буде доведена в даній науковій роботі.

Таким чином, метою даної роботи є розробка нових модифікацій методу PSO та їхнє застосування у задачах інженерії. Варто вказати і те, що обмеження класом задач інженерії не означає, що новостворені модифікації PSO неможливо буде застосування для інших сфер науки і техніки. Всі розроблені модифікації PSO будуть універсальними інструментами придатними до використання для широкого кола оптимізаційних задач у інших галузях знань.

Монографія написана на основі результатів, які отримані у ході виконання україно-ізраїльського науково-дослідного проекту „Розробка нових модифікацій методу оптимізації PSO та їх застосування у задачах інженерії”.

Вклад авторів у написання розділів рівнозначний. Пункт 9.1 написаний у співавторстві з Олексієм Стехном.

РОЗДІЛ 1

КОРОТКИЙ АНАЛІЗ ДОСЛІДЖЕНЬ ЗА ТЕМОЮ PSO

З моменту першої публікації [9], в якій розроблено базовий (класичний / канонічний) алгоритм рою часточок(PSO), пройшло вже майже три десятка років. Однак, цей алгоритм заслужено вважається одним із найефективніших (принаймні найбільш застосовуваний) для розв'язання багатьох задач оптимізації [15, 16] у галузях робототехніки, енергетики, оптимізації виробничих процесів, логістики, ідентифікації систем, обробки зображень та сигналів. І це далеко не повний перелік проблемних сфер, де успішно застосовується PSO.

Вчені з різних країн світу провели колосальний обсяг робіт для того, щоб вивчити динаміку роботи алгоритму, встановити його сильні та слабкі сторони, дати рекомендації стосовно налаштувань параметрів PSO, розробити підходи щодо застосування PSO у різних галузях науки і техніки. Ці дослідження продовжуються і тепер, хоча їхній акцент змістився в сторону модифікації і гібридизації PSO з іншими алгоритмами оптимізації.

Нижче наведено короткий аналіз наукових публікацій, в яких отримано найбільш значимі результати, що стосуються PSO та вказано сучасні тенденції розвитку цієї галузі знань.

1.1 Кількісний аналіз

Для оцінки рівня уваги до PSO-референтних досліджень використовувалися наукометричні бази даних Scopus та Web of Science. Усі дані в наведених нижче таблицях актуальні станом на 9 червня 2023 року.

Оскільки кількість науковців у світі зростає, то кількість опублікованих праць також має тенденцію до зростання. Тому для оцінки величини інтересу до певної тематики запропонуємо користуватись показником „індекс інтересу” (index interest, I_2), який визначається таким чином:

$$I2 = \frac{N_{Y,T}}{N_Y}, \quad (1.1)$$

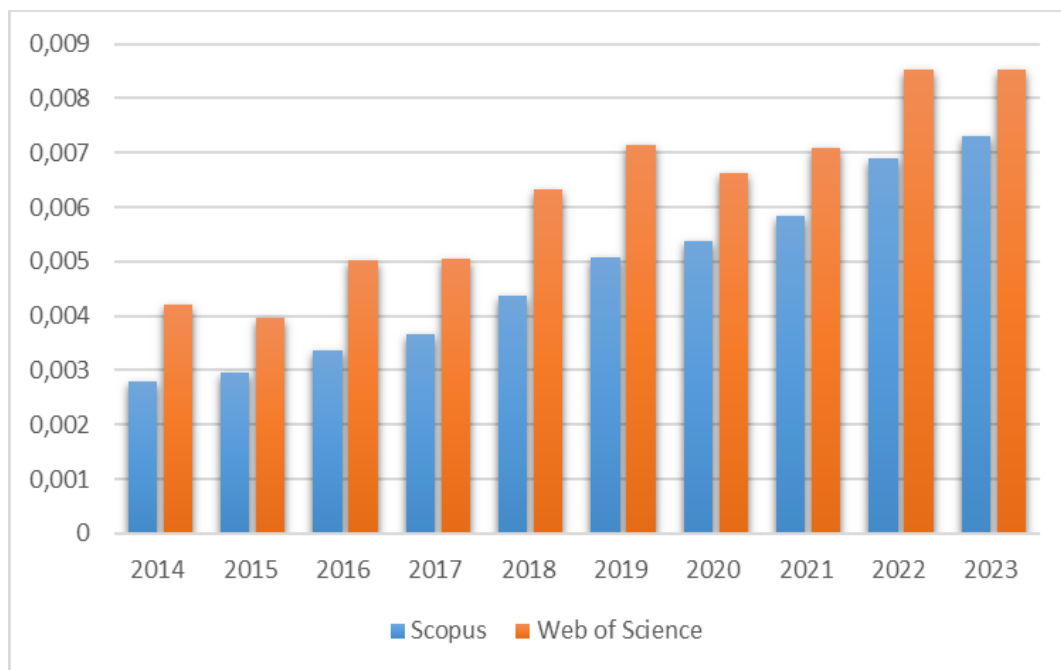
де $N_{Y,T}$ – кількість проіндексованих статей у наукометричній базі за певним запитом за темою „ T ” станом на Y -й рік; N_Y – загальна кількість проіндексованих статей у наукометричній базі станом на Y -й рік.

Індекс $I2$ (1.1) показує частку статей за певною тематикою, яка відображена у запиті, серед усіх статей, що проіндексовані у наукометричній базі даних за певний рік. Звісно індекс (1.1) є відносним показником. Ми вважаємо, що він дозволяє відобразити ступінь зацікавленості наукової спільноти у дослідженнях за певної теми. Динаміка зміни індекса (1.1) за роками дає змогу оцінити зацікавленість певною тематикою зі сторони світової наукової спільноти. Крім того, така оцінка спирається на статистику проіндексованих публікацій у наукометричних базах даних (у нашому випадку будемо використовувати бази Scopus та Web of Science). Такі публікації проходять рецензування і тому є підстави стверджувати, що вони відповідають актуальним і ґрунтовним (у тій чи іншій мірі) дослідженням.

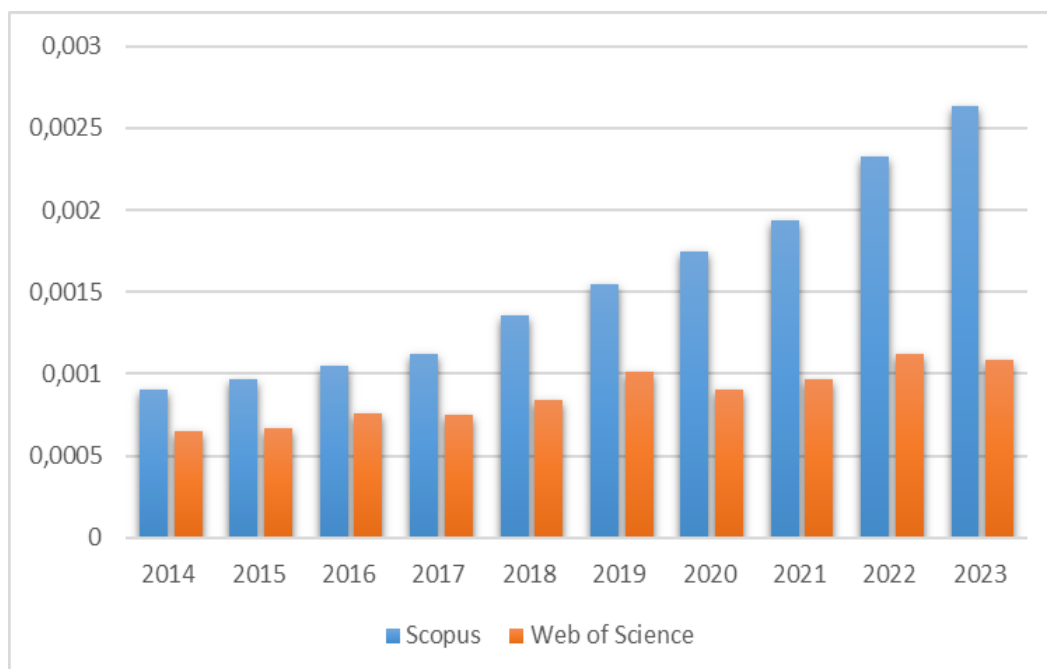
Було зроблено по шість запитів в кожній з баз даних Scopus та Web of Science. Вони чітко демонструють тенденцію збільшення наукових робіт у сфері PSO, як її теоретичних досліджень, так і практичного застосування (Додаток А). Зауважимо, що ми маємо не повні дані для 2023 року. Однак, вони свідчать про те, що тенденція до зростання інтересу $I2$, що відноситься до цього наукового напрямку продовжується. Наприклад, за запитом „PSO” індекс $I2$ за наукометричними базами даних Scopus та Web of Science за десять років (з 2014 по 2023) зріс у 2,6 та 2,0 разів відповідно.

Проведений аналіз дозволяє переконатися, що з моменту першої публікації [9] інтерес наукової спільноти до досліджень та застосувань методу PSO лише зростає. Загальна кількість публікацій (табл. А.1) підтверджує цю тезу. Причому таке зростання відбувається за всіма напрямками, що пов’язані із методом PSO (аналіз самого методу, його модифікації/гібридизації та

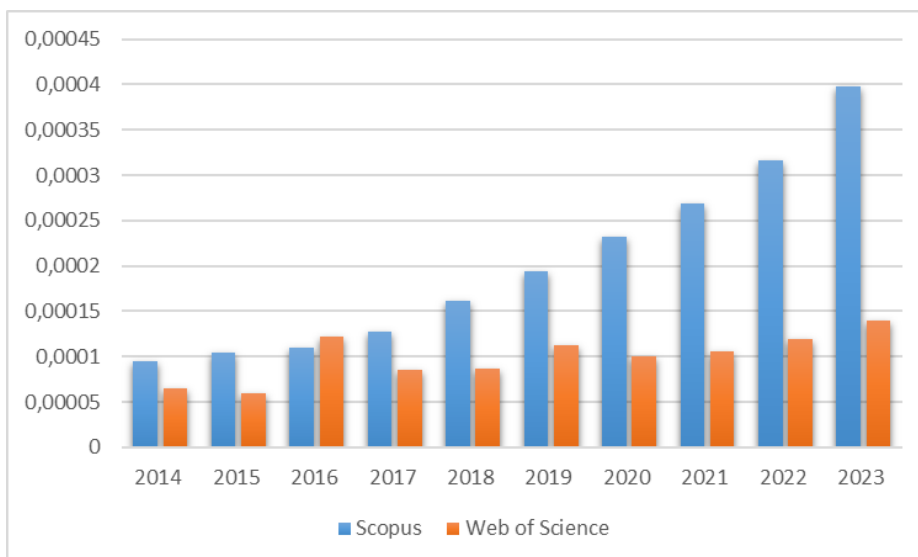
застосування для розв'язання оптимізаційних задач). Варто також відмітити нелінійний характер зростання індексу I_2 для запитів „PSO global optimization” та „PSO modification”, що розрахований за даними наукометричної бази даних Scopus.



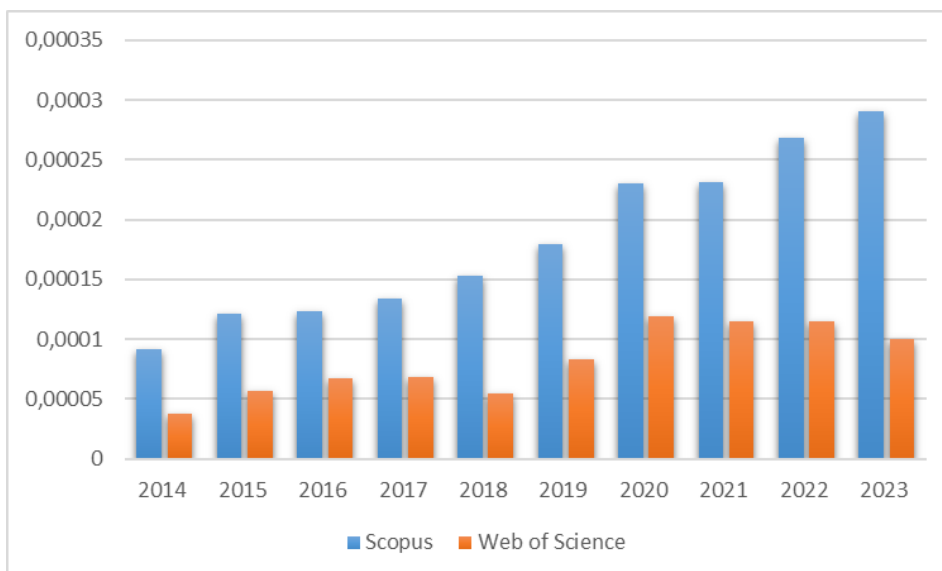
a)



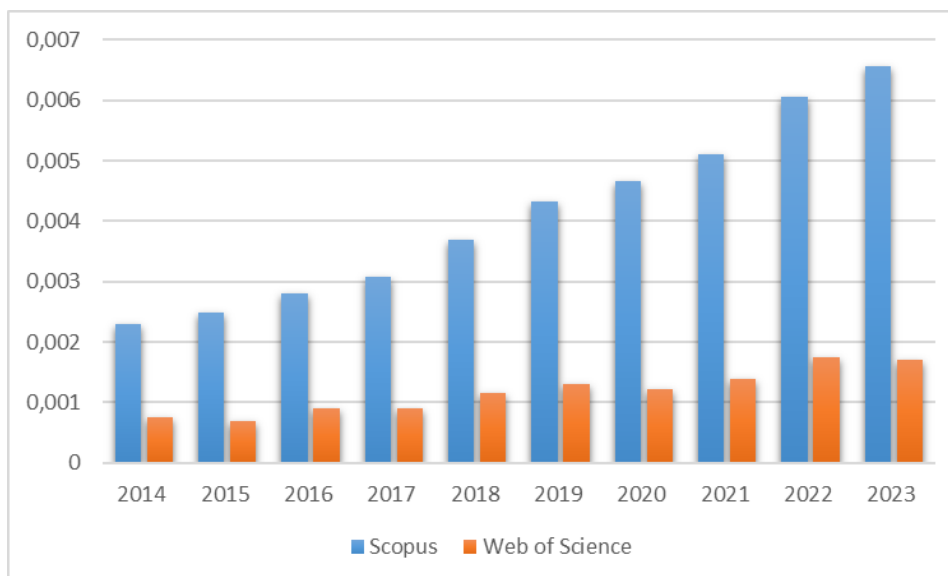
б)



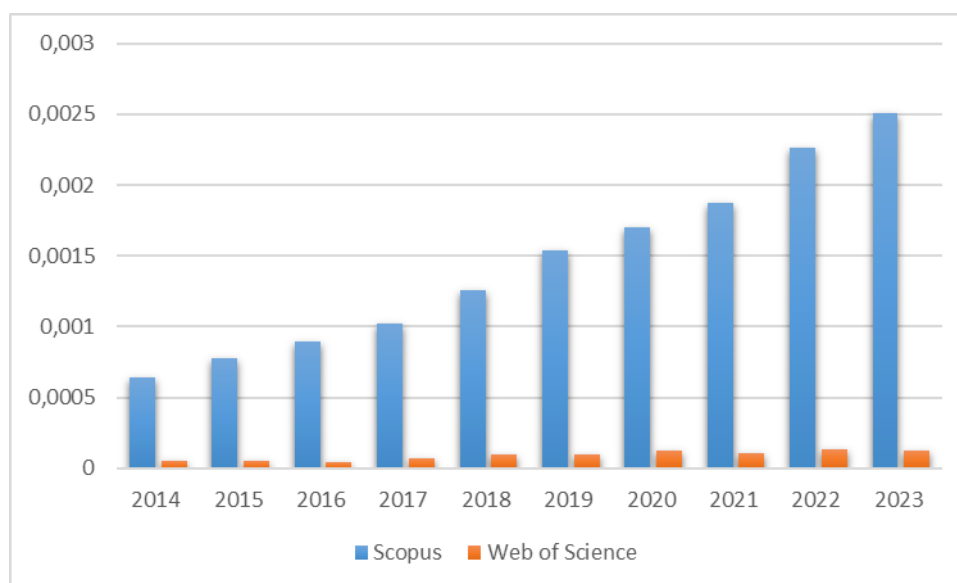
B)



Г)



Д)



е)

Рисунок 1.1 – Гістограми зміни індексу I_2 за запитами: а) „PSO”; б) „PSO global optimization” (PSO та глобальна оптимізація); в) „PSO modification” (PSO модифікація); г) „PSO hybridization” (PSO гібридизація); д) „PSO application” (PSO застосування); е) „PSO survey” (PSO огляд)

Однією з причин інтересу до алгоритму PSO є його сильні пошукові властивості. Вони дають змогу розв’язувати досить складні оптимізаційні задачі. Тут мається на увазі визначення глобального мінімуму цільової функції. Результати даних, які отримані за другим запитом, що наведені у табл. А.1 „PSO global optimization” (глобальна оптимізація за допомогою PSO) підтверджує цю тезу. Однак, наскільки нам відомо, відсутня теоретична основа, яка дозволяє використовувати PSO як оптимізатор, що здатний завжди знаходити глобальний розв’язок задач. Ми припускаємо, що для задач оптимізації з невеликою кількістю аргументів PSO може знайти глобальний мінімум цільової функції. Гадаємо, таке ж твердження справедливе і до задач із великою кількістю аргументів, але з відносно простою топологією цільової функції.

За запитом „PSO global optimization” індекс I_2 за наукометричними базами даних Scopus та Web of Science за десять років зріс у 2,9 та 1,7 разів відповідно.

Канонічний алгоритм PSO має недоліки (відсутність гарантії локалізації глобального мінімуму, передчасна збіжність, відсутність науково-обґрунтованих рекомендацій стосовно кількості ітерацій і розмірності рою) і тому вчені намагаються їх виправити. Одним із можливих шляхів є модифікація алгоритму PSO. Запит „PSO modification” (модифікації PSO) показує значно меншу кількість робіт у наукометричних базах даних. Причина цього може бути в тому, що кожна модифікація пов’язана із розробкою оригінальної ідеї, яка дозволяє подолати один або декілька недоліків PSO. Крім того, лише деякі модифікації призводять до значного покращення пошукових властивостей PSO і автор наукової публікації повинен довести доцільність такої модифікації (порівняти її ефективність із ефективністю відомих модифікацій PSO на досить широкому класі задач оптимізації, або на специфічній і важливій задачі, яка має практичну спрямованість).

За запитом „PSO modification” індекс I_2 за наукометричними базами даних Scopus та Web of Science за десять років зріс у 4,2 та 2,1 разів відповідно.

PSO належить до алгоритмів оптимізації, які нав’язані механізмами живої природи. На даний час розроблено досить значну кількість алгоритмів, в яких реалізовано подібний підхід [5, 17-20]. Кожен такий алгоритм має свої недоліки і переваги. Тому доцільно об’єднувати переваги алгоритмів і уникати їх недоліків, що пов’язано із синтезом гібридних алгоритмів. Дані табл. А.1 чітко свідчать про те, що кількість робіт за тематикою гібридизації PSO можна порівняти з кількістю робіт за попереднім запитом „PSO modification”. Тенденція зміни кількості індексованих статей підштовхує нас до висновку про перспективність цього напрямку дослідження PSO.

За запитом „PSO hybridization” індекс I_2 за наукометричними базами даних Scopus та Web of Science за десять років зріс у 3,2 та 2,7 разів відповідно. Ці дані підтверджують зроблений висновок.

Розробки алгоритмів оптимізації мають за мету ефективно розв’язання реальних задач оптимізації. Тут ми маємо на увазі локалізацію глобального (або доброго локального) мінімуму із залученням доступних (у багатьох

випадках відносно невеликих) обчислювальних ресурсів. Кількість індексованих робіт у базі даних Scopus за запитом „PSO application” (застосування PSO) лише зростає. Це вказує на широку гамму прикладних сфер, де алгоритми PSO застосовуються, а також на ту стійку тенденцію до використання цих алгоритмів у нових проблемних галузях.

За запитом „PSO application” індекс *I2* за наукометричними базами даних Scopus та Web of Science за десять років зріс у 2,9 та 2,2 разів відповідно.

Як видно з табл. А.1 існує велика кількість робіт, що стосуються теоретичного аналізу PSO, його модифікації, гібридизації, застосування тощо. Аналіз цих досліджень можна провести із використанням даних кількості опублікованих праць, що отримані за запитом „PSO survey” (огляд PSO). У цих роботах автори намагалися виділити та класифікувати найбільш суттєві напрямки розвитку PSO. Деякі роботи будуть проаналізовані в наступному підрозділі. Кількість опублікованих робіт викликана загальним інтересом до проблем, що пов’язані з алгоритмом PSO. Слід зазначити, що лише деякі роботи можна віднести до ґрунтовних оглядів. Вони включають десятки або навіть сотні документів, які залучені для аналізу. Решта оглядів дають лише короткий аналіз праць, які пов’язані із алгоритмом PSO.

За запитом „PSO survey” індекс *I2* за наукометричними базами даних Scopus та Web of Science за десять років зріс у 3,9 та 2,3 разів відповідно.

Проводячи аналіз робіт, присвячених PSO, слід поставити запитання: які найважливіші проблемні сфери його застосування? Щоб дати відповідь було зібрано дані з баз Scopus (табл. А.2) і Web of Science (табл. А.3).

Категорії цих баз різні, але можна спостерігати спільну тенденцію: більшість індексованих робіт пов’язані з інженерними та інформаційними науками. База даних Web of Science дозволяє виявити специфічні категорії інженерії (серед них електротехніка та машинобудування, електроніка, автоматизація, телекомунікації тощо) та інформаційних технологій (штучний інтелект, інформаційні системи тощо).

Наступний за значимістю напрям – математика. Слід зазначити, що деякі з робіт, які опубліковані у цій галузі, не можна прямо віднести до тієї чи іншої категорії, оскільки вони отримані в міждисциплінарних дослідженнях (наприклад, інформаційні системи, штучний інтелект тощо, тобто галузі де використовуються методи прикладної математики). Ми вважаємо, що кількість чисто математичних праць, які пов'язані із алгоритмом PSO досить незначна. Тут ми маємо на увазі роботи, в яких закономірності поведінки рою, можливість глобальної мінімальної локалізації (в термінах ймовірностей), збіжність та багато інших питань досліджуються у парадигмі строгого доведення певних математичних теорем.

Задачі, що виникають у енергетиці також є важливим напрямком застосування PSO. Вони переплітаються із зазначеними вище категоріями (наприклад, досить поширена у публікаціях задача оптимізації параметрів вітрогенератора – задача енергетики – відноситься до більш широкого класу інженерних задач).

Існує перелік наук (фізика, астрономія, науки про навколишнє середовище тощо), де алгоритм PSO також активно застосовується (табл. А.2). Іншим значимим напрямком застосування PSO є матеріалознавство. Проведений аналіз показує, що розв'язок цих задач може бути ефективно знайдений методами PSO та модифікаціями цього алгоритму.

Проводячи аналіз робіт, пов'язаних з алгоритмом PSO, ми повинні враховувати географію установ, де вони виконані (за регіонами світу). Для цього також були використані бази даних Scopus і Web of Science. Усі зібрані дані наведено в табл.А.4. У цей аналіз також було включено Україну та Ізраїль, хоча вони не входять у десятку країн-лідерів за кількістю проіндексованих статей.

Необхідно зазначити, що більшість статей є результатом міжнародної співпраці і відносити їх лише до однієї країни не зовсім коректно. Однак, такий аналіз допоможе оцінити вплив авторів з окремих країн на розвиток тематики PSO. Як видно з табл.А.4 трійкою країн-лідерів є Китай, Індія та Іран. Інші

азіатські країни, які потрапили в першу десятку – Тайвань, Малайзія та Саудівська Аравія. На Північноамериканському континенті за кількістю публікацій виділяється США, дослідники з Великобританії та Австралії також активно виконують дослідження у цьому напрямку.

Можна виділити чотири світові центри досліджень PSO: південна Азія (Єгипет, Саудівська Аравія, Іран, Індія, Китай) з акцентом у Китаї, Північна Америка з акцентом у США, Європа з акцентом у Великобританії (Італія, Франція, Німеччина, Іспанія, Польща та інші країни ЄС не входять до першої десятки, але мають багато вчених, які публікують цікаві та впливові праці за тематикою PSO), Тихоокеанський регіон (Малайзія, Тайвань, Австралія).

На жаль, Україна поза цим науковим мейнстрімом. Залишаючи поза увагою причини такої ситуації, необхідно відмітити, що все ж таки в Україні проводяться наукові дослідження де вирішуються питання PSO. Однак, більшість таких робіт далека від вимог бути опублікованими у високоімпактних міжнародних рецензованих журналах. У Ізраїлі за кількістю публікацій у наукометричних базах ситуація досить схожа на описану вище.

Одна з цілей цієї наукової роботи пов'язана з виправленням такої ситуації шляхом залучення зацікавлених представників наукової спільноти України до досліджень, що пов'язані із алгоритмом PSO.

1.2 Контент-аналіз окремих наукових праць

1.2.1 Налаштування базових параметрів PSO

Класичний (канонічний) алгоритм PSO має декілька параметрів, які необхідно налаштувати перед його запуском: інерційний w , соціальний c_1 та когнітивний c_2 коефіцієнти, кількість ітерацій та розмірність рою (кількість часточок в ньому), топологія зав'язків між часточками. Існує значна кількість модифікацій PSO, в яких кількість параметрів можуть змінюватись, про що буде сказано нижче. Однак, перелічені базові параметри присутні майже у

кожній відомій модифікації PSO. Їхній вплив на ефективність роботи алгоритму має важливе значення. Вони дають змогу отримати алгоритм, який буде придатним для розв'язування широкого класу задач оптимізації.

У роботі [20] досліджено вплив розмірності рою на ефективність застосування восьми різних модифікацій PSO. Для них запропоновано використовувати рої із 10-50 часточками. Деякі автори наводять також прив'язку розмірності рою до розмірності задачі (кількість аргументів, які необхідно відшукати, будемо позначати D).

Виходячи з класичної роботи [21] для рою, який включає 20-50 часточок, достатньо для відшукування розв'язку більшості задач. Автори, провівши достатньо значний об'єм розрахунків та проаналізувавши їх, прийшли до висновку, що найкраща ефективність модифікацій досягається тоді, коли рої складаються з 70-500 частинок. Великі рої часто покращують ефективність алгоритмів PSO для складних і практично-орієнтованих задач. Для унімодальних цільових функцій рекомендується трохи зменшувати розмір рою. Однак, і для цих задач модифікації PSO добре працюють із сотнями частинок. Заключною тезою роботи є твердження „в метаевристиці будь-яке налаштування параметра залежить від задачі і його неможливо вважати універсальним”, з яким важко не погодитись.

В праці [22] зазначено: „немає точних правил для вибору розміру рою, але, як правило, коли розмірність задачі зростає, розмір рою також слід збільшити”. Різні автори наводять різні рекомендації стосовно раціональної розмірності рою (табл. 1.1) [23].

Дослідження, предметом яких є вплив розмірності рою на ефективність роботи алгоритму, проведені у роботі [23]. Тут автори рекомендують для застосування великі рої (3000 часточок) при відносно незначній кількості ітерацій (200).

Крім того, встановлено вплив початкової ініціалізації положень частинок на ефективність роботи алгоритму (не тільки для алгоритму PSO, а і інших: ABC – метод штучних бджолиних колоній [29], CS – зозулиний пошук

[30], DE – метод диференціальної еволюції [31], DE-a – метод адаптивної диференціальної еволюції [31], GA – генетичний алгоритм [32]). Широко використовується техніка рівномірного розподілу для ініціалізації рою не обов’язково є найкращим підходом.

Таблиця 1.1 – Деякі рекомендації стосовно розмірності рою

Автори	Наукова праця	Розмірність рою
Weibo W., Chuan L., Yongkang Z.	[24]	від 20 до 50
Carlisle A., Dozier G.	[25]	до 30
Bratton D., Kennedy J.	[10]	від 20 до 100
Clerc M.	[26]	20
Pant, M., Thangaraj, R., Abraham, A.	[27]	від $2D$ до $5D$

Для алгоритму PSO, який, як зазначають автори, чутливий до початкової ініціалізації рою, автори рекомендують використовувати випадкову ініціалізацію рою, застосовувати бета-розподіл і вибірку латинського гіперкуба.

У роботі [33] досліджено вплив параметрів коефіцієнтів рою та величини максимальної швидкості руху часточок на класі трьох відомих задач умовної оптимізації (конструювання швидкісного редуктора, пружини та триланкової ферми). Результати показали, що ефективність PSO є найбільш чутливою до інерційного, соціального та когнітивного коефіцієнтів. Робота [33] дозволила вказати гарні початкові значення параметрів, які можна використати у якості першого наближення для налаштування PSO при розв’язуванні оптимізаційних задач. Тут також наведено аналіз стратегій змін параметрів, які запропоновані іншими дослідниками, для задач оптимізації, що різняться за кількістю аргументів та цільовими функціями. До явних недоліків праці можна віднести обмеженість класу задач, на яких проведено аналіз чутливості ефективності PSO.

У наукових працях [34, 35] отримано рекомендації стосовно конфігурації величин параметрів області у координатах „ c_1+c_2 ” – „ w ”. Для

того, щоб забезпечити гарантовану збіжність алгоритму PSO необхідно виконати наступну нерівність:

$$c_1 + c_2 < \frac{24(1-w^2)}{7-5w}. \quad (1.2)$$

У роботі [36] показано, що при застосуванні умови (1.2) необхідно дещо зменшувати отримані чисельні значення коефіцієнтів c_1 , c_2 та w . Це пов'язано з тим, що при налаштуванні їх значень, які лежать на границях області, що описується виразом (1.2), відбувається дуже повільна збіжність алгоритму PSO. Мірою збіжності виступає середня норма відстані, на яку перемістилась часточка за одну ітерацію.

Деякі рекомендації стосовно задання параметрів c_1 , c_2 та w для забезпечення збіжності рою наведені у табл. 1.2 [36].

У роботі [42] рекомендовано задавати наступні параметри $w=0,7968$, $c_1=c_2=1,4962$. У працях [13, 43] автори прийшли до висновку, що коефіцієнт w необхідно задавати у діапазоні $0,9 \dots 1,2$. Це, як вони зазначають, дає більше шансів знайти глобальний мінімум цільової функції у рамках прийнятної кількості ітерацій.

Крім того, звернемо увагу на роботи, в яких алгоритм PSO використано для розв'язання задач параметричної оптимізації виробничих процесів. В огляді [3] представлено досить значний перелік таких праць. Не заглиблюючись у суть цих робіт, наведемо чисельні значення параметрів PSO, які використані для їх розв'язання:

- 1) розмірність рою від 3 до 250, або від $1D$ до $50D$, залежно від кількості оптимізованих параметрів виробничого процесу;
- 2) інерційний коефіцієнт w становив від 0,1 до 1,2;
- 3) соціальний c_1 і когнітивний c_2 коефіцієнти були у більшості задач рівними і становили від 0,1 до 2,05; 4) кількість ітерацій задана у діапазоні від 50 до 5000.

Таблиця 1.2 – Деякі рекомендації стосовно задання параметрів c_1 , c_2 та w для забезпечення збіжності алгоритму PSO

Автори	Праця	Умови збіжності
F. VandenBergh, A. Engelbrecht	[37, 38]	$c_1 + c_2 < 2(1+w)$, $c_1 > 0, c_2 > 0, 0 < w < 1$
I. Trelea	[39]	$c_1 + c_2 < 4(1+w)$, $c_1 > 0, c_2 > 0, 0 < w < 1$
V. Kadiramanathan, K. Selvarajah, P. Fleming	[40]	$\begin{cases} c_1 + c_2 < 2(1+w), & w \in (-1, 0]; \\ c_1 + c_2 < \frac{2(1-w)^2}{1+w}, & w \in (0, 1) \end{cases}$
V. Gazi	[41]	$\begin{cases} c_1 + c_2 < \frac{24(1+w)}{7}, & w \in (-1, 0]; \\ c_1 + c_2 < \frac{24(1-w)^2}{7(1+w)}, & w \in (0, 1) \end{cases}$

Одним із найбільш впливових параметрів алгоритму PSO є зв'язність частинок, тобто можливість отримувати і передавати інформацію про розв'язки задачі. Існує два основних класи топології такої зв'язності: *gbest* та *lbest*. У першому випадку визначення кращого розв'язку (*gbest*) проводиться у всьому рої, а у другому такий розв'язок (*lbest*) є окремим для окремого сусідства частинок (він може включати, наприклад, лише три частинки).

Менша зв'язність рою призводить до повільного поширення інформації серед частинок і, відповідно, повільної збіжності алгоритму. Чим більша зв'язність рою, тим швидша збіжність. Велика зв'язність рою може викликати ризики передчасної збіжності, що в результаті призведе до поганої розвідки цільової функції. При цьому рій збігається у локальний мінімум, а положення глобального мінімуму залишається невизначеним [44].

Автори праці [45] зазначають, що з у різних евристичних алгоритмів зв'язність агентів становить у більшості випадків 15% від розмірності рою (кожна частинка інформує та отримує інформацію лише від 15% частинок рою).

У роботах [14, 4] проаналізовано ефективність роботи PSO з різними топологіями зв'язків між частинками і встановлено, що топології зірки, кільця та фон Неймана є найкращими.

У огляді [47] відзначено, що за останні роки було отримано нові перспективні топології PSO [14, 48, 49]. Автори рекомендують звернутись до праці [50] для того, щоб краще зрозуміти проблеми вибору раціональної топології зв'язків. У дослідженні [50] автори прийшли до висновку, що оптимальна топологія залежить не тільки від конкретної задачі, але й від обчислювальних ресурсів.

У роботі [51] автори, провівши дослідження на 84 тестових функціях (при цьому обчислювальні ресурси були однакові – кожен алгоритм отримав 150000 доступів/оцінок цільової функції) прийшли до висновку, що для складних задач оптимізації топологія lbest працює краще, ніж gbest. Топологія gbest може показувати гарні результати для деяких цільових функцій (лише у 14% випадків gbest працює краще, ніж lbest). Однак, в умовах мультимодальності (наявності декількох мінімумів) та інших факторів складності цільової функції, застосування lbest є бажаним.

Опис топологій зв'язків між частинками зручно виконувати за допомогою так званої матриці інцидентності:

$$L_{SP \times SP} = \begin{pmatrix} L_{11} & L_{12} & \dots & L_{1j} & \dots & L_{1SP} \\ L_{21} & L_{22} & \dots & L_{2j} & \dots & L_{2SP} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ L_{i1} & L_{i2} & \dots & L_{ij} & \dots & L_{iSP} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ L_{SP1} & L_{SP2} & \dots & L_{SPj} & \dots & L_{SPSP} \end{pmatrix}, L_{ij} = \{0, 1\}, \quad (1.3)$$

де SP – розмірність рою (кількість частинок в ньому).

Якщо $L_{ij}=1$, то j -та частинка інформує i -ту частинку, $L_{ij}=0$ у протилежному випадку. Матриця (1.3) у загальному вигляді не є симетричною, тобто з того, що i -а частинка інформує j -у не впливає наявність зворотного

обміну інформацією [52]. Заповнення матриці (1.3) одиницями показує ступінь зв'язності рою.

Варто також відмітити випадкову організацію зав'язків між частинками [53]. У цьому випадку заповнення матриці інцидентності (1.3) відбувається від ітерації до ітерації випадковим чином. В процесі виконання алгоритму матриця заповнюється одиницями і таке заповнення виконується згідно певного закону. Частинним випадком такої топології є статична випадкова конфігурація зав'язків між частинками. Єдиним додатковим параметром, який при цьому необхідно налаштувати, є ступінь зв'язності, який, як зазначалось раніше [45], повинен складати приблизно 15% від розмірності рою.

З аналізу праць стає зрозумілим, що сама природа впливу топології PSO на ефективність алгоритму погано піддається детальному, повному і надійному дослідженню. Більший вплив, радше, відіграє ступінь зв'язності частинок (необхідно відокремлювати терміни „ступінь зв'язності” та „топология зв'язків”. Перший означає кількість зв'язків між частинками у рої, а другий – їхню конфігурацію).

1.2.2 Застосування PSO у задачах керування та ідентифікації динамічних систем

Значна кількість задач, які виникають на даному етапі розвитку науки і техніки, пов'язана із задачами ідентифікації динамічних систем та керування ними. Зокрема, керування має задовольняти мінімум певного критерію оптимізації (оптимальне керування). Тому у даному пункті зосередимо увагу саме на цих двох класах задач та оцінимо ефективність застосування PSO алгоритмів для знаходження їхніх розв'язків.

У наведеному нижче аналізі застосовано високоцитовані наукові публікації, які проіндексовані у наукометричній базі даних Scopus. Вибір саме таких статей зумовлений тим, що вони відображають загальні тенденції застосування PSO у задачах керування та ідентифікації динамічних систем.

У статті [54] метод LDW-PSO використано для налаштування дрібного ПД-регулятора. Тут проведено постановку задачі, яка включає математичну модель $PI^{\beta(t)}D^{\alpha(t)}$ -регулятора (відзначено, що стандартний ПД-регулятор є його частинним випадком) та комплексний безрозмірний критерій оптимізації. Оскільки відсутні математичні основи аналітичного синтезу вказаних регуляторів, автори застосували алгоритм LDW-PSO для знаходження коефіцієнтів $PI^{\beta(t)}D^{\alpha(t)}$ -регулятора. Останній було застосовано для регулювання трьох осциляторних систем. Результати порівняння якісних показників регулювання $PI^{\beta(t)}D^{\alpha(t)}$ - та ПД-регулятора показали перспективність першого.

У роботі [55] розв'язано аналогічну задачу. Тут для налаштування дрібного $PI^{\lambda}D^{\mu}A$ -регулятора застосовано PSO алгоритм (символ „A” у позначенні регулятора означає другу похідну похибки за часом). Для мінімізації критерію середнього зваженого за часом модуля похибки (ІТАЕ) необхідно було визначити аргументи $PI^{\lambda}D^{\mu}A$ -регулятора: коефіцієнти підсилення кожної зі складових регулятора, а також дрібні порядки його інтегральної λ та диференціальної μ частин. Результати проведеного моделювання динаміки регулювання двох електричних двигунів (система третього порядку), а також їх порівняння за показниками якості регулювання із відомими результатами, показали ефективність такого підходу. Очевидно, що залежність величини ІТАЕ від параметрів налаштування регулятора є суттєво нелінійною. Разом з тим, застосування канонічного алгоритму PSO дозволило авторам досить ефективно розв'язати поставлену задачу.

У роботі [56] метод PSO застосовано для налаштування ПІ-ПД-регулятора, який виконує регулювання чотирьох об'єктів першого та другого порядків. Деякі з них мають досить значні часові затримки. Автори запропонували модифікований інтегральний критерій, який включає складові квадратичної похибки і добутку квадратичної величини швидкості сигналу регулювання та третьої степені часу. Такий критерій, як зазначають автори, є більш робастним для задачі налаштування регулятора.

У роботі [57] розроблено та застосовано нову модифікацію PSO для задачі налаштування ПД-регулятора руху електромеханічного екзоскелету колінного суглоба людини. Швидка збіжність класичного PSO викликала необхідність у розробці модифікації алгоритму, яка полягала у застосуванні трьох соціальних складових руху частинки, що направлені на положення трьох найкращих частинок. При цьому когнітивна та інерційна складові руху відсутні. Крім того, як тільки рій збігається (виконується умова зменшення стандартного відхилення положень частинок нижче порогового значення) виконується локальна розвідка околу положення засобами випадково обраних частинок (десята частина розмірності рою). Порівняльний аналіз показав перевагу застосування розробленої модифікації PSO для таких задач.

У роботі [58] за допомогою модифікації алгоритму PSO розроблено підхід для оптимального керування роботом-маніпулятором. Ідея застосування алгоритму полягає у виконанні розрахунків із використанням моделі робота та знаходженням мінімуму цільової функції. Для апробації такого підходу було виконано відповідні розрахунки оптимальних керувань руху робота-маніпулятора PUMA 560, захват якого переміщувався по різних траєкторіях. Результати моделювання показали високу якість керування.

Метод PSO та його модифікації застосовуються також і для тренування штучних нейронних мереж (ШНМ) [59]. У статті [18] для розв'язування нелінійного диференціального рівняння, відомого як задача Троша, застосовано технологію ШНМ. Традиційні оптимізаційні методи тренування ШНМ не прийнятні для великих власних значень. Запропонований у роботі підхід на основі PSO усуває вказаний недолік та дозволяє отримати кращі (у сенсі точності) результати.

У монографії [60] ШНМ були натреновані за парадигмою „з підкріпленням” для задач синтезу оптимальних регуляторів різноманітних об'єктів регулювання (одно- та багатомірні, нелінійні) із різноманітними критеріями оптимізації. Сукупність отриманих результатів переконливо довела високу ефективність PSO алгоритмів для задач тренування ШНМ.

У статті [61] застосовано модифікацію SEPSO (соціально-емоційна модель алгоритму) для визначення невідомих параметрів маніпулятора робота (задача ідентифікації). Результати показали застосовність та ефективність роботи алгоритму SEPSO за показниками точності знайденого розв'язку та швидкості збіжності алгоритму у порівнянні із стандартними методами PSO та методом генетичного алгоритму (GA).

У статті [62] представлено новий метод ідентифікації нелінійних динамічних систем за допомогою вейвлет-ШНМ. Ця ШНМ була навчена за допомогою методу mPSOWIIR, в якому застосовано підхід елітарного відбору частинок. Крім того, інерційний коефіцієнт w протягом виконання алгоритму mPSOWIIR лінійно спадає. У якості критерію, який відображав розбіжність функції виходу динамічної системи і її ідентифікованої моделі і який необхідно мінімізувати, автори використали середньоквадратичне значення похибки прогнозу. Чотири чисельні експерименти, які були використані для оцінки ефективності розробленого підходу, показали достатньо високу якість апроксимації відгуку ідентифікованих систем.

У роботі [63] розв'язано задачу синтезу робастного керування рухом автономного підводного апарату на основі ідентифікації його моделі. Невідомі параметри цієї динамічної системи оцінені за допомогою адаптивного ідентифікатора, чиї коефіцієнти підсилення оптимізовано за допомогою алгоритму PSO. Інерційний коефіцієнт алгоритму w при виконанні ітерацій експоненціально зменшувався. Критерієм мінімізації було обрано інтегральну оцінку похибок відгуку моделі за трьома компонентами (автори виконали декомпозицію моделі, розділивши її на три частини, які не взаємодіють одна з одною: керування кутом рискання, глибиною занурення у воду та курсовою швидкістю апарату). У цій роботі автори обґрунтували вибір алгоритму PSO (наявність імпульсу в русі частинок, поширення інформації через весь рій, відносну автономність руху частинок, наявність у частинок „пам'яті”, що дозволяє досить швидко знаходити розв'язок). Наведені авторами графіки збіжності ідентифікованих параметрів до дійсних значень динамічної системи

показали ефективність алгоритму PSO, яка дозволила досягнути високої точності ідентифікації параметрів моделі.

У роботі [64] наведено досить детальний огляд застосування методів оптимізації для ідентифікації нелінійних динамічних систем. Серед них GA, PSO, диференціальна еволюція (DE). Автори вказали на тенденцію щодо того, що PSO і DE стають все більш популярними для задач параметричної ідентифікації нелінійних динамічних систем. Наприклад, автори проаналізували п'ятдесят наукових робіт, що були опубліковані станом на 2019 рік, де алгоритми PSO були застосовані для ідентифікації нелінійних моделей Бука-Вена, хаотичних систем, Джилса-Атертона, Лю Грея, Прандтля-Ішлінського, Прейзака, Вінера-Гаммерштайна. Такий широкий спектр моделей свідчить про придатність цих алгоритмів для ефективного розв'язання задач параметричної ідентифікації нелінійних систем.

Проведений аналіз показує два головні напрямки використання методу PSO у задачах, які пов'язані із динамічними системами: керування та ідентифікація. Обидва ці напрямки тут вжиті у широкому сенсі. Наприклад, задачі керування включають синтез різноманітних операторів керування: ПД- та інших лінійних регуляторів, нейро- та fuzzy-регуляторів тощо. Задачі ідентифікації також розв'язуються у різних постановках, наприклад, визначаються параметри динамічної системи (інерційні, пружні та дисипативні коефіцієнти тощо), або проводиться синтез математичної моделі системи у вигляді ШНМ. У роботах наведено множину критеріїв оптимізації, які характеризуються різноманітними топологічними особливостями. Це, однак, не завадило використанню PSO алгоритмів для задач знаходження мінімумів цільових функцій. Описані задачі досить складні для розв'язування, що, безперечно, пояснює схильність авторів до використання ефективного інструменту для їх розв'язання, яким виступає PSO.

Висновки до першого розділу:

1. Аналіз тенденцій зміни кількості наукових публікацій за різноманітними запитами, які пов'язані з PSO тематикою, показує зростання інтересу наукової спільноти до даної проблематики. Причому така тенденція є незалежною від проблемної сфери застосування PSO. Цей факт можна пояснити тим, що, на наш погляд, потенціал алгоритмів PSO все ще повною мірою не розкрито, відбувається активний процес виявлення резервів підвищення пошукових та інших характеристик алгоритмів та їх застосування у надзвичайно широкій гаммі прикладних задач.
2. Аналіз наукових праць, які в основному були виконані на ранніх етапах розвитку PSO, показує, що існують досить прості рекомендації стосовно налаштувань його базових параметрів: інерційного, соціального, когнітивного коефіцієнтів, розмірності рою та топології зв'язків між частинками. Для відносно простих задач оптимізації використання PSO із врахуванням цих рекомендацій ($w=0,7968$, $c_1=c_2=1,4962$, розмірність рою 20-50 частинок) дозволяє знаходити шуканий розв'язок. Для більш складних задач виникає потреба у застосуванні модифікацій і гібридизацій PSO. Для модифікованих/гібридизованих алгоритмів базові значення налаштувань PSO є гарною початковою ітерацією для синтезу раціональних стратегій їх змін. Крім того, для складних задач оптимізації варто розглянути застосування роїв із топологіями lbest, хоча це призведе до більш тривалої збіжності алгоритму.
3. Існує значна кількість досліджень, які спрямовані на вдосконалення оригінального алгоритму PSO. В них описано відносно ефективні модифікації PSO. Вони були застосовані до широкої області задач оптимізації в багатьох теоретичних і практично орієнтованих областях. Наразі у науковій спільноті існує консенсус щодо їхньої високої ефективності. Однак, не існує єдиного алгоритму, який би однаково

ефективно можна було б застосувати до різних за характером (умовами задачі, масштабністю, топологічними характеристиками цільової функції) задач оптимізації. Останню тезу можна до певної міри ослабити шляхом гібридизації алгоритмів.

4. Серед алгоритмів, які застосовуються для розв'язання задач ідентифікації моделей динамічних систем та синтезу систем їхнього оптимального керування, алгоритми PSO займають перше місце, що пояснюється їхніми сильними пошуковими властивостями. Навіть у випадках задання у топологічному сенсі явно складних цільових функцій (наприклад, при навчанні ШНМ із десятками невідомих параметрів) PSO алгоритми дають змогу отримати бажані результати. При цьому, звісно, бажано користуватись модифікаціями PSO.
5. Узагальнений аналіз результатів огляду робіт, які представлені у розділі, показує, що найбільш перспективними напрямками розвитку класу PSO алгоритмів є наступні: 1) механізми збільшення різноманітності рухів часточок; 2) застосування динамічних топологій зав'язків між частинками рою; 3) науково-обґрунтовані рекомендації щодо раціональних параметрів модифікацій PSO; 4) нові механізми та умови переініціалізації рою. Крім того, важливою задачею, яка сприятиме подальшому прогресу у даній галузі, є розробка інструментів дослідження динаміки руху рою та порівняльних оцінок ефективності роботи оптимізаційних алгоритмів.

РОЗДІЛ 2

РОЗРОБКА ТА МАТЕМАТИЧНИЙ ОПИС МОДИФІКАЦІЙ PSO

2.1 Постановка задачі оптимізації

У цьому пункті ми дамо коротку постановку задачам оптимізації. Їхня суть полягає у тому, що існує цільова функція f аргументів $x^1, \dots, x^d, \dots, x^D$, яка визначена в деякій D -вимірній області. Ця множина є обмеженою:

$$x^d \in [x_{\min}^d, x_{\max}^d], \quad d \in (\overline{1, D}), \quad (2.1)$$

де x_{\min}^d і x_{\max}^d – нижня і верхня межі d -того аргументу відповідно; D – розмірність задачі (кількість невідомих аргументів).

Цільова функція f має принаймні один мінімум, який необхідно визначити. Таким чином, необхідно розв'язати наступну задачу:

$$f(x) \rightarrow \min, \quad x = [x^1, \dots, x^d, \dots, x^D]^T. \quad (2.2)$$

У наведеній постановці ми не будемо брати до уваги обмеження, що накладаються, оскільки існує багато підходів, які дозволяють звести задачу із обмеженнями до задачі безумовної оптимізації.

2.2 Відомі оптимізаційні алгоритми, на основі яких виконується розробка нових модифікацій PSO

2.2.1 Алгоритм PSO

Класичний алгоритм PSO оперує терміном „рій” [9], який означає набір можливих розв'язувань задачі (кілька точок на поверхні цільової функції).

Загальна кількість частинок у рої називається популяцією рою. Його позначають як N , номер частинки позначають як n , таким чином $n \in (\overline{1, N})$.

Кожна частинка характеризується двома векторами: вектором-положення частинки $x=[x^1, \dots, x^d, \dots, x^D]^T$ у D -мірній області, що включає всі координати частинок, і вектором-швидкості $v=[v^1, \dots, v^d, \dots, v^D]^T$, що характеризує напрям і швидкість руху частинки.

Запуск алгоритму PSO починається з ініціалізації рою. Стандартний спосіб ініціалізації рою – випадковим чином помістити кожну частинку на поверхню цільової функції, тобто задати компоненти вектору x випадковим чином у межах (2.1). Вектор швидкості v на початку роботи алгоритму може задаватись нульовим.

На наступних ітераціях частинки починають рухатися, тобто вектори частинок x і v змінюють свої компоненти. Рух кожної частинки описується такими формулами:

$$\begin{cases} v_j = w \circ v_{j-1} + c_1 r_1 \circ (p_{j-1} - x_{j-1}) + c_2 r_2 \circ (g_{j-1} - x_{j-1}); \\ x_j = x_{j-1} + v_j, \end{cases} \quad (2.3)$$

де j – номер поточної ітерації ($j \in (\overline{1, J})$); J – загальна кількість ітерацій; p – найкращий вектор-положення частинки, який вона знайшла (він називається особистим найкращим – personal best); g – найкращий вектор-положення, який знайшов весь рій (він називається глобальним найкращим – global best); w – вектор інерційних коефіцієнтів; c_1 і c_2 – когнітивний і соціальний коефіцієнти відповідно; r_1, r_2 – вектори випадкових чисел, які на кожній ітерації рівномірно генеруються на інтервалі $[0, 1]$.

Якщо при застосуванні формул (2.3) якась із частинок залишила область (2.1), то вона примусово повертається на межу цієї області.

Щоб оновити вектори p і g , слід застосувати формули:

$$\begin{cases} p_{j,n} = x_{j,n}, & \text{if } f(x_{j,n}) < f(p_{j,n}); \\ g_{j,n} = p_{j,n}, & \text{if } f(p_{j,n}) < f(g_{j,n}). \end{cases} \quad (2.4)$$

Критерієм зупинки алгоритму PSO є, як правило, виконання всіх ітерацій. Інколи критерієм зупинки є досягнення якогось порогового значення цільової функції.

2.2.2 Алгоритм IA-PSO

Як було зазначено в багатьох наукових роботах, одним із недоліків класичного алгоритму PSO є його передчасна збіжність, яка призводить до неефективного використання обчислювальних ресурсів. Щоб уникнути цього явища були проведені ґрунтовні дослідження. Серед них виділимо стратегію адаптивної зміни інерційного коефіцієнта [65]. Вона передбачає дві модифікації PSO. Перша з них пов'язана зі зміною вектору інерційних коефіцієнтів w (окремо для кожної із частинок). Кожен компонент вектору w на кожній ітерації змінюється за наступною формулою:

$$w_n = rand(0.5, 1) \cdot \left(1 - \frac{\|g_j - x_{j,n}\|}{\max_n (\|g_j - x_{j,n}\|)} \right), \quad (2.5)$$

де w_n – ваговий коефіцієнт n -ї частинки. Як зазначено в [65], ця модифікація сприяє тому, що частинка „притягується” до глобального найкращого. Це пов'язано з тим, що частинка, яка знаходиться далеко від глобального найкращого, рухається повільно і не може далеко відійти від атрактора глобального найкращого. Навпаки, частинка, яка знаходиться поблизу до глобального найкращого, рухається швидше та краще досліджує атрактор (окіл) глобального найкращого. Однак цей ефект може створювати для деяких цільових функцій ризик передчасної збіжності алгоритму. Щоб запобігти цьому, було запропоновано механізм, який полягає у тому, що нове положення частинки оновлюється із певним можливим відхиленням від того місця, куди вказує вектор-швидкість частинки:

$$x_j = (1 - \rho) \circ x_{j-1} + v_j, \quad (2.6)$$

де ρ – вектор рівномірно розподілених випадкових чисел у діапазоні $[-0,25, 0,25]$. Розмірність вектору ρ дорівнює D . Ефект, який описано вище, у вимірювальній техніці називається джиттером [66].

Комбінація описаних алгоритмічних модифікацій PSO надає новому алгоритму IA-PSO досить добрі пошукові властивості. Вони будуть проілюстровані в наступних розділах роботи.

Однак, ми вважаємо, що рівняння (2.6) не відповідає схемі пошуку розв'язку задачі „розвідка-дослідження” (у англійській літературі відома як „exploration-exploitation”). Така схема передбачає огляд всієї області пошуку та виявлення перспективних локальних субобластей цільової функції. Після цього починається їхнє детальне дослідження для локалізації мінімуму. Необхідно відмітити, що межа переходу від розвідки до дослідження цільової функції є різною для різних алгоритмів. Вважається, що це питання до цих пір не вирішене.

Оскільки елементи вектору ρ змінюються у діапазоні, границі якого є константами протягом всього виконання алгоритму, то це може зашкодити пошуковій діяльності IA-PSO як на етапі розвідки, так і на етапі дослідження.

2.2.3 Алгоритм VCT-PSO

Алгоритм VCT-PSO представляє ще одну досить просту модифікацію PSO [67]. Основна ідея цього алгоритму пов'язана зі зміною особистого кращого частинки на певні ітерації. Таким чином, у алгоритмі VCT-PSO зміні підлягає лише когнітивна складова швидкості руху частинки. Загальне правило, яке регулює зміну особистого найкращого p^{λ} λ -ої частинки (верхній індекс λ вказує на те, що вектор p^{λ} – є особистий найкращим λ -ої частинки) на j -й ітерації описується таким виразом:

$$\lambda = \begin{cases} \text{Random}(1, \dots, n, \dots, N) \wedge \bar{\lambda}, & \text{якщо } \frac{j}{RC} \in \text{int}; \\ \lambda, & \text{інакше.} \end{cases} \quad (2.7)$$

де RC – коефіцієнт, який визначає частоту зміни особистого кращого p^{λ} . Як показано в роботі [67] алгоритм VCT-PSO може бути ефективним для багатьох складних задач оптимізації. При цьому у цій же роботі дані рекомендації стосовно раціональних значень RC , які знаходяться у діапазоні від 4 до 6 включно.

Не дивлячись на досить непогані пошукові властивості, які алгоритм проявив і які були відображені у роботі [67], подальші дослідження, що проведені на більш широкому класі тестових функцій, показав деякі слабкі сторони VCT-PSO. Для їхнього усунення у подальшому цей алгоритм буде модифіковано.

2.2.4 Алгоритм Ring-PSO

У попередніх модифікаціях PSO використано топологію повного з'єднання між частинками, коли всі частинки обмінюються інформацією про те, яке значення цільової функції вони знайшли (2.4). У деяких випадках це може зашкодити пошуку через передчасну збіжність алгоритму до локального мінімуму. Щоб уникнути передчасної збіжності була розроблена кільцева топологія зв'язків [14], яка передбачає обмін між частинками інформацією лише із своїми найближчими сусідами (мова йде про сусідство в сенсі номерів частинок, а не положень на поверхні цільової функції). Для його реалізації перший рядок системи (2.3) необхідно замінити таким виразом:

$$v_j = w \circ v_{j-1} + c_1 r_1 \circ (p_{j-1} - x_{j-1}) + c_2 r_2 \circ (gl_{j-1} - x_{j-1}), \quad (2.8)$$

де gl – глобальне найкраще локального рою, тобто частинки та двох її сусідів. Крім того, оскільки частинка обмінюється інформацією не з усім роєм, а лише з двома сусідами, то другий рядок системи (2.4) слід замінити такою формулою:

$$gl_{j,n} = \min_{\text{arg}} (f(p_{j,n-1}), f(p_{j,n}), f(p_{j,n+1})) \quad (2.9)$$

Для верхнього та нижнього індексів частинки (для першої та останньої) необхідно застосувати такі вирази:

- ❖ для першої $n=1$: $n-1=N$;
- ❖ для останньої $n=N$: $n+1=1$.

Алгоритм Ring-PSO характеризується досить повільним, але детальним пошуком розв'язку на поверхні цільової функції. Тобто етап „розвідки” у цього алгоритму є відносно тривалим. Для деяких задач оптимізації ця властивість є бажаною, оскільки вона дозволяє виявляти потенційні області, які включають глобальний або гарний локальний мінімум функції.

2.2.5 Алгоритм ME-D-PSO

Ця модифікація PSO була розроблена та досліджена в статті [68]. Від канонічного PSO вона відрізняється двома поліпшеннями. Перший з них пов'язаний з переініціалізацією рою, коли він показує погану ефективність пошуку (етап стагнації рою). Критерієм наявності такої стагнації є показник:

$$AR \geq \frac{f(g_j) - f(g_{j-1})}{f(g_j)}, \quad (2.10)$$

де AR – прийнятна швидкість зменшення глобального найкращого на одній ітерації.

Переініціалізація рою означає знищення старого рою та ініціалізацію нового (звичайно всі частинки починають рух із випадкових положень з нульовими швидкостями). Однак, краща частинка у знову ініціалізованому рої розміщується у глобальному кращому попереднього покоління рою і саме до неї спрямовуються частинки одразу після ініціалізації нового покоління рою.

Друге вдосконалення алгоритму ME-D-PSO спрямоване на збільшення різноманітності руху частинок, що забезпечує більш ефективну роботу алгоритму по локалізації мінімуму цільової функції. Цей механізм забезпечується постійною (протягом всіх ітерацій) і стохастичною зміною коефіцієнтів w , c_1 і c_2 в деяких межах: w_{\min} і w_{\max} – для w ; $c_{1\min}$ і $c_{1\max}$ – для c_1 ; $c_{2\min}$ і $c_{2\max}$ – для c_2 .

2.3 Розробка нових модифікацій PSO

2.3.1 Алгоритми PV-IA-PSO

Ймовірність переміщення частинки у нове положення згідно виразу (2.6) для початкових і кінцевих ітерацій однакові – вони не залежать від поточної ітерації. Однак схема пошуку розв'язку задачі „розвідка-дослідження” вимагає великих змін положень частинок на ранній стадії пошуку і малих на кінцевій. Тому доцільно від ітерації до ітерації змінювати ймовірність відхилення положення частинки від того, на яке направлений її вектор v .

У даному дослідженні пропонується зв'язати функціональною залежністю деякий параметр ймовірнісного закону генерації нового положення частинки та номери ітерацій j . У дослідженні будемо розглядати нормальний розподіл стандартного відхилення положення частинки, який змінюється за лінійно спадаючим законом:

$$\begin{cases} x_j = G(x_{j-1}, \sigma^2) + v_j; \\ \sigma = \frac{J-j}{J}(\sigma_{\max} - \sigma_{\min}) + \sigma_{\min}, \end{cases} \quad (2.11)$$

де G – D -мірний вектор числових значень, породжених нормальним законом (законом Гауса) розподілу ймовірностей; σ_{\max} і σ_{\min} – максимальне (початкове) і мінімальне (кінцеве) стандартні відхилення координат частинки від положення, яке відповідає математичному очікуванню нормального

розподілу відповідно. Зазначимо, що інші відомі розподіли ймовірностей також можуть бути використані для цілей побудови ймовірної зміни положення частинки.

Для локалізації розв'язку (аргументів, на яких цільова функція набуває мінімуму) на кінцевих ітераціях виконання алгоритму значення σ_{\min} повинно бути досить малим для того, щоб частинки могли ефективно виконати етап „дослідження”. Для всіх досліджень, які проведені у даній роботі, задавали $\sigma_{\min}=10^{-20}$. Саме така величина забезпечує вказану вимогу і при цьому вираз (2.11) практично редукує у другий вираз системи (2.3). Раціональне значення σ_{\max} буде отримано у подальшому.

Модифікований алгоритм, який включає зміну ймовірності потрапляння у інше (на яке не вказує вектор-швидкості v частинки) положення, тобто у якому використано закон (2.7), позначимо як PV-IA-PSO-1. Щільність ймовірності, яка описується законом (2.11) наведено на рис. 2.1. На рис. 2.1 сірою лінією зображено закон зміни стандартного відхилення (2.7) та щільність ймовірностей $\phi_{\mu, \sigma^2}(x)$, яка при цьому отримана. Аргументом цих функцій є номер ітерації j .

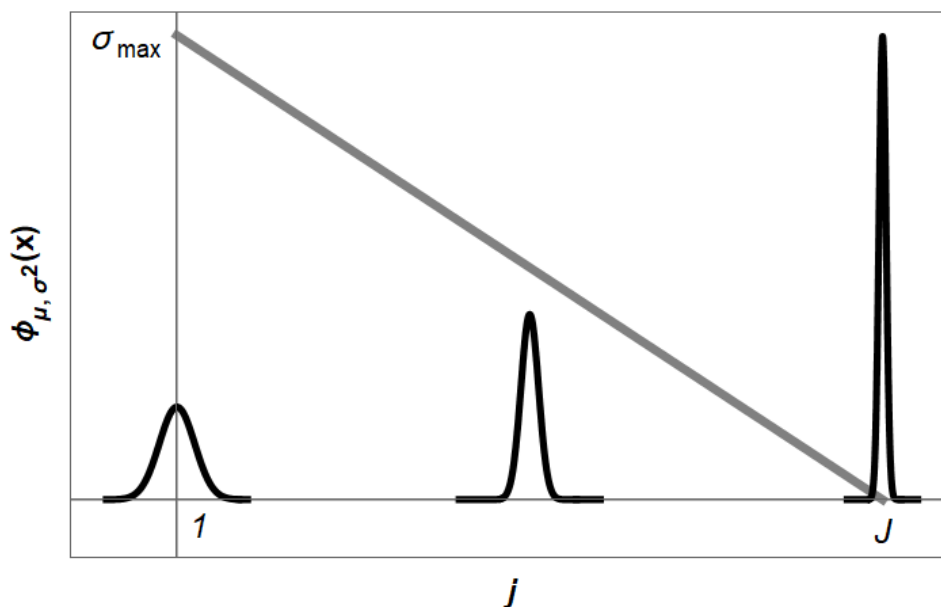


Рисунок 2.1 –Графіки зміни стандартного відхилення σ (сіра лінія) та щільності ймовірності $\phi_{\mu, \sigma^2}(x)$ положення частинки x , як функції номера ітерації j алгоритму

Крім того, будемо розглядати інший варіант зміни параметра σ . При цьому спиратимемось на ідею, яка передбачає самоадаптацію алгоритму. Така стратегія, зокрема, передбачає зміну σ у відповідності до поточних потреб пошуку, тобто в залежності від того, на якій стадії („розвідка” чи „дослідження”) працює алгоритм.

Крім того, будемо розглядати інший варіант зміни параметра σ . При цьому спиратимемось на ідею, яка передбачає самоадаптацію алгоритму. Така стратегія, зокрема, передбачає зміну σ у відповідності до поточних потреб пошуку, тобто в залежності від того, на якій стадії („розвідка” чи „дослідження”) працює алгоритм.

Для того, щоб описати ще одну модифікацію алгоритму IA-PSO необхідно замінити другу стрічку системи (2.7) наступним виразом:

$$\sigma = B^{x_{D+1}}, \quad (2.12)$$

де x_{D+1} – новий аргумент задачі, на що вказує його нижній індекс; B – основа степеневій залежності – параметр алгоритму, який необхідно задавати. Аргумент x_{D+1} може змінюватись у процесі роботи алгоритму таким чином, що під час виконання алгоритму відбувається налаштування параметра σ . Також необхідно пояснити, чому вираз (2.12) включає степеневу залежність. Вона обумовлена тим, що зміна аргументу x_{D+1} веде до значної зміни величини σ , чого необхідно досягнути при виконанні алгоритму PV-IA-PSO-2: якщо на початку його роботи необхідно відносно значна величина σ (аргумент x_{D+1} повинен бути близьким до 1), то у кінці роботи алгоритму різноманітність руху частинок повинна зменшуватись і тому величина σ повинна бути дуже малою (у випадку з PV-IA-PSO-1 $\sigma_{\min}=10^{-20}$). Отриману нову модифікацію позначимо PV-IA-PSO-2.

Передбачається, що алгоритм PV-IA-PSO-2 повинен самостійно відшукати оптимальне значення аргументу x_{D+1} , яке позитивно вплине на

пошукові властивості рою. Як буде показано нижче, параметр σ має значний вплив на ці властивості. Тому нераціональне задання σ , яке викликано стохастичним характером пошуку мінімуму цільової функції, може погіршити пошукові властивості PV-IA-PSO-2. Для того, щоб дати однозначну оцінку алгоритмам PV-IA-PSO-1 та PV-IA-PSO-2 необхідно проводити додаткове дослідження, результати якого наведені у наступному викладі.

Як буде показано в подальшому дослідженні, такі відносно прості модифікації алгоритму IA-PSO значно покращують можливості знаходження мінімуму цільової функції. Це відбувається за рахунок підтримки різноманітності руху частинок, яка закладена у першому рівнянні системи (2.7).

2.3.2 Алгоритм SR-D-PSO

У роботі [69] була запропонована та досліджена відносно проста техніка переініціалізації цілого рою. Його подальший розвиток здійснено в статті [68]: він передбачає випадкову зміну коефіцієнтів w , c_1 , c_2 на кожній ітерації алгоритму (описаний у п.п. 2.2.5 алгоритм ME-D-PSO). Це призвело до збільшення різноманітності рою та покращення його пошукових можливостей.

У цьому пункті розроблено інший механізм переініціалізації, відповідний алгоритм називається SR-D-PSO [70] (м'яка переініціалізація – скорочено SR, від англійської фрази „softreinitialization”; з підтримкою різноманіття рою – скорочено D, від англійського слова „diversity”).

М'яка (або локальна) переініціалізація рою застосовується лише до кількох частинок у рої і їх загальна кількість ζ завжди менша за популяцію всього рою N . Такий підхід дозволяє рою продовжувати пошук, оновлюючи свою популяцію, що, в свою чергу, позитивно впливає на пошукові властивості алгоритму.

Раціональне число ξ невідоме апіорі, воно має бути обґрунтоване в подальших дослідженнях. Для реалізації цього прийому після застосування формул (2.4) необхідно скористатись наступним виразом:

$$\begin{cases} l = \text{Random int}(1, N); l \neq b; \\ x_{l,j} = \text{Random}(x_{l,\min}, x_{l,\max}); \\ v_{l,j} = 0; \\ l \in \overline{(1, \xi)}, \xi < N, \end{cases} \quad (2.13)$$

де $x_{l,\min}$ та $x_{l,\max}$ – нижня та верхня межі області пошуку l -го аргументу цільової функції f ; b – номер частинки, яка пов'язана з вектором глобального найкращого g .

Другою відмінністю між SR-D-PSO [70] та ME-D-PSO [68] є частота повторної ініціалізації: вираз (2.9) слід застосовувати на кожній ітерації. Нагадаємо, що для методу ME-D-PSO рій повторно ініціалізується, коли виконується умова стагнації рою (2.10), що може трапитись не на кожній ітерації.

Підсумовуючи все, можна зробити висновок, що SR-D-PSO відрізняється від ME-D-PSO як частотою, так і об'ємом переініціалізації рою.

2.3.3 Алгоритм M-PSO-Hybrid (гібридизація модифікацій PSO)

Як було відзначено у першому розділі роботи, наразі значна увага науковців приділяється задачам гібридизації PSO. У цьому підпункті опишемо один із можливих гібридів PSO, який включає лише алгоритми, що створені на основі модифікацій PSO. Такий акцент пов'язаний з тим, що досить просто алгоритмічно виконати передачу інформації від однієї модифікації PSO до іншої. Ця передача інформації полягає у передачі від алгоритму до алгоритму цілого рою. Тобто на початку роботи певного алгоритму (модифікація PSO) вже існує зініціалізований рій, тобто рій, який закінчив роботу у попередньому алгоритмі.

Однак, варто зазначити, що розроблений тут алгоритм може бути узагальнений і на інші методи чисельної оптимізації, якщо існує спосіб передати цінну інформацію (розв'язок або множину розв'язків) від одного алгоритму до іншого. Така інформація може бути передана від одного алгоритму до іншого та ефективно ним використана.

Ідея M-PSO-Hybrid (гібридизація на основі множини модифікацій PSO) полягає в наступному: існує множина модифікацій PSO та „інтелектуальний комутатор”, який певним чином обирає та запускає той чи інший алгоритм (модифікацію PSO) із множини алгоритмів відповідно до ефективності його роботи.

Для того, щоб забезпечити різноманітність патернів (схем) пошуку розв'язку необхідно, щоб множина модифікацій PSO містила достатньо велику кількість модифікацій PSO і щоб кожен з них реалізував різний підхід до збільшення пошукових властивостей канонічного методу PSO. Ця вимога пов'язана з тим, що наперед невідомо, яка саме модифікація PSO буде краще працювати і на якому з етапів пошуку.

Задача „інтелектуального комутатора” полягає у тому, щоб з поміж всіх алгоритмів обрати кращий і передати йому обчислювальні ресурси для виконання пошуку розв'язку задачі. Скажімо, якщо певна модифікація PSO протягом ψ ітерацій показує високу ефективність (зменшення цільової функції досягнути протягом ψ ітерацій є значимим), то „інтелектуальний комутатор” працює таким чином, що існує висока ймовірність застосування цієї ж модифікації для наступних ψ ітерацій. „Інтелектуальний комутатор” сприяє застосуванню найбільш ефективних (на конкретному етапі пошуку) модифікацій PSO. Працюючи таким чином, він реалізує стратегію елітарності всього алгоритму M-PSO-Hybrid.

Виконаємо більш формальний опис роботи M-PSO-Hybrid. Для цього позначимо кількість усіх модифікацій PSO як A , а звертання до певної модифікації будемо виконувати використовуючи індекс a ($a = \overline{(1, A)}$). Пояснення M-PSO-Hybrid у вигляді псевдокоду наведено нижче:

*(*Початок роботи алгоритму M-PSO-Hybrid *)*

Задати всі загальні параметри алгоритму (популяція рою N , критерій зупинки) та специфічні параметри алгоритму (тобто такі, що відносяться лише до конкретних модифікацій PSO із множини A);

Ініціалізувати положення і швидкості частинок;

Обчислити особисті найкращі (personalbest) та глобальне найкраще рою (globalbest);

(Визначення ефективності роботи кожного з алгоритмів *)*

Запустити першу модифікацію PSO протягом ψ ітерацій;

Обчислити ефективність її роботи – величину \tilde{P}_1 ;

Передати рій (положення, швидкості частинок, а також особисті найкращі) наступній модифікації PSO;

...

Запустити a -ту модифікацію PSO протягом ψ ітерацій;

Обчислити ефективність її роботи – величину \tilde{P}_a ;

Передати рій (положення, швидкості частинок, а також особисті найкращі) наступній модифікації PSO;

...

Запустити A -ту модифікацію PSO протягом ψ ітерацій;

Обчислити ефективність її роботи – величину \tilde{P}_A ;

(Вибір модифікацій PSO та їх застосування *)*

Робити поки умова зупинки алгоритму не буде виконана

Якщо $\tilde{P}_1=0$ I ... $\tilde{P}_a=0$ I ... $\tilde{P}_A=0$, **то** вибрати модифікацію PSO випадковим чином, передати їй рій (положення, швидкості частинок, а також особисті найкращі) та запустити її на виконання протягом ψ ітерацій,

інакше вибрати модифікацію PSO шляхом зваженого випадкового вибору (a -та модифікація PSO може бути обрана для виконання протягом наступних ψ ітерацій з ймовірністю \tilde{P}_a).

(*Кінець роботи алгоритму M-PSO-Hybrid *)

Для розрахунку ефективності a -ї модифікації PSO використовується наступна формула:

$$P_a = \frac{g_j - g_{j+\psi}}{g_j + \varepsilon}, \quad (2.14)$$

де ε – невелике значення (епсілон), яке дозволяє уникнути ділення на нуль; g_j та $g_{j+\psi}$ – глобальні найкращі на початку роботи a -тої модифікації PSO та у кінці її роботи, тобто після ψ ітерацій.

Якщо всі модифікації PSO відповідають нульовій ефективності (випадок збіжності M-PSO-Hybrid), тоді „інтелектуальний комутатор” проводить незважений випадковий вибір.

Існує шанс уникнути збіжності до локального мінімуму, якщо серед усіх модифікацій PSO із множини застосовуваних існує хоча б один, який забезпечує повну або часткову переініціалізацію рою. У цьому випадку повторно ініціалізовані частинки (або одна частинка) можуть знайти кращий розв’язок, ніж той, що був знайдений до переініціалізації, і рій продовжить пошук.

Величина ψ сильно впливає на ефективність M-PSO-Hybrid. Дійсно, мале значення ψ не дозволяє повною мірою розкрити пошукові властивості модифікацій PSO: для цього потрібно виконати декілька ітерацій. Алгоритм просто не встигне проявити свої бажані на даному етапі роботи пошукові властивості. З іншого боку, велике значення ψ може призвести до неефективного витрачання обчислювальних ресурсів. Модифікації PSO, які не відповідають конкретному етапу пошуку, тобто показують низьку ефективність, витрачають загальні обчислювальні ресурси неефективно. Тому існує необхідність дослідження впливу значення ψ на ефективність роботи всього алгоритму M-PSO-Hybrid. Такі дослідження будуть проведені у наступному викладі.

2.3.4 Алгоритм LG-Ring-PSO

Як було вказано у першому розділі роботи топологія зв'язків між частинками відіграє важливу роль у формуванні пошукових властивостей того чи іншого алгоритму (модифікації) PSO. Топології lbest характеризуються детальним пошуком перспективних областей на поверхні цільової функції, однак вони досить повільно сходяться до розв'язку. Натомість топології gbest мають швидшу збіжність, але вони почасти можуть пропускати області, де знаходиться шуканий розв'язок задачі, а отже така швидка збіжність алгоритму є передчасною.

Існує декілька способів поєднання бажаних властивостей топологій lbest та gbest, однак, вони також мають певні недоліки і не повною мірою відповідають схемі пошуку розв'язку задачі „розвідка -дослідження”.

У даному пункті описано розроблений алгоритм, який характеризується компромісним підходом у застосуванні топологій lbest та gbest. Їхнє застосування відбувається за наперед заданим законом. Саме цей закон відповідає потрібній схемі розв'язку задачі „розвідка -дослідження”.

Розроблений алгоритм (назвемо його LG-Ring-PSO, перші літери відповідають структурі пошуку: спочатку працює топологія lbest, а потім gbest) полягає у тому, що третій доданок виразу (2.8) змінюється. Для цього випадку формула оновлення вектору-швидкості частинки v представляється у такому вигляді:

$$v_j = w \circ v_{j-1} + c_1 r_1 \circ (p_{j-1} - x_{j-1}) + c_2 r_2 \circ (ggl_{j-1} - x_{j-1}), \quad (2.15)$$

де ggl – вектор положення, який має компоненти глобального кращого всього рою g та глобального кращого локального рою gl . Для опису схеми формування d -ї компоненти вектору ggl на j -тій ітерації роботи алгоритму LG-Ring-PSO використовується наступний вираз:

$$\text{ggl}_j^d = \begin{cases} \left(\frac{j}{J}\right)^a \geq r, \text{g}_j^d; \\ \left(\frac{j}{J}\right)^a < r, \text{gl}_j^d, \end{cases} \quad (2.16)$$

де r – випадкове число, що згенероване на проміжку від 0 до 1 з математичним очікуванням рівним 0,5; a – додатковий параметр, який визначає швидкість переходу протягом виконання алгоритму від схеми пошуку $lbest$ до $gbest$. Причому цей перехід від $lbest$ до $gbest$ відбувається плавно, із певною вірогідністю. Він застосовується не до всього вектору положення, який відповідає за соціальну складову руху частинки, а лише до його окремих компонент. Для візуальної оцінки закону зміни вірогідності використання компонентів векторів gl та g у виразі (2.16) наведемо графічні залежності (рис. 2.2). На рис. 2.2 кількість ітерацій задана рівною 100, ця величина задана лише для прикладу, вона не має впливу на форму закону.

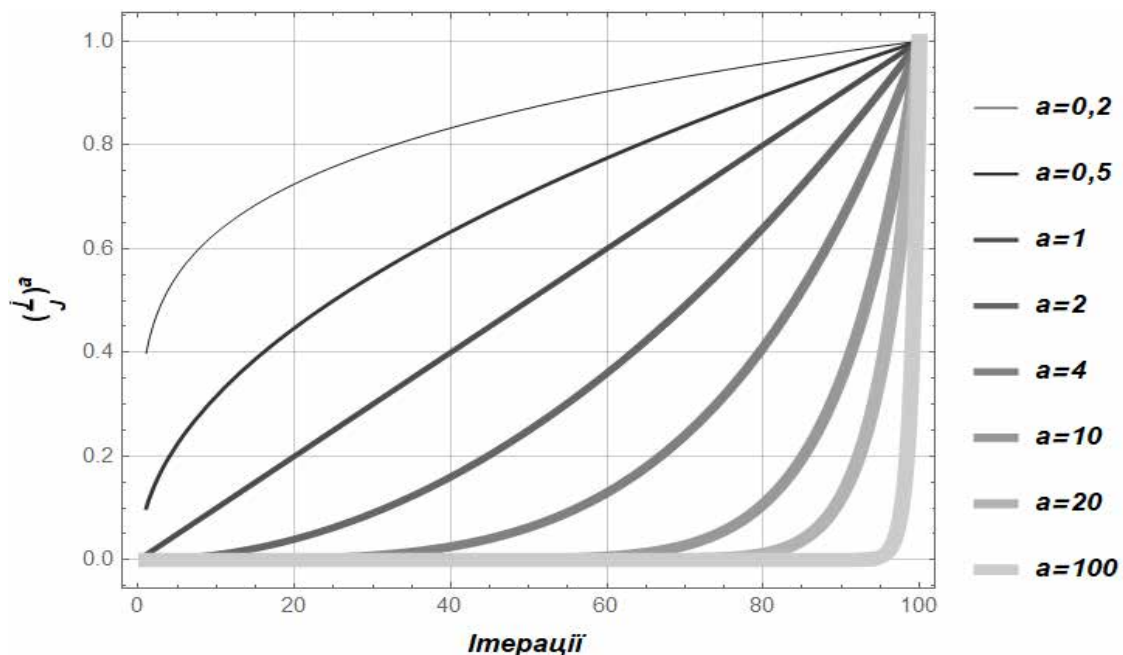


Рисунок 2.2 – Графіки зміни математичного очікування застосування у певному компоненті вектору ggl компонента вектору g

З рис. 2.2 видно, що для варіантів $a > 1$ превалює схема пошуку $lbest$. При $a < 1$ більший вплив залишається за $gbest$. Враховуючи те, що

математичне очікування r рівне 0,5, можемо дати грубу оцінку того, після скількох ітерацій більший вплив матиме схема пошуку gbest (табл. 2.1).

Таблиця 2.1 – Процент виконаних ітерацій алгоритму LG-Ring-PSO, після яких вплив векторуг у формуванні ggl стає домінуючим

Параметр a	0,2	0,5	1	2	4	10	20	100
Процент виконаних ітерацій	3	25	50	71	84	93	97	99

Необхідно підкреслити те, що через ймовірнісний підхід у формуванні вектору ggl застосування окремих компонентів g може виконуватись і на ранніх стадіях пошуку. Однак, вірогідність цього досить незначна.

У граничних випадках можна записати:

$$\begin{cases} \lim_{a \rightarrow \infty} ggl = gl; \\ \lim_{a \rightarrow 0} ggl = g. \end{cases} \quad (2.17)$$

Таким чином, схеми пошуку lbest та gbest можна розглядати як частинні варіанти виразу (2.15).

2.4 Інші модифікації PSO, що використані в дослідженні

Для подальшого дослідження обрано кілька алгоритмів на основі PSO. Вони посилюють ту чи іншу пошукову властивість канонічного алгоритму PSO, їхні пошукові процедури здійснюються з деякими відмінностями. Останні мають сильний вплив на ефективність відшукування мінімуму тієї чи іншої цільової функції. Нижче наведено список алгоритмів на основі PSO та деякі їхні особливості:

1) LDW-PSO. Згідно з роботою [13] інерційний коефіцієнт у першому виразі системи (2.3) описується наступним чином:

$$w = (w_{\min} - w_{\max}) \frac{J - j}{J} + w_{\min}, \quad (2.18)$$

де w_{\min} і w_{\max} – мінімальна та максимальна межі області зміни інерційних коефіцієнтів. Це дуже поширена (можливо найпоширеніша) і відома завдяки своїй простоті модифікація PSO, яка дозволяє покращити ефективність роботи оптимізаційного алгоритму досить простим способом – із застосуванням лінійно-спадаючої характеристики (2.18). Величини w_{\min} і w_{\max} , як правило, приймаються рівними 0,9 та 0,4 відповідно.

2) R-Ring-PSO. Кільцева топологія зв'язків частинок, описана раніше (алгоритм Ring-PSO), є статичною і для багатьох задач оптимізації процедура пошуку мінімуму цільової функції є дуже повільною. Для подолання цього недоліку при умові збереження властивості локального обміну інформації (між частинкою та її сусідами) у роботі [71] була розроблена та досліджена динамічна топологія зв'язків між частинками „обертове кільце”. Обертання кільця зв'язків для окремої частинки можна розглядати як зміну її сусідів. Її реалізація передбачає застосування наступної формули:

$$gl_{j,n} = \min_{\arg} \left(f(p_{j.\text{mod}(\frac{n-1+s \cdot j}{N})}), f(p_{j,n}), f(p_{j.\text{mod}(\frac{n+1+s \cdot j}{N})}) \right), \quad (2.19)$$

де s – параметр, що відповідає за швидкість обертання кільця зв'язків між частинками (ціле число). Чим більше s , тим більше зміщення (відстань обертання), яка припадає на одну ітерацію алгоритму.

У роботі [72] алгоритм R-Ring-PSO успішно застосовано для розрахунку налаштування пристрою плавного пуску приводу стрічкового конвеєра.

Усі параметри алгоритмів, які будуть використані у даному дослідженні, наведено в табл.2.2.

Крім того, у Додатку Б наведено код, що розроблений у середовищі Wolfram Mathematica і який реалізує всі описані у даному розділі алгоритми.

Таблиця 2.2 – Параметри розроблених та використаних у дослідженні алгоритмів PSO

Параметри рою	Алгоритм										
	LDW- PSO	IA- PSO	VCT- PSO	Ring- PSO	R-Ring- PSO	ME-D- PSO	PV-IA- PSO-1	PV-IA- PSO-2	SR-D- PSO	M-PSO- Hybrid*	LG-Ring- PSO
Топологія зв'язків між частинками	повна			неповна		повна				повна+ неповна	неповна
c_1	1,19					змінний	1,19			змінний	1,19
c_2	1,19					змінний	1,19			змінний	1,19
w	змінний		0,72			змінний			0,72	змінний	0,72
Параметри конкретних алгоритмів	$w_{\max}=0,9; w_{\min}=0,4$	-	$RC=5$	-	$s=1$	$AR=0,05; w_{\min}=$ $c_{1\min}=c_{2\min}=0; w_{\max}=1;$ $c_{1\max}=c_{2\max}=2$	σ_{\max}^{**}	B^{**}	ζ^{**}	ψ^{**}	a^{**}

* всі параметри, які відносяться до алгоритму M-PSO-Hybrid, відповідають тим алгоритмам, які в нього включені (LDW-PSO, PV-IA-PSO-1, R-Ring-PSO та VCT-PSO); кожен такий алгоритм працює ψ ітерацій;

** оптимальні значення σ_{\max} , B , ζ , ψ та a будуть знайдені у наступному викладі роботи.

Висновки до другого розділу:

1. У розділі наведено опис канонічного алгоритму PSO та його модифікацій (IA-PSO, VCT-PSO, Ring-PSO, ME-D-PSO), на основі яких проведено подальші розробки модифікацій і гібридизацій алгоритму.
2. Розроблено алгоритми PV-IA-PSO-1 та PV-IA-PSO-2, які ґрунтуються на ефекті коливань вектору положення частинки при його оновленні. У варіанті PV-IA-PSO-1 зміна стандартних відхилень при таких випадкових коливаннях положення частинки відбувається за лінійно-спадаючим законом. Ця стратегія відповідає схемі пошуку мінімуму „розвідка- дослідження”. При цьому в алгоритм PV-IA-PSO-1 вводиться два додаткові параметри – максимальне (початкове) σ_{\max} та мінімальне (кінцеве) σ_{\min} значення стандартних відхилень. Останнє можна задавати достатньо малим (наприклад 10^{-20}) для того, щоб мінімізувати коливання положення частинки на пізніх стадіях пошуку (бажана характеристика, яка дозволяє точно локалізувати мінімум цільової функції). У варіанті PV-IA-PSO-2 зміна стандартного відхилення положення частинки розглядається як додатковий аргумент цільової функції, що має істотне значення лише під час виконання алгоритму.
3. Проведено подальший розвиток механізмів переініціалізації рою, які дали змогу отримати алгоритм SR-D-PSO. В ньому на кожній ітерації виконується переініціалізація певної кількості частинок, за виключенням найкращої.
4. Розроблено гібридний алгоритм M-PSO-Hybrid, суть роботи якого полягає у послідовному включенні у роботу різних алгоритмів (модифікацій PSO, які мають високі пошукові властивості та відрізняються один від одного різними патернами пошуку мінімуму цільової функції). Робота того чи іншого алгоритму виконується протягом ψ ітерацій і надалі рій передається до наступного алгоритму

(складової M-PSO-Hybrid). Вірогідність „вмикання” тієї чи іншої складової M-PSO-Hybrid залежить від ефективності її роботи.

5. Розроблено підхід до вдосконалення пошукових можливостей алгоритмів на основі PSO. Він поєднує lbest і gbest топології зав'язків між частинками, що краще відповідає схемі пошуку розв'язку задачі „розвідка-дослідження”. У розробленій модифікації алгоритму PSO (алгоритм LG-Ring-PSO) на ранніх етапах пошуку переважає топологія lbest, наприкінці пошуку – gbest. Перемикання між цими типами топологій відбувається плавно, воно має ймовірнісний характер і регулюється додатковим параметром алгоритму a . У розділі показано, що відомі топології lbest та gbest є частинними випадками розробленої схеми, яка використана у алгоритмі LG-Ring-PSO.

РОЗДІЛ 3

ОПИС ТЕСТОВИХ ФУНКЦІЙ

У подальших дослідженнях будуть розглянуті алгоритми, що отримані на основі PSO. Їх ефективність можна оцінити шляхом застосування для розв'язування реальних задач оптимізації. Однак їх існує безліч, всі вони відрізняються один від одного кількістю аргументів (D), масштабом і топологією цільової функції, обмеженнями тощо. Тому існують синтетичні (створені штучно) тестові цільові функції, які використовуються для оцінки ефективності різних алгоритмів оптимізації. У цьому розділі дано опис двадцяти відомих тестових функцій, які, зазвичай, використовуються для оцінки роботи алгоритмів оптимізації.

Під час вибору тестових функцій намагалися підтримувати баланс між різними характеристиками (сепарабельністю, унімодальністю тощо). Це було обґрунтовано тим фактом, що алгоритми оптимізації можуть бути застосовані до різних задач з різними цільовими функціями. Щоб оцінити можливу ефективність алгоритмів включили до розгляду двадцять тестових функцій, які мають різні топологічні особливості.

Всі тестові функції отримані із наступних джерел [52, 73-78]. Формули вибраних тестових функцій, їхні області визначень, глобальні мінімуми та їхні розташування (якщо вони відомі), сепарабельність та унімодальність наведені в табл.3.1.

Щоб дати уявлення про те, які тестові функції розглядаються, у табл. 3.2 наведено їхні графіки (як $3D$, так і контурні зображення). Зауважимо, що графічні об'єкти (табл. 3.2) не відображають усіх топологічних особливостей тестових функцій. Наприклад, збільшення розмірності окремих тестових функцій асоціюється зі збільшенням мінімумів, а також створенням областей, які є складними для подолання.

Таблиця 3.1 – Набір тестових функцій та деякі їхні властивості

Тестова функція	Формула	Область визначення	Розташування глобального мінімуму x_i	Значення глобального мінімуму	Сепарабельність	Унімодальність
1	2	3	4	5	6	7
Сферична	$f1 = \sum_{i=1}^D x_i^2$	$-20 \leq x_i \leq 20$	0	0	+	+
Еліптична	$f2 = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2$	$-2 \leq x_i \leq 2$	0	0	+	+
Експоненціальна	$f3 = -\exp\left(-0,5 \sum_{i=1}^D x_i^2\right)$	$-10 \leq x_i \leq 10$	0	-1	-	+
Швефеля 1.2	$f4 = \sum_{i=1}^D \left(\sum_{j=1}^i x_j\right)^2$	$-10 \leq x_i \leq 10$	0	0	-	+
Розенброка	$f5 = \sum_{i=1}^{D-1} \left(100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right)$	$-10 \leq x_i \leq 10$	1	0	-	+
Растригіна	$f6 = \sum_{i=1}^D \left(x_i^2 - 10 \cos(2\pi x_i) + 10\right)$	$-5 \leq x_i \leq 5$	0	0	+	-
Гриванка	$f7 = 4000^{-1} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(x_i i^{-0,5}) + 1$	$-100 \leq x_i \leq 100$	0	0	-	-
Альпайна 1	$f8 = \sum_{i=1}^D x_i \sin(x_i) + 0,1 x_i $	$-10 \leq x_i \leq 10$	0	0	+	-

1	2	3	4	5	6	7
Швефеля 2	$f9 = D^{-1} \sum_{i=1}^D (x_i \sin(x_i ^{0,5})) + 418,983$	$-500 \leq x_i \leq 500$	-420,9687	0	+	-
Еклі	$f10 = -20 \exp\left(-0,2 \left(D^{-1} \sum_{i=1}^D x_i^2\right)^{0,5}\right) -$ $-\exp\left(D^{-1} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	$-30 \leq x_i \leq 30$	0	0	+	-
Вейерштрасса	$f11 = D^{-1} \sum_{i=1}^D \sum_{k=0}^{20} (0,5^k \cos(2\pi 3^k (x_i + 0,5))) -$ $-\sum_{k=0}^{20} (0,5^k \cos(\pi 3^k))$	$-0,5 \leq x_i \leq 0,5$	0	0	+	-
Швефеля 2.20	$f12 = \sum_{i=1}^D x_i $	$-10 \leq x_i \leq 10$	0	0	+	+
Цина	$f13 = \sum_{i=1}^D (x_i^2 - i)^2$	$-500 \leq x_i \leq 500$	$\pm i^{0,5}$	0	+	-
Саломона	$f14 = 1 - \cos\left(2\pi \left(\sum_{i=1}^D x_i^2\right)^{0,5}\right) + 0,1 \left(\sum_{i=1}^D x_i^2\right)^{0,5}$	$-100 \leq x_i \leq 100$	0	0	-	-
Сінь-Ше Яна 2	$f15 = \sum_{i=1}^D x_i \exp\left(-\sum_{i=1}^D \sin(x_i^2)\right)$	$-2\pi \leq x_i \leq 2\pi$	0	0	-	-

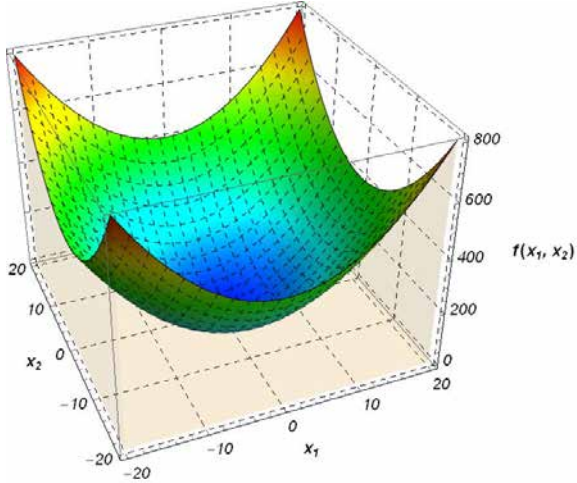
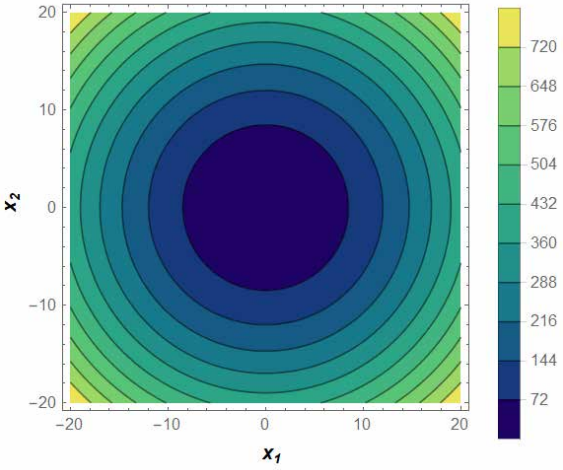
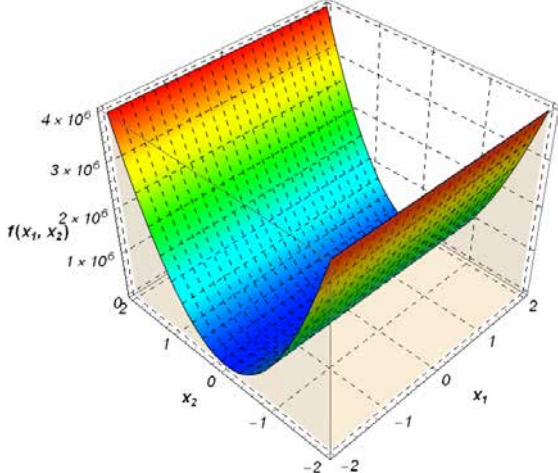
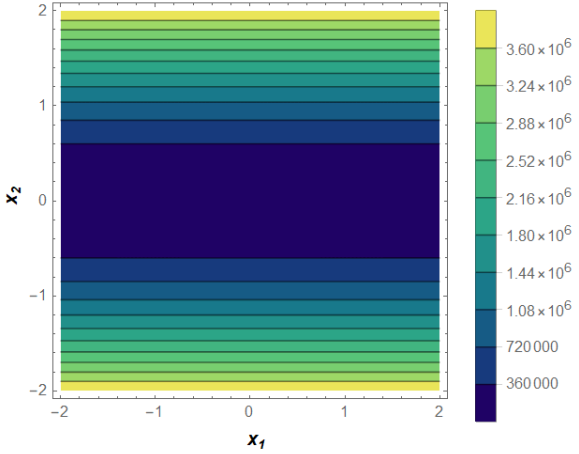
Продовження табл. 3.1

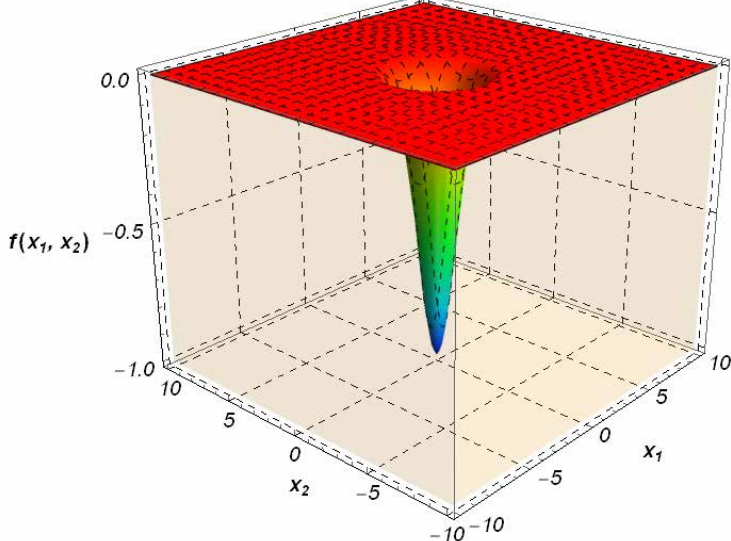
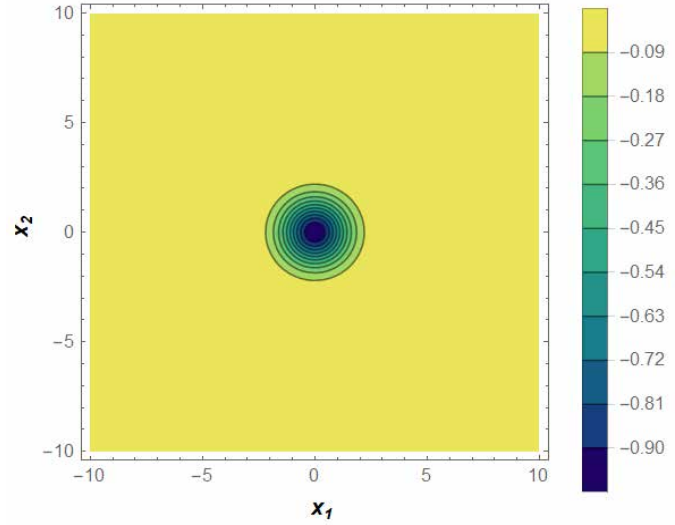
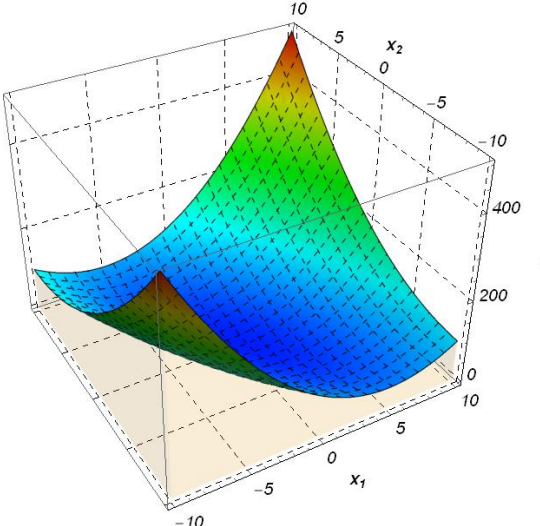
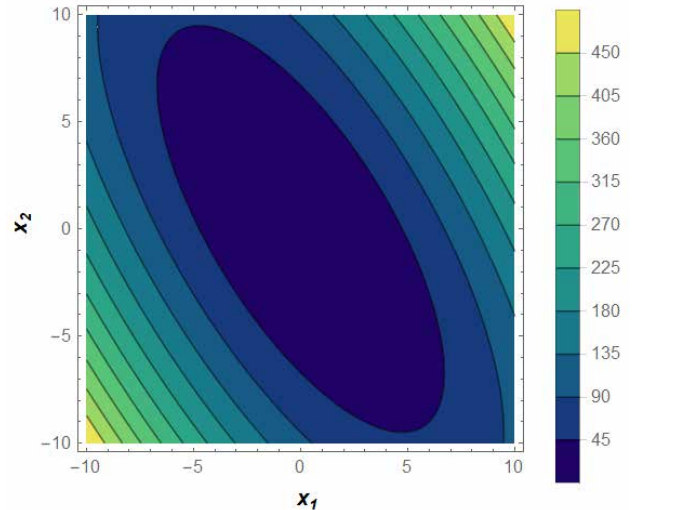
1	2	3	4	5	6	7
Шуберта 4	$f16 = \sum_{i=1}^D \left(\sum_{j=1}^5 j \cos((j+1)x_i + j) \right)$	$-10 \leq x_i \leq 10$	*	**	+	-
Ріджа	$f17 = x_1 + \left(\sum_{i=2}^D x_i^2 \right)^{0,5}$	$-5 \leq x_i \leq 5$	-5	-5	-	+
Щасливий кіт	$f18 = \left(\left(\sum_{i=1}^D x_i^2 - D \right)^2 \right)^{0,125} +$ $+ D^{-1} \left(0,5 \sum_{i=1}^D x_i^2 + \sum_{i=1}^D x_i \right) + 0,5$	$-2 \leq x_i \leq 2$	-1	0	-	-
Гіммельблау	$f19 = \sum_{i=1}^D (x_i^4 - 16x_i^2 + 5x_i)$	$-6 \leq x_i \leq 6$	*	**	-	-
Міхалевича	$f20 = - \sum_{i=1}^D \sin(x_i) \left(\sin \left(\frac{ix_i^2}{\pi} \right) \right)^{20}$	$0 \leq x_i \leq \pi$	*	-49,6248323	-	+

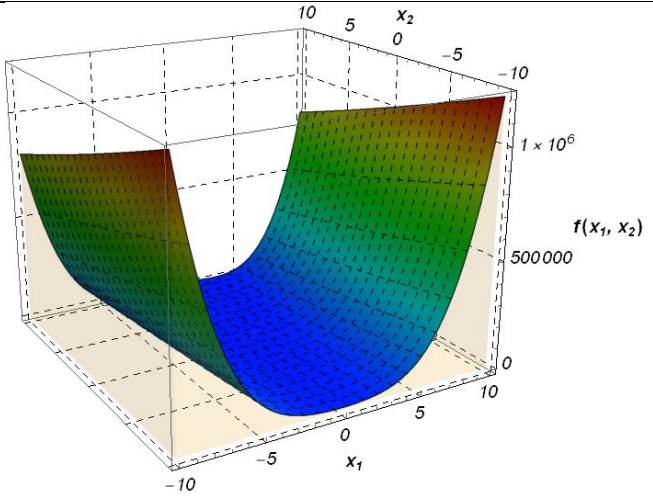
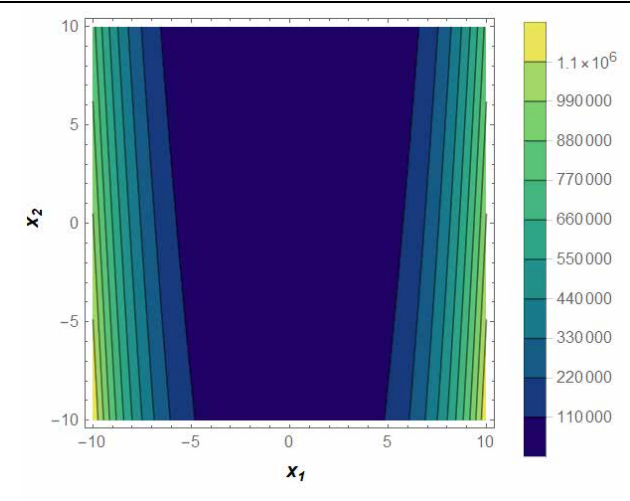
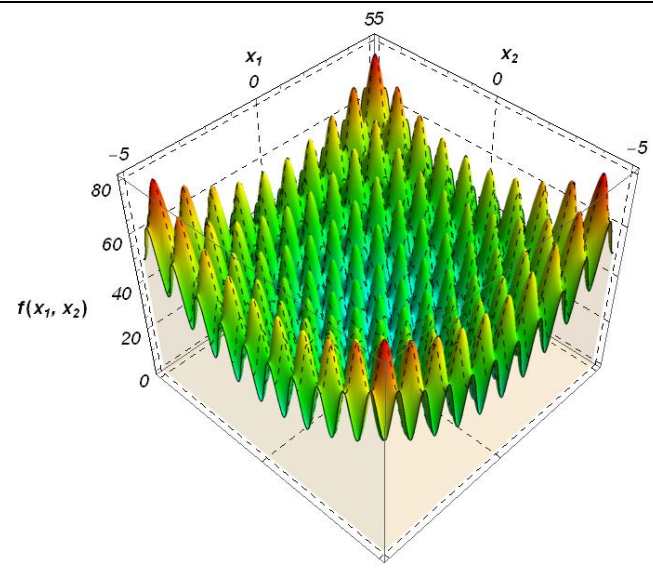
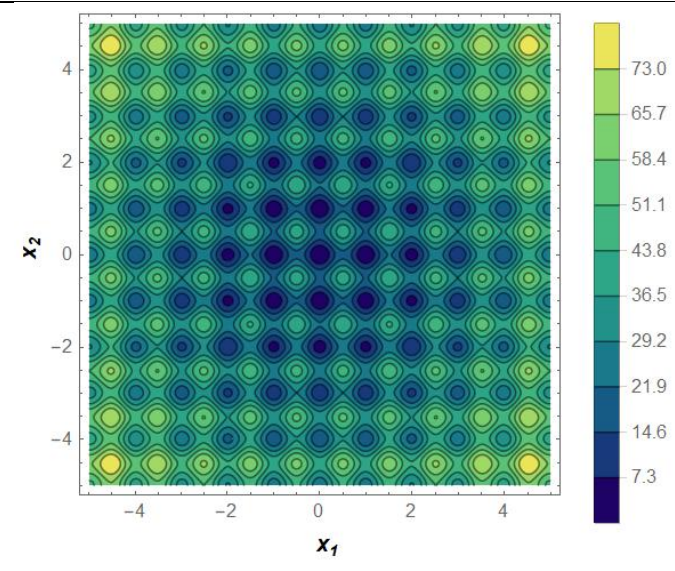
* відсутні надійні дані щодо розташування глобального мінімуму для загального випадку (принаймні для випадку $D=50$);

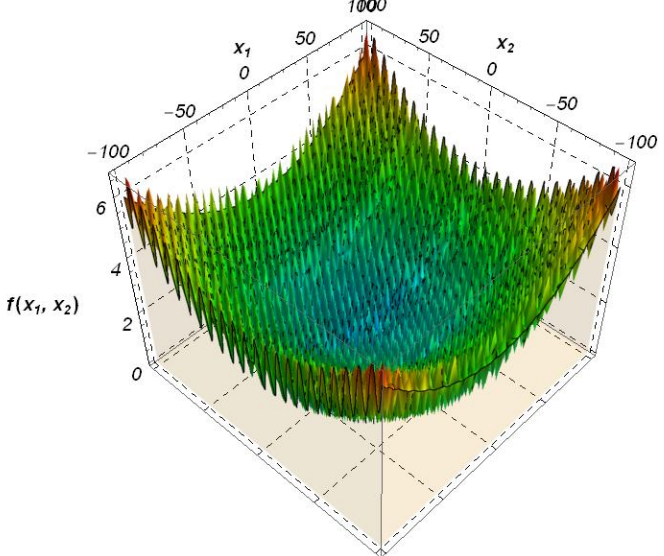
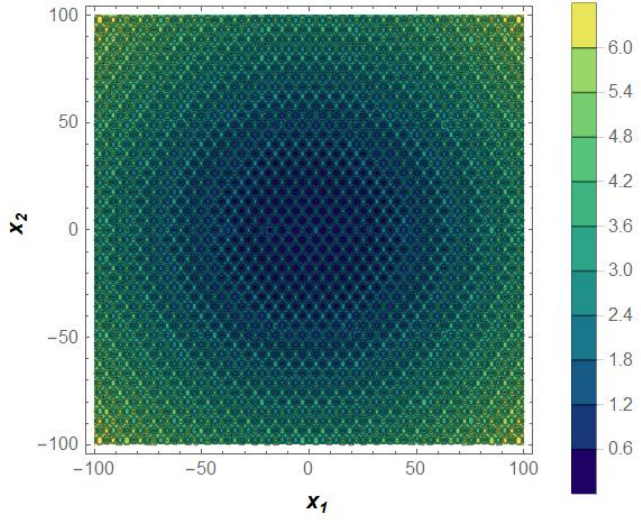
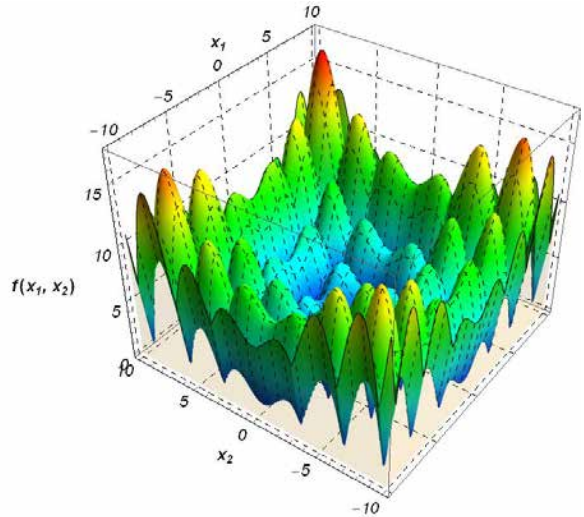
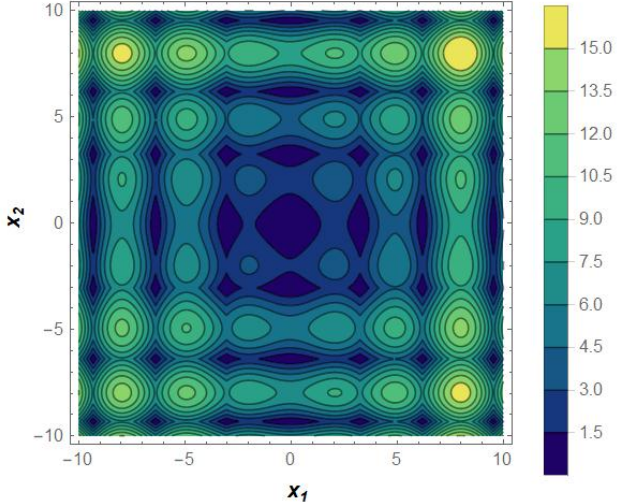
** відсутні надійні дані щодо значення глобального мінімуму для випадку $D=50$.

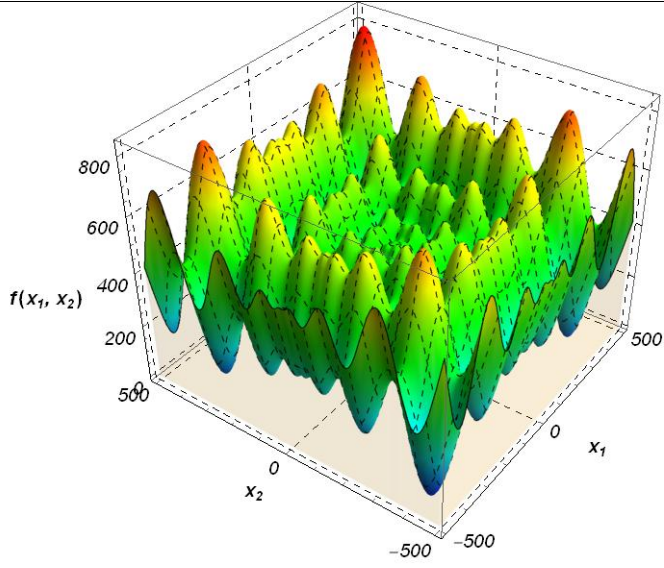
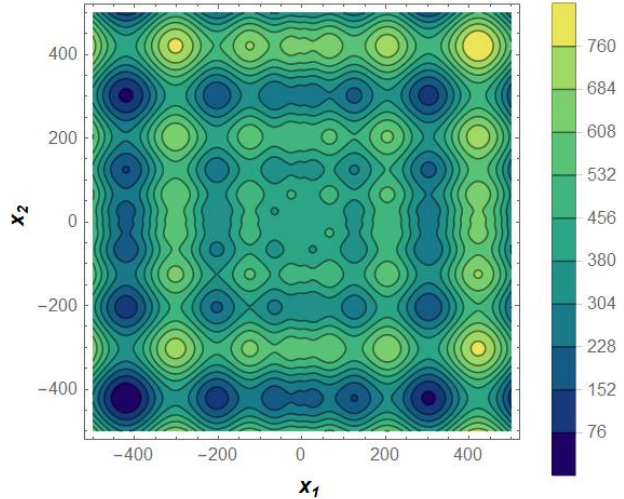
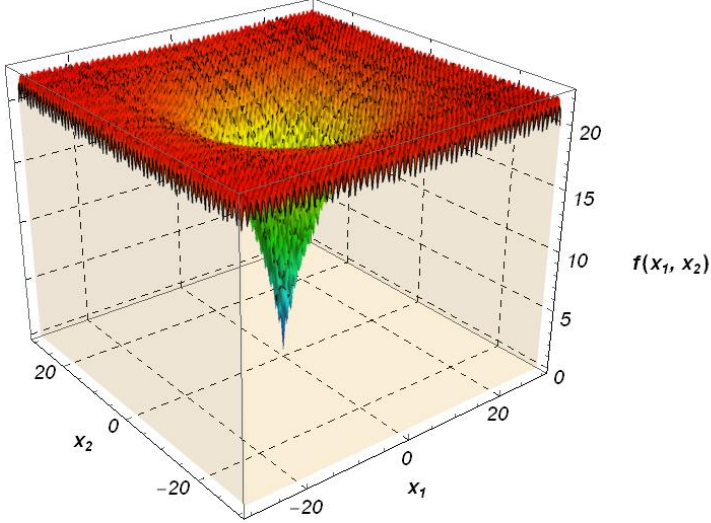
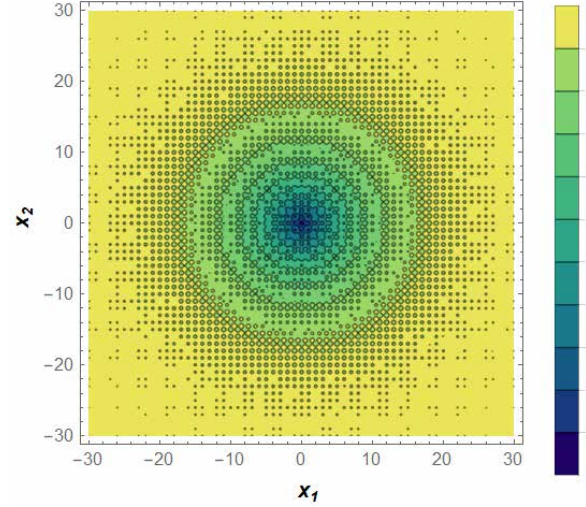
Таблиця 3.2 – Графіки тестових функцій

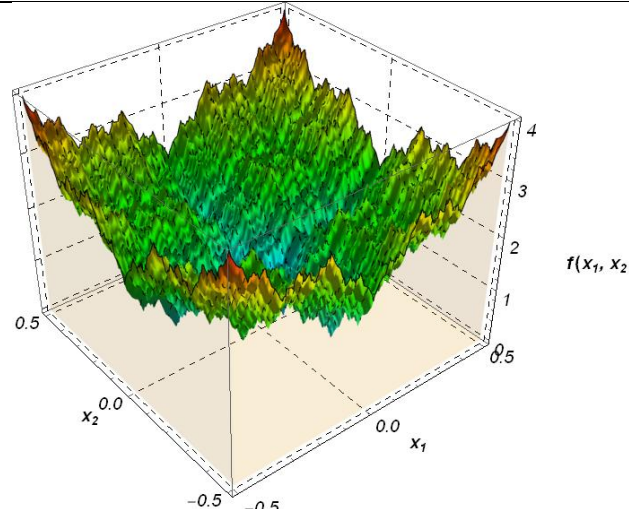
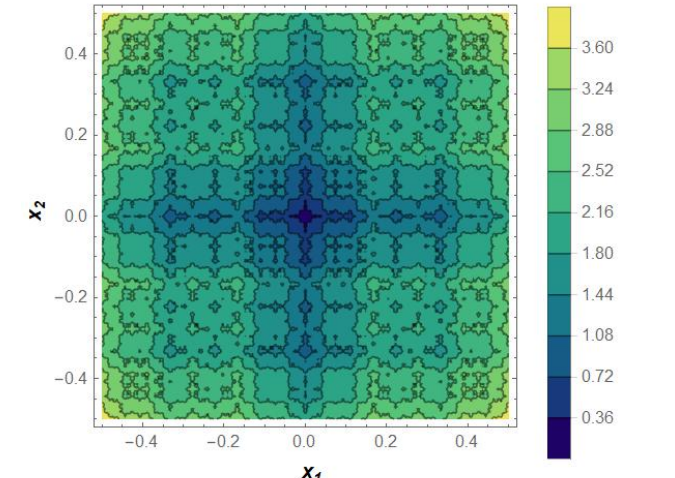
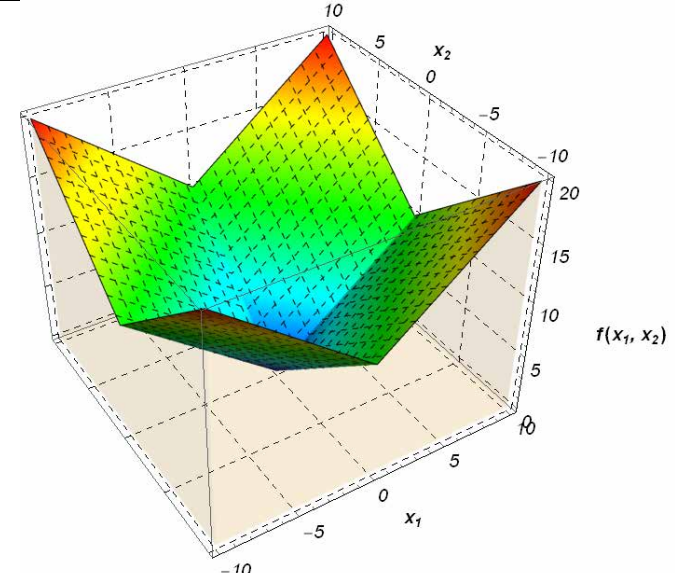
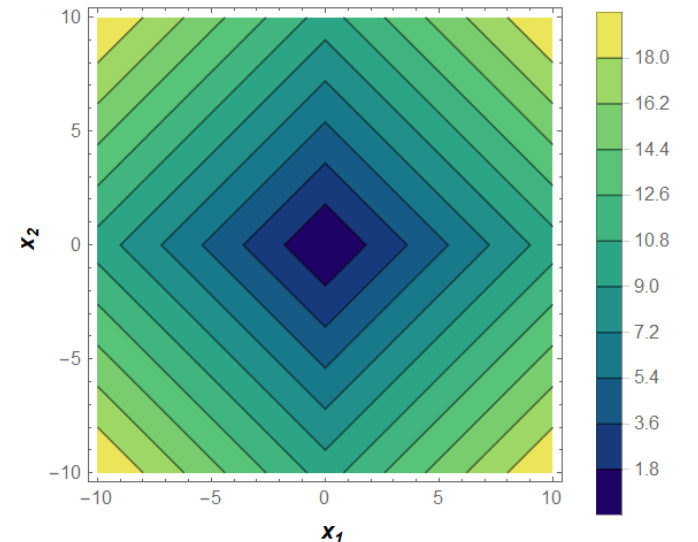
Тестова функція	3D-графік	Контурний графік
1	2	3
Сферична		
Еліптична		

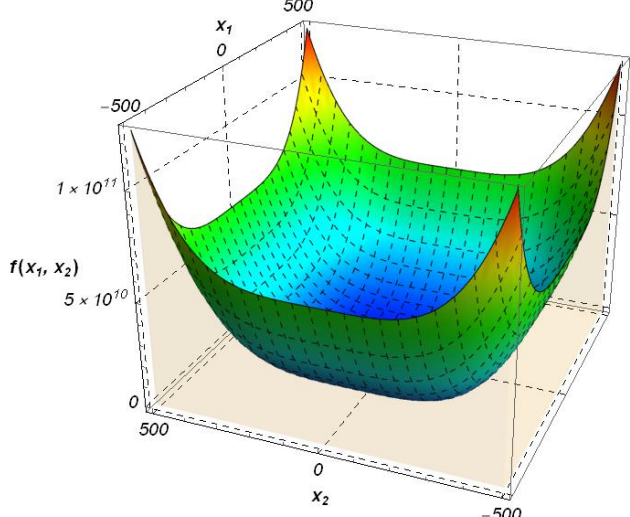
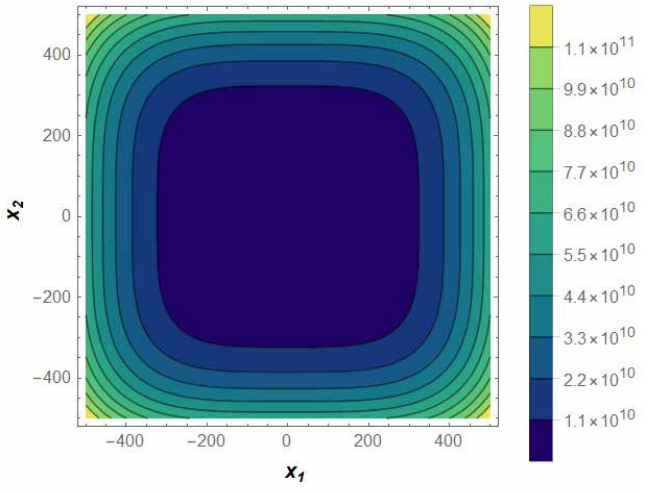
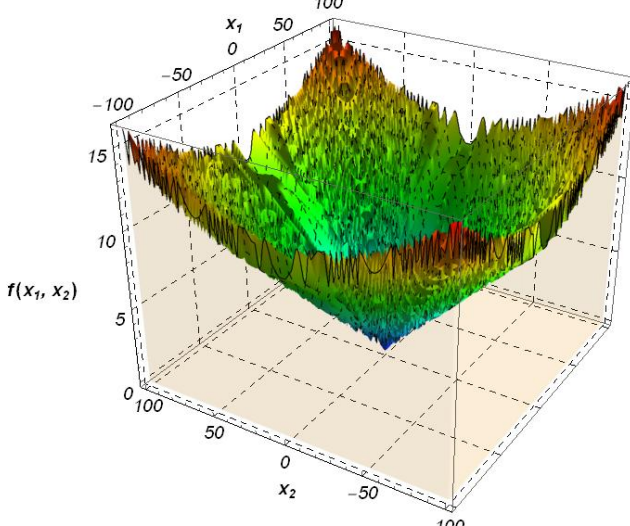
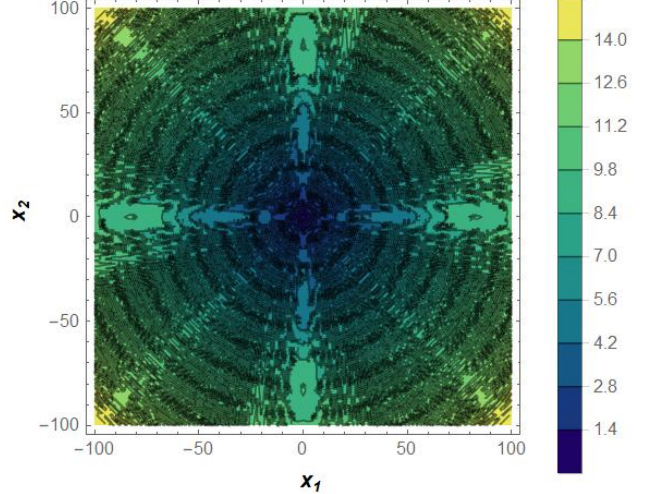
1	2	3
<p>Експоненціальна</p>	<p>3</p> 	
<p>Швефеля 1.2</p>		

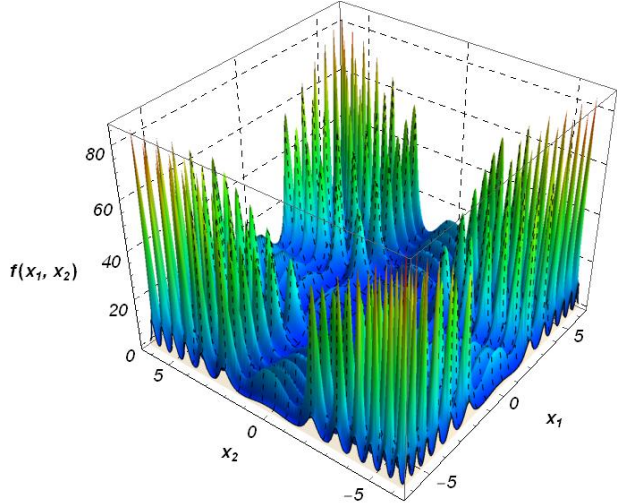
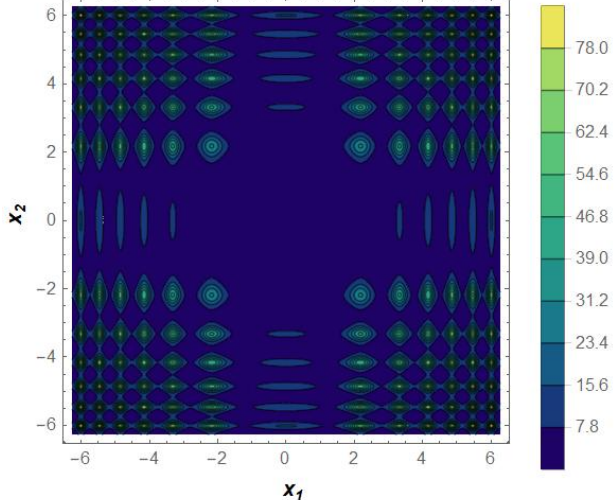
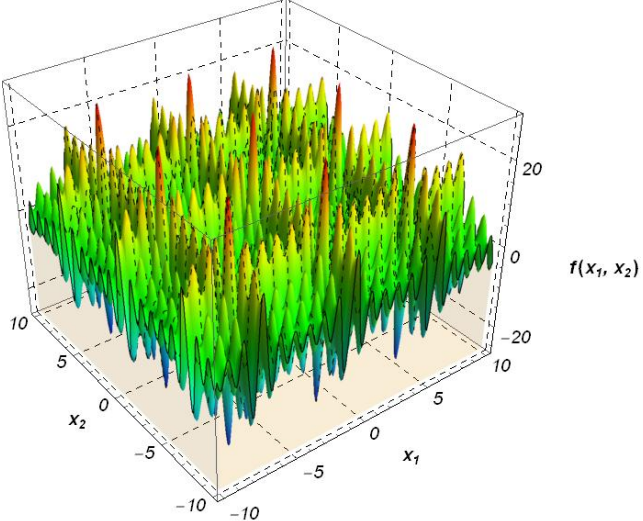
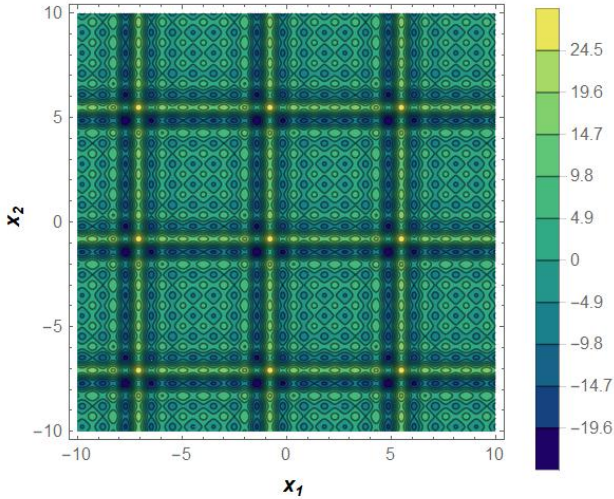
1	2	3
Розенброка	 <p>A 3D surface plot of the Rosenbrock function $f(x_1, x_2)$. The horizontal axes are x_1 and x_2, both ranging from -10 to 10. The vertical axis represents the function value, ranging from 0 to 1×10^6. The surface shows a deep, narrow valley that is difficult to navigate, characteristic of this optimization problem.</p>	 <p>A contour plot of the Rosenbrock function. The horizontal axes are x_1 and x_2, both ranging from -10 to 10. The plot shows a narrow, winding valley. A color scale on the right indicates function values, ranging from 110,000 (dark blue) to 1.1×10^6 (yellow).</p>
Растрігіна	 <p>A 3D surface plot of the Rastrigin function $f(x_1, x_2)$. The horizontal axes are x_1 and x_2, both ranging from -5 to 5. The vertical axis represents the function value, ranging from 0 to 80. The surface is highly oscillatory, with many local minima, making it a challenging optimization problem.</p>	 <p>A contour plot of the Rastrigin function. The horizontal axes are x_1 and x_2, both ranging from -4 to 4. The plot shows a highly oscillatory surface with many local minima. A color scale on the right indicates function values, ranging from 7.3 (dark blue) to 73.0 (yellow).</p>

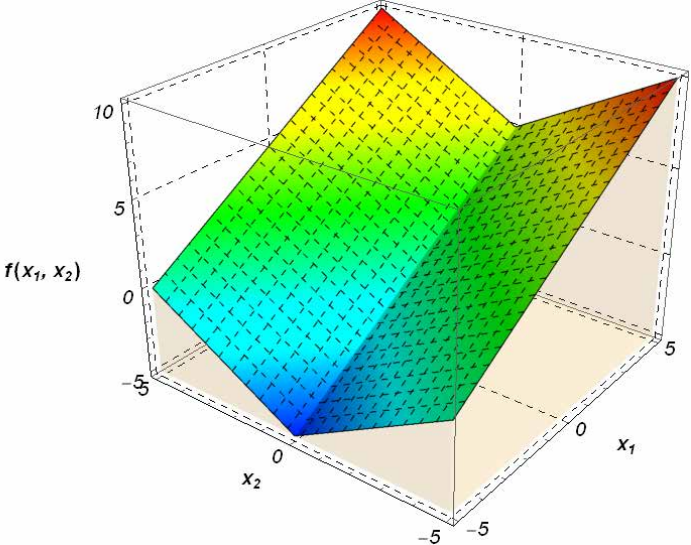
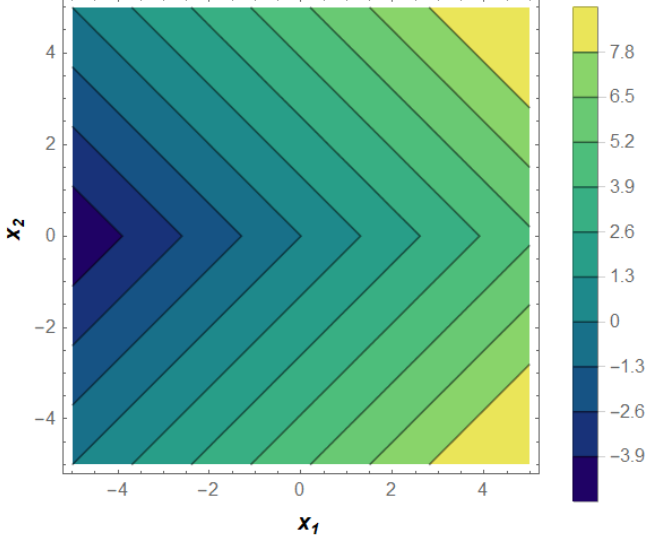
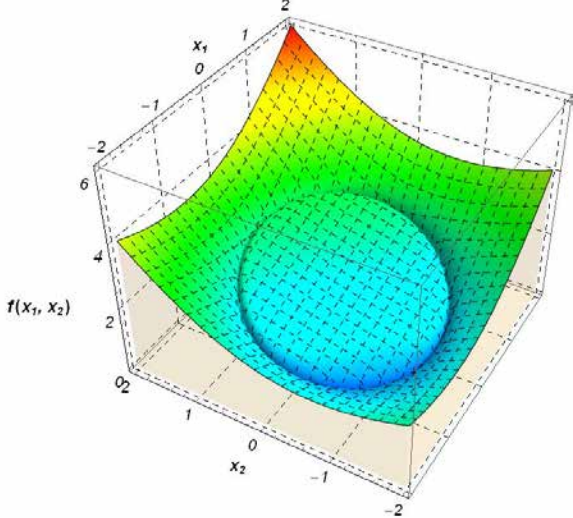
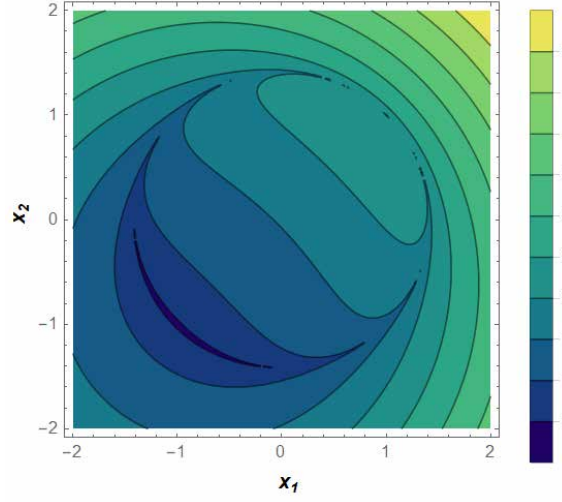
1	2	3
<p>Гріванка</p>		
<p>Альпайна 1</p>		

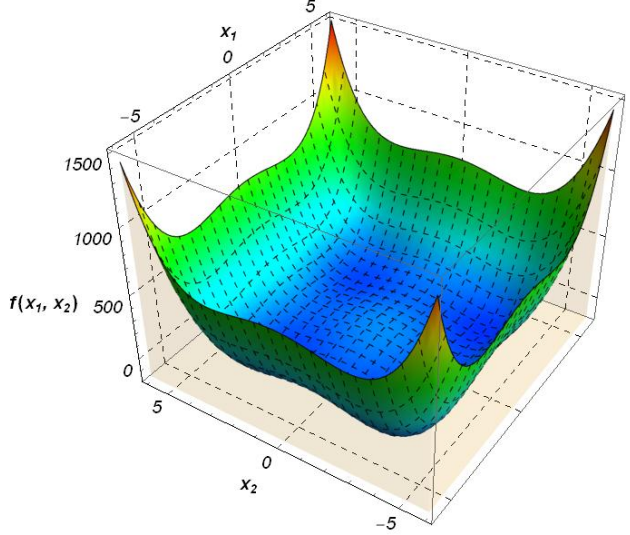
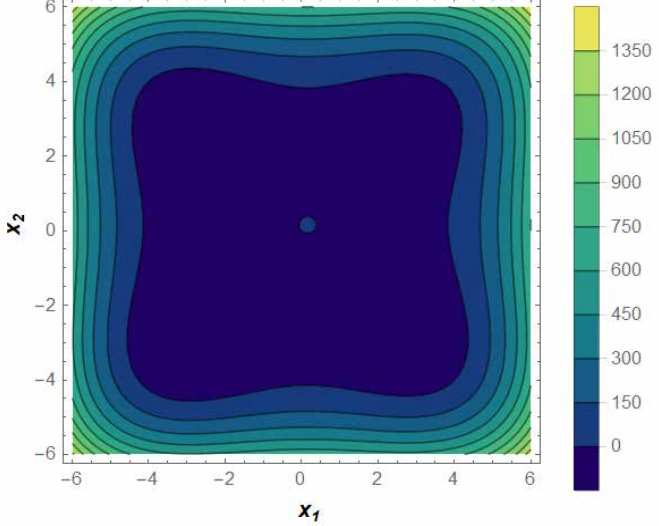
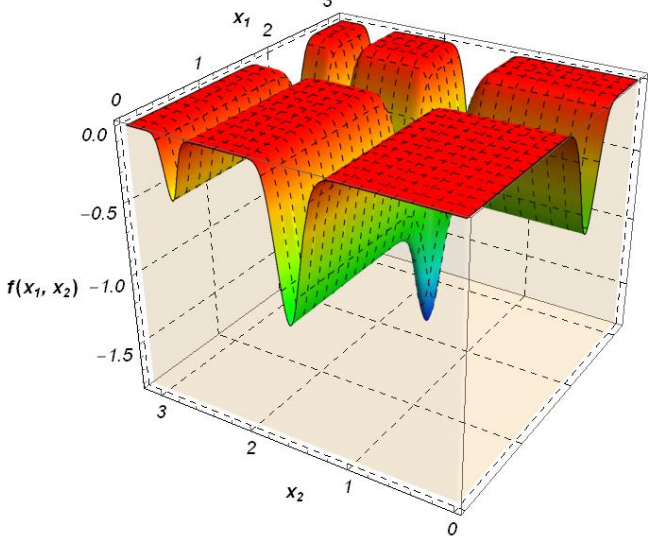
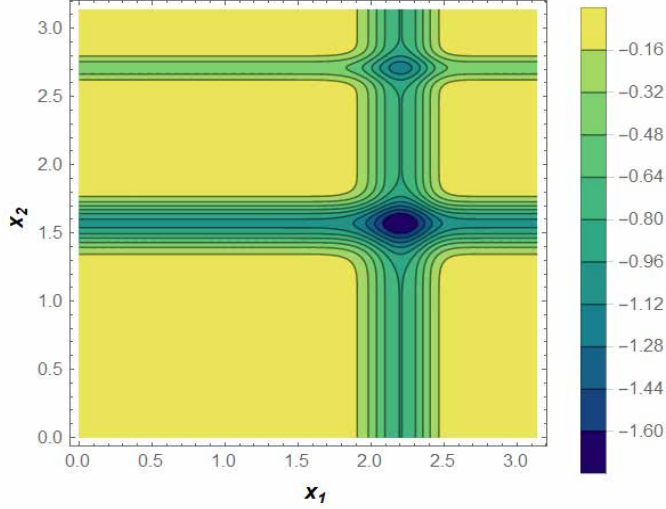
1	2	3
Швефеля 2		
Еклі		

1	2	3
<p>Вейерштрасса</p>	 <p>$f(x_1, x_2)$</p>	
<p>Швефеля 2.20</p>	 <p>$f(x_1, x_2)$</p>	

1	2	3
<p>Цина</p>	<p>3D surface plot of $f(x_1, x_2)$ for Цина. The vertical axis is labeled $f(x_1, x_2)$ with values 0, 5×10^{10}, and 1×10^{11}. The horizontal axes are x_1 and x_2, both ranging from -500 to 500. The surface is a smooth, bowl-shaped paraboloid with a minimum at the center.</p> 	<p>Contour plot of $f(x_1, x_2)$ for Цина. The horizontal axes are x_1 and x_2, both ranging from -400 to 400. The plot shows concentric elliptical contour lines. A color scale on the right indicates values from 1.1×10^{10} (dark blue) to 1.1×10^{11} (yellow).</p> 
<p>Саломона</p>	<p>3D surface plot of $f(x_1, x_2)$ for Саломона. The vertical axis is labeled $f(x_1, x_2)$ with values 0, 5, 10, and 15. The horizontal axes are x_1 and x_2, both ranging from -100 to 100. The surface is highly oscillatory and noisy, with a central peak.</p> 	<p>Contour plot of $f(x_1, x_2)$ for Саломона. The horizontal axes are x_1 and x_2, both ranging from -100 to 100. The plot shows a complex, noisy pattern of contour lines. A color scale on the right indicates values from 1.4 (dark blue) to 14.0 (yellow).</p> 

1	2	3
<p>Сінь-Ше Яна 2</p>	<p>3</p> 	
<p>Шуберта 4</p>		

1	2	3
Ріджа		
Щасливий кіт		

1	2	3
<p>Гіммельблау</p>		
<p>Міхалевича</p>		

Тестова функція f_1 є однією з найпопулярніших і найпростіших, для багатьох алгоритмів оптимізації не виникає труднощів локалізувати її мінімум. Задача її мінімізації може бути пов'язана з точністю до машинного нуля. Градієнтні методи дозволяють успішно визначити точне розташування її глобального мінімуму.

Тестова функція f_2 не надто складна для знаходження мінімуму. Її аргументи мають різний вплив на значення функції: чим більший індекс аргументу i , тим більший вплив. Поверхня функції схожа на каньйон, щодо певної міри ускладнює пошук її мінімуму.

Функція f_3 має один мінімум. Він розташований у початку координат, де функція досягає значення -1 . По осях аргументів функція має глибокі і відносно вузькі каньйони. Таким чином, алгоритм оптимізації може пропускати каньйони і рухатися по плато (верхня частина функції f_3), так і не діставшись до мінімуму.

Тестова функція f_4 має каньйон, який не орієнтований вздовж осей аргументів. Незважаючи на її унімодальність, пошук мінімуму цієї функції є досить складною задачею.

Однією із найважчих для оптимізації є тестова функція Розенброка f_5 . Багато алгоритмів оптимізації здатні знайти один із її локальних мінімумів. Рух до глобального мінімуму дуже повільний. Для більшості випадків не вистачає ітерацій для досягнення глобального мінімуму функції f_5 (принаймні для випадку функції із відносно великою кількістю аргументів).

Тестові функції f_6 та f_7 багато в чому схожі. Вони мають кілька локальних мінімумів. Для випадку f_6 мінімуми рівномірно розподілені на поверхні функції; для випадку f_7 збільшення індексу аргументу i спричиняє збільшення локальних мінімумів по i -й осі. Ці факти пов'язані з тригонометричними функціями, які включені у структуру тестових функцій f_6 і f_7 . Топологічні особливості функцій f_6 і f_7 , що пов'язані з їх мультимодальністю, створюють значні труднощі для локалізації їхніх глобальних мінімумів.

Тестова функція f_8 – це функція з нелінійним оператором модуля. В області її визначення кількість мінімумів дорівнює $7D$ і вони рівномірно розподілені на поверхні функції. Амплітуда коливань функції зростає до меж області її визначення. Функція f_8 має повторювану топологію вздовж кожного з аргументів.

Функція Швевеля 2 (f_9) є однією з найважчих для знаходження її глобального мінімуму. Його розташування, на відміну від більшості тестових функцій, знаходиться поблизу нижньої межі області визначення, далеко від сусідніх добрих локальних мінімумів. Кілька мінімумів функції – ще одна особливість, яка ускладнює пошук серед них глобального.

Функція f_{10} має різко зменшувану (до початку координат) поверхню з кількома мінімумами на ній. Навіть якщо алгоритм знаходить перспективний атрактор (окіл глобального мінімуму), він може потрапити в один із локальних мінімумів, яких дуже багато навколо глобального мінімуму. Згаданий атрактор вузький і оптимізаційний алгоритм може так і не знайти глобальний мінімум функції f_{10} .

Функція Вейерштрасса f_{11} є однією з відомих тестових функцій для оптимізації. Вона має регулярну і, в деякому сенсі, самоповторювану топологію. З цим фактом і з великою кількістю мінімумів пов'язані труднощі знаходження її глобальних мінімумів.

Функція Швевеля 2.20 (f_{12}) є ще однією тестовою функцією, яка використовується для оцінки ефективності алгоритмів оптимізації. Вона неперервна, але недиференційована на всій області визначення. Численні круті каньйони, де схил функції змінює напрямок, ведуть до її мінімуму.

Тестова функція f_{13} має $2D$ глобальні мінімуми, які розташовані на її дні, де вони розподіляються нерегулярним чином. Мінімуми з'єднуються один з одним каньйонами з сідловими точками. Тут алгоритми оптимізації можуть зупинятися або рухатися дуже повільно.

Функція Саломона f_{14} має надзвичайно велику кількість мінімумів. Її мультимодальність зумовлена функцією косинуса з евклідовою нормою

аргументів. Тому функція f_{14} є несепарабельною. Глобальний мінімум оточений багатьма хребтами, які створюють перешкоди для його досягнення.

Тестова функція f_{15} теж несепарабельна. На дні вона покрита численними мінімумами, які з'єднані між собою каньйонами. Ті з них, що знаходяться вздовж осей, є недиференційованими (ефект, що викликаний функцією модуля). Атрактор глобального мінімуму відносно малий, тому алгоритми оптимізації можуть його пропускати.

Функція Шуберта 4 (f_{16}) є несиметричною з кількома мінімумами. Вони розташовані вздовж каньйонів і мають різну глибину. Для випадку $D=2$ глобальний мінімум дорівнює $-25,740858$, він досягається на кількох розв'язках. Локалізація та значення глобальних мінімумів для загальної кількості аргументів є відкритим питанням.

Функція Ріджа (f_{17}) подібна до f_1 . Відмінність пов'язана з першим аргументом, який лінійно входить до функції. Глобальний мінімум функції розташований на нижній межі першого аргументу в поєднанні з середньою точкою області визначень інших аргументів.

Контурний графік тестової функції f_{18} нагадує усміхнену мордочку kota. Знайти її глобальний мінімум досить важко [76]. Причина пов'язана з кільцеподібним похилим каньйоном невеликої ширини. Алгоритм може пропустити атрактор без будь-якого покращення знайденого розв'язку.

Функція Гіммельблау (f_{19}) характеризується кількома мінімумами, які розташовані далеко один від одного на її дні. Цей фактор є перешкодою для досягнення глобальних мінімумів: алгоритм може сходитися до одного з локальних мінімумів і вже не покидає його атрактор.

Функція Міхалевича f_{20} має $D!$ мінімумів, які розташовані відносно далеко один від одного. Значення глобального мінімуму функції пов'язане із кількістю її аргументів (для випадку $D=50$ воно дорівнює $-49,62483$ [79]). Показник степеню великих дужок впливає на крутизну долин і хребтів [75]. Тому в даному дослідженні його було задано рівним двадцяти, щоб ускладнити пошук мінімуму.

Висновки до третього розділу:

1. У розділі наведено математичні вирази двадцяти відомих синтетичних тестових функцій, які у подальшому будуть використані для оцінки ефективності роботи різноманітних алгоритмів оптимізації – модифікацій PSO. Наведено основні характеристики тестових функцій (значення глобального мінімуму, його локалізація, кількість і розташування локальних мінімумів, їхня віддаленість один від одного та від глобального мінімуму, уні- або мультимодальність, сепарабельність тощо). Для наглядної оцінки топологічних властивостей для кожної тестової функції побудовані графіки функцій (3D та контурні).
2. Вибір саме таких функцій зумовлений двома факторами: їх значною поширеністю (це, у свою чергу, дає підстави для проведення порівняльних оцінок із іншими методами оптимізації, для яких вони вже використані) та різноманітністю їхніх топологічних властивостей. Останній фактор важливий з точки зору прогнозу ефективності застосування алгоритмів для різних оптимізаційних задач, оскільки їхні цільові функції можуть мати такі ж топологічні властивості.
3. Крім того, база тестових функцій, які описані у даному розділі, або її частина, може у подальшому використовуватись для аналогічних цілей при оцінці новостворених оптимізаційних алгоритмів, модифікацій або гібридизацій PSO.

РОЗДІЛ 4

ОБҐРУНТУВАННЯ ЧИСЕЛЬНИХ ЗНАЧЕНЬ ПАРАМЕТРІВ РОЗРОБЛЕНИХ МОДИФІКАЦІЙ PSO

4.1 Методика визначення оптимальних значень параметрів розроблених модифікацій PSO

Мета досліджень, представлених в цьому розділі, полягає у визначенні оптимальних (або принаймні раціональних) значень параметрів модифікацій, які були розроблені у другому розділі. Для цього було використано шість ($f1$, $f4$, $f7$, $f8$, $f15$, $f18$) добре відомих тестових функцій оптимізації (табл. 3.1), які характеризуються різними топологічними властивостями. Вибір таких функцій зумовлений саме різноманітністю їхніх топологій, що дасть змогу отримати, до певної міри, загальні рекомендації стосовно налаштування параметрів алгоритмів.

Суть методики полягає у тому, що чисельне значення того чи іншого параметра алгоритму розглядають як незалежний фактор, який чинить вплив на ефективність роботи алгоритму тобто відгук. Останній позначимо аббревіатурою *NEO* (від фрази „numerical experiments output”). Для візуалізації результатів чисельних експериментів будують діаграму Бокса-Віскера десятичного логарифму *NEO*. Остання показує максимальне, мінімальне, медіанне значення та значення першого та четвертого кварталів у даних, що отримані за 50 незалежних запусків алгоритму при певному значенні незалежного фактору. Підкреслимо, що для всіх випадків запуск алгоритму на виконання (пошук мінімуму тієї чи іншої тестової функції $f1$, $f4$, $f7$, $f8$, $f15$ або $f18$) був незалежним. Це означає випадкову ініціалізацію рою на початку роботи алгоритму.

За аналізом графічних залежностей – діаграм Бокса-Віскера – можна встановити діапазон оптимальних значень того чи іншого параметра модифікації PSO.

Для кожного запуску алгоритму кількість ітерацій була однаковою і рівною 1000. Популяція рою N приймалася рівною розмірності задачі (кількості аргументів для визначення) $N=D$. Для кожної тестової функції було використано варіанти розмірності 10, 30 та 50.

У табл. 4.1 наведено незалежні фактори (параметри модифікацій PSO) та діапазони їх зміни, які використані у даному дослідженні.

Таблиця 4.1 – Незалежні фактори чисельних експериментів та діапазони їх зміни

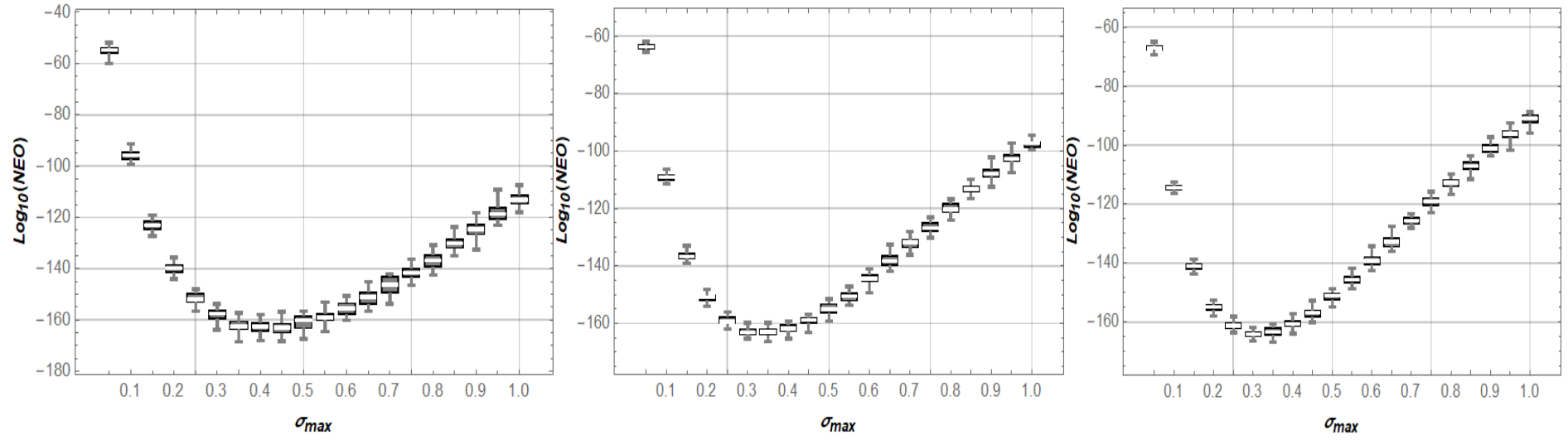
Алгоритм	Параметр	Діапазон зміни	Крок зміни
PV-IA-PSO-1	σ_{\max}	0,05...1,0	0,05
		1,2...5,0	0,2
PV-IA-PSO-2	B	0,2...2,0	0,2
M-PSO-Hybrid	ψ	2...10	1
LG-Ring-PSO	a	0,2...100	0,2; 0,5; 1; 2; 4; 10; 20; 100*

* для алгоритму LG-Ring-PSO крок зміни параметра a є змінним і тому у табл. 4.1 задано значення a , при яких проведено всі чисельні експерименти.

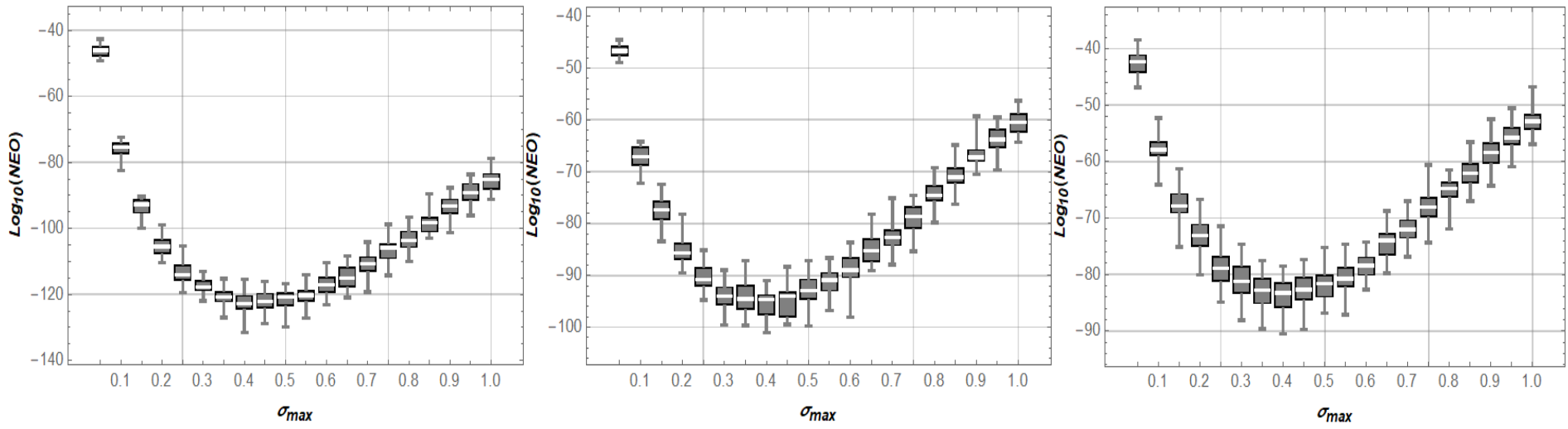
Відмітимо, що друга стрічка у табл. 4.1 відноситься лише до тестових функцій f_{15} та f_{18} .

4.2 Визначення оптимального значення параметра σ_{\max} алгоритму PV-IA-PSO-1

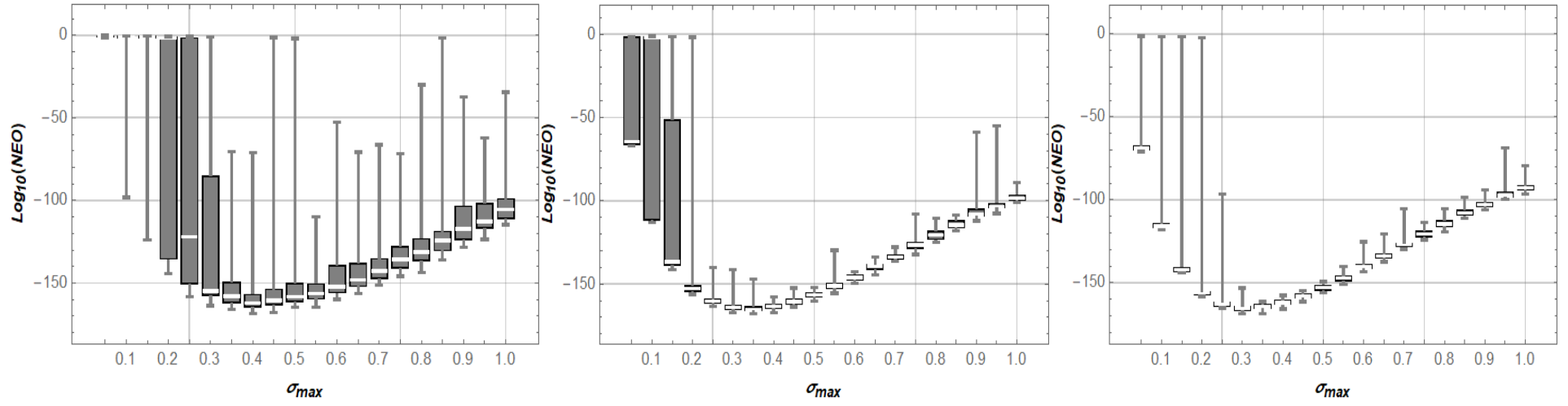
Для того, щоб визначити оптимальне значення σ_{\max} , яке входить у вираз (2.11), було проведено ряд чисельних експериментів. Їхні результати наведено на рис. 4.1: три стовпчики діаграм Бокса-Віскера побудовані для розмірності задач 10, 30 та 50 відповідно. Аналіз графіків, які показані на рис. 4.1 (а, б, в, г), показує, що для $D=50$ оптимальне значення σ_{\max} знаходиться в межах 0,20...0,50. Зсув оптимального діапазону σ_{\max} для інших варіантів ($D=10$ та $D=30$) є незначним.



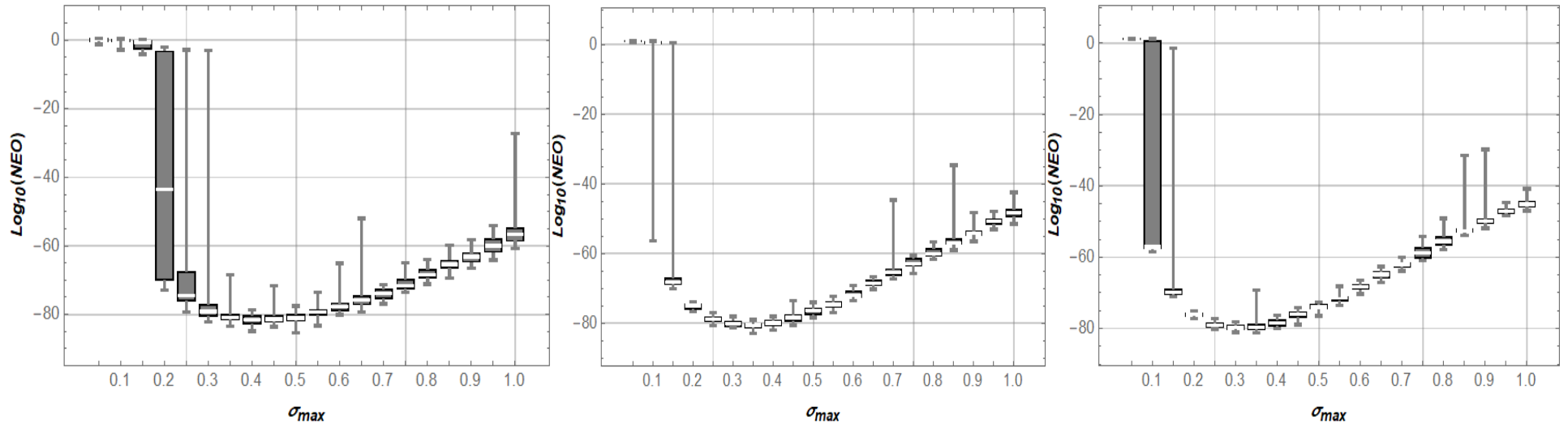
a)



б)



B)



Gamma)

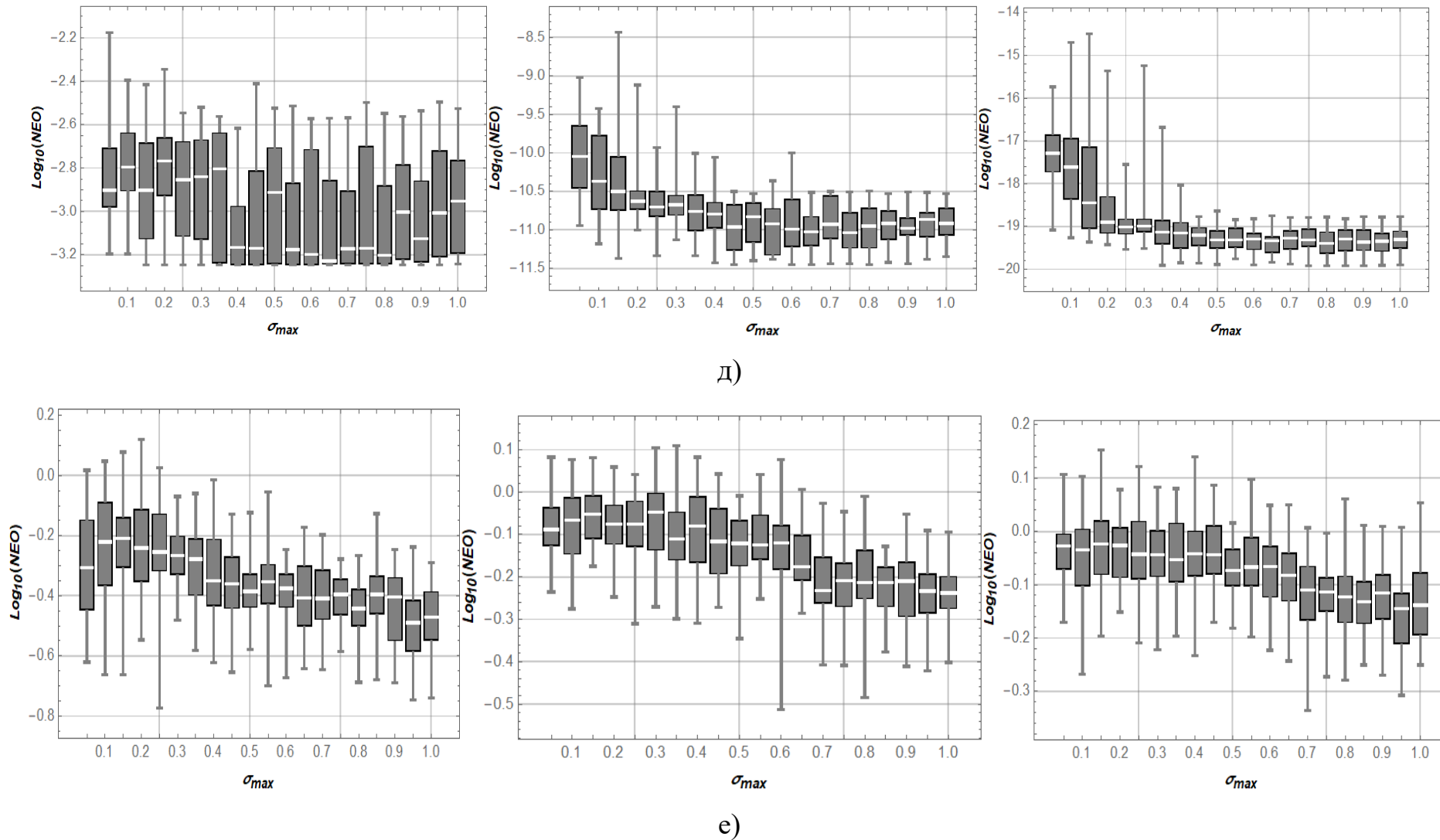


Рисунок 4.1 – Діаграми Бокса-Віскера представлення результатів чисельних експериментів (NEO) по визначенню σ_{\max} , які відповідають тестовим функціям: а) $f1$; б) $f4$; в) $f7$; г) $f8$; д) $f15$; е) $f18$

Для досліджуваних варіантів $D=10\dots50$ існує перекриття діапазонів, яке знаходиться у межах $\sigma_{\max}=0,35\dots0,4$. Проте графіки, які представлені на рис. 4.1. д та 4.1 е, суперечать цьому висновку. Для цих випадків можна спостерігати наступну ситуацію: чим більше значення σ_{\max} , тим краща ефективність роботи алгоритму PV-IA-PSO-1.

Крім того, варто відмітити вплив розмірності рою N та/або розмірності задачі D на ефективність пошуку мінімуму цільових функцій. Для випадків $f1$, $f7$ та $f8$ такий вплив не спостерігається. Для випадків $f4$ та $f18$ збільшення розмірності задачі D призводить до погіршення локалізації мінімуму. Випадок $f15$ показує зворотну залежність: збільшення популяції рою N „перекриває” ефект зростання розмірності оптимізаційної задачі. У цьому випадку можна сподіватись, що подальше зростання N призведе до підвищення ефективності роботи алгоритму PV-IA-PSO-1.

Варто також відмітити і те, що для рекомендованого діапазону $\sigma_{\max}=0,35\dots0,4$ дисперсія отриманих результатів чисельних експериментів для функцій $f1$, $f4$, $f7$ та $f8$ є незначною. Це означає високу повторюваність результатів, що в свою чергу, показує стабільність роботи алгоритму PV-IA-PSO-1.

Підсумовуючи всі наявні дані можемо висунути наступну гіпотезу: лінійний закон спаду σ (2.11) відповідає етапам роботи алгоритму „розвідка - дослідження” лише для деякого класу задач оптимізації. Однак, для ширшого класу оптимізаційних задач, який включає цільові функції $f15$ і $f18$, він демонструє низьку ефективність.

З рис. 4.1, д та е видно, що бажано задавати відносно великі значення σ_{\max} . Тому існує ймовірність того, що при $\sigma_{\max}>1$ алгоритм покаже кращу ефективність. Для того, щоб з'ясувати чи це так насправді були проведені додаткові дослідження, результати яких наведено на рис. 4.2.

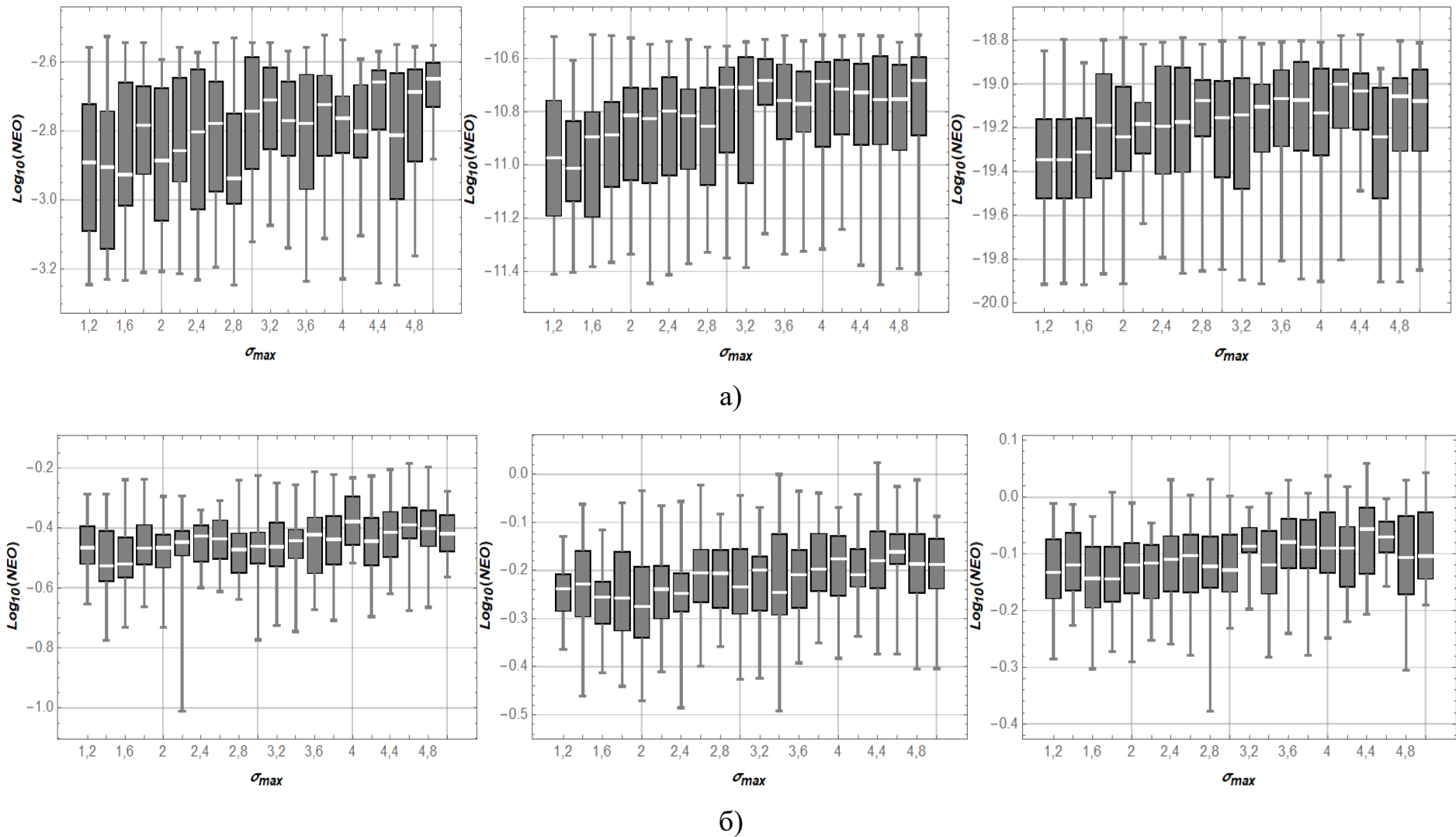


Рисунок 4.2 – Діаграми Бокса-Віскера для представлення результатів чисельних експериментів (NEO) по визначенню σ_{\max} , які відповідають тестовим функціям: а) f_{15} ; б) f_{18}

Порівнюючи графіки, які показані на рис. 4.1, д, е та 4.2, приходимо до висновку, що для тестової функції f_{15} при всіх варіантах її розмірності оптимальний діапазон значень σ_{\max} знаходиться у межах $0,5 \dots 1,6$. Оптимальний діапазон σ_{\max} для функції f_{18} знаходиться у межах $0,8 \dots 2,0$.

Однак, вказані діапазони відповідають лише незначному покращенню ефективності роботи алгоритму PV-IA-PSO-1. Вони вказані через те, що вони не відповідають наведеним вище рекомендаціям $\sigma_{\max}=0,35 \dots 0,4$. Тобто, якщо алгоритм при мінімізації певної цільової функції показує погану ефективність, необхідно відійти від вказаної вище рекомендації стосовно величини σ_{\max} та задати її у діапазоні $0,8 \dots 1,6$.

Згадаємо про теорему про відсутність безкоштовного обіду [80], згідно якої не існує алгоритму, який би добре працював би для кожної можливої задачі оптимізації.

Слід виділити клас оптимізаційних задач, які можна ефективно розв'язувати за допомогою алгоритму PV-IA-PSO-1. Для цього необхідно вказати на деякі спільні властивості тестових функцій f_1, f_4, f_7, f_8 , на яких алгоритм демонструє високу ефективність. Одним із них є сильна унімодальність. Строго кажучи функції f_7 та f_8 не є унімодальними, однак їхні осциляції (у випадку f_7 – дуже слабкі) не справляють істотного впливу на рух рою. Характер руху рою по поверхні цих функцій можна порівняти із рухом по поверхні строго унімодальних функцій, наприклад f_1 або f_4 .

Таким чином, можна рекомендувати застосування алгоритму PV-IA-PSO-1 при налаштуваному параметрі $\sigma_{\max}=0,35 \dots 0,4$ для випадку коли топологія цільової функції є близькою до унімодальної, або коли її мультимодальність не є сильною. Тобто мінімуми функції заходяться близько один від одного, вони не розділені перешкодами, які важно подолати, і перехід часточок від одного до другого мінімуму не представляє значних труднощів.

4.3 Визначення оптимального значення параметра B алгоритму PV-IA-PSO-2

Проведемо визначення оптимального діапазону значень параметра B (2.12) алгоритмом PV-IA-PSO-2. Для цього будемо користуватись методикою, яка описана у п. 4.1 (табл. 4.1.). На рис. 4.3 показано графічну інтерпретацію отриманих результатів чисельних експериментів.

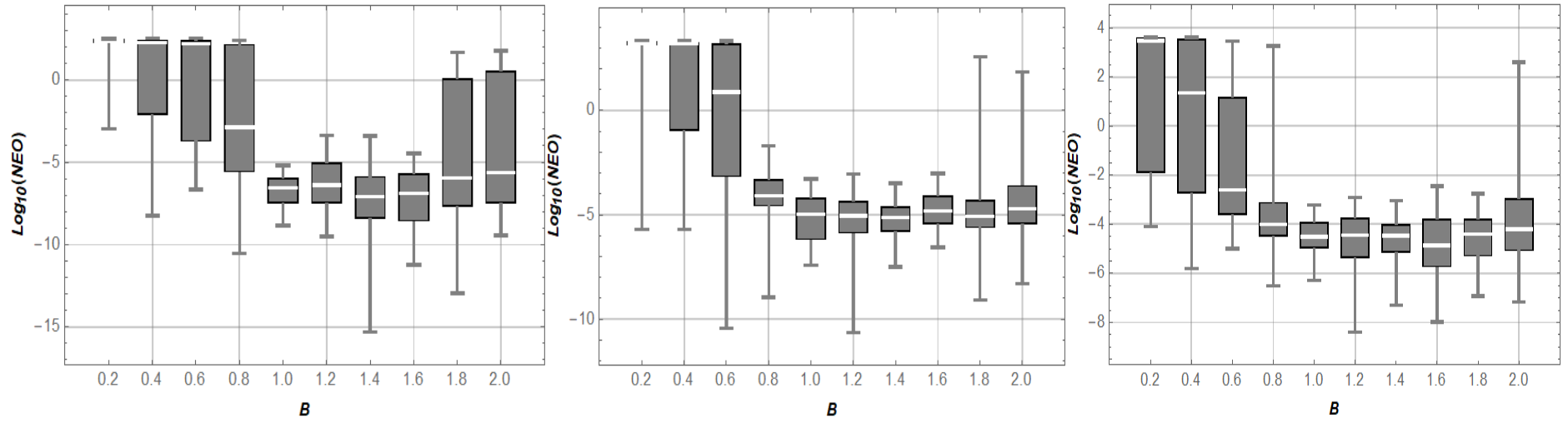
З графіків, які розміщені на рис. 4.3, видно, що, як і у попередньому випадку, тестові функції $f1$, $f4$, $f7$ та $f8$ дають змогу вказати діапазон оптимальних значень параметра B . Для цих функцій діапазон B знаходиться в межах $1 \dots 1,2$. Можемо рекомендувати його для застосування. Цей же діапазон можна рекомендувати і для мінімізації функції $f18$. Однак у цьому випадку параметр B краще задавати у дещо вужчому діапазоні, наприклад $1 \dots 1,1$.

Що стосується функції $f15$, то тут не можна дати однозначної рекомендації, оскільки дані для різної розмірності функцій D є суперечливими. Однак, значення $B=1$ є раціональним для випадків $D=30$ та $D=50$. Для випадку $D=10$ краще задавати параметр B у діапазоні $1,2 \dots 1,4$.

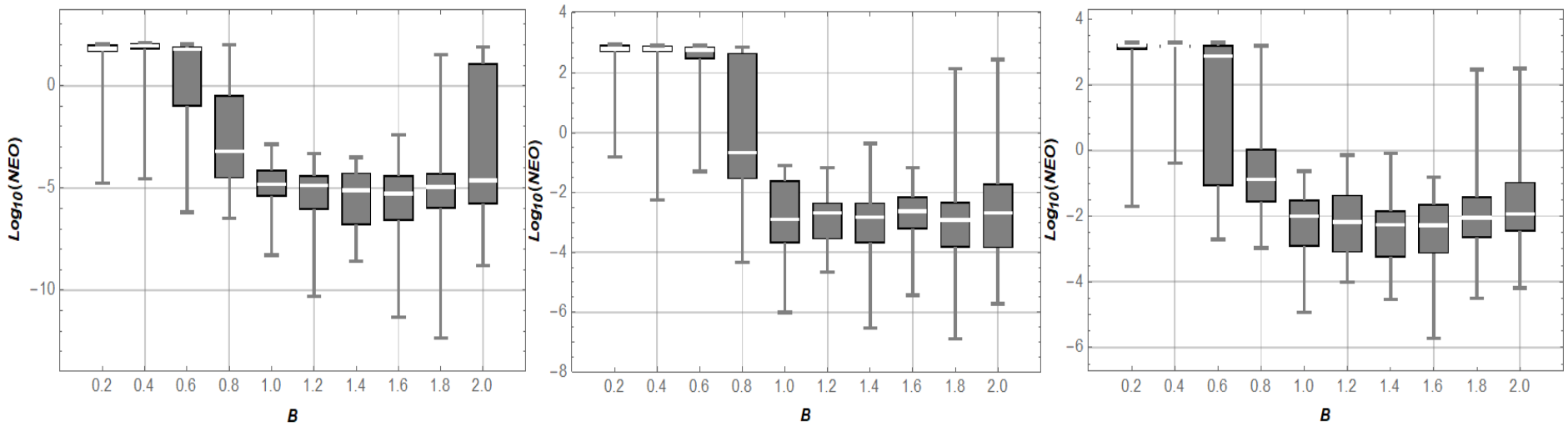
Таким чином, можемо рекомендувати задавати B у діапазоні $1 \dots 1,1$. У випадку поганої ефективності застосування алгоритму PV-IA-PSO-2 при такому налаштуванні варто збільшити значення B до величин $1,2 \dots 1,4$.

Порівнюючи між собою ефективність роботи алгоритмів PV-IA-PSO-1 та PV-IA-PSO-2, можемо відмітити значно кращу (на порядки) роботу алгоритму PV-IA-PSO-1 для випадків тестових функцій $f1$, $f4$, $f7$ та $f8$. Для функцій $f15$ та $f18$ ефективність алгоритму PV-IA-PSO-1 дещо краща, ніж алгоритму PV-IA-PSO-2, але не на порядки.

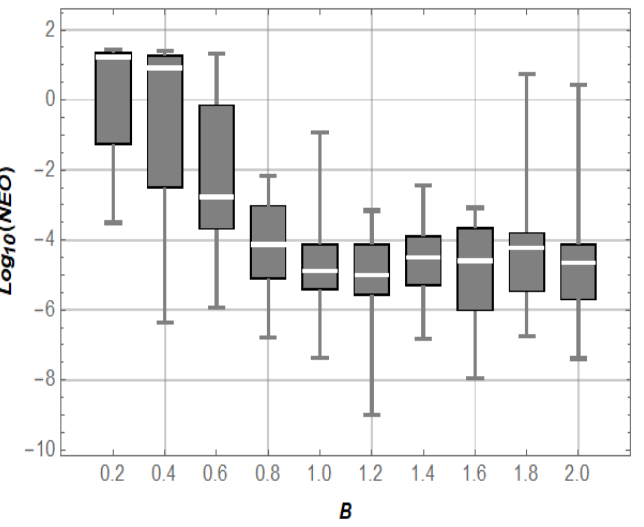
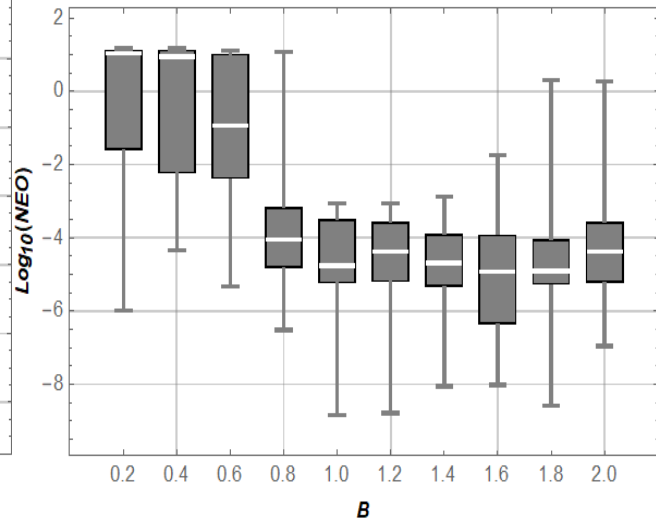
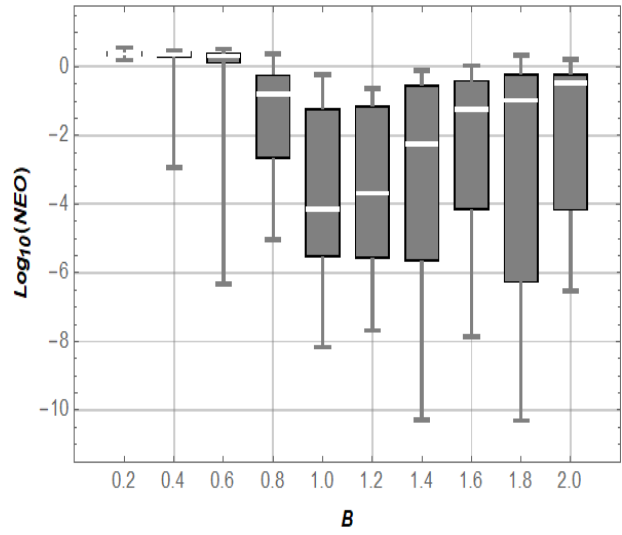
Отже, алгоритм PV-IA-PSO-1 може бути ефективним інструментом для мінімізації різноманітних функцій, у той час як PV-IA-PSO-2 – алгоритм, який не показав високої ефективності на досліджуваному класі функцій.



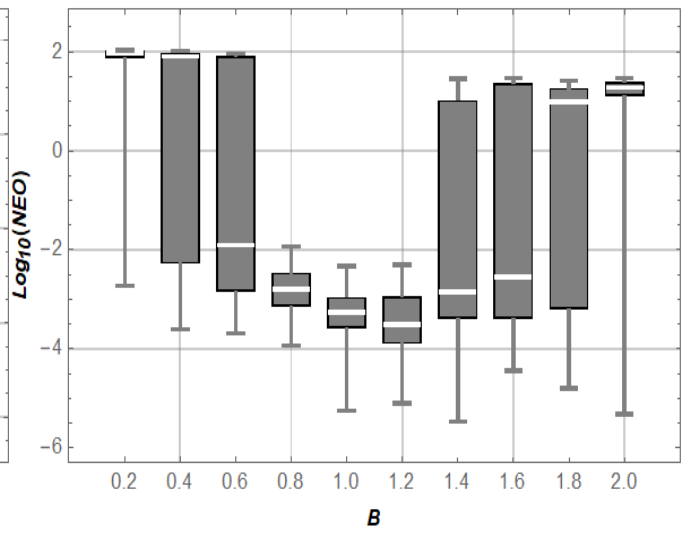
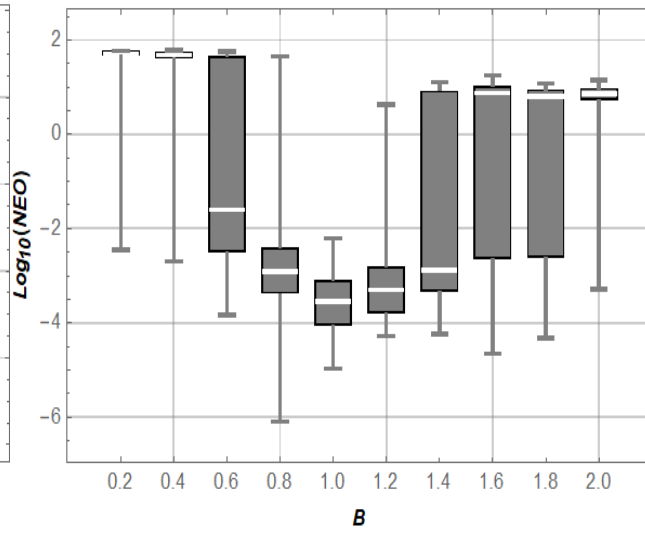
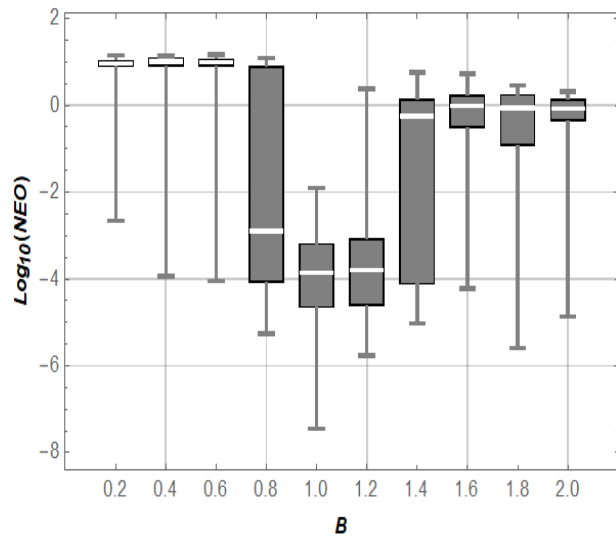
a)



б)



B)



Γ)

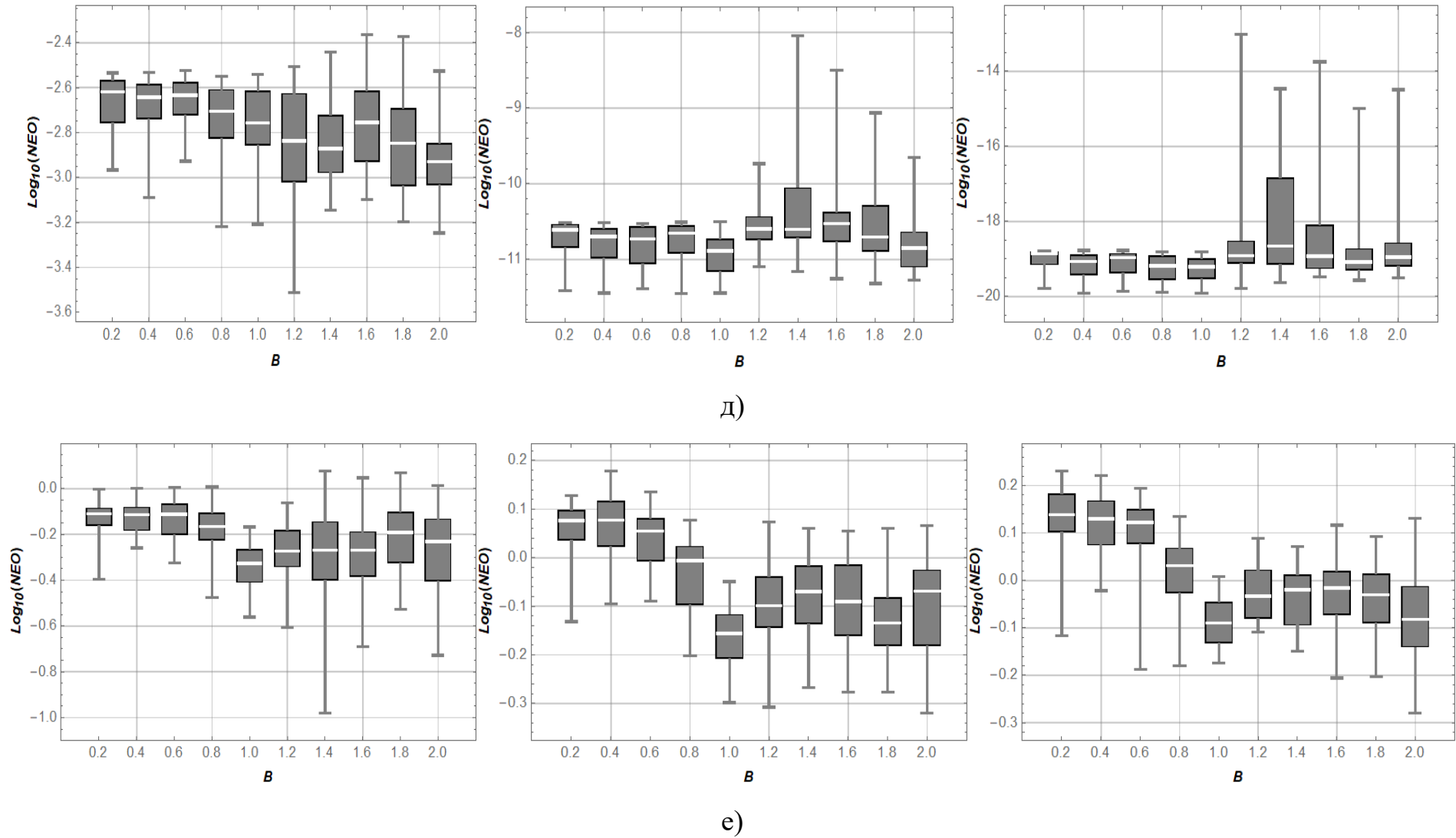


Рисунок 4.3 – Діаграми Бокса-Віскера для представлення результатів чисельних експериментів (NEO) по визначенню B , які відповідають тестовим функціям: а) f_1 ; б) f_4 ; в) f_7 ; г) f_8 ; д) f_{15} ; е) f_{18}

4.4 Визначення оптимального значення параметра ψ алгоритму M-PSO-Hybrid

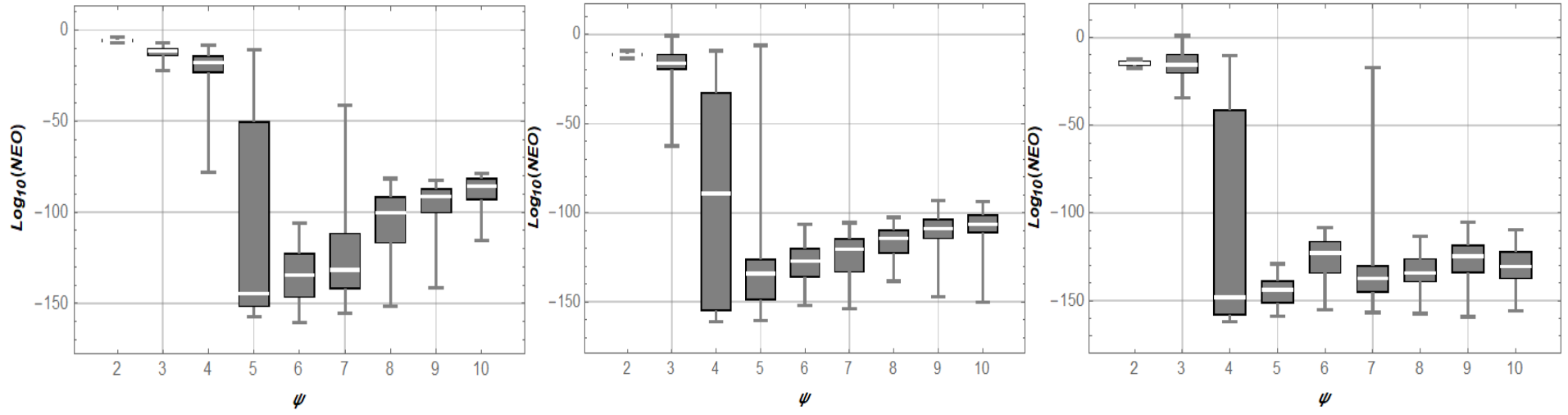
Параметр ψ алгоритму M-PSO-Hybrid визначає кількість ітерацій, протягом яких виконується його окрема складова (один із алгоритмів LDW-PSO, PV-IA-PSO-1, R-Ring-PSO та VCT-PSO). Наперед невідомо оптимальне значення параметра ψ і тому його необхідно визначити. Застосуємо для цього розроблену у п. 4.1 методику. Отримані результати представимо у вигляді графічних залежностей (рис. 4.4).

Аналіз графіків, які показані на рис. 4.4, не дає підстав надати загальні рекомендації стосовно оптимального значення ψ . Однак, можна дати рекомендації, які будуть гарним компромісом, що забезпечує прийнятну ефективність застосування алгоритму M-PSO-Hybrid для певного класу задач оптимізації. Для цього наведемо чисельні дані ψ , які отримані як оптимальні діапазони: на них діаграми Бокса-Віскера набувають мінімуму (табл. 4.2).

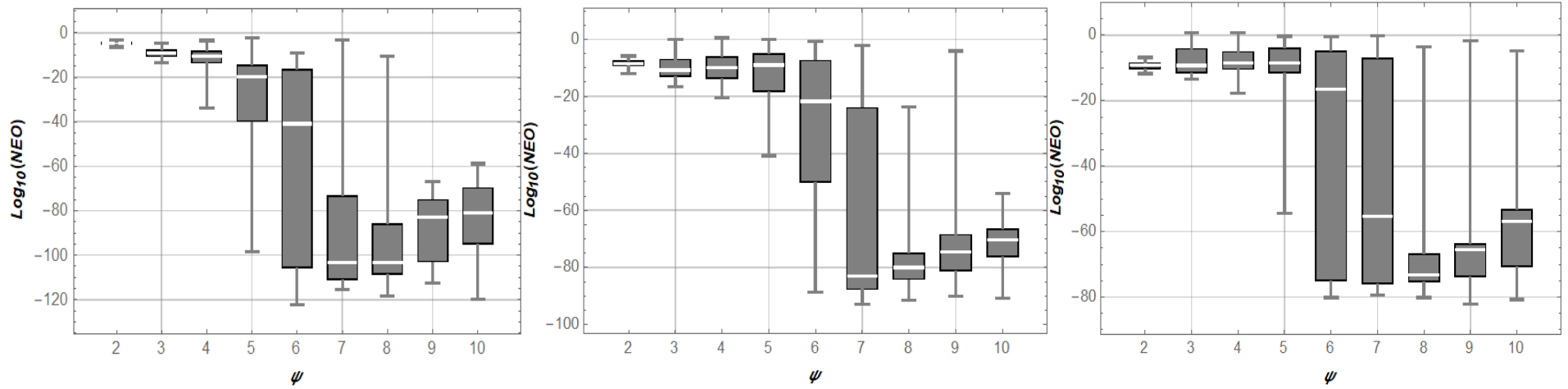
Таблиця 4.2 – Оптимальні діапазони параметра ψ для тестових функцій різної розмірності

Тестова функція	Розмірність функції D		
	10	30	50
$f1$	5...7	5...7	5...7
$f4$	6...9	7...9	7...9
$f7$	5...7	5...7	5...7
$f8$	6...9	5...7	5...7

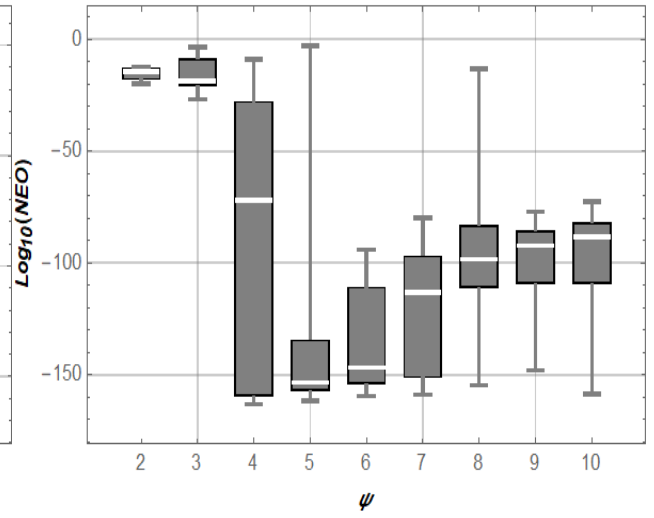
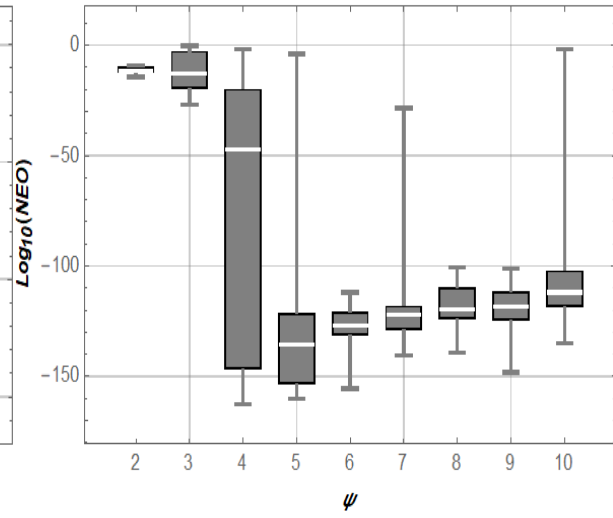
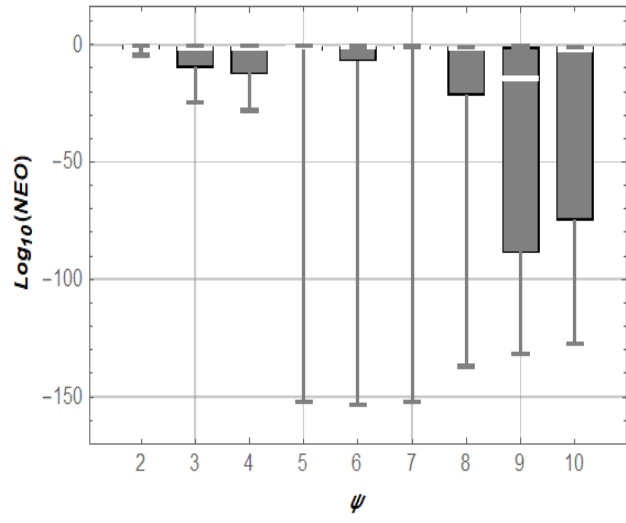
Для функцій $f15$ та $f18$ не отримано стійких результатів впливу параметра ψ на ефективність роботи алгоритму тому вони не наведені у табл. 4.2. Аналіз даних, які наведені у табл. 4.2, показує лише одне число ψ , яке є єдиним раціональним налаштуваннями для тестових функцій $f1, f1, f4, f7, f8$ – $\psi=7$. Саме це чисельне значення параметра ψ можна рекомендувати до застосування.



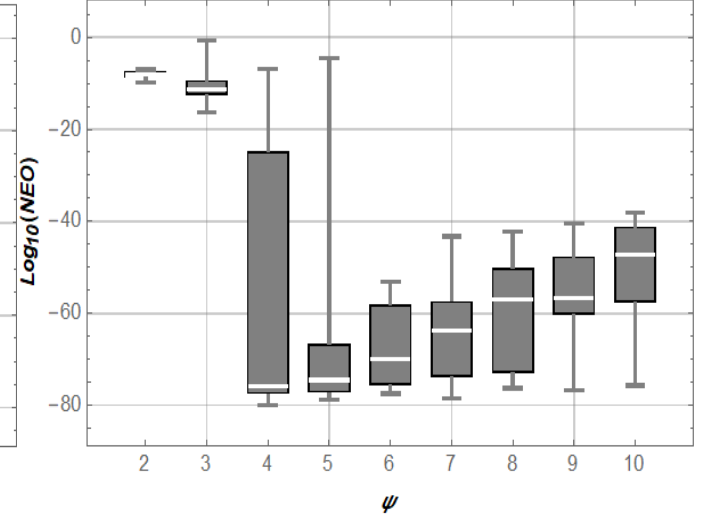
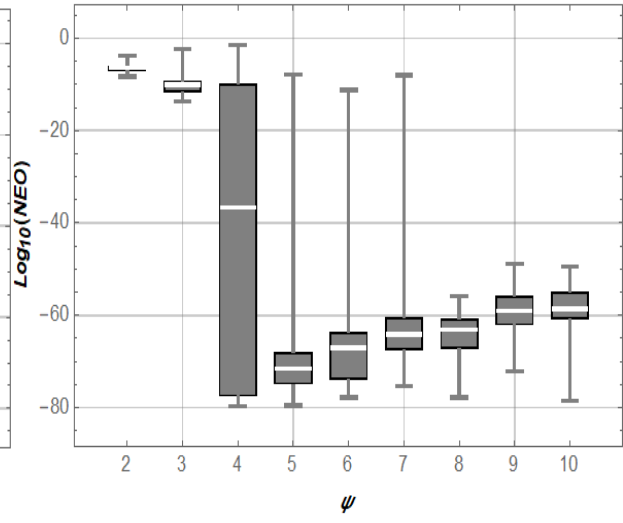
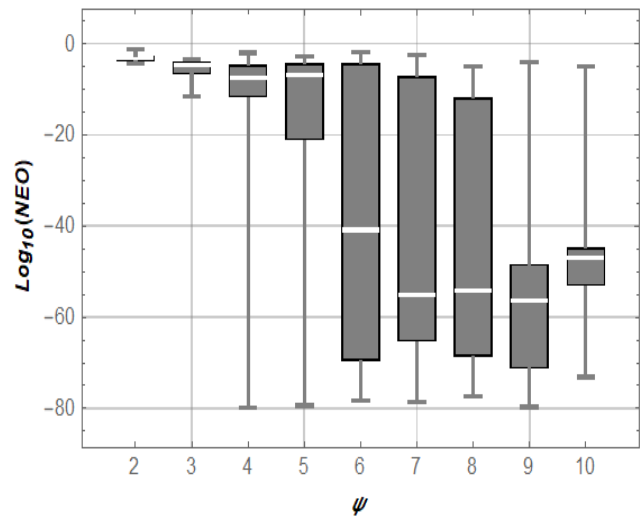
a)



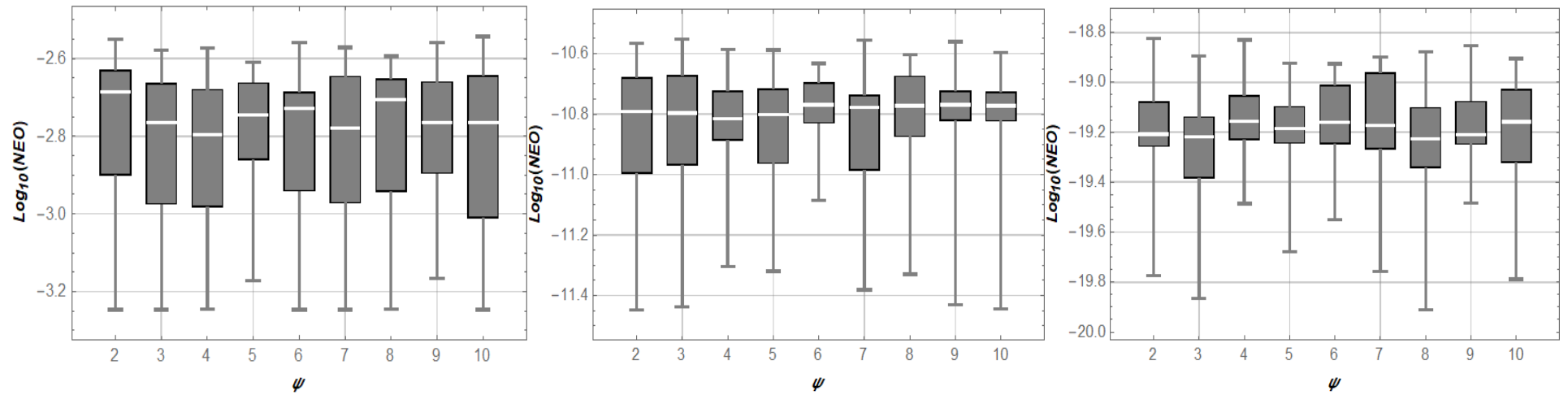
b)



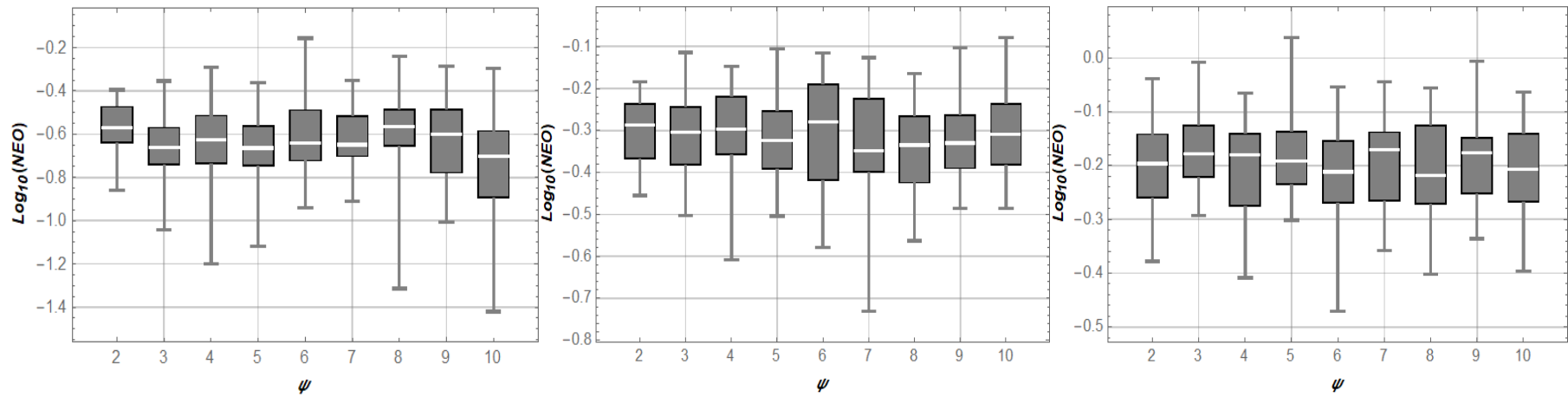
B)



Г)



д)



е)

Рисунок 4.4 – Діаграми Бокса-Віскера для представлення результатів чисельних експериментів (NEO) по визначенню ψ , які відповідають тестовим функціям: а) $f1$; б) $f4$; в) $f7$; г) $f8$; д) $f15$; е) $f18$

Для останніх двох тестових функцій f_{15} та f_{18} чисельний експеримент не виявив оптимального діапазону значень ψ (весь діапазон змін параметра ψ відноситься до майже однакової ефективності роботи алгоритму M-PSO-Hybrid). Можна лише здогадуватися, що воно існує, але умови експерименту, зокрема діапазон зміни ψ , які використані у даному дослідженні, не дає змоги його визначити.

Однак, навіть неоптимальне значення ψ , яке використано для проведення чисельних експериментів, було достатнім для того, щоб підвищити пошукові властивості алгоритму M-PSO-Hybrid у порівнянні із алгоритмом PV-IA-PSO-1 для тестової функції f_{18} . Це можна побачити, провівши порівняння між собою відповідні графічні залежності, які наведено на рис. 4.1, д і 4.4, д. Дійсно, загальний рівень знайдених мінімумів (за всіма чисельними експериментами) за допомогою алгоритму M-PSO-Hybrid є меншим, ніж відповідний показник для алгоритму PV-IA-PSO-1. Очевидно, що алгоритм PV-IA-PSO-1 не достатньо придатний для мінімізації цільової функції f_{18} і його вмикання при роботі алгоритму M-PSO-Hybrid відбувалось не часто.

Натомість частіше працювали інші алгоритми (один з LDW-PSO, R-Ring-PSO або VCT-PSO), які і вплинули на підвищення ефективності роботи M-PSO-Hybrid.

Разом з тим, необхідно відмітити і те, що ефективність роботи M-PSO-Hybrid для тестових функцій f_1, f_4, f_7, f_8 та f_{15} є дещо нижчою, ніж алгоритму PV-IA-PSO-1.

Також з рис. 4.4, д та рис. 4.4, е можна відмітити позитивний вплив збільшення розмірності рою N на ефективність пошукової діяльності алгоритму M-PSO-Hybrid. Вона „переважає” ефект зростання розмірності задачі оптимізації тестових функцій f_{15} та f_{18} .

4.5 Визначення оптимального значення параметра a алгоритму LG-Ring-PSO

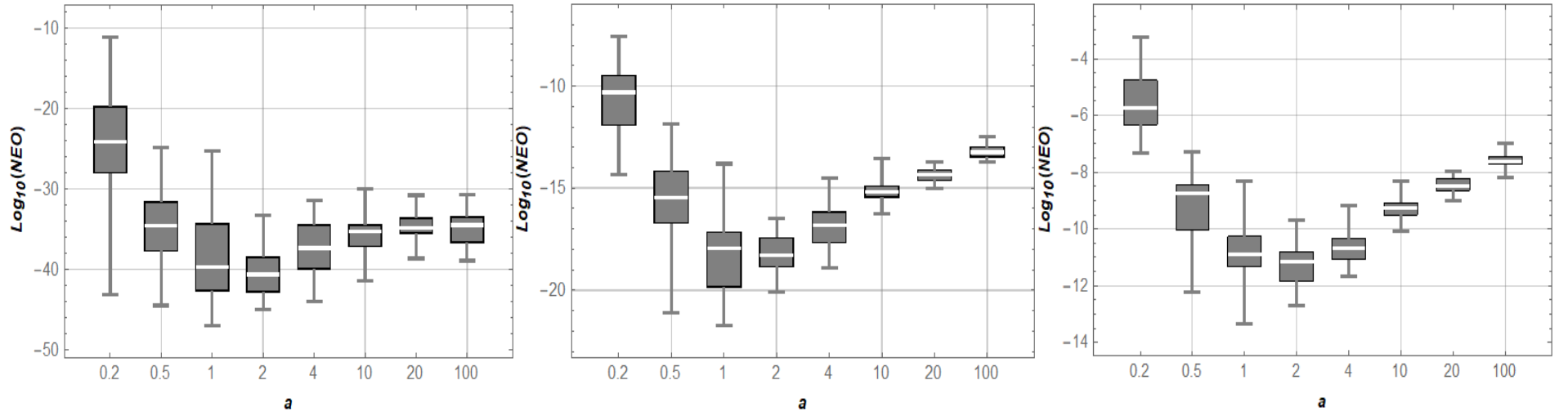
Параметр a алгоритму LG-Ring-PSO є показником степеню у законі, за яким відбувається перехід від локального обміну інформацією між частинками (схема lbest) до глобального (схема gbest). Результати проведення чисельних експериментів, які становлять масив даних для аналізу, наведені на рис. 4.5. З цих графіків видно, що для різних тестових функцій існують різні значення шуканого параметра a . Крім того, розмірність задачі або кількість часточок у рої також має вплив на цю величину. У табл. 4.3 наведено дані для всіх тестових функцій та їхньої розмірності.

Таблиця 4.3 – Оптимальні значення параметра a для тестових функцій різної розмірності

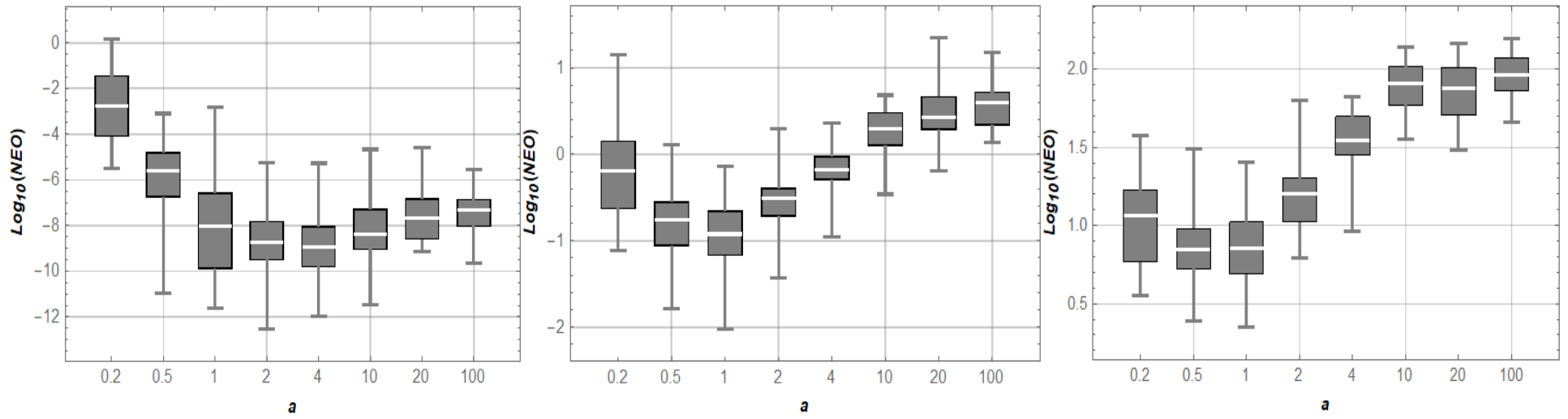
Тестова функція	Розмірність функції D		
	10	30	50
$f1$	2		
$f4$	2...4	1	0,5...1
$f7$	2*	1...2	1...2
$f8$	10...20	0,5...1	0,5...1

* дуже слабкий вплив параметра a на результати роботи алгоритму.

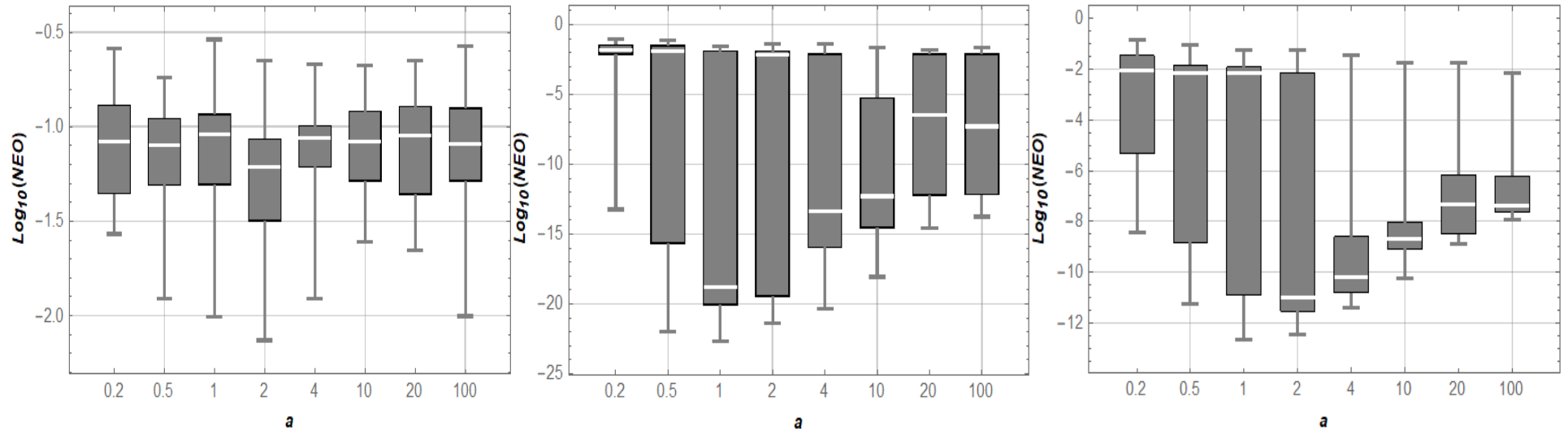
Робота алгоритму на тестових функціях $f15$ та $f18$ не дає змоги рекомендувати оптимальний діапазон зміни налаштування параметра a . Хоча, можна відмітити зростання ефективності роботи алгоритму при збільшенні розмірності рою. Таким чином, за деякими виключеннями, можна рекомендувати налаштувати параметр a у діапазоні 1...2. Необхідно також зазначити, що навіть у вказаному діапазоні незначна зміна величини a може викликати як покращення так і погіршення пошукових властивостей алгоритму LG-Ring-PSO. Тому бажано виконувати декілька незалежних пусків алгоритму з налаштуваннями $a=1$ та $a=2$.



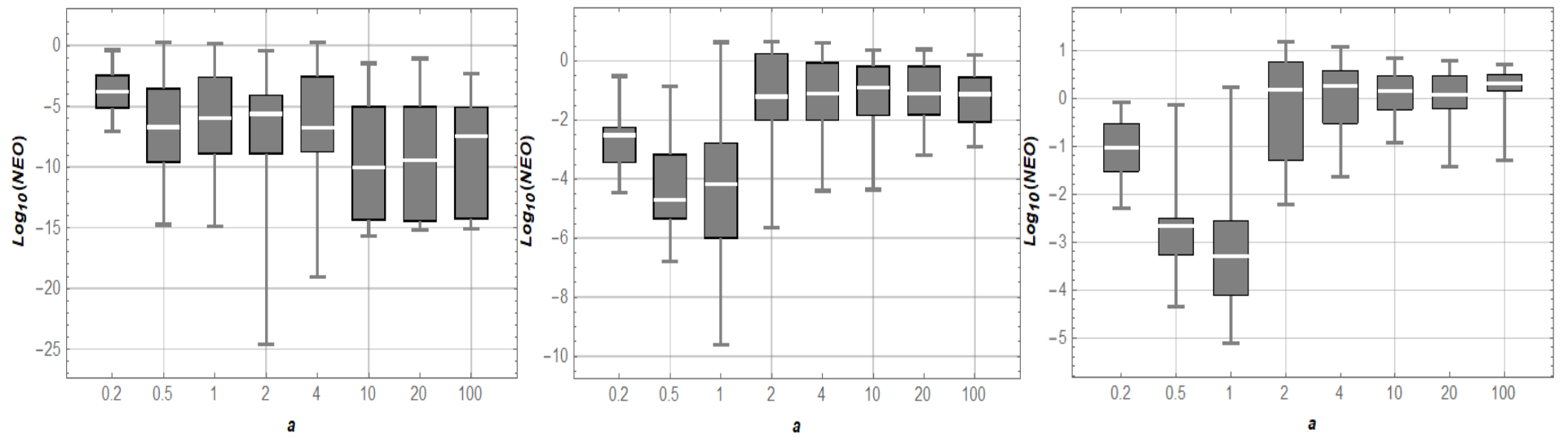
a)



b)



B)



Г)

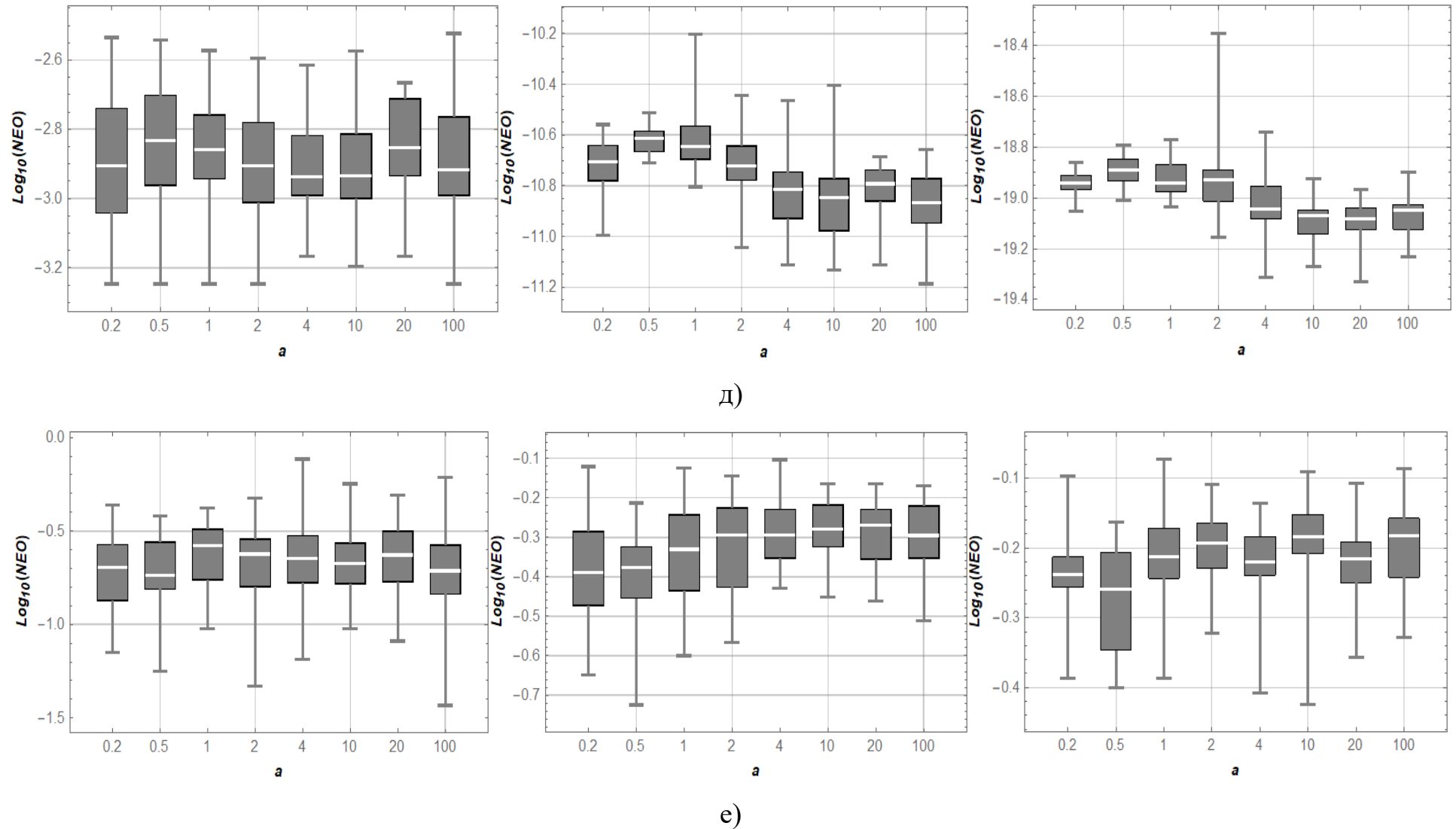


Рисунок 4.5 – Діаграми Бокса-Віскера для представлення результатів чисельних експериментів (NEO) по визначенню a , які відповідають тестовим функціям: а) f_1 ; б) f_4 ; в) f_7 ; г) f_8 ; д) f_{15} ; е) f_{18}

Висновки до четвертого розділу:

1. У розділі розроблено методику визначення оптимальних діапазонів чисельних значень параметрів розроблених раніше модифікацій PSO. Вона ґрунтується на проведенні чисельних експериментів по мінімізації тестових цільових функцій. Незалежним фактором цих експериментів є налаштовуваний параметр алгоритму. Надалі визначаються десяткові логарифми всіх отриманих при проведенні чисельних експериментів результатів і будуються діаграми Бокса-Віскера. Характер їх зміни дає змогу отримати оптимальний діапазон налаштування параметра алгоритму.
2. Користуючись розробленою методикою отримано наступні рекомендації стосовно параметрів модифікацій PSO: для PV-IA-PSO-1 – $\sigma_{\max}=0,35\dots0,4$ ($0,8\dots1,6$ – для сильно мультимодальних цільових функцій); для PV-IA-PSO-2 – $B=1\dots1,1$; для M-PSO-Hybrid – $\psi=7$; для LG-Ring-PSO – $a=1\dots2$. Всі наведені дані валідні для тестових функцій $f1, f4, f7, f8$. Визначення оптимального значення параметра ζ ($\zeta=1$) для алгоритму SR-D-PSO наведено у Додатку В.
3. Тестові функції $f15$ та $f18$ мають досить складну топологію, яка не дала змогу отримати рекомендації стосовно налаштувань параметрів алгоритмів. Відмітимо лише досить сильний вплив на ефективність роботи алгоритмів по мінімізації функції $f15$ розмірності рою.
4. Порівнюючи між собою ефективність модифікацій приходимо до висновку стосовно сильних пошукових властивостей алгоритмів PV-IA-PSO-1 та M-PSO-Hybrid, середніх пошукових властивостей алгоритмів SR-D-PSO та LG-Ring-PSO і слабких алгоритму PV-IA-PSO-2. Для розв'язання оптимізаційних задач рекомендується використовувати алгоритми PV-IA-PSO-1 та M-PSO-Hybrid. У випадку їхньої низької ефективності необхідно використовувати алгоритми SR-D-PSO та LG-Ring-PSO.

РОЗДІЛ 5

РОЗРОБКА ОЦІНОК ЕФЕКТИВНОСТІ РОБОТИ МОДИФІКАЦІЙ PSO

5.1 Класи оцінок роботи алгоритмів

Для того, щоб проводити подальше вивчення поведінки оптимізаційних алгоритмів необхідно мати певні індикатори, які в тому чи іншому сенсі відображають ефективність роботи алгоритмів. У даному розділі такими індикаторами будуть виступати декілька оцінок, які доволі поширені у дослідженнях такого роду.

Роботу алгоритму прийнято оцінювати по тому розв'язку, який він знайшов. Оскільки у всіх алгоритмах, які ґрунтуються на канонічному методі PSO, присутня стохастична складова, то виконують декілька (як правило більше п'ятдесяти) незалежних запусків алгоритму (наприклад, як це було зроблено у розділі 4) і надалі проводять статистичний аналіз отриманих розв'язувань задачі. При цьому залучають показники: мінімальну, максимальну, середньоквадратичну та медіанну величини, середньоквадратичне відхилення. Такі базові індикатори дають змогу зорієнтувати дослідника щодо середньої ефективності роботи досліджуваного алгоритму, виявити його робастність (чутливість до стохастичних факторів, що спричинені факторами r_1 та r_2), оцінити перспективи застосування (найкращий розв'язок задачі серед отриманих результатів) тощо.

Розв'язання будь-якої задачі пов'язане з компромісом, який можна сформулювати у вигляді запитання: „Яку кількість обчислювальних ресурсів дослідник може витратити для розв'язання задачі?” Під обчислювальними ресурсами, як правило, розуміють кількість доступів до цільової функції, кількість ітерацій алгоритму, тривалість виконання програмного коду, який реалізує алгоритм тощо. При інших рівних умовах саме останній показник буде визначати практичність застосування алгоритму. Дійсно, важко уявити ситуацію, коли дослідник чекатиме довгі тижні і навіть місяці поки бажаний

розв'язок задачі буде отриманий. Натомість краще отримати можливо і не глобальний мінімум функції, витративши при цьому небагато часу. Звісно все залежить від важливості задачі. Однак, тривалість розв'язання задачі у багатьох випадках вартує не менше, ніж знайдений розв'язок. Тому будемо також оцінювати роботу алгоритмів за показником тривалості виконання програмного коду.

Не применшуючи роль вказаних вище індикаторів, зазначимо, що вони не дають у повному обсязі провести оцінку роботи алгоритму, оскільки лише фіксують остаточний результат його роботи. Поза увагою дослідника при цьому залишається динаміка і патерни (схеми, шаблони) руху рою, його різноманітність та інші фактори. Тому у даному розділі будуть розроблені інструменти, які дадуть змогу оцінити ці фактори. Підкреслимо, оцінки, які несуть нові вигоди, дають можливість більш глибоко вивчити роботу алгоритму, помічати ті фактори і явища, які були раніше не явними. Саме тому питання розробки цих оцінок є важливим. Відзначимо також і те, що їх, до певної міри, можна застосовувати до інших метаевристичних (зокрема, агентно-орієнтованих) методів оптимізації.

Одразу можна вказати загальні класи, до яких відносяться оцінки, що будуть розроблені. Це динамічні та статичні класи. Перші показують зміну індикаторів під час виконання алгоритму, а другі – фіксують результат його виконання. У таблиці 5.1 показано, які саме оцінки будуть розроблені та класи, до яких вони відносяться.

Таблиця 5.1 – Класи оцінок, які будуть розроблені у дослідженні

Назва оцінки	Клас оцінки	
	Статичний	Динамічний
1	2	3
Відносний успіх алгоритму	●	
Відносна кількість успішних запусків алгоритму	●	
Залученість частинок у пошук розв'язку задачі	●	●

1	2	3
Зміна базових показників рою* за ітераціями		●
Рух частинок на контурній поверхні функції		●

* до базових показників віднесемо максимальне, мінімальне значення, глобальне найкраще, медіану, 0,25- та 0,75-квантили.

Надалі детально опишемо вказані у табл. 5.1 оцінки.

5.2 Клас статичних оцінок

Одним із показників, за яким часто оцінюють роботу того чи іншого алгоритму оптимізації є сума рангів, які він отримує при мінімізації певного класу цільових функцій. Звичайно таке ранжування проходить на основі залученості у „змагання з оптимізацією” інших алгоритмів. У зв'язку з тим, що дану оцінку досить легко розрахувати і вона наглядно демонструє ефективність алгоритму у порівнянні з іншими оптимізаційними алгоритмами, тому її також будемо застосовувати у дослідженні. Ранжування, як правило, проводять за середньою або медіанною величинами (рис. 5.1). Таке обмеження базовими показниками, які використані для розрахунку рангів, має за мету відобразити середню ефективність роботи алгоритму.

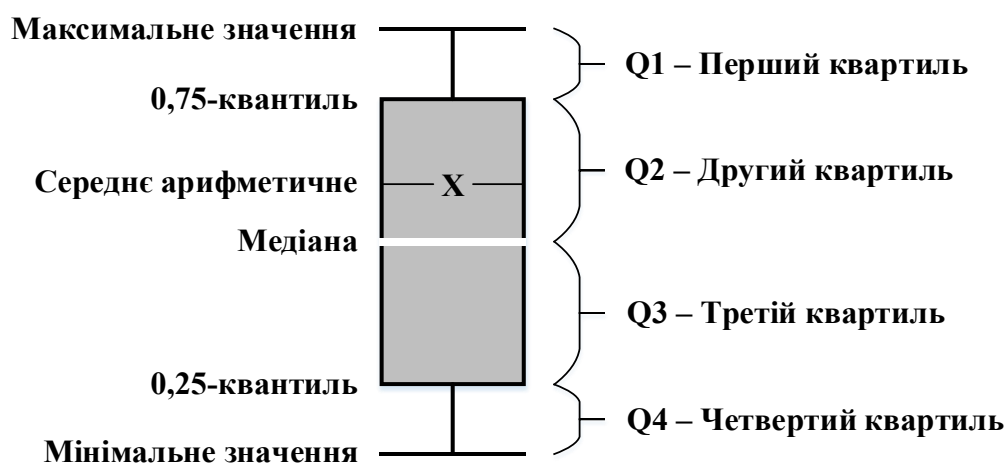


Рисунок 5.1 – Умовне позначення статистичних показників

Однак, якщо дослідник хоче провести оцінку по максимально можливій (і досягнутій під час проведення чисельних експериментів) ефективності роботи алгоритму, то необхідно застосовувати мінімальне значення цільової функції із масиву отриманих розв'язків задачі.

Враховуючи стохастичний характер роботи алгоритмів, такий розв'язок може бути і не знайденим під час повторних пусків алгоритму. Тому бажано пом'якшити умови щодо оцінки і перейти від мінімального значення цільової функції, отриманого за певну кількість незалежних пусків (наприклад, п'ятдесяти), до 0,25-квантиля у масиві отриманих розв'язків. Така оцінка має перевагу над попередньою, оскільки статистично кожен четвертий отриманий розв'язок задачі буде кращим за цю оцінку. Отже, вона не буде „білою вороною” – чимось таким, чого важко досягнути і що з'являється лише за випадковим збігом обставин. Останній фактор виникає завдяки стохастичності роботи алгоритму.

Описані показники також будуть застосовані для оцінки роботи алгоритмів. Вони не є оригінальними розробками авторів. Інші оцінки, які описані нижче, представляють результати напрацювань авторів даної роботи.

5.2.1 Відносний успіх алгоритму

У роботі [68] розроблено критерій, який дає змогу на основі медіанної величини провести оцінку роботи алгоритму на певному класі цільових функцій. Така оцінка передбачає залученість до „змагання з оптимізаціі” інших алгоритмів і дозволяє порівнювати ефективність їх роботи.

У даному пункті виконаємо узагальнення оцінки. По-перше, у якості вихідних даних для розрахунку оцінки рекомендується застосовувати не лише медіанну величину, а і інші статистичні показники, наприклад, середню, мінімальну величини, або 0,25-квантиль. Як було зазначено вище, мінімальну величину необхідно використовувати з обережністю, оскільки розрахунки, які будуть на ній ґрунтуватись, можуть призводити до занадто оптимістичних

результатів. Натомість 0,25-квантиль або середня величина є гарними кандидатами (як і медіана) для виконання розрахунків оцінки роботи алгоритму.

Перший етап у розрахунку оцінки полягає у побудові матриці логарифмічних оцінок, яка представляється у такому вигляді:

$$Cr = \begin{pmatrix} Cr_{1.1} & \dots & Cr_{1.b} & \dots & Cr_{1.B} \\ \dots & \dots & \dots & \dots & \dots \\ Cr_{a.1} & \dots & Cr_{a.b} & \dots & Cr_{a.B} \\ \dots & \dots & \dots & \dots & \dots \\ Cr_{A.1} & \dots & Cr_{A.b} & \dots & Cr_{A.B} \end{pmatrix}, \quad (5.1)$$

$$Cr_{a.b} = \log_{10} \left(\frac{S(f_{a.b,j=1})}{S(f_{a.b,j=J})} \right),$$

де a та b – індекси, що показують номер алгоритму та номер тестової функції відповідно; A та B – кількість оптимізаційних алгоритмів, які залучені у оцінку („змагання з оптимізаціє”) та кількість тестових функцій, на яких це змагання проводиться; S – умовне позначення деякої статистичної величини, про яку вище йшла мова (наприклад, медіана, і тоді $S=Me$). Чисельник аргументу логарифмічної функції (5.1) показує статистичний показник S , який відповідає значенню функції на початку роботи алгоритму (тобто коли $j=1$). Тобто це статистичний показник, який обчислений при п'ятдесяти незалежних ініціалізаціях рою. Знаменник аргументу логарифмічної функції (5.1) показує кінцевий стан мінімізації функції (при виконанні всіх ітерацій, тобто коли $j=J$).

Підсумовуючи все вище наведене, можемо стверджувати, що матриця (5.1) містить повну інформацію стосовно ефективності роботи кожного алгоритму по кожній функції. Така ефективність є чисельним показником, який показує на скільки порядків алгоритм a зміг мінімізувати функцію b за J ітерацій. Оскільки різні тестові функції мають різні топологічні властивості, то задачі їхньої мінімізації будуть мати різну складність. Тому доцільно встановити і використати у подальшому певні вагові коефіцієнти, які б

враховували цей фактор. Для цього на наступному кроці розрахунків необхідно отримати вектор Q вагових коефіцієнтів:

$$\begin{aligned} Q &= (Q_1, \dots, Q_b, \dots, Q_B)^T, \\ Q_b &= \frac{A}{\|Cr_{a,b}\|_L}, \quad a \in (\overline{1, A}), \quad b = \text{const}, \end{aligned} \quad (5.2)$$

де L – норма вектору, який відповідає b -тому рядку матриці Cr (5.1), причому для всіх рядків матриці Cr він однаковий. У випадку $L=\infty$ ми приходимо до відомого результату, який описаний у роботі [68]. Однак, така оцінка є лише частинним випадком, який показує ефективність того чи іншого алгоритму по відношенню до ефективності найкращого з них, яку він показав при мінімізації певної тестової функції. У випадку $L=1$ або $L=2$ отримуємо Манхеттенську та Евклідову норми вектору Q відповідно. Вони не такі жорсткі, як у випадку $L=\infty$, що дає змогу більш адекватно оцінити ефективність роботи алгоритму.

Надалі необхідно визначити матрицю зважених логарифмічних оцінок \tilde{Cr} , яка подібна до Cr (5.1) і яку можна отримати з Cr шляхом підрахунку добутків:

$$\tilde{Cr}_{a,b} = Cr_{a,b} \cdot Q_b. \quad (5.3)$$

Кожен із добутків (5.3) є елементом матриці зважених логарифмічних оцінок \tilde{Cr} . Останній крок у розрахунку оцінки є визначення суми $\sum_{b=1}^B \tilde{Cr}_{a,b}$, яку отримують для кожного алгоритму. Для цього додаються всі елементи у кожному стовпці матриці \tilde{Cr} . Отримана величина показує відносний успіх a -того алгоритму по мінімізації всіх тестових функцій. Відносність такої оцінки впливає з того, що вона буде більшою для випадків, коли алгоритм зміг успішно мінімізувати складні тестові функції, і меншою у випадках, коли така мінімізація (хоч і успішна) була проведена для не складних для мінімізації тестових функцій.

Приведемо приклад розрахунку. Дані, які для цього використано, отримані у роботі [68]. Тестові функції, які використані у цьому дослідженні, наведені у відповідності до позначень, що введені у другому розділі роботи.

Розраховані оцінки за нормами $L=1$ та $L=2$, що ґрунтуються на медіані ($S=Me$), наведені у табл. 5.2. Як видно з табл. 5.2 алгоритмом-лідером за обома оцінками є IA-PSO (результат лідера показано у комірках сірого кольору), хоча за оцінкою $L=\infty$, яка розрахована у статті [68], найкращим виявився алгоритм ME-D-PSO.

Таблиця 5.2 – Оцінки роботи алгоритмів на класі тестових функцій $f1-f15$

Тестові функції	Алгоритми				
	LDW-PSO	IA-PSO	Ring-PSO	ME-PSO	ME-D-PSO
1	2	3	4	5	6
Норма $L=1$					
$f1$	0,312	2,218	0,762	0,782	0,924
$f5$	0,931	1,082	1,031	0,977	0,980
$f6$	0,956	1,052	1,163	0,608	1,221
$f7$	0,216	2,874	0,718	0,375	0,817
$f8$	0,732	0,363	0,851	1,239	1,814
$f9$	1,200	0,313	0,917	1,033	1,537
$f10$	0,346	0,408	2,273	0,029	1,944
$f11$	0,155	3,775	0,730	0,089	0,251
$f12$	0,197	3,159	0,742	0,167	0,735
$f13$	0,707	0,757	0,965	1,204	1,367
$f14$	0,781	1,128	1,036	0,990	1,065
$f15$	1,057	0,738	1,058	1,072	1,075
Сумарна оцінка $\sum_{b=1}^B \tilde{C}r_{a,b}$	7,592	17,868	12,247	8,564	13,730

1	2	3	4	5	6
Норма $L=2$					
$f1$	0,263	1,866	0,641	0,658	0,778
$f5$	0,930	1,080	1,029	0,976	0,978
$f6$	0,934	1,028	1,137	0,594	1,194
$f7$	0,156	2,071	0,517	0,270	0,589
$f8$	0,656	0,326	0,763	1,111	1,627
$f9$	1,114	0,290	0,851	0,958	1,426
$f10$	0,255	0,300	1,672	0,021	1,431
$f11$	0,090	2,188	0,423	0,051	0,145
$f12$	0,132	2,117	0,497	0,112	0,492
$f13$	0,685	0,734	0,936	1,167	1,325
$f14$	0,776	1,120	1,029	0,983	1,057
$f15$	1,048	0,732	1,049	1,063	1,066
Сумарна оцінка $\sum_{b=1}^B \tilde{C}r_{a,b}$	7,039	13,852	10,546	7,964	12,107

Рекомендовано для застосування цю оцінку, яка ґрунтується на медіанному показнику для норми $L=2$.

5.2.2 Відносна кількість успішних запусків алгоритму

У багатьох задачах оптимізації існують обмеження, які тим чи іншим способом вдається звести до нової цільової функції $f(x)$, наприклад, із використанням штрафних компонентів. Забезпечення обмежень є обов'язковим і задача не може вважатись розв'язаною, якщо хоча б одне з обмежень порушене. Робота алгоритму вважається успішною, якщо вдалось знайти область допустимих значень. Можна привести інший (більш жорсткий)

приклад, де вимагається не лише виконання обмежень, а і покращення відомого розв'язку задачі (зменшення вихідної цільової функції при забезпеченні всіх обмежень).

Обидва ці приклади демонструють підхід, згідно якого робота алгоритму вважається успішною у випадку, коли знайдений розв'язок задачі забезпечує значення цільової функції, яке є не більшим, ніж наперед задане її порогове значення:

$$f_J(x) \leq \Delta, \quad (5.4)$$

де $f(x)$ – значення цільової функції у кінці роботи алгоритму (тобто на J -тій ітерації); Δ – деяке наперед задане порогове значення цільової функції. Це дає підстави оцінити роботу алгоритму за показником відносної кількості його успішних запусків:

$$FAR = \left(\frac{OR}{SR} - 1 \right) 100\%, \quad (5.5)$$

де OR – загальна кількість запусків алгоритму для певної оптимізаційної задачі; SR – кількість успішних запусків алгоритму, для яких справедлива нерівність (5.4). У роботі [81] така оцінка використана для порівняння дванадцяти метаевристичних алгоритмів.

Суть задачі, яка розв'язана у [81], полягала у знаходженні таких моментів перемикавання функції керування рухом крана, при яких маятникові коливання вантажу у кінці розгону були б рівні нулю, а швидкість крана дорівнювала б номінальній. Задачу зведено до мінімізації нелінійної функції. Для мінімізації цієї функції застосовано дванадцять алгоритмів: VCT-PSO [67], LDW-PSO [13], ME-D-PSO [68], DEbest/1/bin [82], DErand/1/bin [31, 83], SADE [84], mGWO [85], GWOEPD [86], RaFA [87], ABHS [88], PAHS [89], ACS [90]. Умовою (5.4), за якою запуск того чи іншого алгоритму вважався успішним, була наступна нерівність $T < 5,02$ с. Вона відповідає умові, при якій тривалість

розгону динамічної маятникової системи типу „кран-вантаж” є не більшою, ніж період вільних коливань вантажу на гнучкому підвісі. Результати оцінки роботи згадуваних вище алгоритмів наведені у табл. 5.3.

Таблиця 5.3 – Оцінка (5.5), яка отримана для різних алгоритмів

Кількість ітерацій алгоритму J	Оцінювані алгоритми								
	VCT-PSO	LDW-PSO	ME-D-PSO	DE best/1/bin	DE rand/1/bin	mGWO	GWOEPD	RaFA	ABHS
40	4,8	4,5	15,0	4,2	*	3,5	37,2	64,2	*
100	4,0	3,5	8,8	3,8	8,0	5,2	53,0	19,2	*
200	4,0	3,8	6,0	3,0	5,0	3,8	*	6,2	77,0
400	3,8	3,0	5,8	4,5	7,5	3,2	69,0	4,8	22,8

* показник FAR більше, ніж 100%. У деяких випадках алгоритм не знаходив усіх необхідних розв’язків, і його наступні запуски були припинені.

Алгоритми CS, PAHS і SADE для задачі оптимізації „провалилися” – вони не показали ефективної роботи і для них оцінка $FAR < 100\%$. Через це вони не були показані в табл.5.3.

Як видно з табл. 5.3 алгоритми на основі PSO, DE та GWO показали сильні пошукові властивості, які практично не залежать (принаймні для даної задачі) від кількості ітерацій.

5.2.3 Клас статично-динамічних оцінок (показник залученості частинок у пошук розв’язку задачі)

Однією із характеристик будь-якого мультиагентного методу оптимізації, до якого належить і PSO, є те, що всі частинки приймають участь у пошуку розв’язку задачі. Однак, не всі вони покращують знайдений роєм розв’язок задачі.

Звісно, наперед не відомо, які частинки будуть успішними у пошуку розв'язку задачі, а які – ні. Однак, для виконання алгоритму необхідно виділяти певні обчислювальні ресурси, які рівномірно розподіляються для пошуку розв'язку задачі між всіма частинками.

Було б корисно оцінити, яка кількість частинок під час виконання алгоритму брала участь у покращенні розв'язку задачі та співвіднести її до всієї популяції рою:

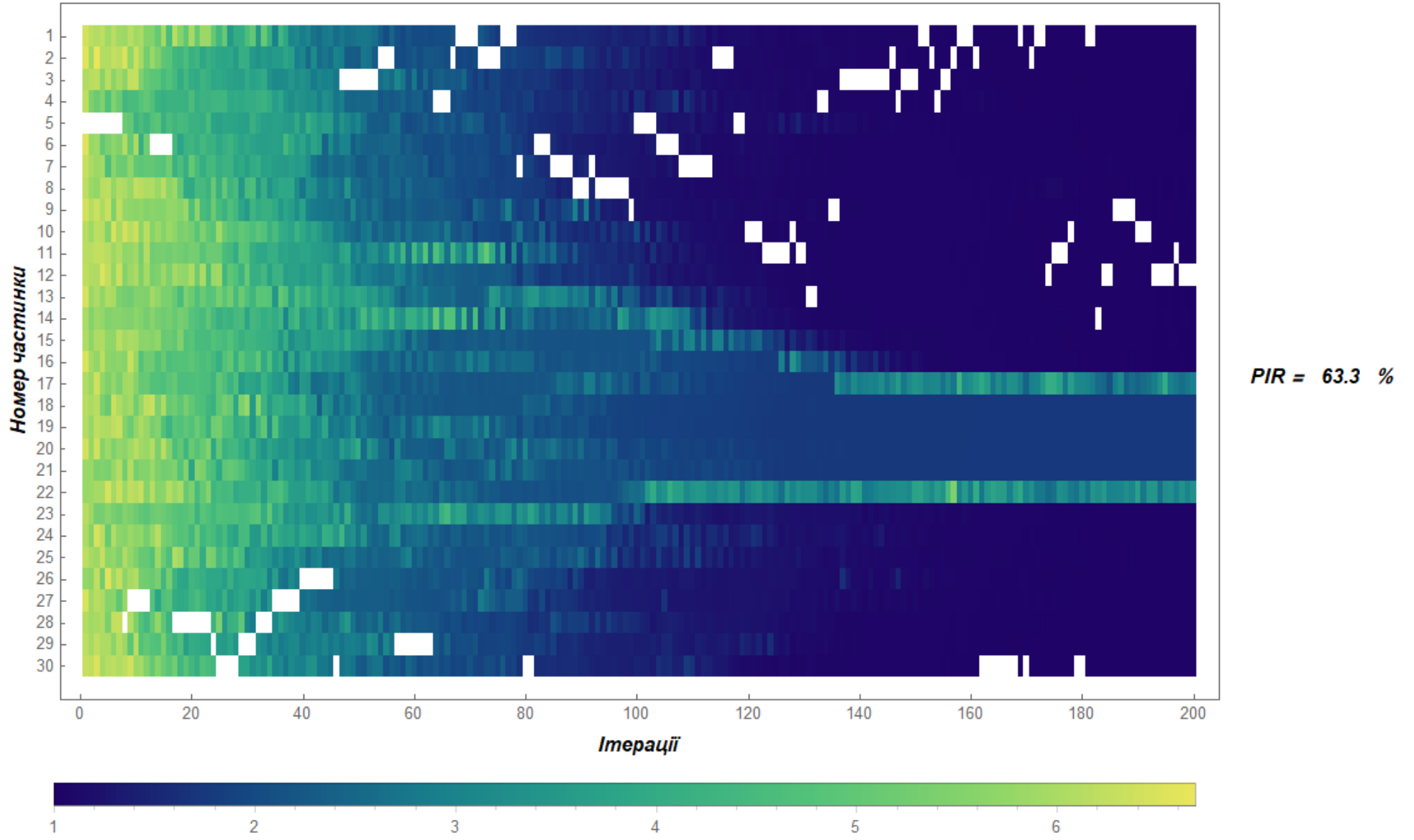
$$PIR = \frac{N_s}{N} 100\%, \quad (5.6)$$

де N_s – кількість частинок у рої, які хоча б на одній ітерації знайшли глобальне найкраще рою g . Аббревіатура PIR відповідає фразі „particles involvement ratio”, яка відповідає назві українською – показник залученості частинок.

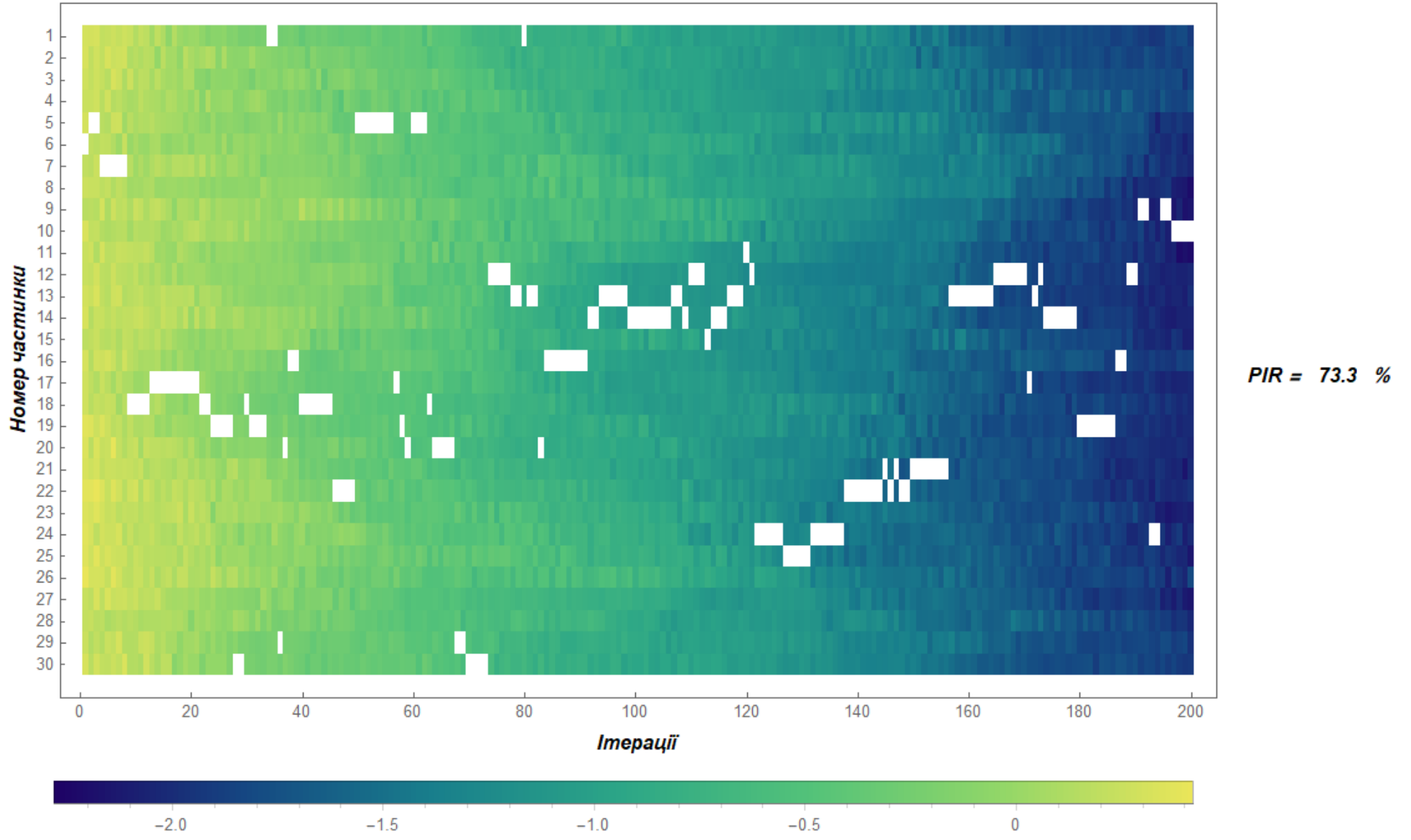
Оцінку (5.6) назвемо показником залученості частинок у пошук розв'язку задачі. Він не дозволяє встановити ефективність роботи алгоритму, однак, показує наскільки ефективно використовуються обчислювальні ресурси для розв'язання задачі. Це так би мовити „обчислювальний коефіцієнт корисної дії алгоритму”.

Показник (5.6) – це число, яке можна було б віднести до класу статичних показників (табл. 5.1). Однак, корисно подавати цей показник разом із певним графічним представленням того, які частинки і на яких ітераціях знаходили глобальне найкраще рою g . Для цього було розроблено інструмент, який дозволяє це зробити (рис. 5.2). Тут для мінімізації обрано тестові функції $f5$, $f11$ та $f15$. Кожна з них має десять незалежних аргументів ($D=10$). Розмірність рою алгоритму PSO-Ring, який оцінюється з використанням даного інструменту, рівна 30 ($N=30$), а кількість ітерацій 200 ($J=200$).

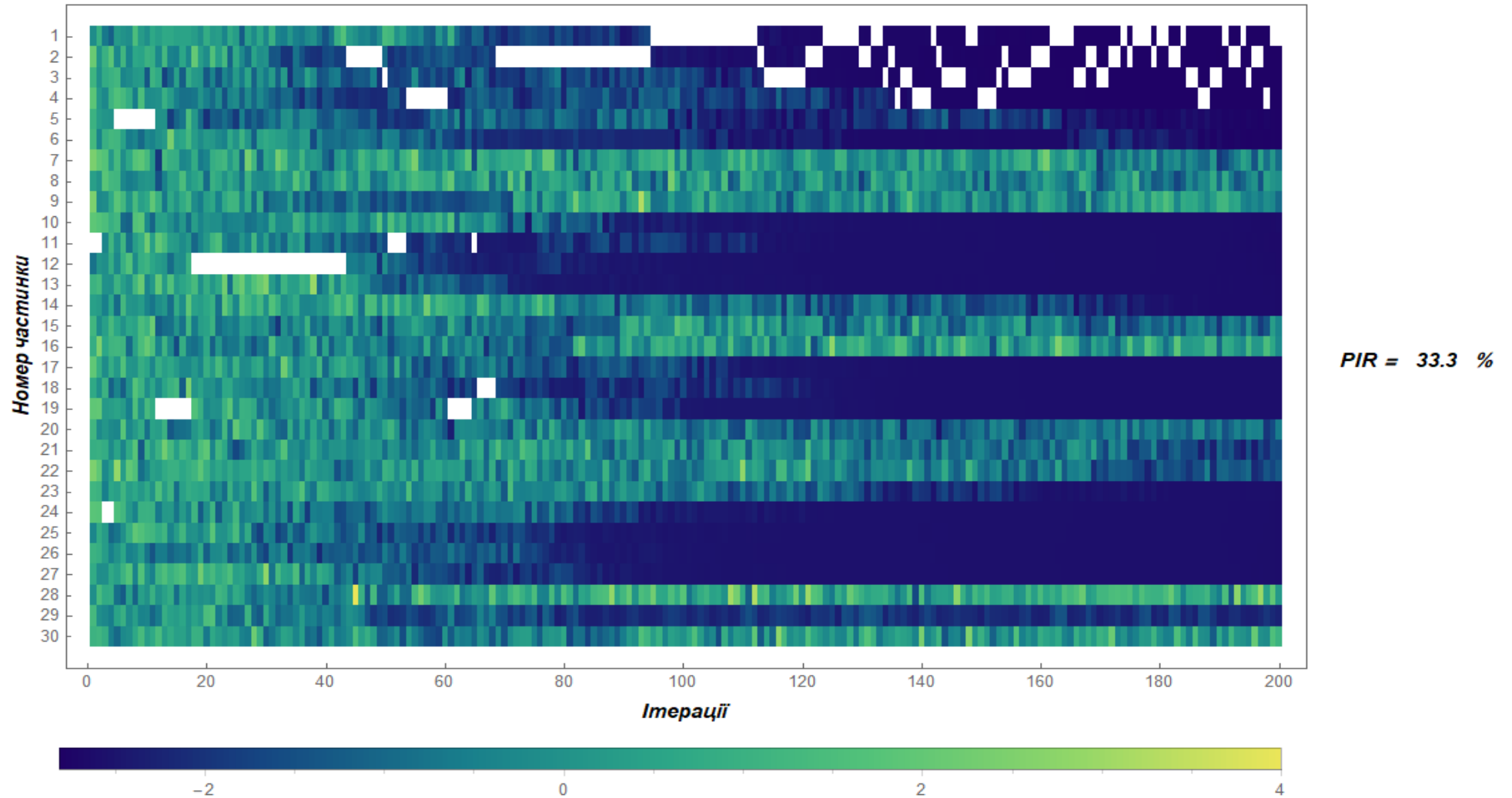
Полоса індикації величини цільової функції знизу дозволяє оцінити значення функції для кожної із частинок.



a)



б)



в)

Рисунок 5.2 – Графічна інтерпретація залученості частинок у пошук розв’язку задачі для алгоритму PSO-Ring, який виконує мінімізацію тестових функцій: а) f_5 ; б) f_{11} ; в) f_{15}

На рис. 5.2 із застосуванням певної кольорової гамми показано значення десяткового логарифму цільової функції кожної частинки на кожній ітерації. Частинка, яка на даній ітерації є кращою з усього рою (її положення відповідає вектору g), показана білим прямокутником. Зліва від графічних об'єктів наведено значення показника PIR . Як видно з рис. 5.2, робота алгоритму PSO-Ring характеризується різною залученістю частинок для пошуку мінімуму функції. Для перших двох випадків ($f5$, $f11$) більше половини частинок покращили розв'язок задачі, у той час як для останнього випадку ($f15$) – лише третина.

Прогрес мінімізації цільової функції кожної частинки рою також можна спостерігати за допомогою інструменту (рис. 5.2), оцінюючи при цьому як швидко рій збігається і наскільки стрімко кращі розв'язки задачі знаходяться роєм.

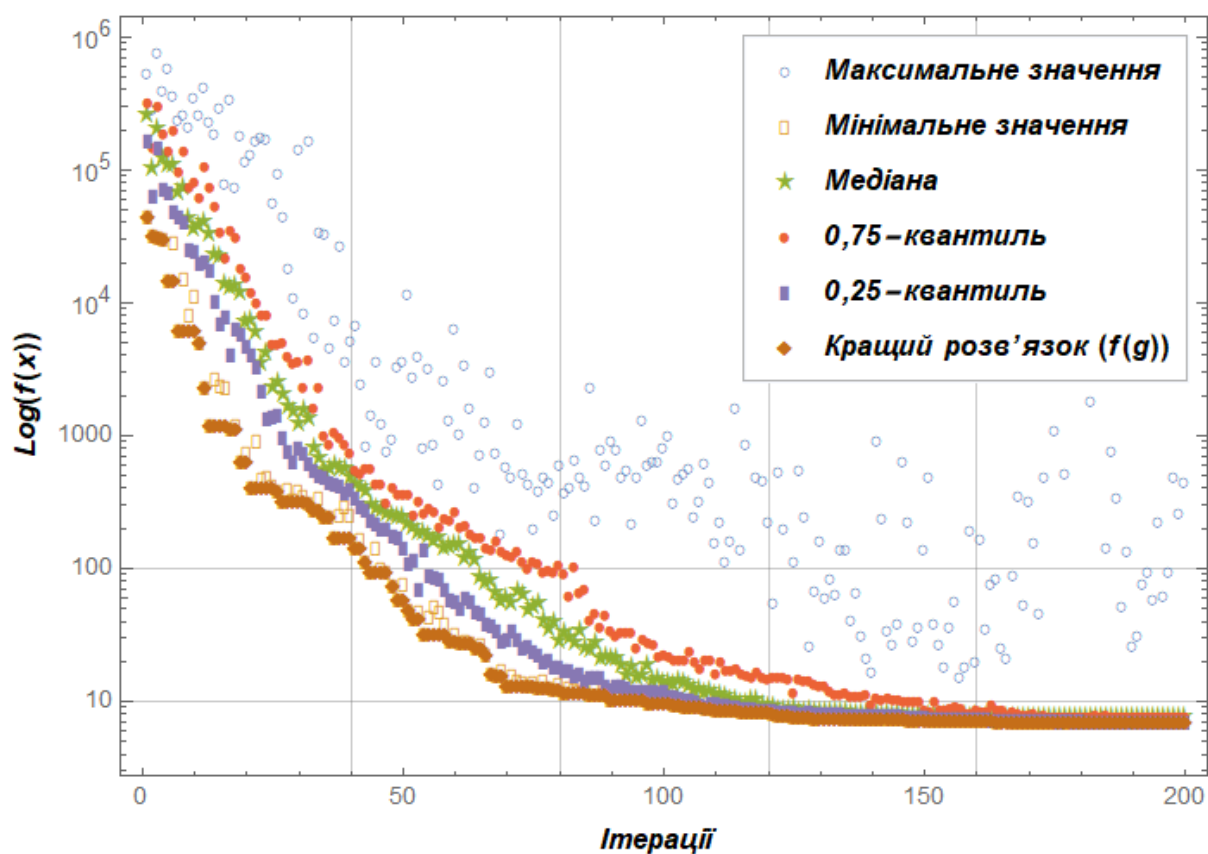
5.3 Клас динамічних оцінок

Клас динамічних показників оцінки роботи алгоритмів, які будуть розроблені у даному пункті, спирається на графічні об'єкти. Вони показують динаміку руху частинок або фіксують певну статистичну характеристику рою від ітерації до ітерації. Крім того, оскільки досліджувані алгоритми характеризуються стохастичним пошуком, то вони можуть бути для різних запусків алгоритму.

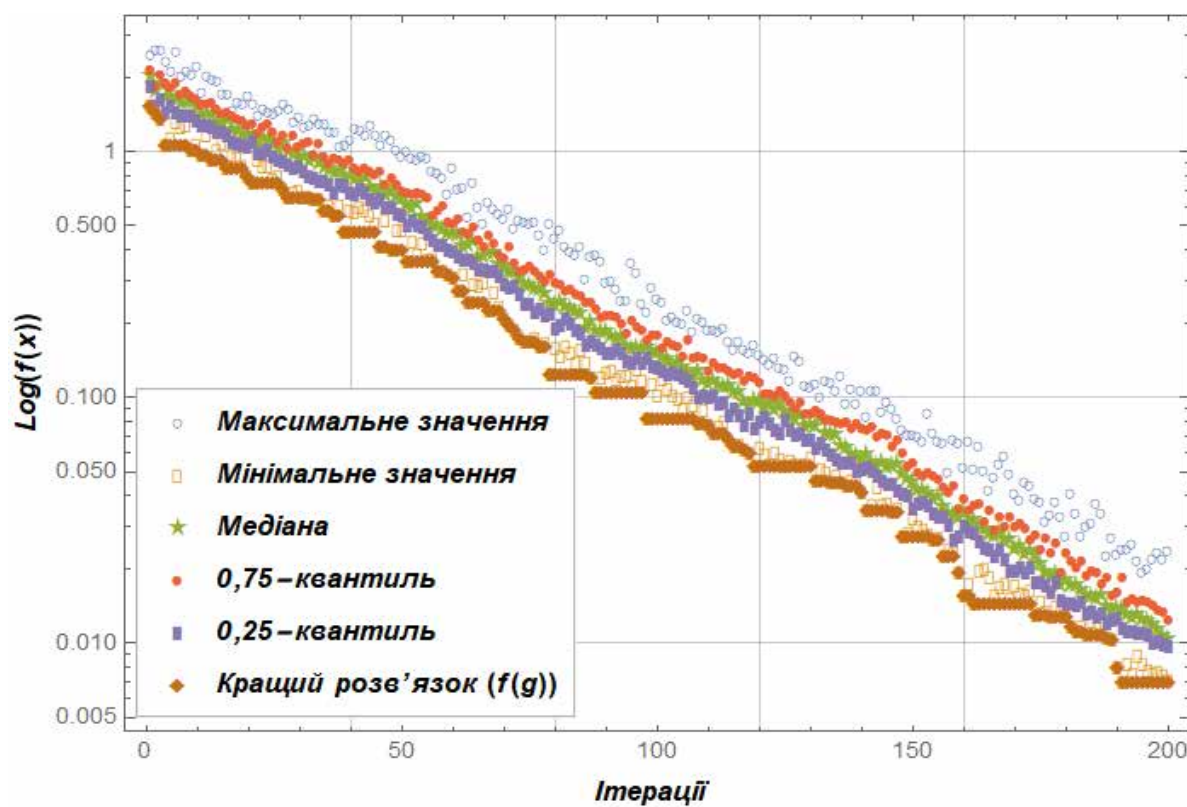
5.3.1 Зміна базових показників рою за ітераціями

Одним із показників, які будемо використовувати для аналізу пошукової діяльності рою є зміна базових (статистичних) показників рою за ітераціями. Для того, щоб зрозуміти яку цінну інформацію може нести цей показник, наведемо його для випадків, які описані у попередньому підпункті (такі ж

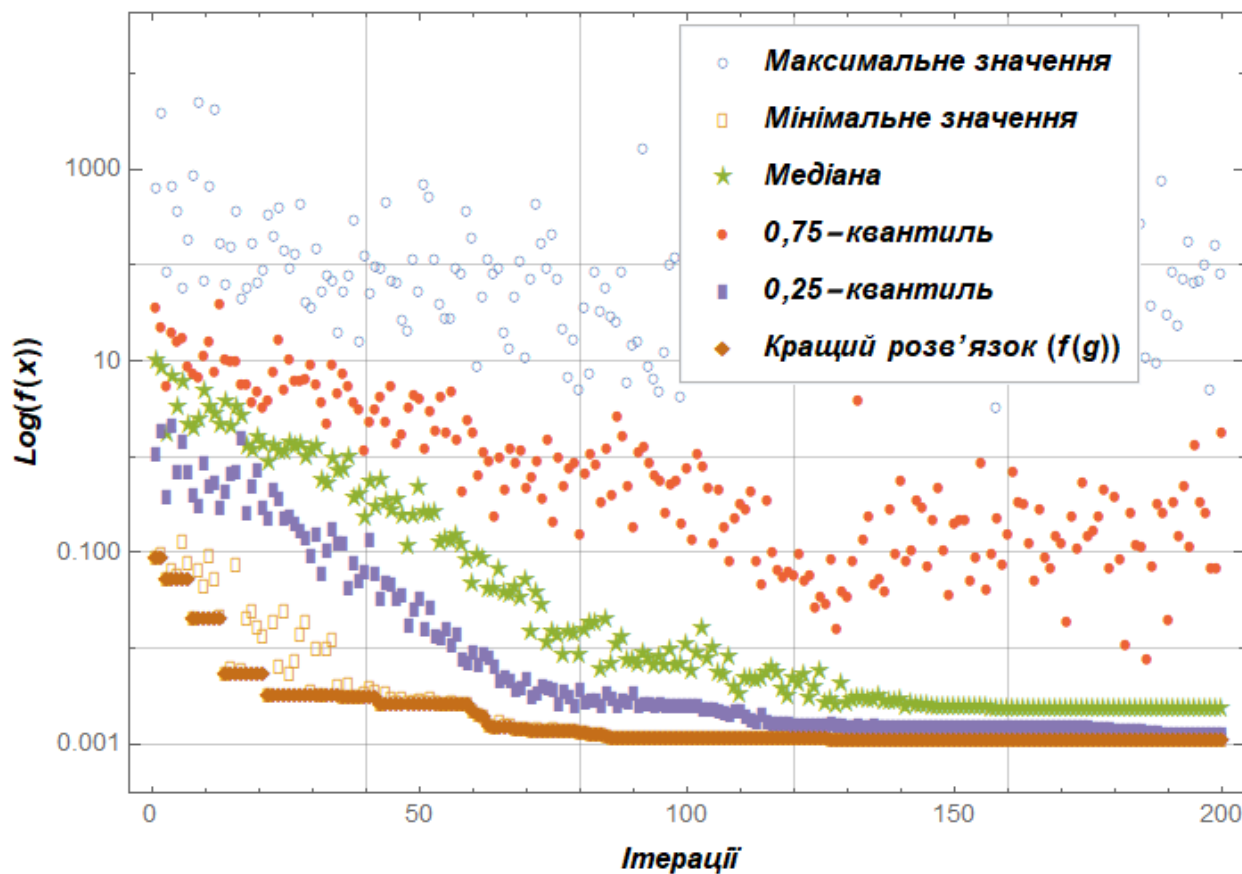
тестові функції, їхня розмірність, кількість ітерацій, алгоритм та популяція рою).



а)



б)



в)

Рисунок 5.3 – Графічна інтерпретація зміни базових показників рою за ітераціями алгоритму PSO-Ring ($J=200$), який виконує мінімізацію тестових функцій: а) f_5 ; б) f_{11} ; в) f_{15}

Зазначимо, що у більшості публікацій з аналізу пошукової діяльності оптимізаційних алгоритмів мультиагентного типу дослідники наводять лише нижній графік (рис. 5.3), який показує найкраще досягнення алгоритму на певній ітерації. Однак, якщо предметом дослідження є пошукова діяльність алгоритму, то такої інформації не достатньо.

Інструмент, який показано на рис. 5.3, дає більшу кількість інформації стосовно динаміки пошуку алгоритмом мінімуму функції. Зокрема, для випадку f_5 (рис. 5.3) видно, що алгоритм практично повністю збігається, хоча гірші розв'язки показують те, що деякі (одна чи декілька) частинки знаходяться далеко від глобального найкращого g . Це також можна пояснити

і тим, що функція f_5 має круті схили і навіть якщо частинки знаходяться близько одна біля одної, значення цільової функції відрізняються на порядки.

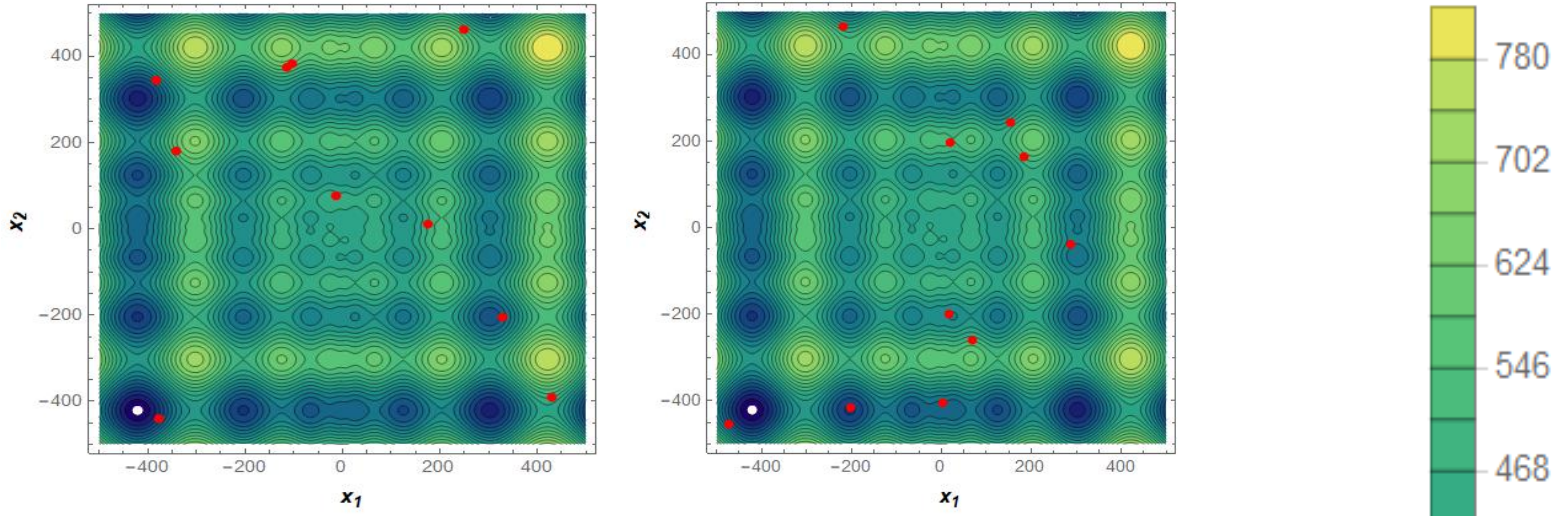
Для випадку f_9 видно, що рій на кінцевих ітераціях ще не збігся. Близькість базових статистичних показників між собою свідчить про певні топологічні характеристики цільової функції, наприклад про те, що коливання функції незначні за амплітудою. Мінімізація тестової функції f_{15} показує збіжність алгоритму, однак певна фракція рою знаходиться досить далеко (за величиною цільової функції на знайденому розв'язку) від глобального найкращого g .

5.3.2 Рух частинок по контурній поверхні функції

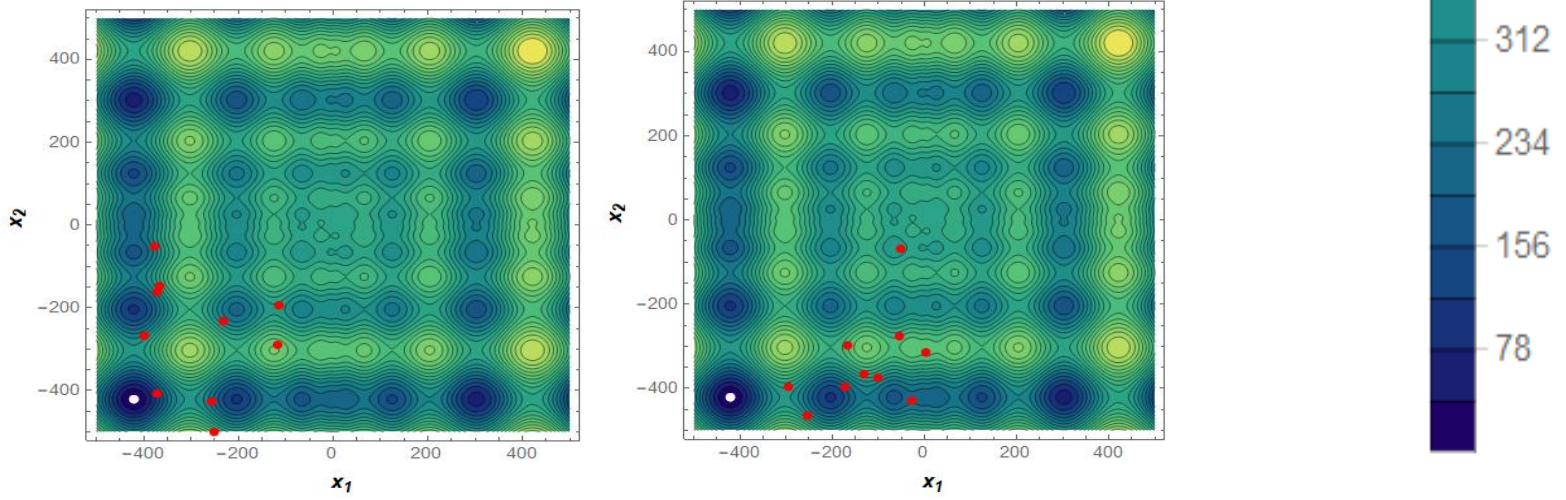
Для того, щоб з'ясувати вплив певної модифікації на динаміку руху частинок рою бажано було б виконати порівняння положень частинок для базового та модифікованого алгоритмів. Таке порівняння необхідно проводити для різних ітерацій, наприклад для першої та наступних, які розраховані як частини від загальної кількості ітерацій (наприклад, $0,1J$, $0,25J$, $0,5J$, $0,75J$, $0,9J$, $0,95J$ та J). Загалом можуть бути використані і інші номери ітерацій для порівняння алгоритмів. Однак, їхня сукупність повинна відображати схему пошуку мінімуму функції „розвідка - дослідження”.

Наведемо приклад використання такого інструменту (рис. 5.4). Для цього порівняємо динаміку руху роїв для алгоритмів IA-PSO та PV-IA-PSO-1. Алгоритми використані для мінімізації тестової функції f_9 , кількість ітерацій $J=100$, розмірність рою $N=10$. На контурних графіках, які показані на рис. 5.4, положення частинок позначені червоними точками. Глобальний мінімум функції показано точкою білого кольору.

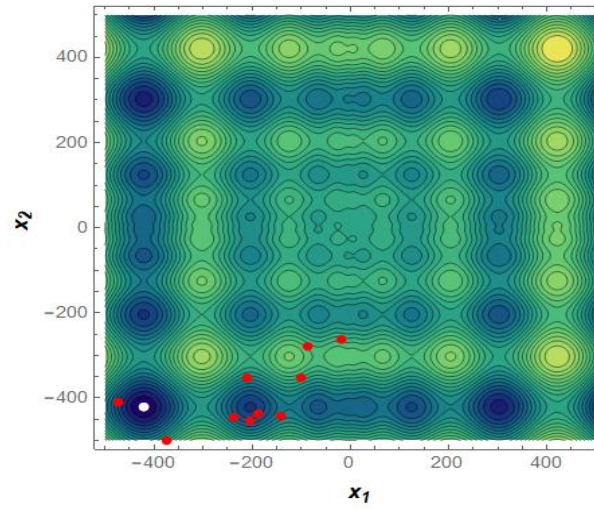
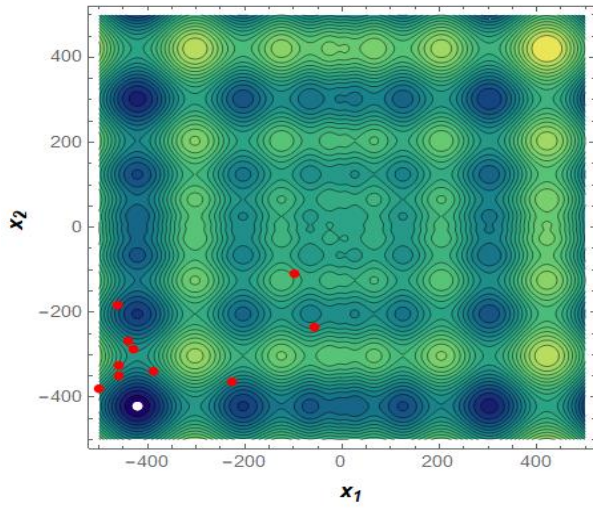
Ліва колонка графіків, які показані на рис. 5.4, відповідає руху рою за алгоритмом IA-PSO, а права – за алгоритмом PV-IA-PSO-1. Порівняння положень частинок на певних ітераціях дає змогу оцінити вплив проведеної модифікації алгоритму на динаміку руху рою.



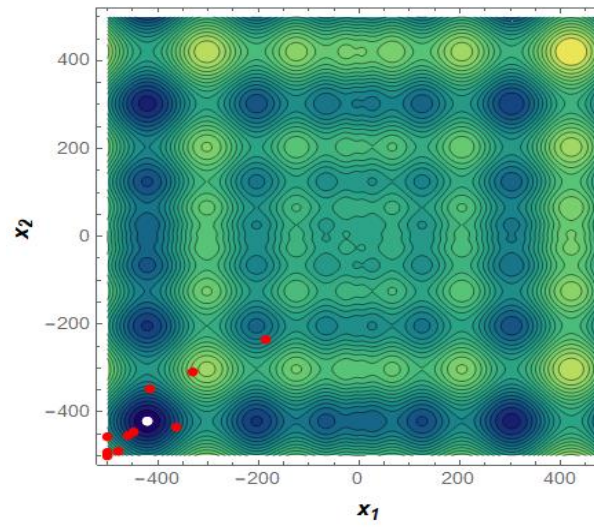
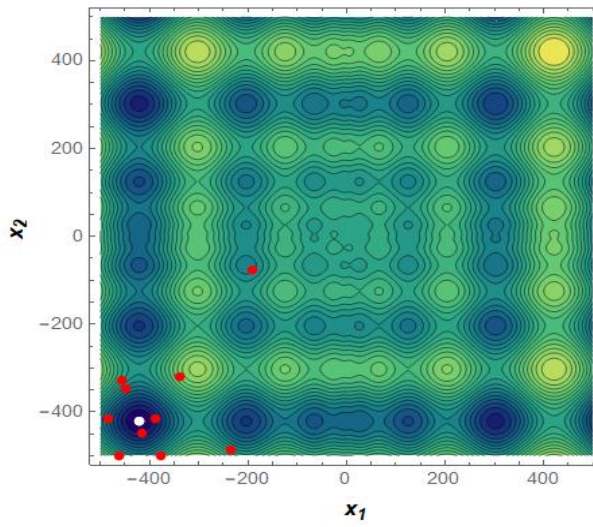
a)



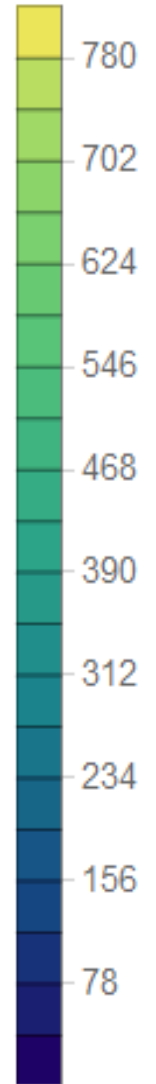
b)

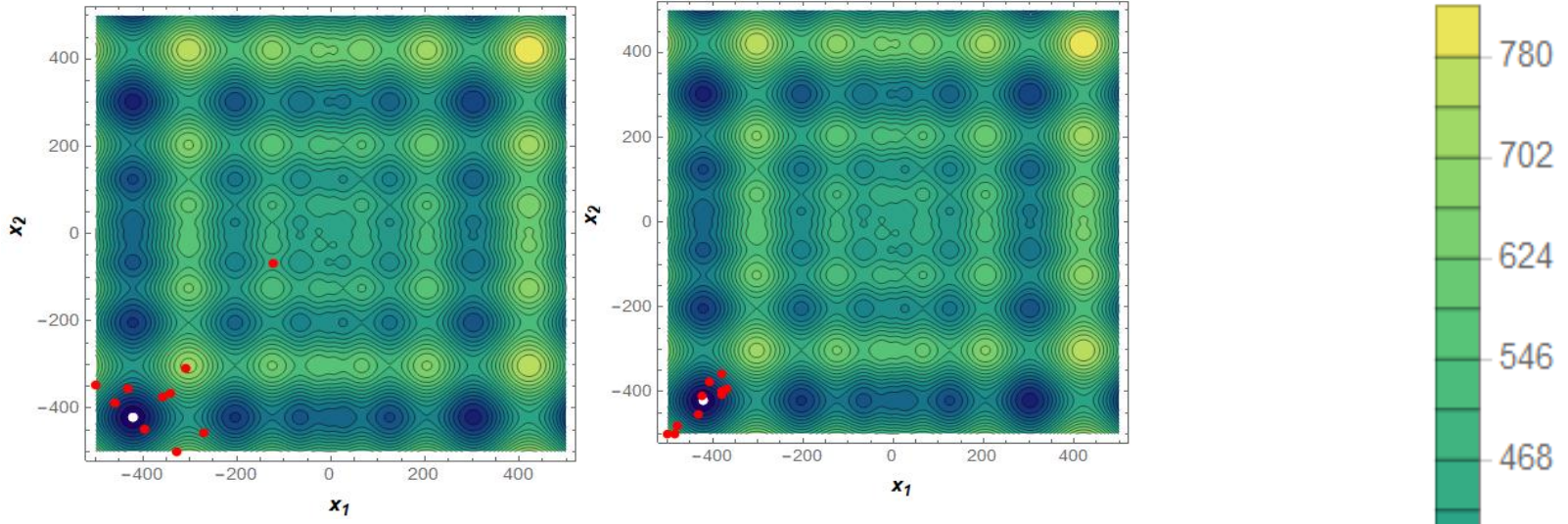


B)

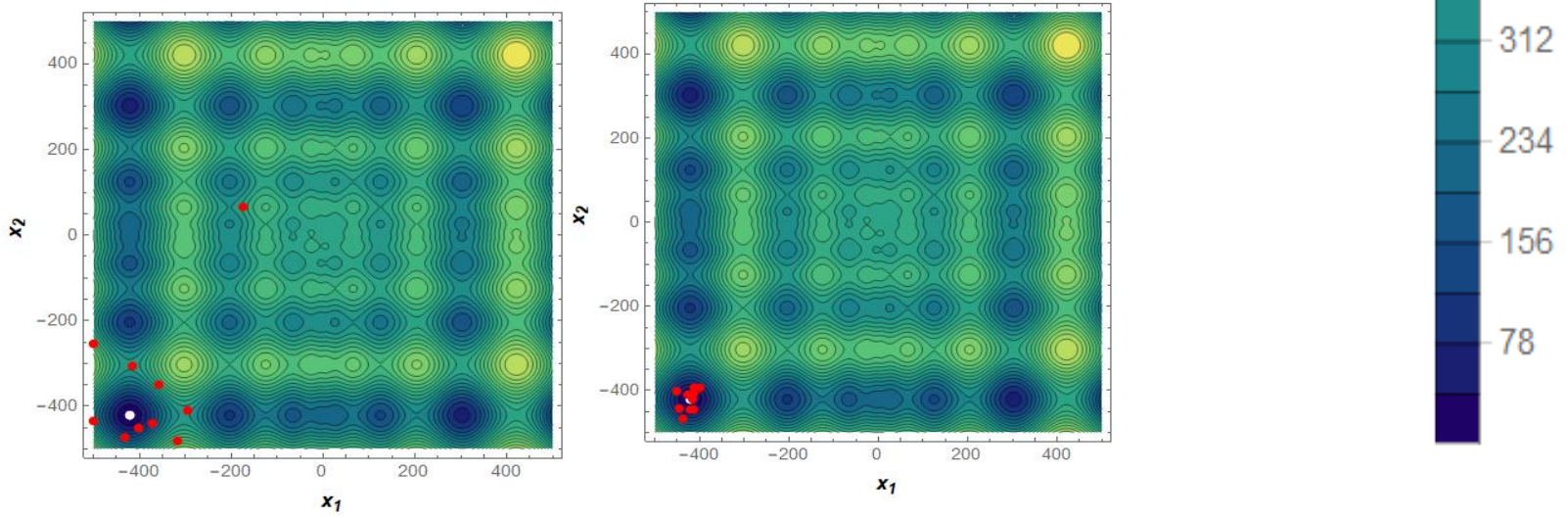


Γ)





Д)



е)

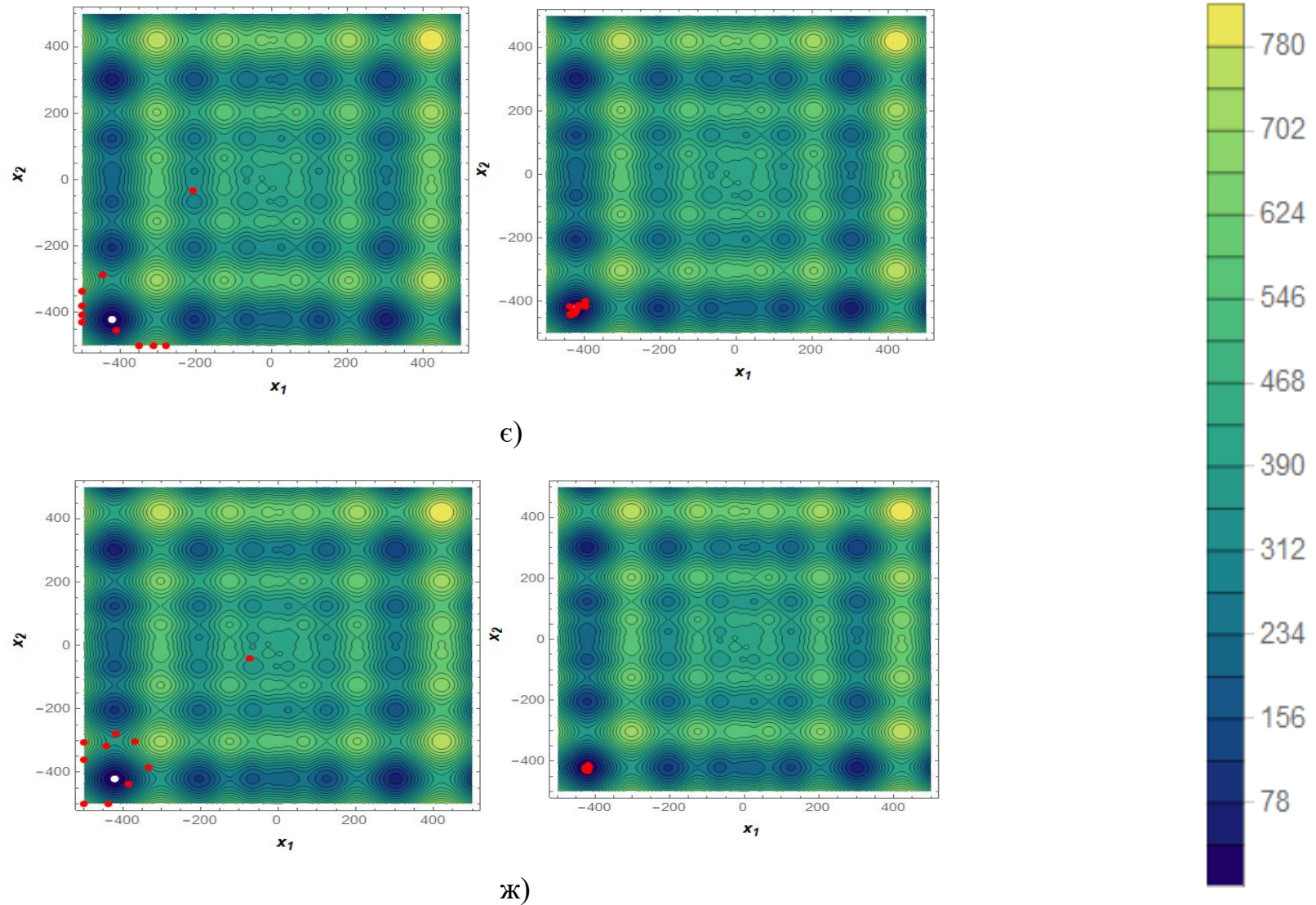


Рисунок 5.4 – Графічна інтерпретація положення частинок для алгоритмів IA-PSO та PV-IA-PSO-1 для окремих ітерацій:

а) $j=1$; б) $j=10$; в) $j=25$; г) $j=50$; д) $j=75$; е) $j=90$; е) $j=95$; ж) $j=100$

Наприклад, для випадку алгоритму PV-IA-PSO-1, який, нагадаємо, є подальшою модифікацією алгоритму IA-PSO, видно, що рій на кінцевих ітераціях збігається і досить точно локалізує глобальний мінімум f_9 , в той час як рій, що рухається за алгоритмом IA-PSO хоч і знаходиться у аттракторі глобального мінімуму тестової функції f_9 , не знаходить глобального мінімуму. Більше того, як видно з рис. 5.5, рій, що рухається за алгоритмом IA-PSO може збігатись у напрямку локального мінімуму (розрахунки проведені для випадку $J=1000$). При цьому, навіть для простого випадку функції f_9 двох аргументів, її глобальний мінімум залишається недосяжним для алгоритму IA-PSO.

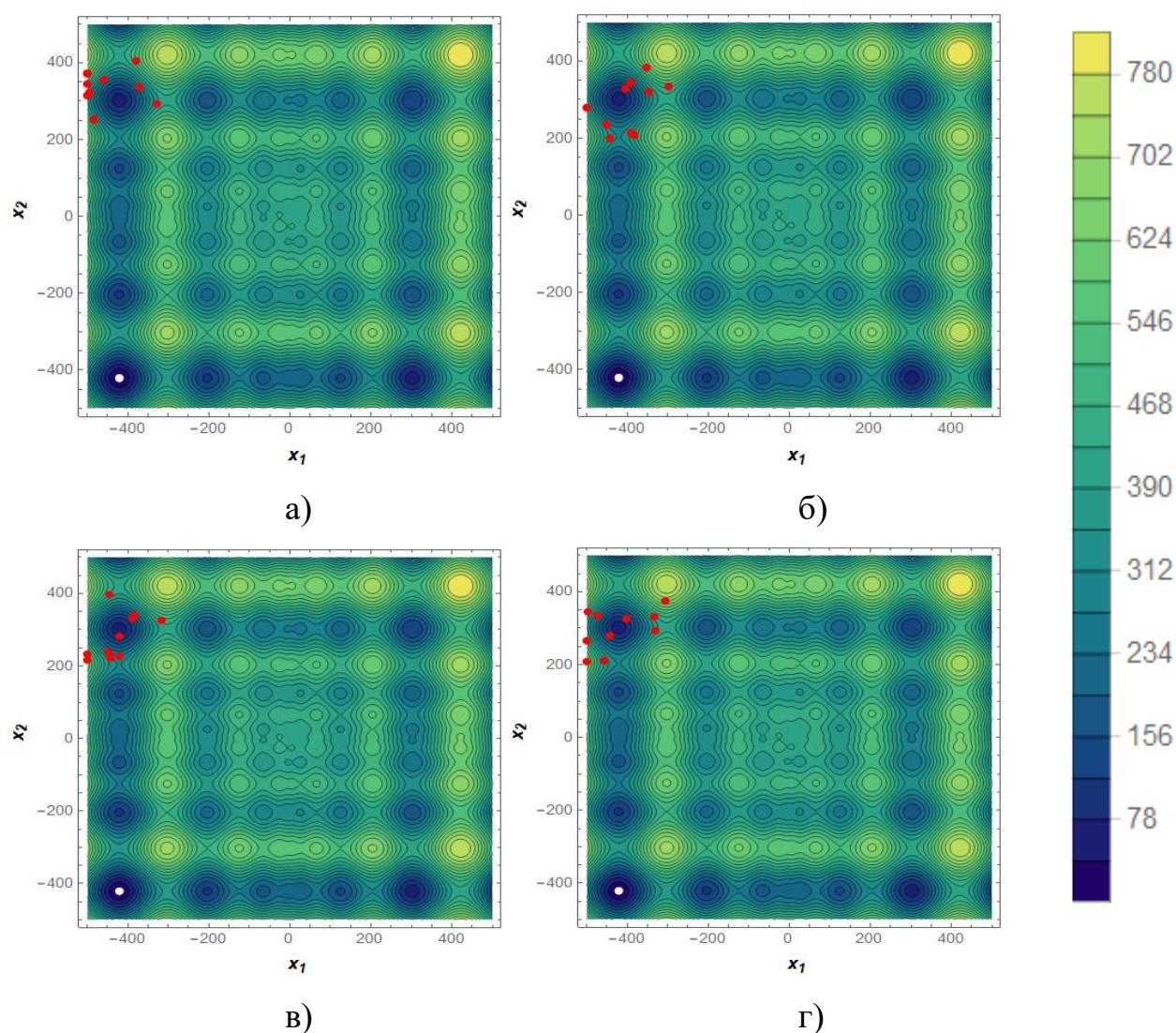


Рисунок 5.5 – Графічна інтерпретація положення частинок для алгоритму IA-PSO для окремих ітерацій при збіжності до локального мінімуму: а) $j=100$; б) $j=500$; г) $j=900$; д) $j=J=1000$

Висновки до п'ятого розділу:

1. У розділі дано загальну характеристику класам (статичний, динамічний та статично-динамічний) оціночних показників ефективності роботи оптимізаційних алгоритмів, у тому числі мультиагентного типу.
2. Для того, щоб використати оціночні показники статичного класу (відносний успіх алгоритму, відносна кількість успішних запусків алгоритму) необхідно виконувати багаторазові запуски оптимізаційних алгоритмів та за цими результатами виконувати розрахунки оціночних показників. Цей клас показників дозволяє нівелювати стохастичний фактор результатів роботи того чи іншого алгоритму. Однак, він не дає жодної інформації стосовно динаміки руху рою, фіксуючи лише досягнутий алгоритмом результат.
3. Динамічні оціночні показники (зміна базових показників рою за ітераціями, рух частинок по контурній поверхні функції) дозволяють спостерігати еволюцію рою та його статистичних характеристик під час виконання алгоритму. Перевагами такого підходу є можливість порівнювати два алгоритми і вказувати на переваги одного над іншим за схемою пошуку мінімуму функції „розвідка - дослідження”. У випадку порівняння батьківського алгоритму і його модифікації, ці показники дають змогу вказати етап (область ітерацій), на якому один із алгоритмів починає переважати інший. Недоліком таких оціночних показників є залежність від стохастичного фактору, який не дає змогу отримувати повторювані результати.
4. Показник залученості частинок у пошук розв'язку задачі (клас статично-динамічних оціночних показників) дозволяє оцінити ефективність використання розрахункових ресурсів для отримання розв'язку задачі та, у деякій мірі, складність самої задачі. Остання теза відноситься також і до показника зміни базових показників рою за ітераціями.

РОЗДІЛ 6

ПРОВЕДЕННЯ КОМПЛЕКСУ ЧИСЕЛЬНИХ ЕКСПЕРИМЕНТІВ ТА СТАТИСТИЧНА ОЦІНКА ОТРИМАНИХ ДАНИХ

6.1 Умови проведення чисельних експериментів щодо оптимізації тестових функцій

Усі експерименти проводилися для двох різних кількостей ітерацій: $J=1000$ і $J=10000$. Перший з експериментів дозволяє оцінити ефективність алгоритмів при ранній зупинці, другий – їх ефективність при пізній зупинці (коли кількість ітерацій достатньо велика). Другий чисельний експеримент включає результати оптимізації тестових функцій, які було найважче оптимізувати при $J=1000$. Вони були відібрані після проведення першої серії дослідів.

Для обох випадків ($J=1000$ і $J=10000$) усі алгоритми запускалися незалежно 50 разів.

Для кожного із зібраних даних було розраховано п'ять статистичних показників: максимальне значення (найгірший знайдений розв'язок), мінімальне значення (найкращий знайдений розв'язок), середнє, медіана та стандартне відхилення.

Всі отримані дані наведені у Додатку Г.

6.2 Оцінка результатів чисельних експериментів

У цьому пункті ми представимо результати чисельних експериментів (NEO) для ранніх (табл. Г.1) і пізніх (табл. Г.2) зупинок модифікацій PSO. Всі чисельні експерименти були проведені для наступних тестових функцій: сферичної ($f1$), еліптичної ($f2$), експоненціальної ($f3$), Швевеля 1.2 ($f4$), Розенброка ($f5$), Растригіна ($f6$), Гриванка ($f7$), Альпайна 1 ($f8$), Швевеля 2 ($f9$), Еклі ($f10$), Вейерштрасса ($f11$), Швевеля 2.20 ($f12$), Цина ($f13$), Саломона ($f14$),

Сінь-Ше Яна 2 (f_{15}), Шуберта 4 (f_{16}), Ріджа (f_{17}), щасливого кота (f_{18}), Гіммельблау (f_{19}), Міхалевича (f_{20}).

Для другої серії експериментів було обрано лише кілька найважливіших для оптимізації тестових функцій ($f_5, f_9, f_{13}, f_{18}, f_{19}, f_{20}$). Усі алгоритми на першій серії експериментів для вказаних тестових функцій показали не високу ефективність, і виникає наступне запитання: алгоритмам було не достатньо ітерацій ($J=1000$), чи причина пов'язана з їхньою низькою пошуковою здатністю? Спробуємо дати відповідь.

У табл. Г.1 та Г.2 найкращі значення (серед порівнюваних) виділені жирним шрифтом. Для випадків тестових функцій f_3, f_{11} та f_{17} деякі алгоритми мають однакові кращі показники.

Аналіз отриманих даних свідчить про ефективність PV-ND-PSO у порівнянні з IA-PSO. Для тестових функцій $f_1, f_2, f_4, f_6, f_7, f_8, f_{10}, f_{12}$ і f_{15} покращення знайденого розв'язку є суттєвим. Зазначимо, що єдина різниця між ними полягає в лінійно-спадному законі зміни стандартного відхилення координат частинки при оновленні її положення. Цей факт підтверджує, що навіть незначними змінами алгоритму можна отримати значне підвищення його продуктивності. Однак, це твердження не є універсальним: справді, спостережувані покращення стосуються не всіх задач мінімізації тестових функцій.

Для вказаних тестових функцій алгоритм PSO-Hybrid не перевершує алгоритм PV-IA-PSO, хоча він показав досить добрі пошукові можливості для них тестових функцій. Для решти функцій PV-IA-PSO показує кращу ефективність, ніж PSO-Hybrid. Причину цих фактів можна виявити за допомогою аналізу активності PSO-Hybrid: при розробці цього алгоритму прийнято компроміс між роботою окремих PSO-алгоритмів. Як і будь-який компроміс, він лише до певної міри подолав недоліки задіяних алгоритмів.

Тестові функції f_3 і f_{17} досить прості для оптимізації: майже всі алгоритми на основі PSO знайшли їхні глобальні мінімуми. Крім того, всі порівнювані алгоритми локалізували глобальний мінімум функції f_{15} .

Що стосується найскладніших оптимізаційних задач, які асоціюються із мінімізацією функцій f_5, f_9, f_{18}, f_{19} та f_{20} , то збільшення кількості ітерацій у десять разів ($J=10000$) порівняно з першою серією експериментів ($J=1000$) не дозволяють значно покращити пошукові характеристики PV-IA-PSO та PSO-Hybrid (єдиним винятком є лише функція f_{13}). Справа пов'язана з особливостями топології згаданих функцій.

Ми не можемо зробити загальний висновок щодо впливу J на ефективність алгоритмів. Тут більш доцільним було б таке твердження: низька різноманітність рою призводить до того, що його пошукова діяльність практично не продовжується (оскільки підтримка різноманітності є необхідною умовою пошуку). Тому треба вивчати показник різноманітності роїв, щоб дати відповідь: чи є шанс знайти глобальний мінімум для найскладніших для оптимізації тестових функцій ($f_5, f_9, f_{13}, f_{18}, f_{19}, f_{20}$).

6.3 Аналіз різноманітності рою

У роботі [91] запропоновано три визначення терміну «різноманітність»: положення частинок, швидкість частинок і персональне краще частинок. У поточному дослідженні проведемо аналіз різноманітності лише за різноманітністю положень частинок. Проте вважаємо, що у формулу різноманітності, запропоновану в [91], слід ввести дві модифікації.

Можна зауважити, що в реальних оптимізаційних задачах і в деяких синтетичних задачах масштаби областей аргументів (від нижньої до верхньої межі) різні. Таким чином, різновиди аргументів можуть істотно відрізнятися один від одного. Щоб нормалізувати їх, запропоновано розділити все різноманіття на відповідну довжину області визначення (відстань від верхньої до нижньої межі того чи іншого аргументу).

Друга модифікація пов'язана зі зміщенням центральної точки рою (точки ваги). У роботі [91] його координати розраховуються як середні аргументів по всьому рою. Вважаємо, що така центральна точка повинна бути

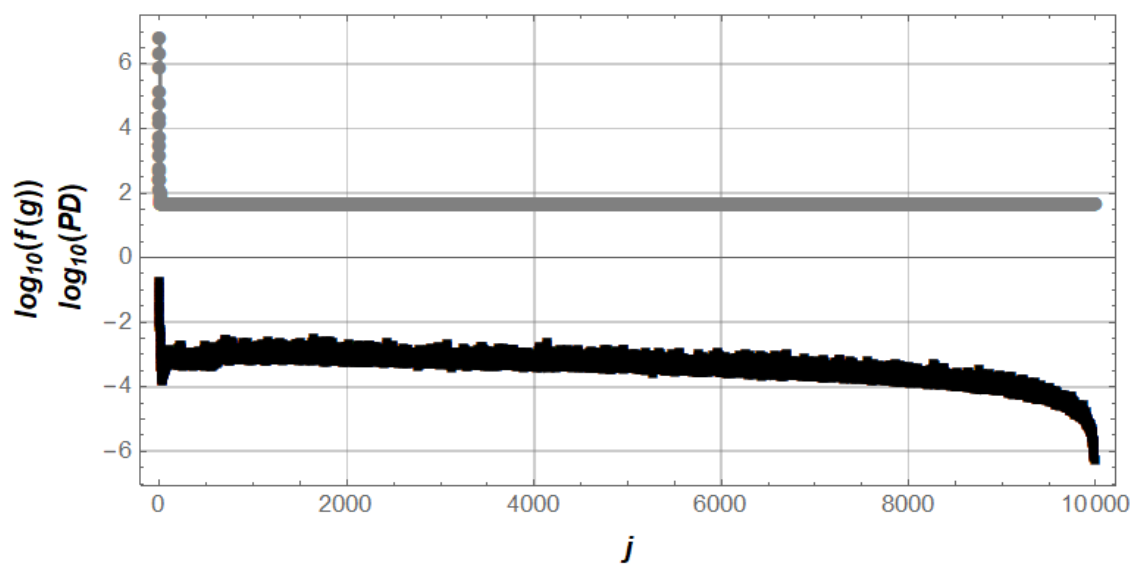
переміщена в глобальне найкраще (g), оскільки вона притягує всі частинки рою. Поєднання згаданих ідей дає наступну формулу:

$$PD_j = \frac{1}{D \cdot N} \sum_{d=1}^D \sum_{n=1}^N \frac{|x_{j,n,d} - g_{j,d}|}{x_{d,max} - x_{d,min}}, \quad (6.1)$$

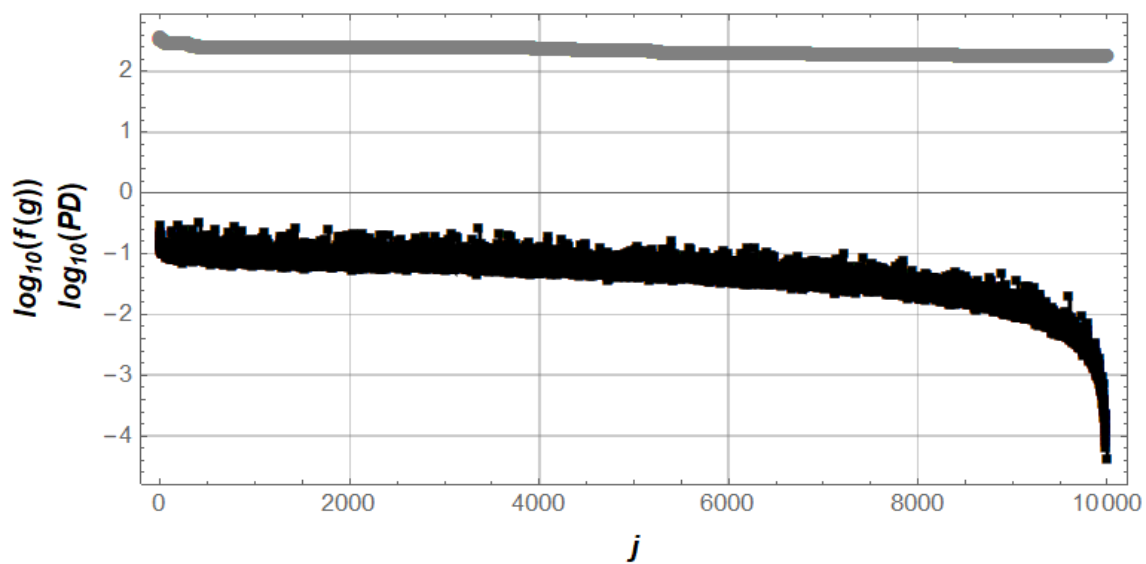
де D – розмірність цільової функції (кількість її аргументів); N – кількість частинок у рої; $x_{d,max}$ і $x_{d,min}$ – нижня і верхня межі d -того аргументу відповідно; j – поточний номер ітерації; $x_{j,n,d}$ – значення d -того аргументу n -ої частинки на j -тій ітерації; $g_{j,d}$ – значення d -того аргументу глобального найкращого на j -тій ітерації.

На рис. 6.1 показані графіки збіжності та різноманіття роїв для різних порівнюваних алгоритмів. Усі дані представлені в логарифмічних масштабах. Для побудови відповідних графіків для функції $f20$ (рис. 6.1, е) віднімали сертифіковане (встановлене теоретично) значення глобального мінімуму від масиву $f20(g_j)$ (табл. 2.1): ця операція змістила всі дані $f20(g_j)$ вгору, що, в свою чергу, дозволило представити дані в логарифмічних масштабах. Для випадку функції $f19$ всі дані масиву $f19(g_j)$ додали до величини 3500 для того, щоб отриманий результат став масивом додатних чисел.

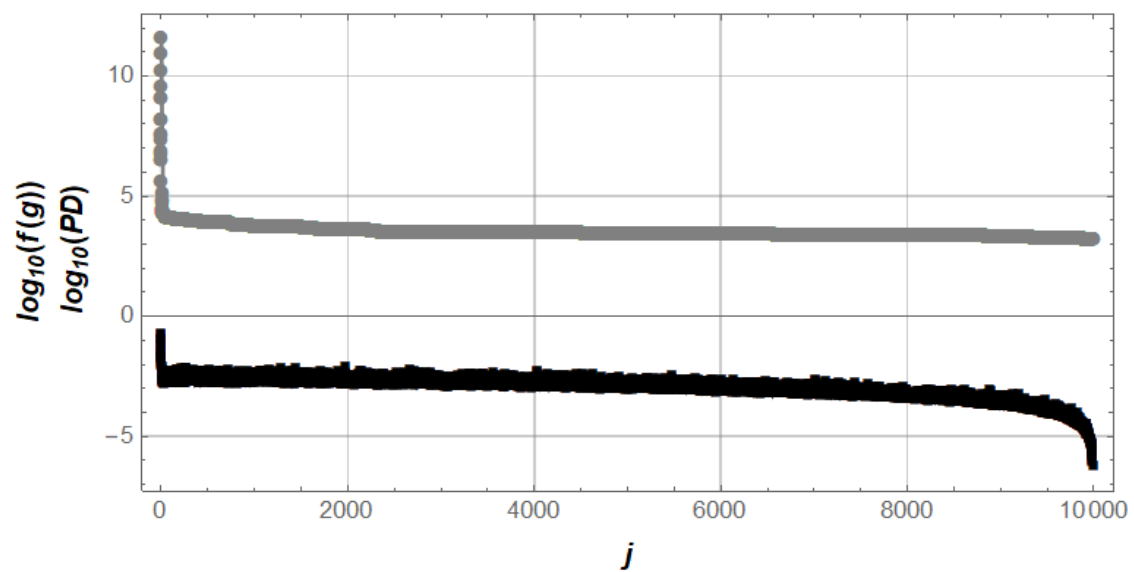
Майже для всіх випадків (крім $f18$) наприкінці пошукової процедури різноманітність різко зменшувалася. Для функцій $f5$, $f9$, $f13$ і $f18$ PD має слабкий вплив на ефективність алгоритму PV-IA-PSO. Останні два випадки ($f19$ і $f20$) є цікавими через чіткі докази сильного впливу малих значень σ на високу ефективність алгоритму PV-IA-PSO. Таким чином, залежність $\sigma = \sigma(j)$, яка використана при розробці цього алгоритму, підходить для ефективної мінімізації не всіх тестових функцій.



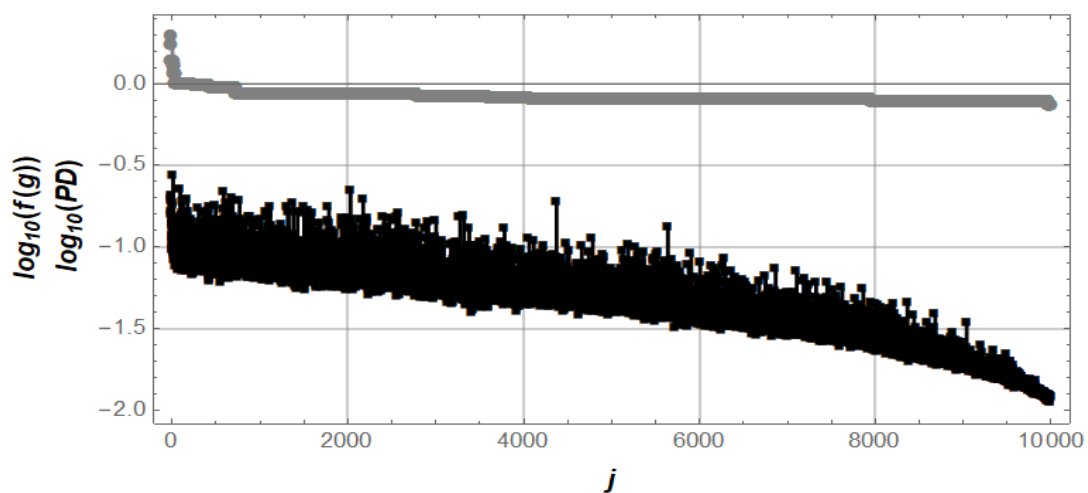
a)



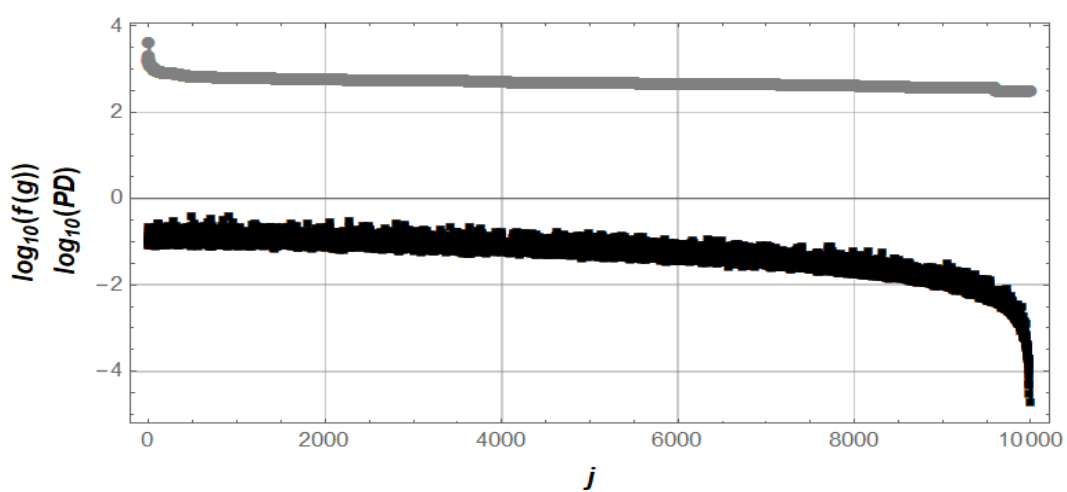
б)



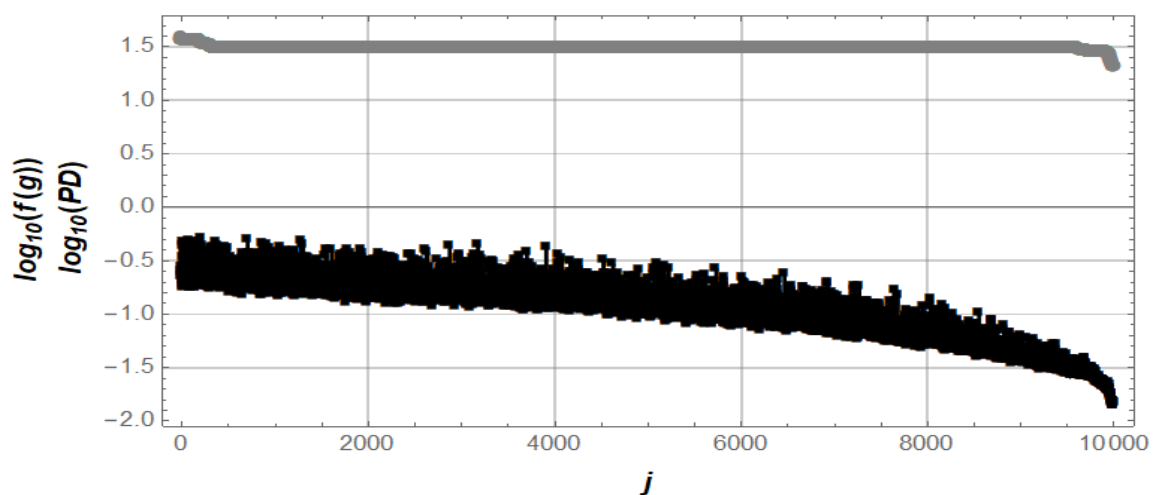
B)



г)



д)



е)

Рисунок 6.1 – Графіки збіжності алгоритму PV-IA-PSO (сірий графік) і різноманітності PD (чорний графік) для функцій: а) f_5 ; б) f_9 ; в) f_{13} ; г) f_{18} ; д) f_{19} ; е) f_{20}

Було проведено аналогічний аналіз для алгоритму PSO-Hybrid (рис. 6.2). Тут можна спостерігати зовсім іншу пошукову поведінку алгоритму. Різноманітність для всіх розглянутих тестів практично на одному рівні. Немає різкого зменшення різноманіття положень часточок наприкінці пошуку. Цей факт підштовхує нас до висновку, що PV-IA-PSO не використовується на останніх етапах активності PSO-Hybrid. Очевидно, що на пізніх стадіях роботи PSO-Hybrid були використані інші алгоритми.

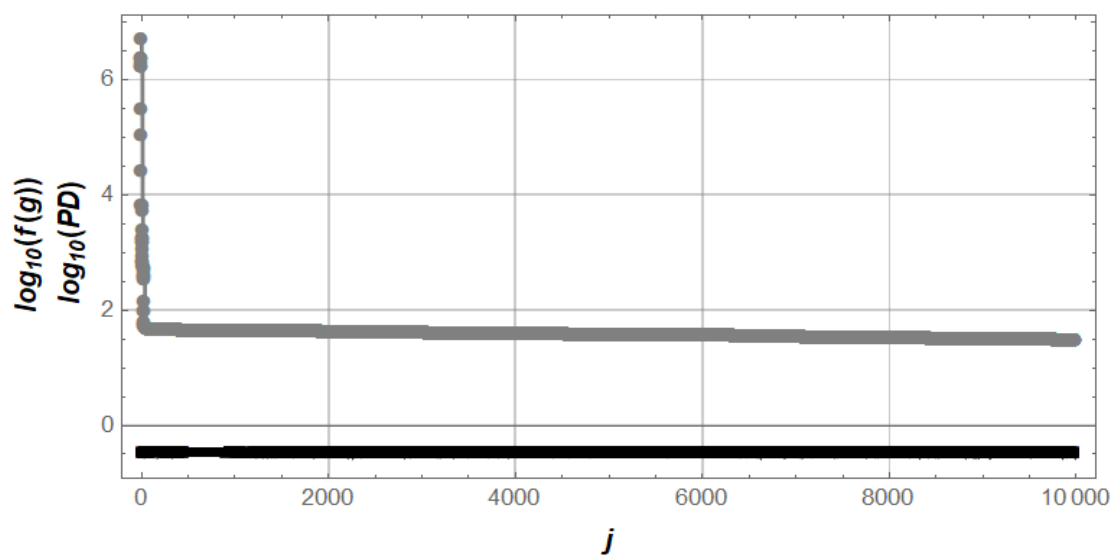
Тепер наведемо короткий порівняльний аналіз PSO-Hybrid і PV-IA-PSO в контексті мінімізації тестових функцій. При мінімізації f_5 PSO-Hybrid демонструє повільне покращення розв'язку, у той час як PV-IA-PSO не має значних змін після двадцятої ітерації.

При мінімізації функції f_9 PSO-Hybrid трохи перевершує PV-IA-PSO. Алгоритм PSO-Hybrid має шанси продовжити ефективну процедуру пошуку, через достатній рівень PD . Для PV-IA-PSO в кінці пошуку значення PD дуже мале.

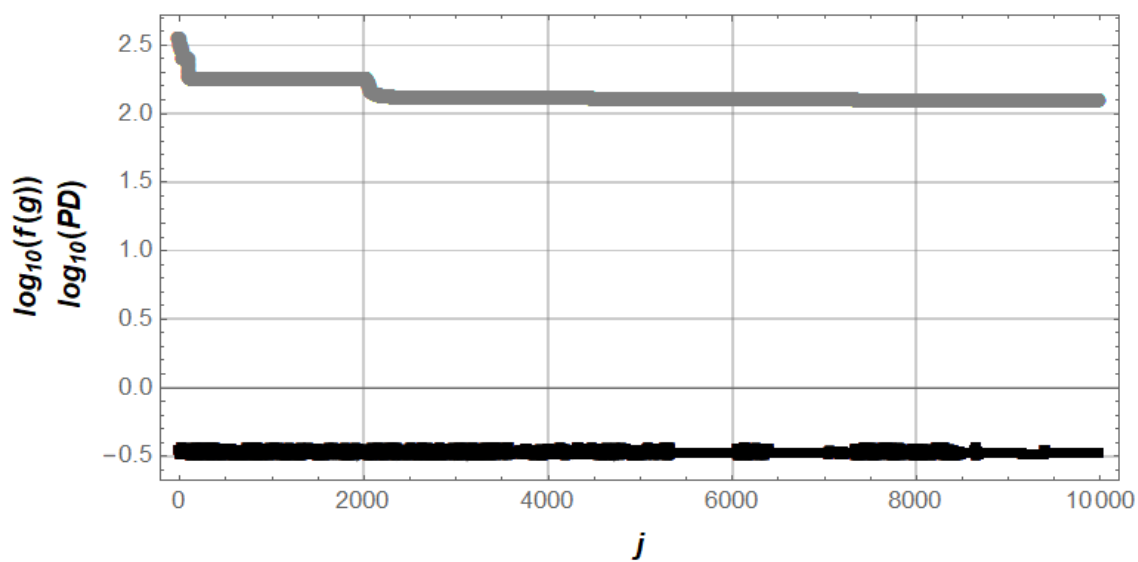
При мінімізації f_{13} можна побачити велику різницю в ефективності алгоритмів. Розумне поєднання алгоритмів на основі PSO (стратегія PSO-Hybrid) дозволяє поступово мінімізувати цю функцію. PV-IA-PSO показує дуже повільне зменшення f_{13} .

Для f_{18} PSO-Hybrid активно відшукував кращі розв'язки до 500-ї ітерації. Після цього спостерігається лише незначне поліпшення отриманих результатів. PV-IA-PSO демонструє слабку пошукову активність під час виконання ітерацій з 700-ї і до останньої.

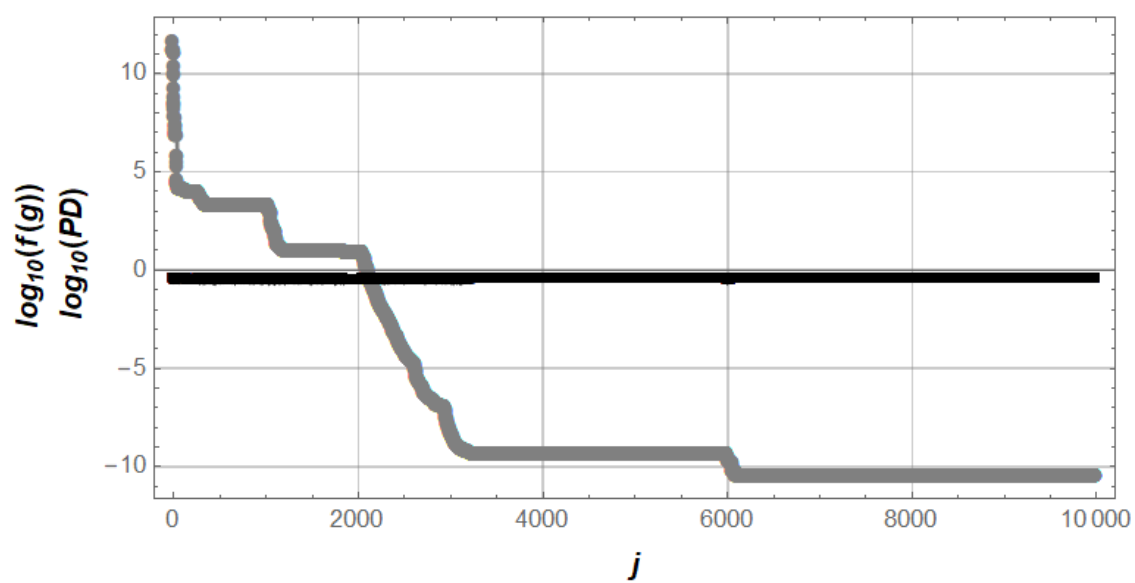
Для f_{19} алгоритм PSO-Hybrid після 600-ї ітерації майже припинив пошукову активність, на відміну від цього PV-IA-PSO продовжує мінімізацію, а швидкість мінімізації тестової функції збільшує в кінці процедури (коли σ мала). Однак, досягнутий за допомогою алгоритму PV-IA-PSO мінімум функції f_{19} є гіршим, ніж той, що був знайдений за допомогою PSO-Hybrid.



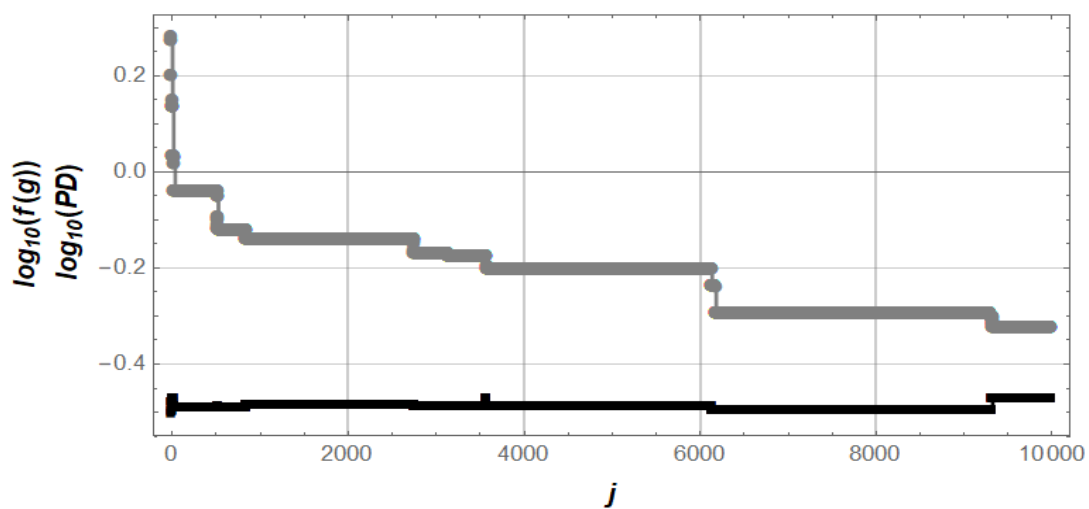
a)



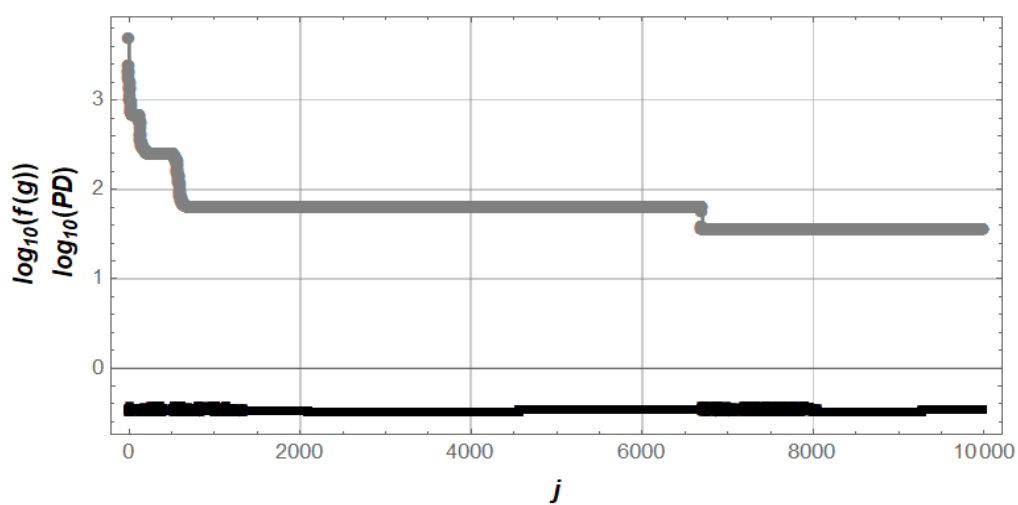
б)



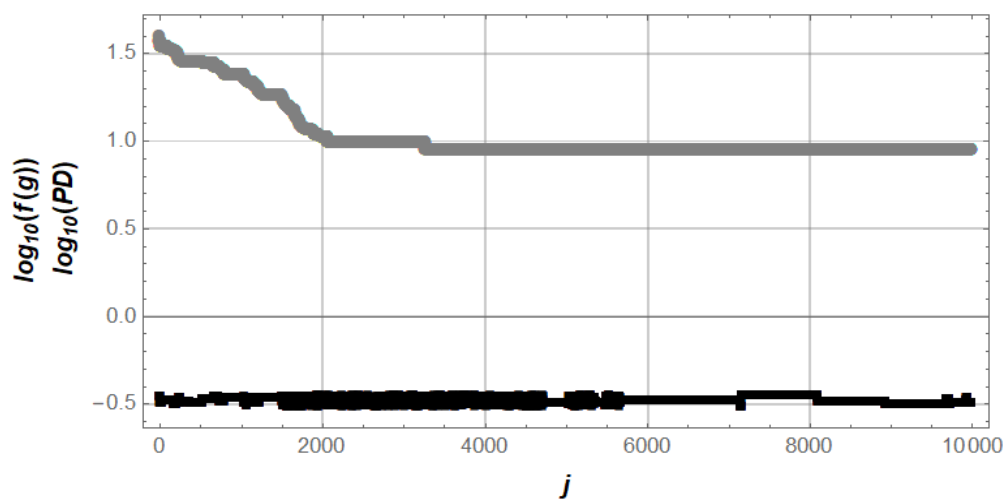
B)



г)



д)



е)

Рисунок 6.2 – Графіки збіжності алгоритму PSO-Hybrid (сірий графік) і різноманітності PD (чорний графік) для функцій: а) $f5$; б) $f9$; в) $f13$; г) $f18$; д) $f19$; е) $f20$

Подібний ефект можна спостерігати для мінімізації функції f_{20} . Приблизно після 2050-ї ітерації PSO-Hybrid припиняє практично всі покращення розв'язків. Навпаки, як було зазначено раніше, зниження σ викликає відродження пошукової активності PV-IA-PSO. Однак, розв'язок, який знайдено за допомогою алгоритму PSO-Hybrid, краще, ніж той, що знайдено за допомогою PV-IA-PSO.

Беручи до уваги всі дані, можна зробити наступні висновки: 1) лише для тестової функції f_{13} алгоритм PSO-Hybrid може покращити результат за рахунок збільшення кількості ітерацій J (отже, для деяких реальних задач оптимізації збільшення J може позитивно вплинути знайдений розв'язок задачі); 2) для ефективної мінімізації функцій f_{19} і f_{20} за допомогою алгоритму PV-IA-PSO необхідно суттєво зменшувати діапазон значень σ .

6.4 Оцінювання за додатковими показниками

6.4.1 Вибір і розробка додаткових показників

Дані, які наведені в табл. Г.1 і Г.2, є необробленими. Їх можна проаналізувати лише коротко. Для більш глибокого дослідження потрібні конкретні показники, які мають базуватися на зазначених даних. Вони запропоновані нижче.

У роботі [68] було застосовано два критерії для оцінки ефективності порівнянних PSO-алгоритмів. Вони базуються на медіанних значеннях серій незалежних пусків алгоритмів. Вони будуть використані в поточному дослідженні. Перший показник – це сума рангів алгоритму, які отримані на всіх тестових функціях (табл. Г.3). Однак, як зазначено в роботі [68], такий показник не відноситься до досягнутих значень медіан, а оцінює лише їх порядок. Тому будемо застосовувати другий показник, який розроблено в статті [68]:

$$LogCr = \frac{\log_{10} \left(\frac{f_{zj=1}}{f_{zj=J}} \right)}{\max \left(\left\{ \log_{10} \left(\frac{f_{zj=1}}{f_{zj=J}} \right) \right\} \right)}, \quad (6.2)$$

де $f_{zj=1}$ – медіана значення z -го алгоритму (для його п'ятдесяти незалежних пусків), яке пов'язане з першою ітерацією алгоритму (перша ітерація всіх алгоритмів однакова – це обчислення цільової функції для випадково ініціалізованих аргументів); $f_{zj=J}$ – медіана значення z -го алгоритму, яке пов'язане з J -ю (останньою) ітерацією алгоритму. Як зазначено в [68] значення (6.2) показує ефективність роботи алгоритму по відношенню до найкращого досягнутого розв'язку задачі (звичайно, для конкретної цільової функції). Для найкращого алгоритму (з точки зору найменшого середнього значення конкретної цільової функції) $LogCr$ дорівнює одиниці. Розраховані значення показника (6.2) наведено у табл. Г.4.

Зауважимо, що перший і другий показники можна застосовувати не лише до медіанних значень, а й до інших. Наприклад, до мінімальних значень: вони показують потенціал алгоритму (з точки зору ефективності пошуку). Його можна отримати за допомогою кількох незалежних пусків. У табл. Г.5 і Г.6 наведено відповідні показники.

Подальше узагальнення показників приводить нас до розгляду векторів. Досі всі показники були скалярними. Усі вони відображають небажані значення, тобто чим вони менші, тим краща ефективність алгоритму. У отриманих даних (табл. Г.1 та Г.2) також міститься показник стандартного відхилення – він відповідає стабільності роботи алгоритму, його низькій кореляції з стохастичним фактором пошуку.

Таким чином, далі ми розглянемо вектор, який включає п'ять основних показників: максимальне значення (найгірший варіант), мінімум (найкращий варіант), середнє значення, медіана та стандартне відхилення. Всі ці дані отримані за 50 незалежних запусків того чи іншого алгоритму.

Підсумовуючи все, третій показник *ind* обчислюється за наступними виразами:

$$\begin{aligned}
 \text{Max}_{1.\text{max}} &\geq \dots \geq \text{Max}_{\alpha.\text{max}} \geq \dots \geq \text{Max}_{Z.\text{max}}, \alpha \in \overline{(1, Z)}; \\
 \text{Min}_{1.\text{min}} &\geq \dots \geq \text{Min}_{\beta.\text{min}} \geq \dots \geq \text{Min}_{Z.\text{min}}, \beta \in \overline{(1, Z)}; \\
 \text{Average}_{1.\text{av}} &\geq \dots \geq \text{Average}_{\gamma.\text{av}} \geq \dots \geq \text{Average}_{Z.\text{av}}, \gamma \in \overline{(1, Z)}; \\
 \text{Median}_{1.\text{med}} &\geq \dots \geq \text{Median}_{\delta.\text{med}} \geq \dots \geq \text{Median}_{Z.\text{med}}, \delta \in \overline{(1, Z)}; \\
 \text{SD}_{1.\text{st.dev.}} &\geq \dots \geq \text{SD}_{\varepsilon.\text{st.dev.}} \geq \dots \geq \text{SD}_{Z.\text{st.dev.}}, \varepsilon \in \overline{(1, Z)}; \\
 \bar{u}_z &= [\alpha.\text{max}, \beta.\text{min}, \gamma.\text{av}, \delta.\text{med}, \varepsilon.\text{st.dev.}]^T; \\
 \text{ind} &= \left\langle \bar{v}_z, \bar{e} \right\rangle,
 \end{aligned} \tag{6.3}$$

де $\alpha, \beta, \gamma, \delta, \varepsilon$ – нижні індекси, що відображають відповідний ранг показника в списку порівнюваних показників; z – індекс z -го алгоритму; Z – загальна кількість порівнюваних алгоритмів ($Z=8$); \bar{e} – вектор вагових коефіцієнтів, які відображають значущість відповідної складової у векторі \bar{u}_z . У поточному дослідженні встановлені такі значення вагових коефіцієнтів: $\bar{e}=[0,05, 0,3, 0,2, 0,4, 0,05]^T$.

Максимальне значення та стандартне відхилення є найменш значимими показниками. Максимальне значення є результатом невдалого запуску алгоритму і існує можливість досягнення кращого розв'язку задачі при повторних пусках. Стандартне відхилення пов'язане зі стабільністю роботи алгоритму (в умовах багаторазових пусків), але не з пошуковою діяльністю самого алгоритму.

Мінімально досягнуте значення є найбільш значущим, як було зазначено раніше. Цей показник показує потенційну можливість знайти добрий розв'язок задачі оптимізації.

Порівнюючи середнє та медіанне значення, віддаємо перевагу останньому, оскільки воно більш стійке до окремих викидів у даних (великі відхилення від математичного очікування), про що свідчать відповідні вагові коефіцієнти. Проте обґрунтований набір компонентів \bar{e} залишається відкритим

питанням. Застосування формул (6.3) до даних (табл. Г.1) дозволив отримати значення *ind*, що наведені у табл. Г.7.

Одним із найважливіших параметрів виконання алгоритму є його тривалість протягом заданої кількості ітерацій, яка є четвертим показником. Це дозволяє оцінити ефективність обчислювальних ресурсів, які були витрачені для знаходження розв'язку оптимізаційної задачі. У дослідженні запускали кожен алгоритм на ПК з такими параметрами: ЦП Intel Core i5-8400, 2,8 ГГц; 8 Гб оперативної пам'яті; 64-бітна операційна система Windows 10 Pro. Усі отримані дані наведено в табл. Г.8.

6.4.2 Оцінка роботи алгоритмів за числовими значеннями додаткових показників

Порівнюючи дані, наведені в табл. Г.3 і Г.5, можна зробити висновок, що великої різниці у відхиленні алгоритмів (за їх ефективністю) немає. Наприклад, перші три алгоритми в цих таблицях однакові: PSO-Hybrid, PV-IA-PSO та R-Ring-PSO. Таким чином, у рамках поточного дослідження медіанні та мінімальні значення відображають одну тенденцію.

За критерієм (6.2) медіанний та мінімальний показники відносяться до близьких результатів (табл. Г.4 та Г.6). Три найкращі алгоритми: PV-IA-PSO, PSO-Hybrid та IA-PSO. Лідер за медіанним показником (6.2) (PV-IA-PSO) переміг за рахунок дуже точної локалізації глобальних мінімумів тестових функцій. PSO-Hybrid є другим. Розрахунок показника на основі мінімуму (табл. Г.6) показує перевагу алгоритму PSO-Hybrid (PV-IA-PSO є другим), тобто багаторазові пуски алгоритму дозволяють отримати трохи кращі мінімальні значення. Це, у свою чергу, означає, що PSO-Hybrid потенційно може знайти кращий розв'язок тієї чи іншої оптимізаційної задачі.

Розширені (векторні) показники (6.3), да за якими наведено у табл. Г.7, свідчать про перевагу PSO-Hybrid над усіма порівнюваними алгоритмами. Дуже близьке значення *ind* відноситься до PV-IA-PSO, третє місце посідає

алгоритм R-Ring-PSO. Найгірший серед порівнюваних алгоритмів – ME-D-PSO.

Отже, незначне покращення алгоритму IA-PSO, яке реалізовано у PV-IA-PSO, призводить до збільшення ефективності до 57,1% (за показником *ind*). Слід зазначити обмеженість цього твердження: воно базується лише на даних, що отримані при мінімізації тестових функцій $f1-f20$.

Отже, PV-IA-PSO є потужним алгоритмом для розв'язання багатьох задач оптимізації; однак PSO-Hybrid показує кращу загальну ефективність.

Одним із важливих показників, який пов'язаний не з ефективністю алгоритмів, а з необхідними обчислювальними ресурсами для їх виконання, є тривалість виконання алгоритму. За цим показником три найкращі алгоритми: LDW-PSO, VCT-PSO та Ring-PSO (табл. Г.8). Вони є простими модифікаціями канонічного PSO. Цей факт пояснює швидкість обчислень. Розроблені у даному дослідженні алгоритми (PV-IA-PSO і PSO-Hybrid) працюють повільно. Найповільнішим є ME-D-PSO через необхідність переініціалізації рою (на задання положень частинок витрачаються відносно великі обчислювальні ресурси). Ще один момент, який необхідно підкреслити: PSO-Hybrid може мати різну тривалість виконання при мінімізації різних цільових функцій. Це пояснюється тим, що під час запуску PSO-Hybrid будуть вмикатися різні PSO-алгоритми (вони вибираються відповідно до ефективності алгоритмів; однак, який з них буде кращим для мінімізації тієї чи іншої цільової функції априорі невідомо).

Аналіз графіка, представленого на рис. 6.3, чітко свідчить, що великі значення ψ відповідають швидкому виконанню PSO-Hybrid і навпаки. Цей ефект пояснюється тим, що після ітерації ψ виникає потреба переписати матрицю, яким представлено рій (положення частинок, швидкості тощо). Таким чином, чим менше число ψ , тим частіше необхідно виконувати ці обчислення, при цьому тривалість виконання алгоритму збільшується.

Підсумовуючи аналіз наведемо наступну тезу: краще витратити додатковий час на виконання алгоритму або залучити додаткові

обчислювальні ресурси і отримати (з високою ймовірністю) точну локалізацію глобального мінімуму цільової функції, ніж витративши мало часу на розв'язування задачі, не знайти її добрий розв'язок.

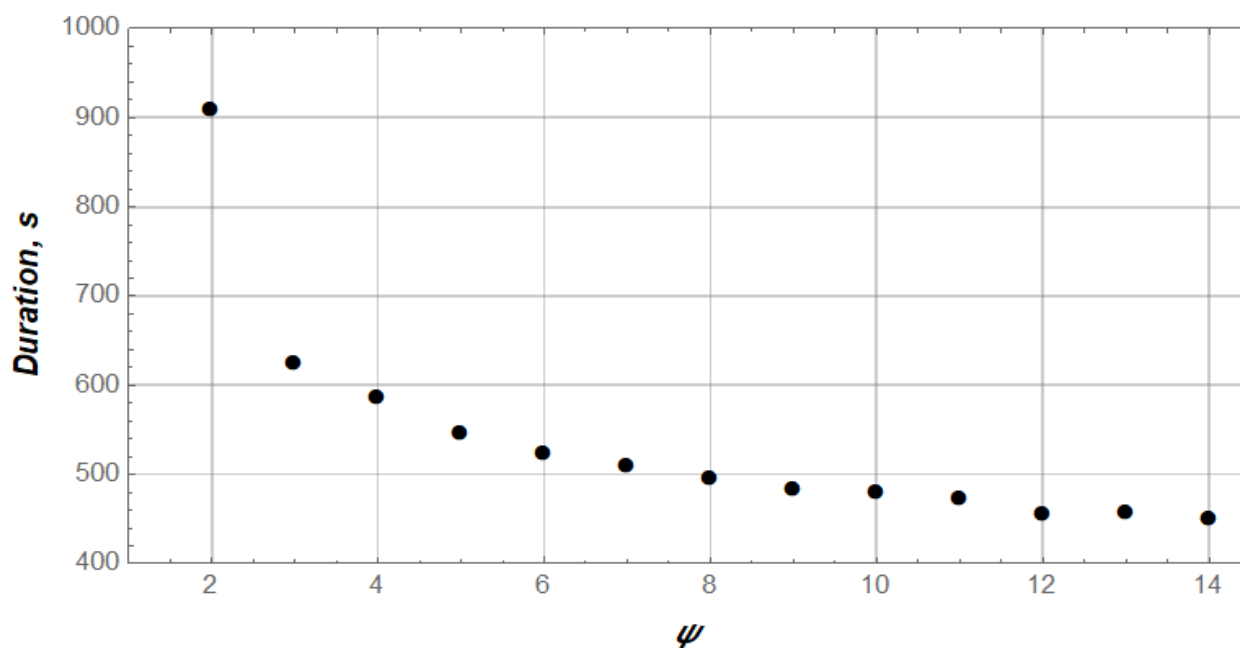


Рисунок 6.3 – Діаграма тривалості виконання алгоритму PSO-Hybrid (для всіх тестових функцій) як функція параметра ψ

6.5 Дослідження впливу ступеня зв'язності рою алгоритмів l-best PSO на їхню ефективність

Сучасний рівень розвитку науки і техніки пов'язаний з гострою потребою у вирішенні різноманітних задач оптимізації. Для цього розроблено безліч методів. Потік наукових робіт щодо цього напрямку не зменшується, а навпаки, зростає. Серед величезного класу методів оптимізації є «зірковий» підклас: оптимізація рою частинок (PSO) [9], диференціальна еволюція [31], оптимізатор сірого вовка [92], зозулиний пошук [32] тощо. Цей перелік алгоритмів добре відомий завдяки десятиліттям розробки та успішного застосування цих алгоритмів до значної кількості задач оптимізації [47].

Одним із найвпливовіших параметрів PSO (у сенсі ефективності роботи алгоритму) є зв'язність частинок – здатність отримувати та передавати інформацію про знайдений розв'язок задачі. Менша зв'язність рою призводить

до повільного поширення інформації між частинками i , відповідно, повільної збіжності алгоритму. Чим більша зв'язність рою, тим швидша збіжність. Висока зв'язність роїв може спричинити ризики передчасної збіжності, що призведе до поганого дослідження цільової функції. Наприклад, у роботі [45] зазначено, що для різних метаевристичних алгоритмів зв'язність агентів у рої повинна становити 15% від її розміру. У роботі [14] проаналізовано ефективність PSO з різними топологіями з'єднань між частинками. Тут встановлено, що найкращими є зіркова, кільцева та фон-нейманівська топології. Також отримано нові перспективні топології PSO [48, 93]. Автори дослідження [50] дійшли висновку: оптимальна топологія з'єднань залежить не тільки від конкретної задачі, але й від доступного обчислювального бюджету.

Інший перспективний напрям у цьому відношенні пов'язаний з динамічними топологіями зв'язків, коли зв'язки між частинками перебудовуються під час роботи алгоритму [94]. Деякі з них базуються на поточній ітерації [71], інші – на особливостях еволюції рою [95]. Однак ці підходи вимагають глибокого аналізу, щоб отримати стійке підвищення ефективності пошуку алгоритму PSO із локальними зв'язками між частинками (l-best PSO) для широкого класу цільових функцій.

Тим не менш, існує незадоволена потреба в розумінні впливу рівня зв'язності між частинками у l-best PSO зі статичними топологіями з'єднань. Цьому питанню присвячено дане дослідження.

6.5.1 Топології зв'язків PSO l-best та g-best

Алгоритм PSO є типовим прикладом багатоагентного методу, де множина агентів (часток) поводить як рій. Рухаючись по поверхні цільової функції, рій виявляє інтелектуальні особливості, які дозволяють знайти мінімум (або максимум залежно від задачі) цільової функції. Кожна частинка описується вектором положення x і вектором швидкості v . Обидва вони

містять D компонентів (D – кількість аргументів цільової функції, які потрібно знайти). Оновлення x і v описується такими виразами:

$$\begin{cases} v_d^{i+1} = wv_d^i + c_1r_1(p_d^i - x_d^i) + c_2r_2(g_d^i - x_d^i); \\ x_d^{i+1} = x_d^i + v_d^{i+1}, \quad d \in (\overline{1, D}), \end{cases} \quad (6.4)$$

де v_d і x_d – d -ті компоненти x і v відповідно; p_d – d -та складова так званого особистого кращого, тобто найкращого положення, яку частинка знайшла до i -ї ітерації; g_d – d -й компонент так званого глобального кращого, тобто найкращого положення, яку рій знайшов на i -ї ітерації; w , c_1 , c_2 – коефіцієнти (інерційний, когнітивний і соціальний відповідно), які вказують на величину відповідного чинника, що впливає на рух частинки; r_1 , r_2 – випадкові числа, які генеруються на інтервалі $0 \dots 1$. Загальна кількість ітерацій алгоритму дорівнює I , а верхній індекс i вказує на поточний номер ітерації, $i \in (\overline{1, I})$.

Після завершення розрахунку за виразом (6.4) вектори p і g необхідно оновити:

$$\begin{cases} p^{i+1} = x^{i+1}, & \text{якщо } f(x^{i+1}) < f(p^{i+1}); \\ g^{i+1} = p^{i+1}, & \text{якщо } f(p^{i+1}) < f(g^{i+1}), \end{cases} \quad (6.5)$$

де f – цільова функція, яку необхідно мінімізувати.

Представлений вище алгоритм відомий як g-best, через те, що глобальне краще g використовується в першому рядку системи (6.4). Це може призвести до передчасної збіжності алгоритму, оскільки кожна частинка «знає» про найкраще рішення, знайдене на даний момент, і вона змушена рухатися до g . Виправити зазначений недолік можна шляхом застосування так званого l-best підходу, де вектор g замінюється на вектор gl :

$$gl_k = \min_{\text{arg}} (f(p_k), f(p_j)), \quad \forall j < K - 2, \quad j \in (\overline{1, J}), \quad k \in (\overline{1, K}), \quad (6.6)$$

де k – номер частинки; K – загальна кількість частинок у рої (популяція рою); J – загальна кількість частинок у локальному рої (частина цілого рою). В

алгоритмі, який використовує вектор g_l , інформація про кращий знайдений розв'язок поширюється набагато повільніше (чим менше J , тим повільніше це поширення). У багатьох випадках це запобігає передчасній збіжності, оскільки рій має достатньо ітерацій для дослідження цільової функції.

У відомих і широко застосовуваних алгоритмах l-best зв'язки частинок розташовуються в деякі структури: кільцеву, фон Неймана (сіткову), зіркову, колесоподібну тощо. Найбільш популярними є перші дві. Однак, незважаючи на численні дослідження топології зв'язків між частинками, є кілька важливих питань:

- 1) чи існують якісь загальні рекомендації щодо налаштування J для широкого класу цільових функцій;
- 2) чи є необхідність в організації спеціальних структур з'єднань (як зазначено вище), або на ефективність роботи l-best алгоритму впливає лише значення J .

Надалі розглянемо зв'язки випадкової статичної топології між частинками (скорочено RSTC_{J/K}-PSO). У позначенні „RSTC_{J/K}-PSO” нижній індекс вказує на кількість частинок у локальному рої (J) та загальну популяцію рою (K). Якогось особливого порядку в з'єднаннях частинок немає. Для кожної частинки її сусідні частинки мають бути випадковим чином встановлені перед першою ітерацією алгоритму. Єдина вимога: індекси (порядкові номери) частинок у сусідстві не повторюються.

6.5.2 Постановка чисельного експерименту

Найпоширеніший спосіб дослідження ефективності алгоритмів стохастичної оптимізації пов'язаний з їх багаторазовими запусками на різних тестових функціях. Останні є деякими синтетичними цільовими функціями для мінімізації, які мають складну топологію. У поточному дослідженні було використано шість добре відомих тестових функцій: еліптичну ($f2$), Швевеля 2 ($f9$), Еклі ($f10$), Вейерштрасса ($f11$), щасливого кота ($f18$), Міхалевича ($f20$).

Сподіваємось, що таке різноманіття тестових функцій дозволить виявити загальну ефективність пошуку порівнюваних алгоритмів і стане основою для до певної міри загальних висновків.

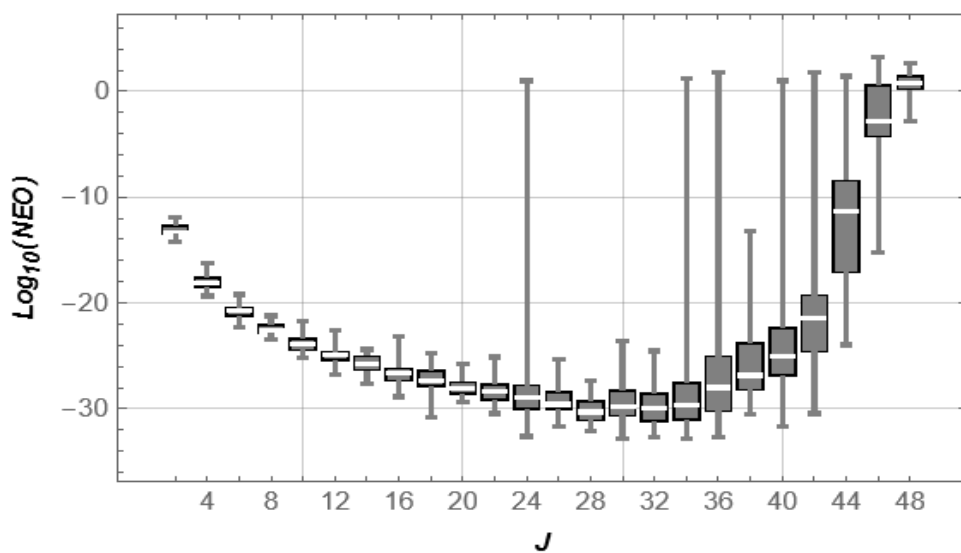
Кількість аргументів усіх тестових функцій у рамках поточного дослідження дорівнює 30 ($D=30$). Порівняли кілька найкращих PSO: Ring-PSO (кільцьову), фон-Неймана PSO, $RSTC_{J/50}$ -PSO. Загальні параметри для порівнюваних алгоритмів $c_1=c_2=1,19$, $w=0,72$, $K=50$. Щоб отримати статистично достовірні результати, кожен з алгоритмів на кожному з контрольних тестів було незалежно запущено 50 разів.

Для l-best PSO було проведено низку чисельних експериментів. Кожний з експериментів проводився з постійною J , вона змінювалася від 2 до 48 з кроком 2.

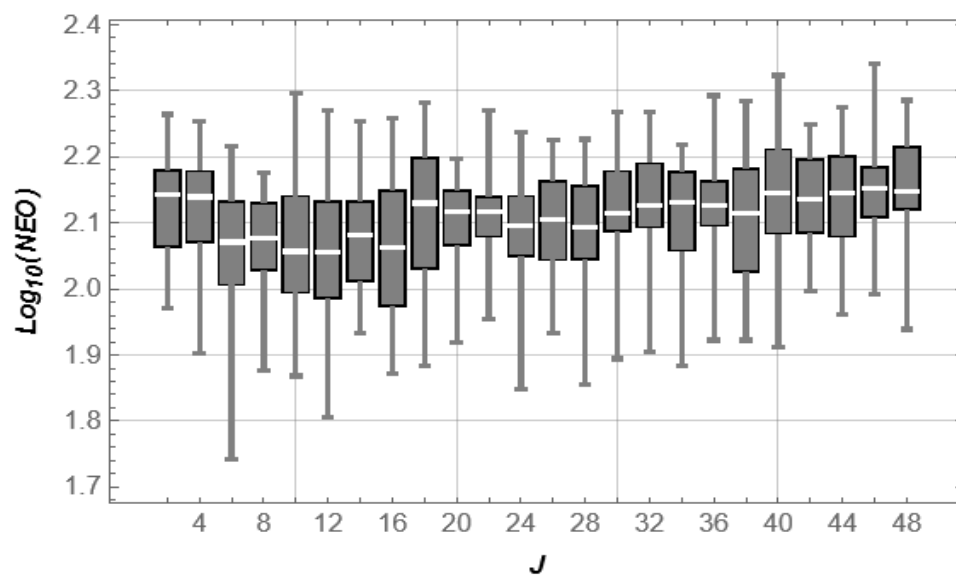
6.5.3 Визначення впливу J на ефективність $RSTC_{JK}$ -PSO

Першим кроком в аналізі отриманих результатів є їх візуалізація, яка дозволяє оцінити вплив значення J на показники $RSTC_{JK}$ -PSO. З цієї причини використали діаграми Бокса-Вішкера (рис. 6.4), які були побудовані на основі десяткового логарифму отриманого результату числового експерименту (скорочено NEO). Використовували таку обробку даних через значну (на порядки) різницю величин NEO .

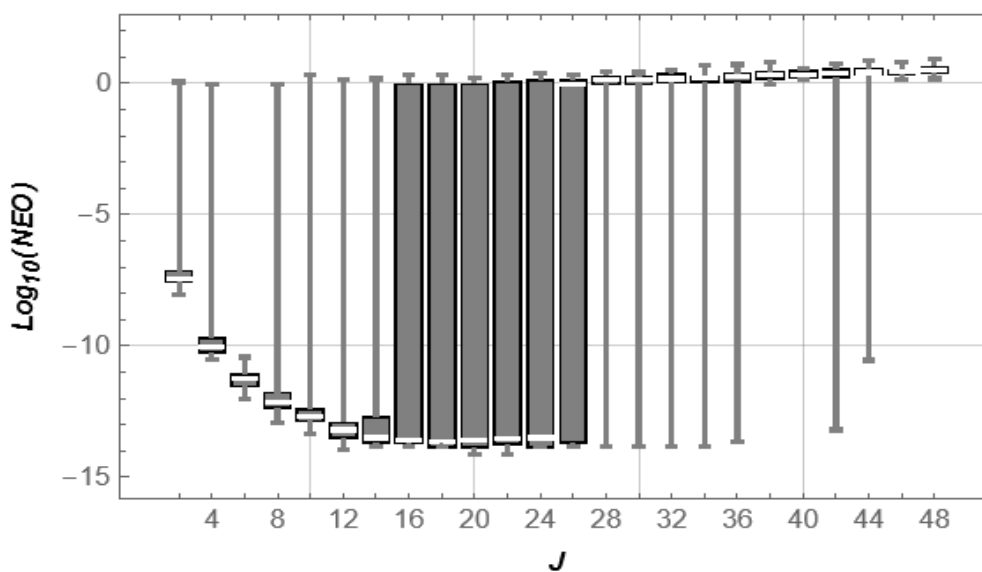
Спостерігаючи за діаграмами (рис. 6.4), можемо розділити тестові функції на два класи: для першого J сильно впливає на NEO ($f2, f10, f11$), для другого – спостерігається слабкий ($f18, f20$) або відсутній ($f9$) вплив J на ефективність роботи алгоритму $RSTC_{JK}$ -PSO. Таким чином, існує клас цільових функцій, які не чутливі до алгоритмів l-best.



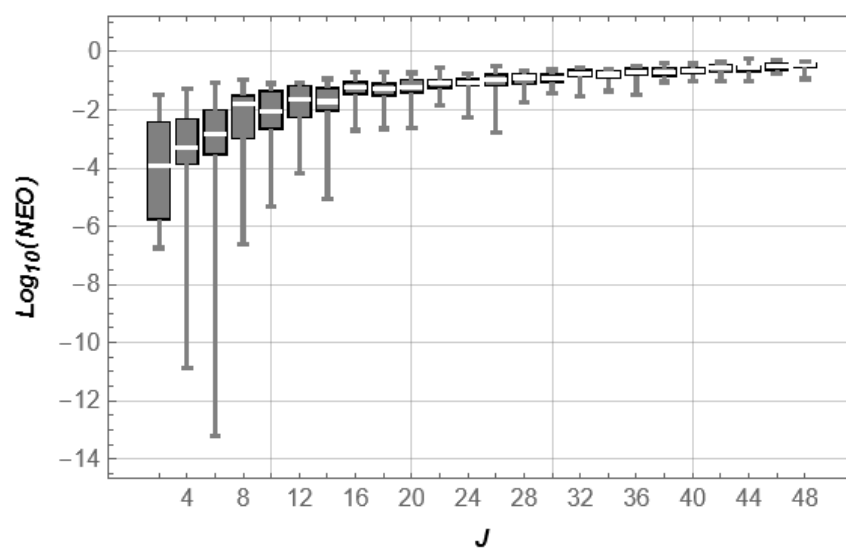
a)



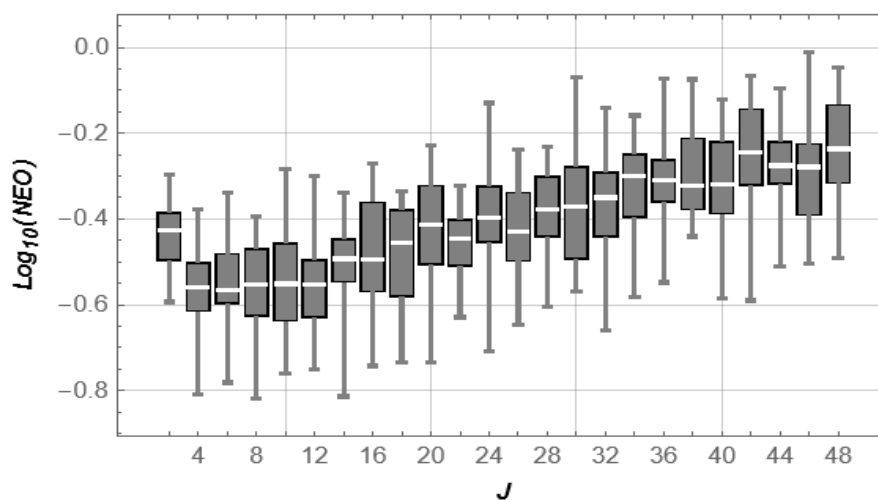
б)



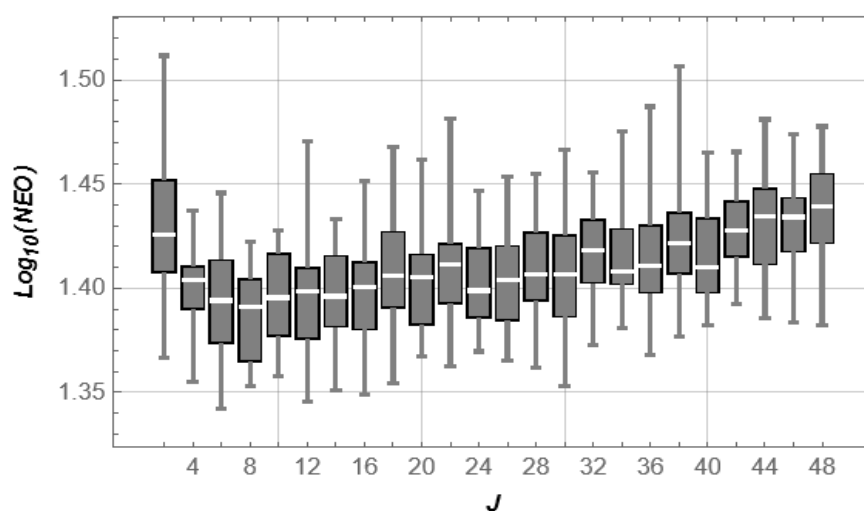
B)



г)



д)



е)

Рисунок 6.4 – Діаграми Бокса-Вішкера величин $\text{Log}_{10}(\text{NEO})$ для тестових функцій: а) f_2 ; б) f_9 ; в) f_{10} ; г) f_{11} ; д) f_{18} ; е) f_{20}

У даному дослідженні звернемо увагу на перший клас функцій. Враховуючи ці дані, можна сказати, що немає загальних рекомендацій щодо раціонального налаштування J . Наприклад, ефективна мінімізація f_2 вимагає $J=28\dots34$, $J=2$ для f_{10} та $J=4\dots12$ для f_{11} . Для всіх розглянутих варіантів $J \approx (0,04\dots0,68)K$. Таким чином, для мінімізації цих функцій немає раціональних підстав застосовувати g-best. Щоб знайти раціональний рівень зв'язності частинок (значення J), слід спробувати кілька пробних запусків алгоритмів з $J=0,04K$, $J=0,15K$, $J=0,25K$, $J=0,50K$ і $J=0,7K$. Однак слід мати на увазі, що отримані результати справедливі для $K=50$ і $D=30$. Для інших випадків рекомендації можуть відрізнятись.

6.5.4 Визначення впливу структури зв'язків частинок на ефективність RSTC_{J/K}-PSO

Тепер слід з'ясувати, чи існує будь-який вплив структури з'єднань частинок (наприклад, кільцевих або фон-нейманівських) на ефективність алгоритмів l-best. Порівняння *NEO* може дати відповідь. Щоб провести статистично достовірне дослідження, розширили набір контрольних показників. Додані цільові функції для мінімізації [2]: f_1 – сферична, f_3 – експоненціальна, f_4 – Швєфела, f_5 – Розенброка, f_6 – Растрігіна, f_7 – Гріванка, f_8 – Альпайна, f_{12} – Швєфела 2.2, f_{13} – Квінга, f_{14} – Саломона, f_{15} – Хі-Ше-Янга, f_{16} – Шуберта, f_{17} – Ріджа, f_{19} – Хімельблау. Медіанні значення, розраховані на 30 незалежних запусках алгоритмів, наведені в табл. 6.1.

Таблиця 6.1 – Медіани результатів незалежних пусків алгоритмів при оптимізації тестових функцій

Тестова функція	Алгоритм			
	Ring-PSO	RSTC _{3/50} -PSO	фон Неймана-PSO	RSTC _{5/50} -PSO
1	2	3	4	5
f_1	7,00E-13	7,88E-10	1,08E-20	8,15E-23
f_2	3,52E-12	1,01E-15	4,70E-21	3,38E-20

1	2	3	4	5
f_3	-1,00E+00	-1,00E+00	-1,00E+00	-1,00E+00
f_4	5,07E+00	1,38E+00	2,23E-01	1,62E-01
f_5	2,56E+01	2,48E+01	2,45E+01	2,43E+01
f_6	6,71E+01	3,83E+01	4,48E+01	4,83E+01
f_7	3,27E-12	5,61E-16	4,17E-22	4,71E-24
f_8	2,17E-02	8,34E-08	4,83E-09	3,28E-12
f_9	1,34E+02	1,24E+02	1,29E+02	1,14E+02
f_{10}	7,12E-07	1,92E-08	1,54E-11	2,06E-11
f_{11}	4,03E-05	6,46E-05	1,65E-04	9,14E-04
f_{12}	3,49E-08	5,29E-10	1,44E-11	8,69E-13
f_{13}	2,85E-05	3,43E-10	2,73E-13	1,42E-16
f_{14}	5,00E-01	4,00E-01	3,00E-01	5,00E-01
f_{15}	1,18E-11	1,35E-11	1,19E-11	1,88E-11
f_{16}	-2,39E+02	-2,34E+02	-2,47E+02	-2,45E+02
f_{17}	-5,00E+00	-5,00E+00	-5,00E+00	-5,00E+00
f_{18}	4,89E-01	3,03E-01	3,05E-01	2,89E-01
f_{19}	-2,12E+03	-2,10E+03	-2,12E+03	-2,12E+03
f_{20}	2,71E+01	2,54E+01	2,55E+01	2,44E+01

Щоб встановити статистично достовірну різницю, застосували знаковий критерій Вілкоксона. Порівнюються дві пари алгоритмів: Ring-PSO RSTC_{3/50}-PSO – перша пара; фон Неймана-PSO і RSTC_{5/50}-PSO – друга пара. Результати наведені в табл. 6.2.

Беручи до уваги розраховані цифри, можна бути впевненим у достатніх доказах для прийняття нульової гіпотези. Це означає, що немає різниці між порівнюваними алгоритмами з точки зору їх середньої продуктивності. Таким чином, можемо розглядати алгоритми PSO зі структурами зав'язків Ring-PSO і фон Неймана-PSO лише як окремі випадки RSTC_{JK}-PSO.

Таблиця 6.2 – Результати тесту Вілкоксона

Тестові функції	Порівнювані алгоритми	
	Ring-PSO та RSTC _{3/50} -PSO	фон Неймана-PSO та RSTC _{5/50} -PSO
f_1	-6	5
f_2	5	-6
f_3	-1	-1
f_4	16	14
f_5	14	15
f_6	20	-19
f_7	4	4
f_8	11	11
f_9	18	20
f_{10}	8	-8
f_{11}	-9	-12
f_{12}	7	10
f_{13}	10	7
f_{14}	12	-16
f_{15}	-3	-9
f_{16}	-17	-18
f_{17}	-1	-1
f_{18}	13	13
f_{19}	-19	-1
f_{20}	15	17
Додатна сума	153	116
Від'ємна сума	-56	-91
Статистика тесту	56	91
Граничне значення тестової величини при $\alpha=0,05$ та $N=20$	52	

Ring-PSO і фон Неймана-PSO вимагають спеціальних структур зв'язків частинок з точки зору порядкових номерів частинок. Проте випадкова ініціалізація (реалізована в алгоритмі RSTC_{J/K}-PSO) забезпечує майже таку ж саму ефективність. Слід підкреслити, що розглянуті алгоритми (Ring-PSO та фон Неймана-PSO) – це найпоширеніші l-best PSO. Для інших (зірка, піраміда, випадкова тощо) результати можуть відрізнятися і для того, щоб зробити більш загальні висновки необхідно провести додаткові дослідження.

Висновки до шостого розділу:

1. Аналіз даних, які отримані у ході виконання експериментів, показав сильні пошукові властивості алгоритмів PV-IA-PSO та PSO-Hybrid. Вони дозволили досить точно локалізувати глобальні мінімуми тестових функцій $f1-f4$, $f6-f8$, $f10-f17$. Розв'язання найскладніших оптимізаційних задач, які асоціюються із мінімізацією функцій $f5$, $f9$, $f18$, $f19$ та $f20$, при збільшенні кількості ітерацій у десять разів ($J=10000$) порівняно з першою серією експериментів ($J=1000$) не дозволяють значно покращити отримані результати і тому існують певні обмеження на застосування розроблених модифікацій PV-IA-PSO та PSO-Hybrid.
2. Для тестових функцій $f1$, $f2$, $f4$, $f6$, $f7$, $f8$, $f10$, $f12$ і $f15$ покращення знайдених розв'язків оптимізаційних задач із використанням вказаних алгоритмів є суттєвим. Наприклад, за показником *ind* ефективність алгоритму PV-IA-PSO збільшилась на 57,1%, у порівнянні з ефективністю алгоритму IA-PSO, хоча відмінність між обома алгоритмами у сенсі проведеної модифікації, є мізерною.
3. Розроблено загальний l-best алгоритм RSTC_{JK}-PSO. На першій ітерації RSTC_{JK}-PSO для кожної частинки встановлюється її сусідство (частинки, з якими вона буде обмінюватись інформацією). Він включає частинки $J-1$ і не змінюється під час роботи алгоритму. Дослідження RSTC_{JK}-PSO показало, що існує два класи цільових функцій, які чутливі і не чутливі до налаштування J . Для класу чутливих до l-best PSO функцій рекомендується виконати кілька пробних запусків алгоритму з $J=0,04K$, $J=0,15K$, $J=0,25K$, $J=0,50K$ і $J=0,7K$, щоб оцінити прийнятний J .
4. Доведено відсутність сильного впливу на роботу алгоритму фактору структури зв'язків між частинками. На ефективність роботи l-best PSO впливає лише значення J . Таким чином, добре відомі алгоритми Ring-PSO і фон Неймана-PSO є лише окремими частинними випадками RSTC_{JK}-PSO.

РОЗДІЛ 7

ЗАСТОСУВАННЯ PSO-АЛГОРИТМІВ ДЛЯ ЗАДАЧ ОПТИМАЛЬНОГО КЕРУВАННЯ ДИНАМІЧНИХ СИСТЕМ

Щоб довести здатність застосування розроблених алгоритмів до задач оптимального керування були розв'язані лінійні та нелінійні задачі. Методика досліджень полягала у багаторазовому розв'язанні задач оптимального керування, збором отриманих даних і проведенням аналізу.

Усі розв'язані задачі пов'язані з визначенням деякої кількості числових значень (від 2 до 31), які необхідні для побудови оптимального регулятора.

Кожну задачу розв'язували 50 разів за допомогою алгоритмів ME-D-PSO, R-Ring-PSO, IA-PSO, PV-IA-PSO, PSO-Hybrid. Перший з цих алгоритмів вже був застосований до деяких задач оптимального керування, які досліджувались у даному розділі. Отже було цікаво спостерігати за тим, як зміна алгоритму може вплинути на ефективність (з точки зору досягнутого мінімуму критерію) процесу розв'язання.

7.1 Постановка лінійних задач (налаштування ПІ-регуляторів)

Першим із підкласу розглянутих задач є оптимальне налаштування ПІ-регуляторів. У цих задачах необхідно визначити лише два параметри: інтегральний і пропорційний коефіцієнти підсилення. Їхні числові значення впливають на динаміку регулювання об'єктом і, як наслідок, на значення критерію мінімізації (вихідне значення цільової функції).

Усі елементи задач наведено в табл. 7.1. У першому ряду табл. 7.1 присутня передавальна функція об'єкта регулювання (ОР). Цей вид математичної моделі зазвичай використовується в задачах автоматизації, тому дотримуємося цієї форми представлення моделі. У другому рядку табл. 7.1 наведено деякі особливості вибраних ОР.

Для всіх задач задане значення (уставка) однакове, воно дорівнює одиниці ($r=1$).

Граничні умови для всіх задач наведені в четвертому рядку табл. 7.1. Вони схожі: на початку ОР характеризується станом спокою, а наприкінці регульована величина повинна вийти в окіл уставки, а вищі похідні регульованого параметру за часом повинні дорівнювати нулю.

Таблиця 7.1 – Задачі оптимального регулювання лінійними ОР за допомогою ПІ-регуляторів

Елемент задачі	ОР		
1	2	3	4
Передавальна функція	$G_1(s) = \frac{1}{(s+1)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}$	$G_2(s) = \frac{\omega_0^2}{(s+1)(s^2 + 2\xi\omega_0 s + \omega_0^2)}$	$G_3(s) = \frac{1}{(s+1)^3}$
Особливості	Ця система має чотири полюси, відстань між якими визначається параметром α . За малих значень α спостерігаються суттєві покращення якості регулювання при переході від ПІ- до ПІД-регулятора [96].	Використання ПІД-регуляторів для систем цього класу з низьким демпфуванням ξ не дозволяє отримати високі показники якості регулювання. ОР, які описані цими передавальними функціями, мають кращу якість регулювання при значному ω_0 . Якість регулювання можна підвищити за рахунок використання більш загальних структур регуляторів [96].	Ці системи дуже поширені. Для $n=1$ і 2 можна досягти високих показників якості регулювання за допомогою ПІ- або ПІД-регуляторів відповідно. Для великих значень n ОР поводитья як системи з великими запізненнями. Такі ОР протягом тривалого часу використовувалися виробниками контролерів для тестування [96].
Уставка	$r=1$		

Продовження табл. 7.1

1	2	3	4
Крайові умови	$\begin{cases} x(0) = \dot{x}(0) = \ddot{x}(0) = \ddot{x}(0) = 0; \\ x(t_s) = r; \dot{x}(t_s) = \ddot{x}(t_s) = \ddot{x}(t_s) = 0. \end{cases}$		$\begin{cases} x(0) = \dot{x}(0) = \ddot{x}(0) = 0; \\ x(t_s) = r; \dot{x}(t_s) = \ddot{x}(t_s) = 0. \end{cases}$
Допустимі значення кінцевих координат	$\begin{aligned} x_a &= 0,002; \\ \dot{x}_a &= 0,002; \\ \ddot{x}_a &= 0,002; \\ \ddot{x}_a &= 0,002 \end{aligned}$	$\begin{aligned} x_a &= 0,005; \\ \dot{x}_a &= 0,005; \\ \ddot{x}_a &= 0,005 \end{aligned}$	$\begin{aligned} x_a &= 0,002; \\ \dot{x}_a &= 0,002; \\ \ddot{x}_a &= 0,002 \end{aligned}$
Критерій оптимізації	$Cr = \frac{1}{u_{\max}} \sqrt{\frac{1}{t_s} \int_0^{t_s} u^2 dt}$	$Cr = \frac{t_s}{T}$	$Cr = \frac{1}{r} \sqrt{\frac{1}{t_s} \int_0^{t_s} e^2 dt}$
Обмеження	$\max(u_i) \leq u_{\max}$		
Термінальний критерій Ter	$Ter = \begin{cases} \frac{ x_{t_s/\Delta t} }{x_a} + \frac{ \dot{x}_{t_s/\Delta t} }{\dot{x}_a} + \frac{ \ddot{x}_{t_s/\Delta t} }{\ddot{x}_a} + \frac{ \ddot{x}_{t_s/\Delta t} }{\ddot{x}_a} & \text{якщо} \\ x_{t_s/\Delta t} > x_a \vee \dot{x}_{t_s/\Delta t} > \dot{x}_a \vee \ddot{x}_{t_s/\Delta t} > \ddot{x}_a \vee \ddot{x}_{t_s/\Delta t} > \ddot{x}_a; \\ 0, & \text{якщо } x_{t_s/\Delta t} \leq x_a \wedge \dot{x}_{t_s/\Delta t} \leq \dot{x}_a \wedge \ddot{x}_{t_s/\Delta t} \leq \ddot{x}_a \wedge \ddot{x}_{t_s/\Delta t} \leq \ddot{x}_a. \end{cases}$		$Ter = \begin{cases} \frac{ x_{t_s/\Delta t} }{x_a} + \frac{ \dot{x}_{t_s/\Delta t} }{\dot{x}_a} + \frac{ \ddot{x}_{t_s/\Delta t} }{\ddot{x}_a} & \text{якщо} \\ x_{t_s/\Delta t} > x_a \vee \dot{x}_{t_s/\Delta t} > \dot{x}_a \vee \ddot{x}_{t_s/\Delta t} > \ddot{x}_a; \\ 0, & \text{якщо } x_{t_s/\Delta t} \leq x_a \wedge \dot{x}_{t_s/\Delta t} \leq \dot{x}_a \wedge \\ \wedge \ddot{x}_{t_s/\Delta t} \leq \ddot{x}_a. \end{cases}$

Продовження табл. 7.1

1	2	3	4
Перерегулювання OS	$OS = \begin{cases} \frac{x_{\max} - r}{x_{\max}}, & \text{якщо } \max(x_i) > r; \\ 0, & \text{якщо } \max(x_i) \leq r, \end{cases}$ $i \in \overline{(1, t_s / \Delta t)}$		
Узагальнений критерій $GenCr$	$GenCr = 10^8 Ter + 10^2 OS + Cr$		
Параметри	$\alpha=0,2; u_{\max}=2$	$\xi=0,1; \omega_0=1,0; u_{\max}=2$	$n=3; u_{\max}=1,2$
Область задання коефіцієнтів ПІ- регулятора	$K_p \in [10^{-5}, 10^2]; T_i \in [10^{-5}, 10^2]$		
Тривалість моделювання T, c	4	60	10
Крок за часом $\Delta t, c$	0,005	0,05	0,05

Точне виконання кінцевих умов є досить складною проблемою, тому замінюємо нульові значення на деякі прийнятні (малі) величини для всіх кінцевих координат. Вони представлені в п'ятому рядку табл. 7.1.

У наступному рядку наведено критерії для мінімізації. Для різних ОР були обрані різні критерії для того, щоб з'ясувати, як алгоритми можуть впоратися із задачами мінімізації цих критеріїв. Крім того, вихідне значення ПІ-регулятора (функція регулювання u) має бути обмежене. Беручи це до уваги, можна написати повний вираз вихідного значення ПІ-регулятора:

$$u_{out} = \begin{cases} K_p(r-x) + \frac{1}{T_i} \int (r-x) dt, & \text{якщо } u_{max} \geq K_p(r-x) + \frac{1}{T_i} \int (r-x) dt \geq 0; \\ u_{max}, & \text{якщо } u_{max} < K_p(r-x) + \frac{1}{T_i} \int (r-x) dt; \\ 0, & \text{якщо } K_p(r-x) + \frac{1}{T_i} \int (r-x) dt < 0, \end{cases} \quad (7.1)$$

де K_p і T_i – пропорційні та інтегральні коефіцієнти підсилення ПІ-регулятора; u_{max} – максимальний вихідний сигнал ПІ-регулятора; r – уставка; x – регульована змінна. У восьмому рядку табл. 7.1 описаний термінальний критерій Ter . Оскільки другий і третій ОР третього порядку для них Ter однакові. Для кожної із задач обов'язковою умовою є відсутність перерегулювання OS (дев'ятий рядок табл. 2.1).

Для виконання всіх вказаних умов використовується умова мінімізації узагальненого критерію $GenCr$. Усі доданки $GenCr$ є безрозмірними. Узагальнений критерій $GenCr$ має специфічну топологію: він придатний для застосування алгоритмів на основі PSO [97].

Параметри ОР наведено в одинадцятому рядку табл. 7.1. У дванадцятому рядку представлені області значень, де здійснювався пошук чисельних значень K_p і T_i . У тринадцятому рядку табл. 7.1 наведено тривалість моделювання для кожної задачі T , а в останньому – часовий крок моделювання Δt .

7.2 Постановка нелінійних задач

У цьому пункті розглянемо певні задачі керування, де ОР можуть бути описані системами нелінійних диференціальних рівнянь. Оскільки відповідні задачі важко розв'язати, вони утворюють особливий клас задач керування, де для пошуку прийняттого розв'язку необхідно застосувати деякі нелінійні

оператори. Такі підходи отримали розвиток у роботі [60]. Для розв'язання цих задач використовувався метод ME-D-PSO [69]. У цьому пункті зосередимося не на розв'язанні задач, а на ефективності застосування для цього PSO-алгоритмів.

Дамо коротке пояснення сутності задач керування [60]. Перша з них – це задача керування рухом перевернутого маятника. Суть задачі зводиться до того, що необхідно побудувати керування рухом маховика таким чином, щоб маятникова ланка стабілізувалась у верхньому (нестійкому) положенні рівноваги.

Друга задача аналогічна: необхідно знайти оптимальне керування рухомим візком, при якому ланка маятника перейде у верхнє (вертикальне) нестійке положення рівноваги, а візок буде знаходитись у стані спокою і займатиме положення, з якого він починав рух.

Третя задача полягає в наступному: необхідно знайти керування маятником Фурути, яке стабілізує ланку маятника у верхньому (нестійкому) стані спокою, а колону необхідно повернути у стан, з якого вона починала рух.

У табл. 7.2 наведено відповідні динамічні та математичні моделі. Крім того, п'ятий рядок табл. 7.2 відображає описані крайові умови задач (початкові умови відповідають стану спокою систем; маятникові ланки знаходяться в нижніх, тобто стійких положеннях рівноваги).

Для виконання кінцевих умов розроблено методику, яка дозволяє знайти керування, яке, в свою чергу, переводить систему з початкового стану у кінцевий з деякими допустимими відхиленнями (для всіх фазових координат). Останні наведені в шостому рядку табл. 7.2.

Щоб завершити постановку задач керування, слід записати критерій мінімізації Cr та обмеження. Вони наведені в сьомому та восьмому рядках табл. 7.2. Критерій Cr для всіх задач пропорційний середньоквадратичному (RMS) значенню функції керування (крутного моменту або сили) і часу регулювання t_s . Ці значення є небажаними, оскільки середньоквадратичне значення керування пропорційне втратам енергії в електроприводі, а час регулювання відображає період часу, який необхідний для виконання кінцевих умов руху тієї чи іншої динамічної системи.

Відповідно до розробленої методики кінцеві умови слід замінити на термінальний критерій Ter (для досліджуваних задач Ter наведено в дев'ятому рядку табл. 7.2). Для першої і другої задач перерегулювання OS фазової

координати необхідно мінімізувати. Формули, що відображають цю вимогу, наведено в десятому рядку табл. 7.2.

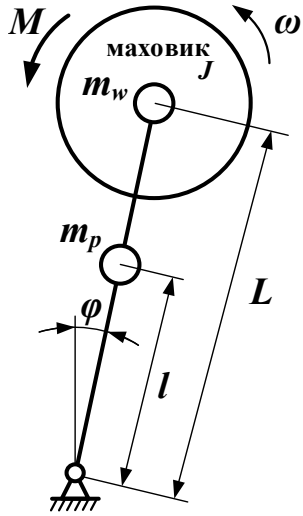
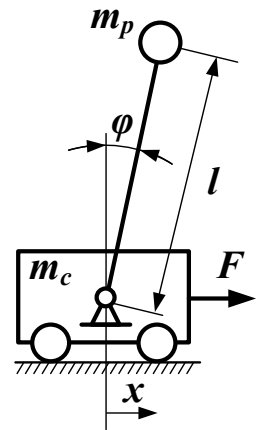
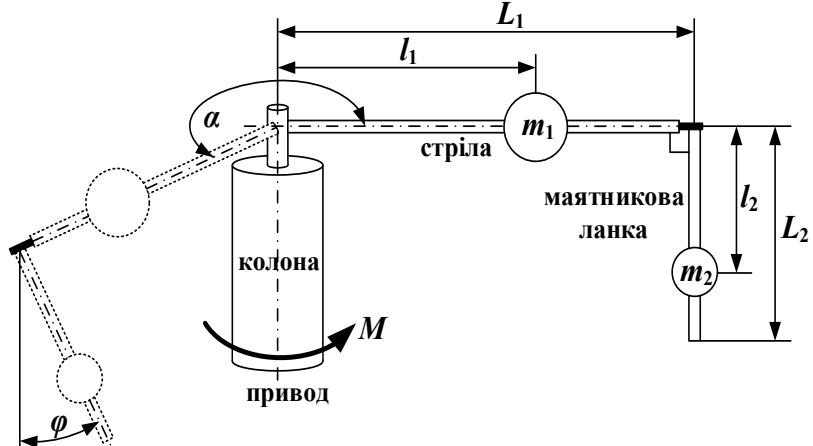
На практиці реалізація оптимального керування пов'язана з керуванням приводним механізмом. У даному дослідженні мається на увазі електропривод. З метою обмеження напруги на обмотках приводу регулювання не повинно бути швидко змінним. Швидка зміна функції керування є небажаною. Величину, яка показує кількість перевищень граничного значення швидкості зміни керування CG , необхідно мінімізувати. Вона наведена в одинадцятому рядку табл. 7.2. Введення у розгляд такого небажаного показника є подальшим розвитком методології розв'язання задач оптимального керування, яка розроблена у монографії [60].

Для формування загальної функції мінімізації $GenCr$ у дванадцятому рядку табл. 7.2 наведено відповідні формули. Вони дозволяють врахувати всі необхідні умови задачі в одному виразі. Біля кожного доданку $GenCr$ стоять коефіцієнти, які дозволяють сформулювати топологію цільової функції «яма в ямі». Така топологія досить складна, оскільки містить окремі топологічні властивості доданків Ter , OS , CG та Cr , які проявляються на різних етапах мінімізації $GenCr$. Усі доданки $GenCr$ є безрозмірними величинами.

Усі постійні параметри, які використовуються в розрахунках, наведені в табл. 7.2: у тринадцятому рядку – параметри динамічної системи; у чотирнадцятому рядку – коефіцієнт підсилення виходу штучної нейронної мережі (ШНМ) (оскільки вихідне значення ШНМ змінюється в обмеженому діапазоні, його слід збільшити шляхом використання відповідного коефіцієнта); у п'ятнадцятому рядку наведено тривалість моделювання динаміки регулювання системи, а у шістнадцятому – часовий крок (крок чисельного інтегрування). Наведені значення досить малі, щоб забезпечити виконання розрахунків без значних чисельних похибок.

Для тренування ШНМ було використано парадигму „з підкріпленням”. Тобто всі значення $GenCr$ були отримані у результаті визначення руху тієї чи іншої динамічної системи під дією певного керування. У свою чергу керування формувалось на виході ШНМ. У такий спосіб функція керування пов'язана зі значенням $GenCr$.

Таблиця 7.2 Задачі оптимального керування нелінійними ОР

Елемент задачі	ОР		
	Перевернутий маятник з маховиком	Перевернутий маятник з рухомою точкою підвісу	Маятник Фурути
1	2	3	4
Динамічна модель			
Математична модель	$\begin{cases} \alpha_1 \ddot{\varphi} + J \dot{\omega} = \alpha_2 \sin(\varphi); \\ J(\ddot{\varphi} + \dot{\omega}) = M \end{cases}$	$\begin{cases} (m_c + m_p) \ddot{x} + m_p l \times \\ \times (\ddot{\varphi} \cos \varphi - \dot{\varphi}^2 \sin \varphi) = F; \\ \ddot{x} \cos \varphi + l \ddot{\varphi} + g \sin \varphi = 0 \end{cases}$	$\begin{cases} \ddot{\alpha}(J_0 + J_{21} \sin^2(\varphi)) + \dot{\varphi} m_2 L_1 l_2 \cos(\varphi) - \\ - m_2 L_1 l_2 \sin(\varphi) \dot{\varphi}^2 + \dot{\alpha} \dot{\varphi} J_{21} \sin(2\varphi) = M; \\ \ddot{\alpha} m_2 L_1 l_2 \cos(\varphi) + \ddot{\varphi} J_{21} - \\ - \frac{1}{2} \dot{\varphi}^2 J_{21} \sin(2\varphi) g m_2 l_2 \sin(\varphi) = 0. \end{cases}$

Продовження табл 7.2

1	2	3	4
<p>Пояснення позначень</p>	<p>φ – узагальнена координата маятника; ω – кутова швидкість маховика маятника; l – відстань від центра мас маятника до осі його обертання; L – еквівалентна довжина маятника; J – момент інерції маховика і ротора двигуна; M – електромагнітний момент, створюваний двигуном; α_1 і α_2 – коефіцієнти, які визначаються з таких виразів: $\alpha_1 = J_p + m_w L^2$, $\alpha_2 = (m_p l + m_w L)g$; m_p – маса маятника; m_w – маса маховика та електродвигуна; J_p – момент інерції маятника</p>	<p>x – узагальнена координата візка; φ – узагальнена координата маятника; l – еквівалентна довжина маятника; m_p і m_c – зведені маси вантажу та візка відповідно; F – рушійна сила, що діє на візок</p>	<p>φ, α – узагальнені координати системи; M – рушійний момент, створюваний двигуном; L_1 і L_2 – довжини відповідно стріли і маятничкової ланки відносно осей їх обертання; l_1 і l_2 – відстань від центра мас m_1 до осі обертання стріли і відстань від центра мас m_2 до осі обертання маятничкової ланки відповідно; m_1 і m_2 – зведені маси, що приєднані до стріли та маятничкової ланки відповідно; J_0 – зведений момент інерції колони відносно осі її обертання за умови, що ланка маятника знаходиться в нижньому (стійкому) положенні рівноваги; J_1 – сумарний момент інерції колони і двигуна відносно осі їх обертання; J_2 – зведений момент інерції ланки маятника відносно осі його обертання; J_{21} – зведений момент інерції ланки маятника відносно осі його обертання.</p>

Продовження табл. 7.2

1	2	3	4
Загальні позначення	g – прискорення вільного падіння; t_s – тривалість регулювання; Ω – частота власних коливань системи		
Крайові умови	$\begin{cases} \varphi(0) = \pi; \dot{\varphi}(0) = \omega(0) = 0; \\ \varphi(t_s) = \dot{\varphi}(t_s) = \omega(t_s) = 0. \end{cases}$	$\begin{cases} \varphi(0) = \pi; \dot{\varphi}(0) = x(0) = \dot{x}(0) = 0; \\ \varphi(t_s) = \dot{\varphi}(t_s) = x(t_s) = \dot{x}(t_s) = 0. \end{cases}$	$\begin{cases} \varphi(0) = \pi; \dot{\varphi}(0) = \alpha(0) = \dot{\alpha}(0) = 0; \\ \varphi(t_s) = \dot{\varphi}(t_s) = \alpha(t_s) = \dot{\alpha}(t_s) = 0. \end{cases}$
Допустимі значення кінцевих координат	$\varphi_a = 0,02$ рад; $\dot{\varphi}_a = 0,02$ рад/с; $\omega_a = 0,1$ рад/с	$\varphi_a = 0,005$ рад; $\dot{\varphi}_a = 0,005$ рад/с; $x_a = 0,005$ м; $\dot{x}_a = 0,005$ м/с	$\varphi_a = 0,005$ рад; $\dot{\varphi}_a = 0,005$ рад/с; $\alpha_a = 0,005$ м; $\dot{\alpha}_a = 0,005$ м/с
Критерій мінімізації Cr	$Cr = 2\pi\Omega t_s + 10t_s^{-1}M_{\max}^{-1}\Delta t \sum_{i=1}^{t_s/\Delta t} M_i^2$	$Cr = 2\pi\Omega t_s + 10t_s^{-1}F_{\max}^{-1}\Delta t \sum_{i=1}^{t_s/\Delta t} F_i^2$	$Cr = 2\pi\Omega t_s + 10t_s^{-1}M_{\max}^{-1}\Delta t \sum_{i=1}^{t_s/\Delta t} M_i^2$
Обмеження	$\max(M_i) \leq M_{\max};$ $\max(\omega_i) \leq \omega_{\max}$	$\max(F_i) \leq F_{\max};$ $\max(x_i) \leq x_{\max}$	$\max(M_i) \leq M_{\max}$

Продовження табл. 7.2

1	2	3	4
Термінальний критерій Ter	$Ter = \begin{cases} \frac{ \varphi_{t_s/\Delta t} }{\varphi_a} + \frac{ \dot{\varphi}_{t_s/\Delta t} }{\dot{\varphi}_a} + \frac{ \omega_{t_s/\Delta t} }{\omega_a}, & \text{якщо} \\ \varphi_{t_s/\Delta t} > \varphi_a \vee \dot{\varphi}_{t_s/\Delta t} > \dot{\varphi}_a \vee \\ \vee \omega_{t_s/\Delta t} > \omega_a; & \\ 0, & \text{якщо} \\ \varphi_{t_s/\Delta t} \leq \varphi_a \wedge \dot{\varphi}_{t_s/\Delta t} \leq \dot{\varphi}_a \wedge \\ \wedge \omega_{t_s/\Delta t} \leq \omega_a. & \end{cases}$	$Ter = \begin{cases} \frac{ \varphi_{t_s/\Delta t} }{\varphi_a} + \frac{ \dot{\varphi}_{t_s/\Delta t} }{\dot{\varphi}_a} + \frac{ x_{t_s/\Delta t} }{x_a} + \frac{ \dot{x}_{t_s/\Delta t} }{\dot{x}_a}, & \\ \text{якщо } \varphi_{t_s/\Delta t} > \varphi_a \vee \dot{\varphi}_{t_s/\Delta t} > \dot{\varphi}_a \vee \\ \vee x_{t_s/\Delta t} > x_a \vee \dot{x}_{t_s/\Delta t} > \dot{x}_a; & \\ 0, & \text{якщо } \varphi_{t_s/\Delta t} \leq \varphi_a \wedge \\ \wedge \dot{\varphi}_{t_s/\Delta t} \leq \dot{\varphi}_a \wedge & \\ \wedge x_{t_s/\Delta t} \leq x_a \wedge \dot{x}_{t_s/\Delta t} \leq \dot{x}_a. & \end{cases}$	$Ter = \begin{cases} \frac{ \varphi_{t_s/\Delta t} }{\varphi_a} + \frac{ \dot{\varphi}_{t_s/\Delta t} }{\dot{\varphi}_a} + \frac{ \alpha_{t_s/\Delta t} }{\alpha_a} + \frac{ \dot{\alpha}_{t_s/\Delta t} }{\dot{\alpha}_a}, & \text{якщо} \\ \varphi_{t_s/\Delta t} > \varphi_a \vee \dot{\varphi}_{t_s/\Delta t} > \dot{\varphi}_a \vee \alpha_{t_s/\Delta t} > \alpha_a \vee \\ \vee \dot{\alpha}_{t_s/\Delta t} > \dot{\alpha}_a; & \\ 0, & \text{якщо } \varphi_{t_s/\Delta t} \leq \varphi_a \wedge \dot{\varphi}_{t_s/\Delta t} \leq \dot{\varphi}_a \wedge \\ \wedge \alpha_{t_s/\Delta t} \leq \alpha_a \wedge \dot{\alpha}_{t_s/\Delta t} \leq \dot{\alpha}_a. & \end{cases}$
Перерегулювання OS	$OS = \begin{cases} \frac{\omega_{\max} - \max(\omega_i)}{\omega_{\max}}, & \text{якщо} \\ \max(\omega_i) > \omega_{\max}; & \\ 0, & \text{якщо } \max(\omega_i) \leq \omega_{\max}, \end{cases}$ $i \in \overline{(1, t_s/\Delta t)}$	$OS = \begin{cases} \frac{x_{\max} - \max(x_i)}{x_{\max}}, & \text{якщо} \\ \max(x_i) > x_{\max}; & \\ 0, & \text{якщо } \max(x_i) \leq x_{\max}, \end{cases}$ $i \in \overline{(1, t_s/\Delta t)}$	-

Продовження табл. 7.2

1	2	3	4
Кількість недопустимих перевищень швидкості керування	$CG = \begin{cases} CG+1, & \text{якщо } \frac{ M_i - M_{i+1} }{\Delta t} > \dot{M}_{\max}; \\ CG, & \text{якщо } \frac{ M_i - M_{i+1} }{\Delta t} \leq \dot{M}_{\max}, \end{cases}$ $i \in \overline{(1, t_s / \Delta t - 1)}$	$CG = \begin{cases} CG+1, & \text{якщо } \frac{ F_i - F_{i+1} }{\Delta t} > \dot{F}_{\max}; \\ CG, & \text{якщо } \frac{ F_i - F_{i+1} }{\Delta t} \leq \dot{F}_{\max}, \end{cases}$ $i \in \overline{(1, t_s / \Delta t - 1)}$	$CG = \begin{cases} CG+1, & \text{якщо } \frac{ M_i - M_{i+1} }{\Delta t} > \dot{M}_{\max}; \\ CG, & \text{якщо } \frac{ M_i - M_{i+1} }{\Delta t} \leq \dot{M}_{\max}, \end{cases}$ $i \in \overline{(1, t_s / \Delta t - 1)}$
Узагальнений критерій	$GenCr = 10^8 Ter + 10^4 OS + 10^2 CG + Cr$		
Параметри	$L = 2,9 \cdot 10^{-1} \text{ м}, l = 2,6 \cdot 10^{-1} \text{ м},$ $J = 1,5 \cdot 10^{-3} \text{ кгм}^2, J_p = 4,2 \cdot 10^{-2} \text{ кгм}^2,$ $m_p = 3,0 \cdot 10^{-1} \text{ кг}, m_w = 2,8 \cdot 10^{-1} \text{ кг},$ $M_{\max} = 6,0 \cdot 10^{-1} \text{ Нм}; \dot{M}_{\max} = 2,0 \cdot 10^0$ $\text{Нм/с}; \omega_{\max} = 15 \text{ рад/с}.$	$l = 5,0 \cdot 10^{-1} \text{ м}, m_p = 3,0 \cdot 10^{-1} \text{ кг},$ $m_c = 2,0 \cdot 10^{-1} \text{ кг}; F_{\max} = 5,0 \text{ Н};$ $x_{\max} = 0,5 \text{ м}; \dot{F}_{\max} = 0,5 \text{ Н/с}.$	$L_1 = 2,78 \cdot 10^{-1} \text{ м}, L_2 = 3,0 \cdot 10^{-1} \text{ м}, l_1 = 1,5 \cdot 10^{-1} \text{ м}, l_2 = 1,48 \cdot 10^{-1}$ $\text{ м}, m_1 = 3,0 \cdot 10^{-1} \text{ кг}, m_2 = 7,5 \cdot 10^{-2} \text{ кг}, J_1 = 2,48 \cdot 10^{-2} \text{ кг} \cdot \text{м}^2,$ $J_2 = 3,86 \cdot 10^{-3} \text{ кг} \cdot \text{м}^2; M_{\max} = 1,0 \cdot 10^0 \text{ Нм}.$
Підсилення виходу ШНМ	10	5	1
Тривалість моделювання, с	2	8	6
Крок часу, с	0,01	0,025	0,02

7.3 Розробка оператора керування (структура ШНМ)

Лінійні регулятори (наприклад, ПІ, які описані в пункті 7.1) можуть бути застосовані лише для обмеженого класу ОР. Для нелінійних систем (тобто тих, що описані в пункті 7.2) потрібні більш досконалі оператори керування. Один із можливих шляхів вирішення задачі оптимального керування пов'язаний із застосуванням ШНМ. Вони використовуються для формування сигналу керування через входні сигнали. Останні є сигналами про поточні фазові координати ОР.

Розробку ШНМ можна розглядати як двоетапну процедуру: встановлення структури та навчання (тренування) ШНМ. Друга процедура зводиться до розв'язання задачі оптимізації, де будуть задіяні порівнювані PSO-алгоритми. Відповідні результати будуть наведені в наступних пунктах.

Основна мета цього пункту – розкрити структуру ШНМ і виконати опис всіх вагомих елементів ШНМ.

Будемо розглядати ШНМ прямого поширення лише з одним прихованим шаром із п'ятьма нейронами у ньому (рис. 7.1). У такій ШНМ вихідним є шар із одним нейроном. У входньому шарі ШНМ не обробляються сигнали, він поширює їх на прихований шар.

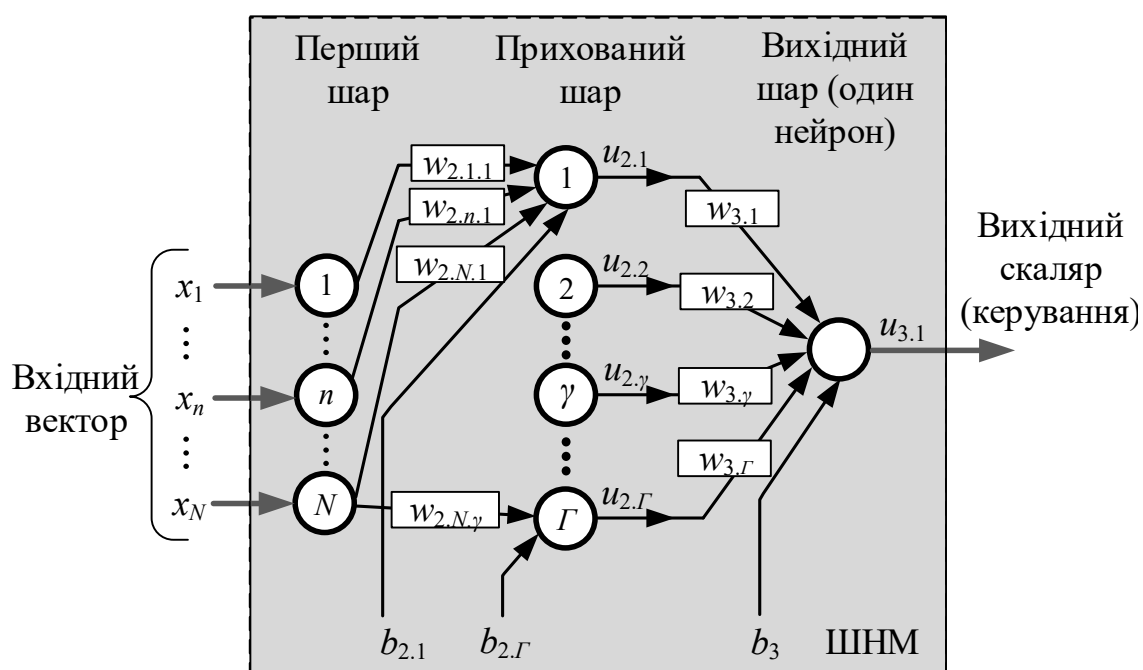


Рисунок 7.1 – ШНМ прямого поширення з одним прихованим шаром нейронів (показано не всі ваги та біаси)

Для всіх задач структура ШНМ однакова – різні лише числові значення ваг і біасів.

З урахуванням схеми (рис. 7.1) можна побудувати математичну модель ШНМ (вираз, що описує функціональну залежність вихідних і вхідних величин):

$$\begin{cases} u_{3.1} = \arctan \left(\sum_{\gamma=1}^{\Gamma} u_{2.\gamma} w_{3.\gamma} + b_3 \right); \\ u_{2.\gamma} = \arctan \left(\sum_{n=1}^N x_n w_{2.n.\gamma} + b_{2.\gamma} \right), \gamma \in (\overline{1, \Gamma}), \end{cases} \quad (7.2)$$

де $u_{3.1}$ – вихідний сигнал ШНМ; $u_{2.\gamma}$ – вихідний сигнал γ -того нейрона прихованого шару; x_n – n -та фазова координата ОР; N – загальна кількість фазових координат (порядок ОР); Γ – кількість нейронів у прихованому шарі; b_3 – біас нейрона вихідного шару; $b_{2.\gamma}$ – біас γ -того нейрона прихованого шару; $w_{3.\gamma}$ – вага зв'язку між нейроном вихідного шару та γ -тим нейроном прихованого шару; $w_{2.n.\gamma}$ – вага зв'язку між n -м нейроном вхідного шару та γ -м нейроном прихованого шару.

Як можна помітити з виразу (7.2), функція активації нейронів – арктангенс.

Задачі, описані в пункті 7.2, мають різну кількість фазових координат: три для першої ($N=3$), і чотири для другої та третьої ($N=4$). Таким чином, навчання (тренування) ШНМ потребує встановлення різної кількості параметрів (ваги та біаси). Для першої задачі їх загальна кількість дорівнює 26, для решти задач – 31.

7.4 Результати розв'язання задач оптимального керування

7.4.1 Лінійні задачі

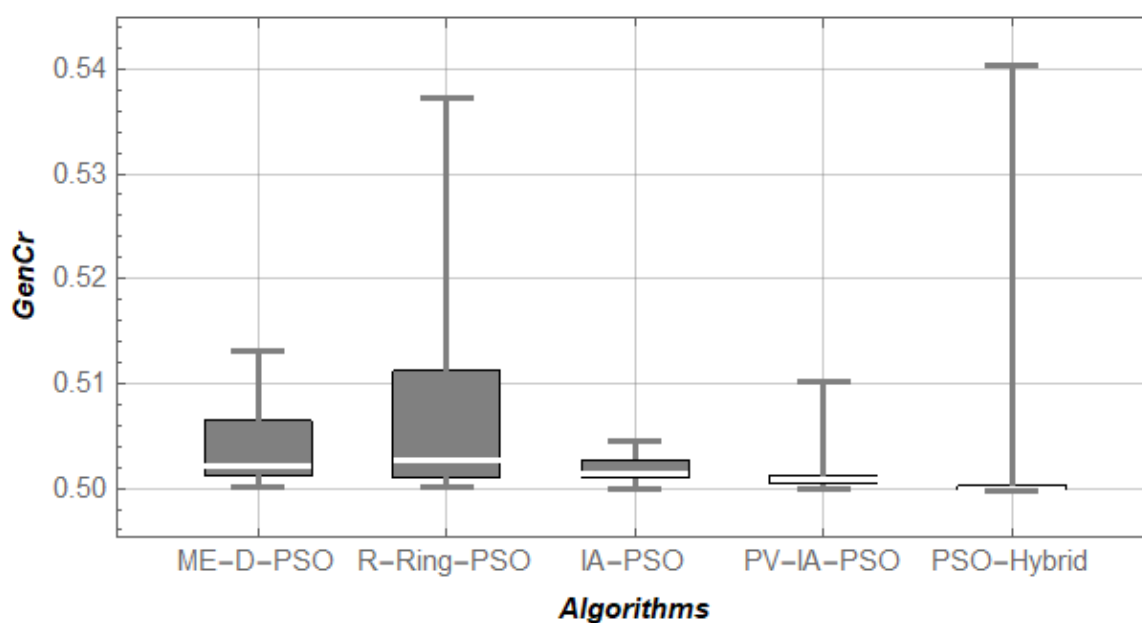
Дані, що отримані в результаті чисельних експериментів (оптимальне налаштування ПІ-регулятора), представлені нижче (рис. 7.2). Наведені діаграми відображають лише найбільш успішні результати, тобто ті, що відповідають допустимим значенням кінцевих координат (термінальний критерій Ter для всіх представлених нижче варіантів дорівнює нулю). Загальна кількість найбільш вдалих розв'язань задач наведена в табл. 7.3.

Для другої задачі алгоритм застосування алгоритму PV-IA-PSO дозволили забезпечити умову $Ter=0$ для всіх незалежних запусків алгоритму.

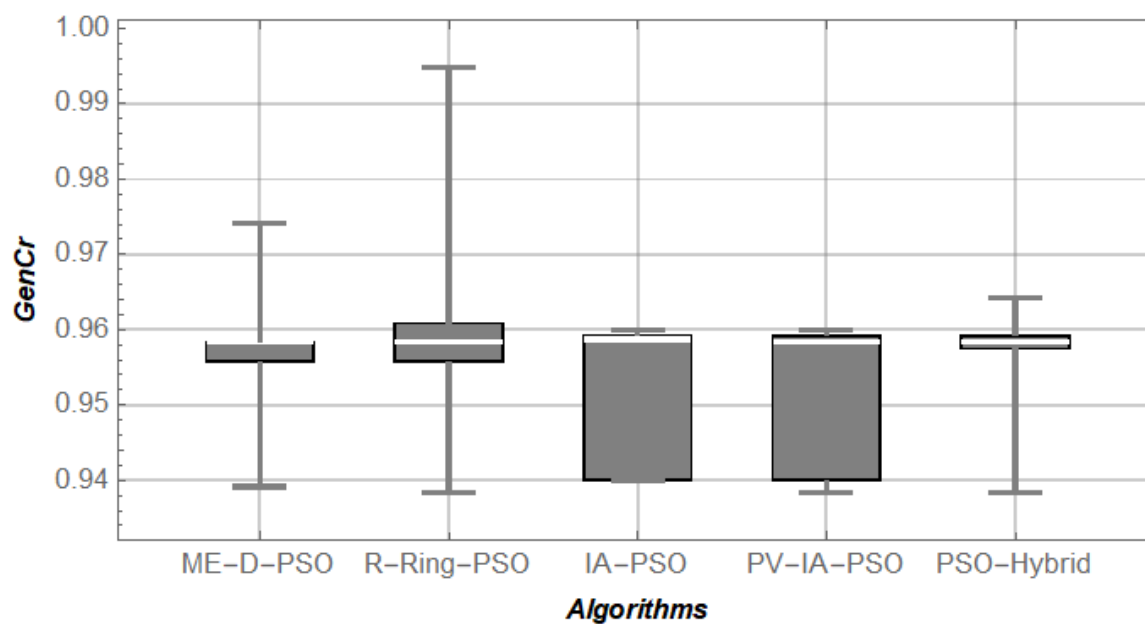
Другим за кількістю найуспішніших розв'язань задач є алгоритм R-Ring-PSO, третім – PSO-Hybrid.

Таблиця 7.3 Кількість найуспішніших розв'язань задач
(випадки, коли алгоритм досяг поверхні критерію Cr)

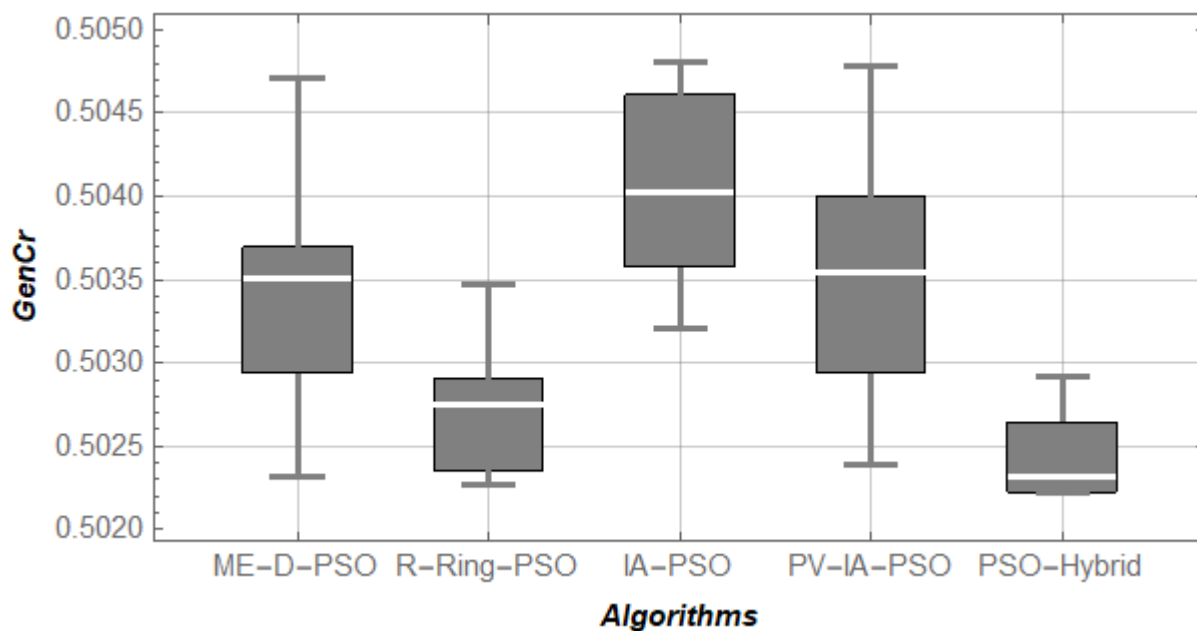
Алгоритми	Передавальна функція OP			Всього
	$G_1(s)$	$G_2(s)$	$G_3(s)$	
ME-D-PSO	38	37	30	105
R-Ring-PSO	42	43	38	123
IA-PSO	43	10	10	63
PV-IA-PSO	45	50	38	133
PSO-Hybrid	40	30	42	112



a)



б)



в)

Рисунок 7.2 Діаграми Бокса-Вішкера, які побудовані за результатами незалежних запусків алгоритмів для задач оптимального керування ОР, що описуються передавальними функціями: а) $G_1(s)$; б) $G_2(s)$; в) $G_3(s)$

Наведені діаграми (рис. 7.2) підтверджують рекомендації застосування PV-IA-PSO для розглянутих задач оптимізації. Дійсно, для першої задачі (де ОР, описаний передавальною функцією $G_1(s)$) PV-IA-PSO дає найкращі (за всіма значеннями критеріїв) результати (рис. 7.2, а). Однак PSO-Hybrid трохи кращий за середнім значенням. Якщо є можливість запустити PSO-алгоритм кілька разів, слід вибрати PSO-Hybrid, якщо такої можливості не існує, то варто надати перевагу застосуванню алгоритма PV-IA-PSO.

Для другої задачі (пов'язаної з регулюванням ОР з передавальною функцією $G_2(s)$ і критерієм оптимізації тривалості регулювання) PV-IA-PSO та IA-PSO показують дуже добрі результати. Серед усіх порівнюваних алгоритмів PV-IA-PSO є найкращим.

Для третьої задачі PSO-Hybrid є найкращим алгоритмом, R-Ring-PSO трохи гірший за отриманими даними. Алгоритм PV-IA-PSO для цієї задачі проявив високі пошукові властивості.

Тут слід зауважити, що різниця в досягнутих значеннях критеріїв не є настільки значимою. Для першої задачі найбільше значення критерію лише на 8,01% більше, ніж для гіршого, для другої задачі така різниця становить 6,08%, а для третьої – 3,07%. Таким чином, ефективність для всіх алгоритмів майже однакова і всі вони більшою чи меншою мірою підходять для розв'язання такого роду задач оптимізації.

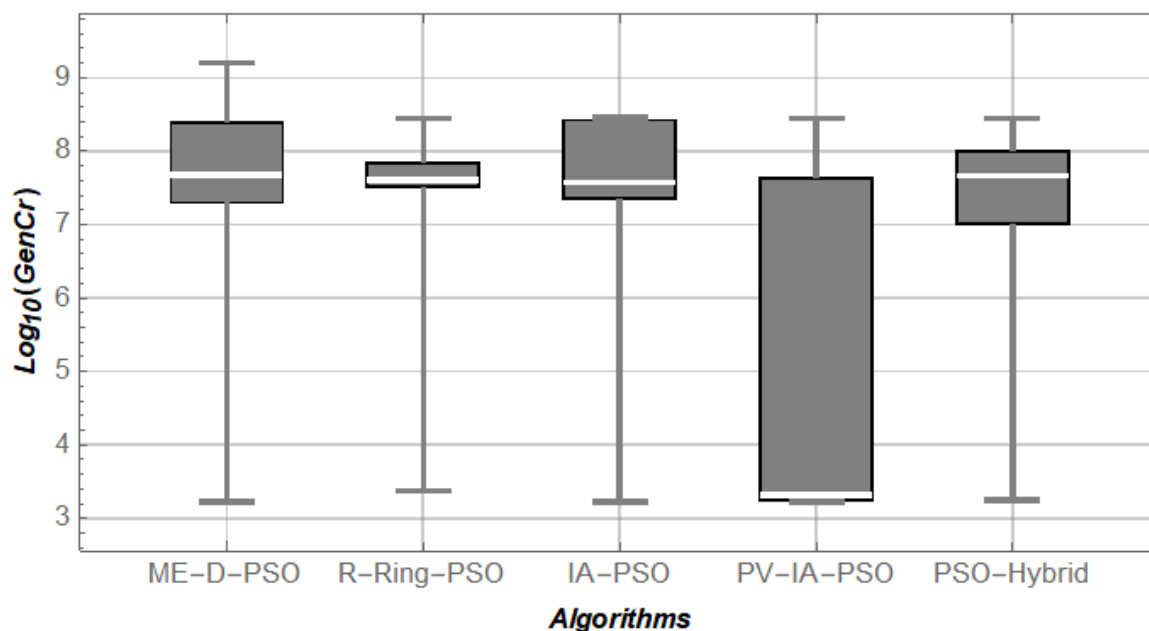
Підсумовуючи все, можемо стверджувати, що PV-IA-PSO є найкращим алгоритмом для оптимального налаштування ПІ-регуляторів, оскільки він може з високою ймовірністю забезпечити прийнятні значення кінцевих координат і мінімізувати різного роду критерії.

Графіки зміни регульованої змінної та функції регулювання для найкращих знайдених результатів наведені в Додатку Д (рис. Д.1). Для всіх отриманих результатів перерегулювання дорівнює нулю.

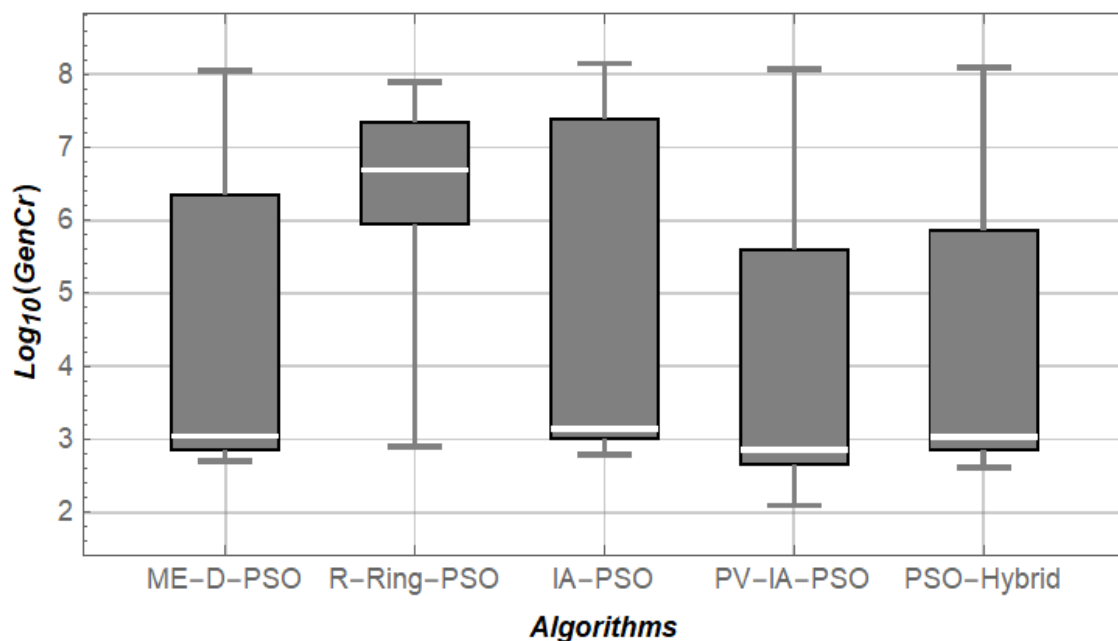
7.4.2 Нелінійні задачі

Для розв'язання нелінійних задач оптимального керування були застосовані лише найкращі PSO-алгоритми (R-Ring-PSO, IA-PSO, PV-IA-PSO, PSO-Hybrid) та алгоритм ME-D-PSO, який використовувався в попередніх дослідженнях [97] для розв'язання перелічених задач (табл. 7.1 і 6.2).

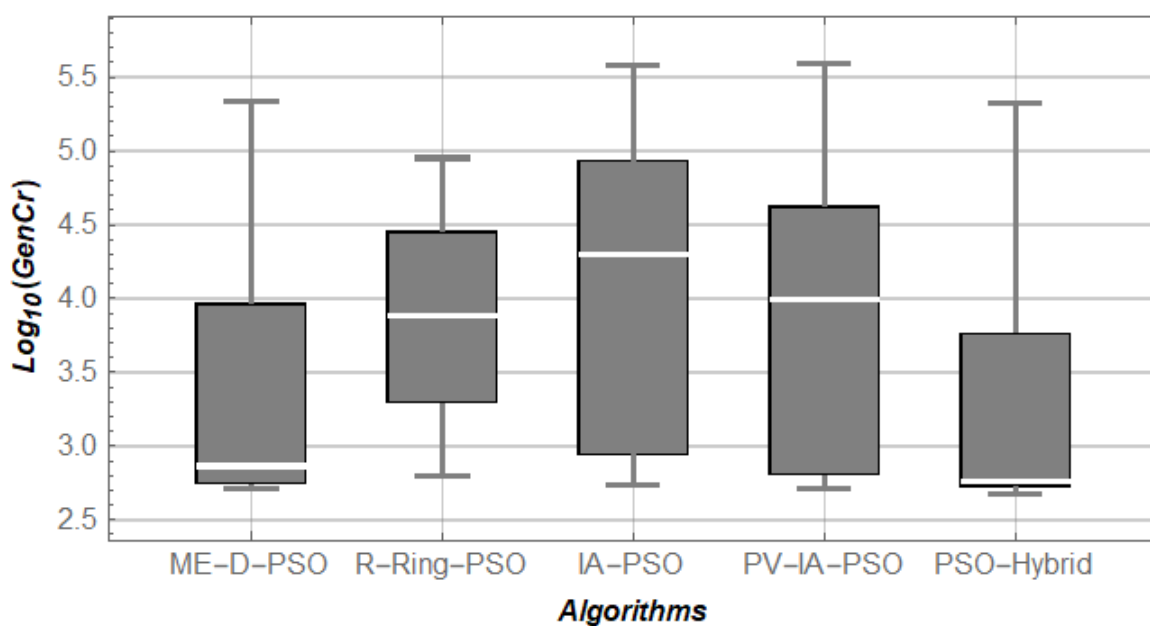
Результати їх розв'язання (в сенсі досягнутих значень критерію *GenCr*) представлені на рис. 7.3. Для першої задачі оптимального керування беззаперечним лідером є PV-IA-PSO. Лише незначна кількість рішень, знайдених за допомогою алгоритму PV-IA-PSO, є невдалими. Для цієї задачі алгоритм PSO-Hybrid не зміг розкрити свій потенціал; його ефективність була на середньому рівні.



a)



б)



в)

Рисунок 7.3 – Діаграми Бокса-Вішкера результатів розв'язування задач оптимального керування рухом: а) перевернутого маятника з маховиком; б) перевернутого маятника з рухомою точкою підвісу; в) маятника Фурути

Для другої задачі алгоритм PV-IA-PSO також є найкращим (з точки зору середнього та мінімального значень критерію $GenCr$). Однак алгоритми PSO-Hybrid і ME-D-PSO були дуже близькі.

Для третьої задачі лідерство за ефективністю розв'язування задач отримав алгоритм PSO-Hybrid. PV-IA-PSO не підходить для такого роду задач.

Крім того, у табл. 7.4 надали кількість рішень, де алгоритми досягли нижньої «ями», тобто поверхні критерію $GenCr$, де залишається лише функція Cr .

Таблиця 7.4 – Кількість найбільш вдалих розв’язань задачі
(випадки, коли алгоритм досяг поверхні критерію Cr)

Алгоритми	ОР			Всього
	Перевернутий маятник з маховиком	Перевернутий маятник з рухомою точкою підвісу	Маятник Фурути	
ME-D-PSO	6	22	36	64
R-Ring-PSO	1	5	20	26
IA-PSO	5	17	22	44
PV-IA-PSO	17	23	22	62
PSO-Hybrid	7	22	37	66

Аналіз даних, які наведені у табл. 7.4, підтверджує те, що алгоритм PV-IA-PSO можна ефективно застосовувати для процедур навчання ШНМ, зокрема для розробки оптимальних регуляторів. Це може зменшити тривалість навчання ШНМ, оскільки ймовірність знайти прийнятне рішення досить висока і немає необхідності перезапускати алгоритм знову.

Добрі результати показує і алгоритм PSO-Hybrid. Однак для першої контрольної задачі у цього алгоритму були деякі труднощі з досягненням поверхні критерію Cr . Найгірший результат на цих задачах показав R-Ring-PSO.

Оскільки перша і друга задачі оптимального керування мають складні топологічні властивості цільової функції, то загальна кількість успішних розв’язувань менша, ніж для третьої задачі (табл. 7.4). Можемо очікувати, що подальше ускладнення топології критерію $GenCr$ (наприклад, шляхом додавання інших умов і обмежень, які необхідно виконати) призведе до зменшення кількості успішних рішень.

Підсумовуючи отриманий результат, можемо стверджувати: PV-IA-PSO та PSO-Hybrid є ефективними алгоритмами для розв’язання описаних задач оптимального керування. Їх сумісне застосування дозволяє відносно швидко і легко знаходити розв’язки для задач такого роду.

Графіки регульованої змінної та регулювання для найкращих варіантів отриманих розв’язувань наведена в Додатку Д (рис. Д.2).

Висновки до сьомого розділу:

1. Для оцінки ефективності застосування PSO-алгоритмів були поставлені три лінійні та три нелінійні задачі оптимального керування (регулювання). Постановка задач включала: математичну модель ОР, крайові умови (для випадку початку руху зі стану спокою і досягнення усталеного режиму функціонування), обмеження (на функцію керування, перегулювання) та комплексний критерій оптимізації (для нелінійних задач він включав кількість перевищень допустимого значення швидкості зміни функції керування).
2. Для розв'язання лінійних задач запропоновано використати ПІ-регулятор (кількість аргументів цільової функції рівна двом), а для нелінійних – ШНМ (кількість аргументів для ОР третього порядку дорівнює 26, а для ОР четвертого порядку – 31).
3. Крім того, отримала подальший розвиток методологія розв'язання задач керування, яка полягає у введенні додаткових обмежень на швидкість зміни керування за часом та накладанні штрафної функції при порушенні цих обмежень.
4. Було проведено 50 незалежних запусків алгоритмів ME-D-PSO, R-Ring-PSO, IA-PSO, PV-IA-PSO, PSO-Hybrid для розв'язання вказаних задач. За показниками кількості найуспішніших розв'язувань лінійних задач (в яких ОР описуються передавальними функціями $G_1(s)$ та $G_2(s)$) та за аналізом діаграм Бокса-Вішкера найкращим виявився алгоритм PV-IA-PSO. Для задачі з ОР з передавальною функцією $G_3(s)$ найкращим серед порівнюваних алгоритмів став PSO-Hybrid.
5. Аналогічні дослідження проведено для нелінійних задач. Для задач керування перевернутим маятником з маховиком та перевернутим маятником з рухомою точкою підвісу за кількістю найбільш вдалих розв'язувань задачі та аналізом діаграм Бокса-Вішкера алгоритм PV-IA-PSO серед порівнюваних є найкращим. Для задачі керування маятника Фурути найкращим є алгоритм PSO-Hybrid.
6. Наявність складних топологічних властивостей цільових функцій задач оптимального керування не створює принципових складнощів при їх розв'язуванні із застосуванням вказаних алгоритмів.

РОЗДІЛ 8

ЗАСТОСУВАННЯ PSO-АЛГОРИТМІВ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧ ІНЖЕНЕРІЇ

8.1 Постановка задач оптимізації інженерних конструкцій

8.1.1 Зварна балка

Ця тестова задача використовувалася багатьма дослідниками [98-100]. На рис. 3.1 представлена схема балки [98]. Консольна зварна балка піддається впливу вертикальної сили P на своєму вільному кінці. Мета цієї задачі оптимізації пов'язана з мінімізацією собівартості виготовлення балки. Цільова функція задачі є функцією чотирьох аргументів, в задачі накладаються сім обмежень, пов'язаних із напруженням зсуву τ , напруженням згину в балці σ , навантаженням на вигин балки P_c , кінцевим прогином балки δ . Аргументами цільової функції є: товщина зварного шва h (x_1), довжина частини балки, що зварюється l (x_2), висота балки t (x_3) і товщина балки b (x_4) (рис. 8.1).

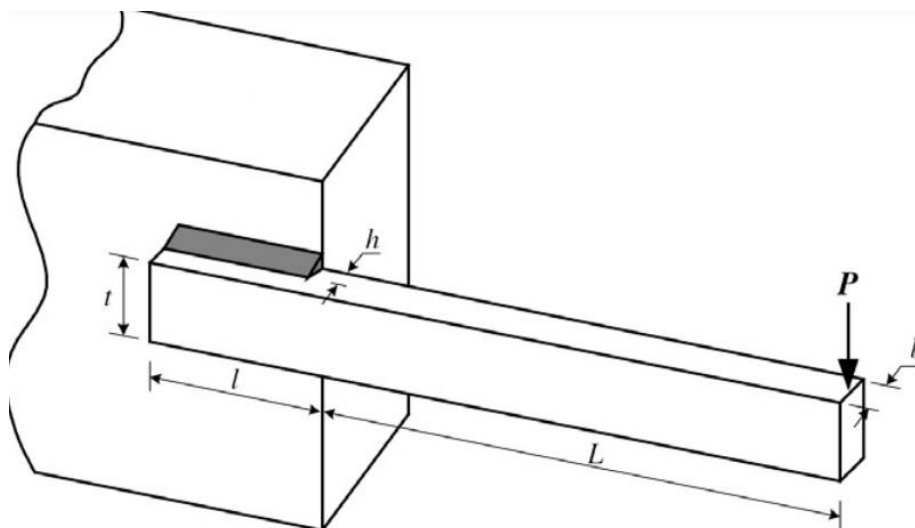


Рисунок 8.1 – Схема конструкції зварної балки

Цільова функція, яку необхідно мінімізувати, може бути виражена так:

$$f(x_1, x_2, x_3, x_4) = 1,10471 x_1^2 x_2 + 0,04811 x_3 x_4 (14 + x_2). \quad (8.1)$$

Згадані раніше обмеження представлені нижче:

$$g_1 = \tau - \tau_{\max} \leq 0;$$

$$g_2 = \sigma - \sigma_{\max} \leq 0;$$

$$g_3 = \delta - \delta_{\max} \leq 0;$$

$$g_4 = x_1 - x_4 \leq 0;$$

$$g_5 = P - P_c \leq 0;$$

$$g_6 = 0,125 - x_1 \leq 0;$$

$$g_7 = 1,10471x_1^2 + 0,04811x_3x_4(14 + x_2) - 5 \leq 0;$$

$$\tau = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2};$$

$$\tau' = \frac{P}{\sqrt{2x_1x_2}}; \quad \tau'' = \frac{MR}{J};$$

$$M = P \left(L + \frac{x_2}{2} \right); \quad R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2} \right)^2}; \quad J = 2 \left(\sqrt{2x_1x_2} \left(\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2} \right)^2 \right) \right);$$

$$\sigma = \frac{6PL}{x_4x_3^2}; \quad \delta = \frac{6PL^3}{Ex_4x_3^2};$$

Константи задачі такі: $P=6000$ фунтів, $L=14$ дюймів, $\delta_{\max}=0,25$ дюймів, $E=3 \times 10^7$ фунт/дюйм², $G=1,2 \times 10^7$ фунт/дюйм², $\tau_{\max} = 1,36 \times 10^4$ фунт/дюйм², $\sigma_{\max}=3 \times 10^4$ фунт/дюйм². Аргументи цільової функції необхідно знайти у таких областях:

$$0,1 \leq x_1, x_4 \leq 2; \quad 0,1 \leq x_2, x_3 \leq 10.$$

8.1.2 Ємність під тиском

Іншою задачею інженерної оптимізації, яка привертає увагу багатьох науковців, є проектування ємності (балона), що знаходиться під тиском. Ця задача пов'язана з мінімізацією загальної вартості ємності під тиском. Ємність утворена циліндром, закритим з обох кінців напівсферичними головками (рис.

8.2) [98]. Загальна вартість ємності під тиском включає вартість матеріалу, формування та зварювання:

$$f(x_1, x_2, x_3, x_4) = 0,6224x_1x_3x_4 + 1,7881x_2x_3^2 + 3,1661x_1^2x_4 + 19,84x_1^2x_3, \quad (8.2)$$

де x_1, x_2, x_3, x_4 – змінні цільової функції, які відповідають товщині оболонки T_s , товщині головки T_h , внутрішньому радіусу R та довжині циліндричного перерізу ємності L (без головки) відповідно.

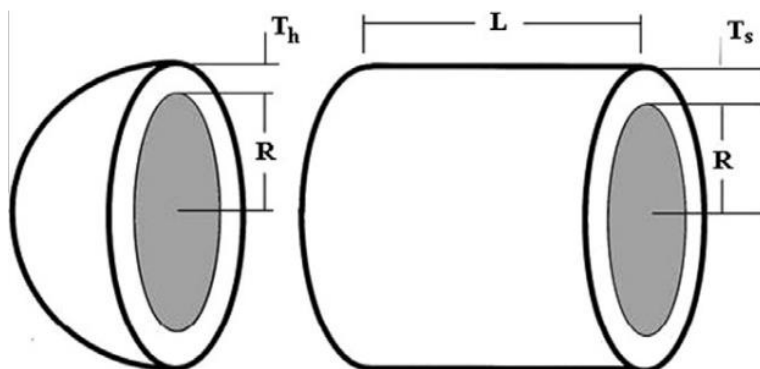


Рисунок 8.2 – Схема задачі проектування ємності під тиском

Ця задача містить чотири обмеження:

$$g_1 = 0,0193x_3 - x_1 \leq 0;$$

$$g_2 = 0,00954x_3 - x_2 \leq 0;$$

$$g_3 = 1296000 - \pi x_3^2 x_4 - \frac{4}{3} \pi x_3^3 \leq 0;$$

$$g_4 = x_4 - 240 \leq 0.$$

Змінні x_1 і x_2 пропорційні 0,0625 дюймам. Аргументи x_3 і x_4 змінюються неперервно:

$$1 \times 0,0625 \leq x_1, x_2 \leq 1600 \times 0,0625; 10 \leq x_3 \leq 200; 10 \leq x_4 \leq 240.$$

8.1.3 Пружина

Іншою добре відомою тестовою інженерною задачею є конструкція пружини (рис. 8.3) [101]. Цільова функція така:

$$f(x_1, x_2, x_3) = (x_3 + 2)x_2x_1^2, \quad (8.3)$$

де x_1 , x_2 , x_3 – розрахункові змінні, які відповідають діаметру дроту d , середньому діаметру котушки D і кількості активних витків P .

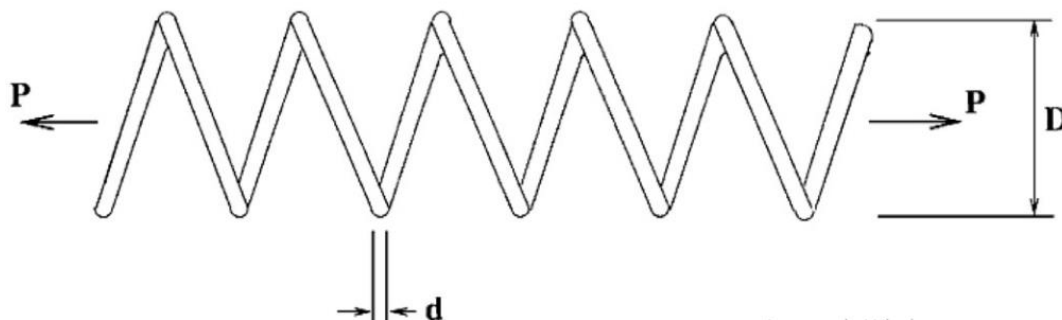


Рисунок 8.3 – Схема пружини

Повинні бути дотримані наступні обмеження, що пов'язані з мінімальним прогином, напруженням зсуву, частотою коливань, обмеженнями зовнішнього діаметра та конструктивними змінними:

$$g_1 = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0;$$

$$g_2 = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} - \frac{1}{5108 x_1^2} - 1 \leq 0;$$

$$g_3 = 1 - \frac{140,45 x_1}{x_2^2 x_3} \leq 0;$$

$$g_4 = \frac{x_1 + x_2}{1,5} \leq 0.$$

Область визначення цільової функції:

$$0,05 \leq x_1 \leq 2; 0,25 \leq x_2 \leq 1,3; 2 \leq x_3 \leq 15.$$

8.1.4 Дерев'яна ферма

Ця інженерна оптимізаційна задача була запропонована Nowski [102]. Метою розв'язання цієї задачі є мінімізація об'єму тристержневої ферми (рис. 8.4) [103], яка перебуває під статичним навантаженням P .

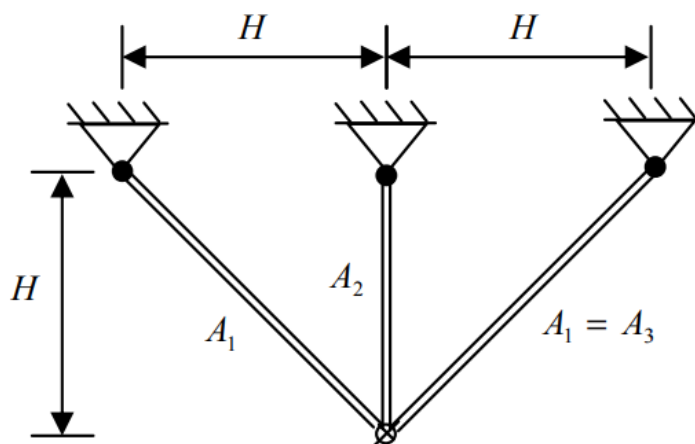


Рисунок 8.4 – Схема тристержевої дерев'яної ферми

Змінні x_1 і x_2 є площами поперечного перерізу (A_1 , A_2), а цільову функцію можна записати наступним чином:

$$f(x_1, x_2) = (2\sqrt{2}x_1 + x_2)h, \quad (8.4)$$

де h – висота стрижневої ферми.

Ця задача має обмеження, які викликані напруженнями у стержнях σ :

$$g_1 = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0;$$

$$g_2 = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0;$$

$$g_3 = \frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \leq 0.$$

Константи задачі: $h=100$ см, $P=2$ кН, $\sigma=2$ кН/см². Аргументи цільової функції можуть змінюватися в таких областях:

$$0 \leq x_1, x_2 \leq 1.$$

8.1.5 Редуктор

Редуктор є невід'ємною частиною багатьох механізмів і його раціональна конструкція важлива для експлуатації машини. Одним із критеріїв дизайну є маса редуктора, яку слід мінімізувати з дотриманням

деяких обмежень. Конструкційними змінними цієї задачі є ширина грані b (x_1), модуль зубів m (x_2), кількість зубів шестерні z (x_3), довжина першого вала між підшипниками l_1 (x_4), довжина другого вала між підшипниками l_2 (x_5), діаметр першого вала d_1 (x_6) і діаметр другого вала d_2 (x_7) (рис. 8.5) [99].

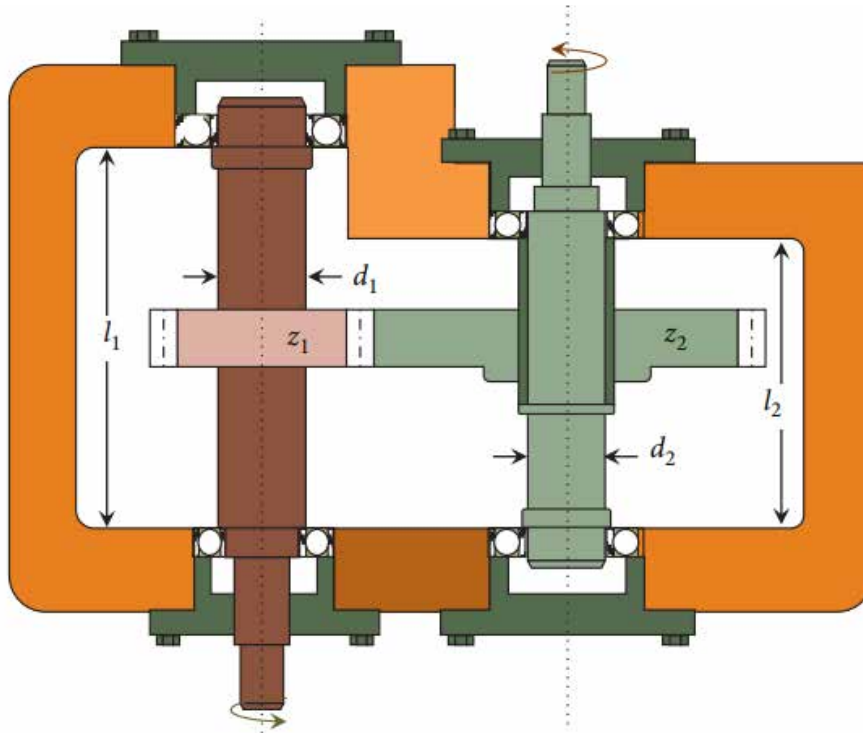


Рисунок 8.5 – Схема редуктора

Задача включає сім нелінійних і чотири лінійні обмеження-нерівності:

$$g_1 = \frac{27}{x_1 x_2^2 x_3} - 1 \leq 0;$$

$$g_2 = \frac{397,5}{x_1 x_2^2 x_3^2} - 1 \leq 0;$$

$$g_3 = \frac{1,93x_4^3}{x_2 x_6^4 x_3} - 1 \leq 0;$$

$$g_4 = \frac{1,93x_5^3}{x_2 x_7^4 x_3} - 1 \leq 0;$$

$$g_5 = \frac{\sqrt{(745x_4 / x_2 x_3)^2 + 16,9 \times 10^6}}{110x_6^3} - 1 \leq 0;$$

$$g_6 = \frac{\sqrt{(745x_5 / x_2 x_3)^2 + 157,5 \times 10^6}}{85x_7^3} - 1 \leq 0;$$

$$g_7 = \frac{x_2 x_3}{40} - 1 \leq 0;$$

$$g_8 = \frac{5x_2}{x_1} - 1 \leq 0;$$

$$g_9 = \frac{x_1}{12x_2} - 1 \leq 0;$$

$$g_{10} = \frac{1,5x_6 + 1,9}{x_4} - 1 \leq 0;$$

$$g_{11} = \frac{1,1x_7 + 1,9}{x_5} - 1 \leq 0.$$

Перші чотири обмеження викликані граничними значеннями: напруження згину зубців шестерні g_1 , поверхневого напруження g_2 , поперечного прогину валів g_3 , напруження у валах g_4 .

Цільова функція визначається таким чином:

$$f(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = 0,7854x_1x_2^2(3,3333x_3^2 + 14,9334x_3 - 43,0934) - 1,508x_1(x_6^2 + x_7^2) + 7,4777(x_6^3 + x_7^3) + 0,7854(x_4x_6^2 + x_5x_7^2). \quad (8.5)$$

Змінні цільової функції неперервні, крім x_3 – це ціле число. Область визначень цільової функції:

$$2,6 \leq x_1 \leq 3,6; 0,7 \leq x_2 \leq 0,8; 17 \leq x_3 \leq 28; 7,3 \leq x_4; x_5 \leq 8,3;$$

$$2,9 \leq x_6 \leq 3,9; 5 \leq x_7 \leq 5,5$$

8.1.6 Зубчаста передача

Одна із задач необмеженої оптимізації була представлена у роботі [104]. У цій оптимізаційній задачі необхідно визначити цілі числа зубів шестерень A , B , C та D (рис. 8.6) [99]. Вони позначаються як x_1 , x_2 , x_3 , x_4 .

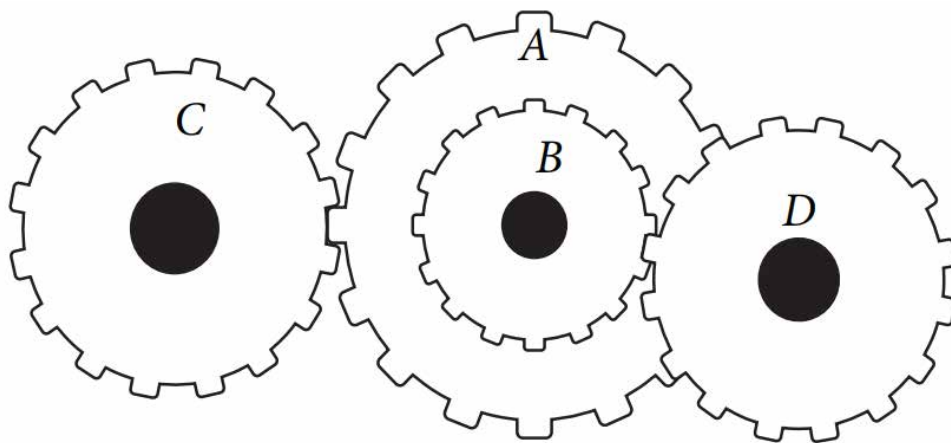


Рисунок 8.6 – Схема зубчастої передачі

Задача полягає в тому, щоб підібрати такі значення x_1, x_2, x_3, x_4 , щоб передавальне число двоступінчатої зубчастої передачі було близьким до $1/6,931$ (необхідне передавальне число). Отже, цільова функція записується так:

$$f(x_1, x_2, x_3, x_4) = \left(\frac{1}{6,931} - \frac{x_3 x_2}{x_1 x_4} \right)^2. \quad (8.6)$$

Усі аргументи задаються в одній області пошуку:

$$12 \leq x_1, x_2, x_3, x_4 \leq 60.$$

8.1.7 Консольна балка

Одним із прикладів задач проектування конструкцій є мінімізація маси консольної балки [105] (рис. 8.7) [99]. На один кінець консольної балки діє вертикальна сила (на рис. 8.7 показано стрілкою). Його поперечний переріз змінний по довжині: він складається з п'яти квадратних блоків постійної товщини ($2/3$). Висота (або ширина) блоків є змінними цільової функції, які потрібно знайти.

Вираз цільової функції такий:

$$f(x_1, x_2, x_3, x_4, x_5) = 0,0624(x_1 + x_2 + x_3 + x_4 + x_5). \quad (8.7)$$

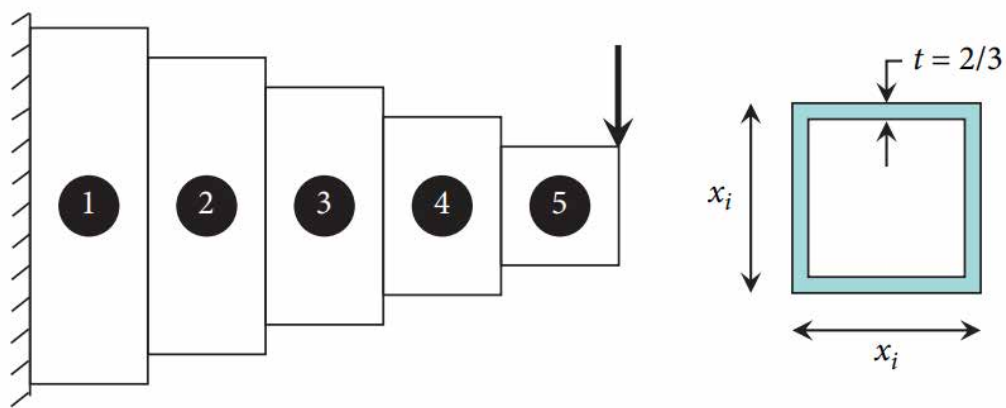


Рисунок 8.7 – Схема консольної балки

У цій задачі розглядається лише одне обмеження:

$$g = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0.$$

Усі змінні необхідно відшукати в одній області:

$$0,01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100.$$

8.1.8 Двотаврова балка

Двотаврові балки широко використовуються в будівництві, оскільки вони можуть витримувати значні навантаження. Розміри такої балки (рис. 8.8) [99] впливають на її жорсткість і, як наслідок, на її деформацію.

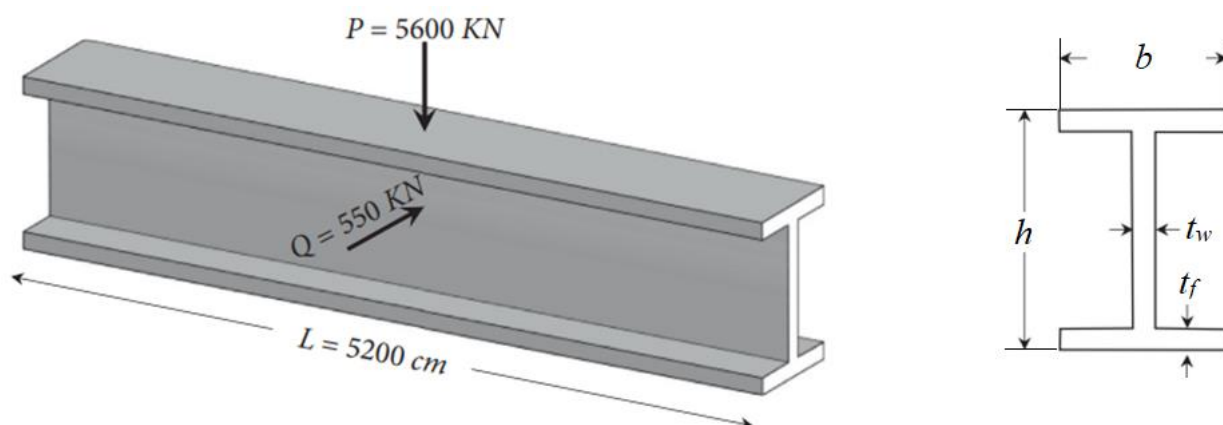


Рисунок 8.8 – Схема двотаврової балки

Таким чином, оптимізаційна задача полягає у визначенні ширини полиці $b(x_1)$, висоти балки $h(x_2)$, товщини стінки $t_w(x_3)$ і товщини полиці $t_f(x_4)$, щоб прогин балки під дією вертикальних P і горизонтальних Q сил було мінімізовано. Формула цільової функції має такий вигляд:

$$f(x_1, x_2, x_3, x_4) = \frac{5000}{x_3(x_2 - 2x_4)^3 / 12 + x_1x_4^3 / 6 + 2x_1x_4(x_2 - x_4 / 2)^2}. \quad (8.8)$$

При цьому накладено умови, що максимальний вертикальний прогин балки становить $PL^3/48EI$, коли довжина балки становить $L=5200$ см і модуль пружності $E=523,10^4$ кН/см².

Обмеження цієї задачі наступні:

$$g_1 = 2x_1x_3 + x_3(x_2 - 2x_4) - 300 \leq 0;$$

$$g_2 = \frac{18x_2 \times 10^4}{x_3(x_2 - 2x_4)^3 + 2x_1x_3(4x_4^2 + 3x_2(x_2 - 2x_4))} + \frac{15x_1 \times 10^3}{(x_2 - 2x_4)x_3^2 + 2x_3x_1^3} - 56 \leq 0.$$

Аргументи обмежені:

$$10 \leq x_1 \leq 50; 10 \leq x_2 \leq 80; 0,9 \leq x_3, x_4 \leq 5.$$

8.1.9 Гофрована перегородка

Задача оптимальної конструкції гофрованої перегородки (елемента хімічного танкера) [106], вимагає визначення ширини (x_1), глибини (x_2), довжини (x_3) і товщини плити (x_4), щоб маса гофрованої перегородки була зведена до мінімуму. Таким чином, відповідну цільову функцію можна представити у вигляді:

$$f(x_1, x_2, x_3, x_4) = \frac{5,885x_4(x_1 + x_3)}{x_1 + \sqrt{|x_3^2 - x_2^2|}}, \quad (8.9)$$

з урахуванням обмежень:

$$g_1 = -x_4x_2(0,4x_1 + x_3/6) + 8,94(x_1 + \sqrt{|x_3^2 - x_2^2|}) \leq 0;$$

$$g_2 = -x_4x_2^2(0,2x_1 + x_3/12) + 2,2\left(8,94(x_1 + \sqrt{|x_3^2 - x_2^2|})\right)^{4/3} \leq 0;$$

$$g_3 = -x_4 + 0,0156x_1 + 0,15 \leq 0;$$

$$g_4 = -x_4 + 0,0156x_3 + 0,15 \leq 0;$$

$$g_5 = -x_4 + 1,05 \leq 0;$$

$$g_6 = -x_3 + x_2 \leq 0.$$

Області зміни аргументів задачі наступні:

$$0 \leq x_1, x_2, x_3 \leq 100; 0 \leq x_4 \leq 5.$$

8.1.10 Залізобетонна балка

У роботі [107] представлено спрощену оптимізаційну задачу проектування залізобетонної балки (рис. 8.9) [99].

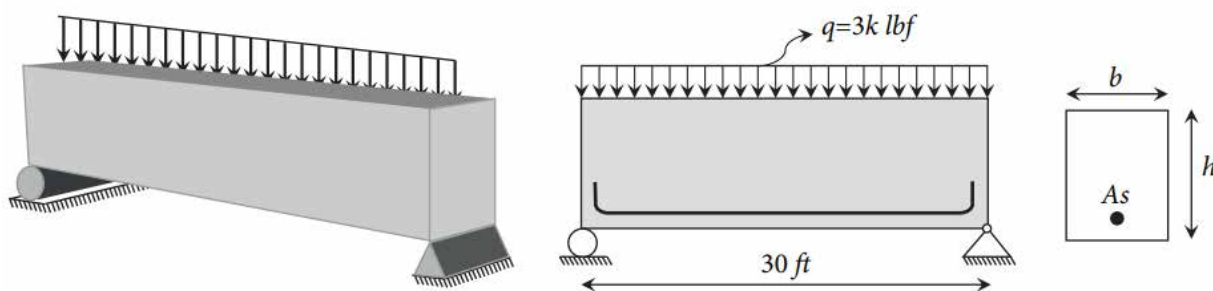


Рисунок 8.9 – Схема залізобетонної балки

Балка має проліт довжиною 30 футів, вона несе розподілене навантаження 2000 фунтів та власну вагу 1000 фунтів. Цільова функція задачі – загальна вартість балки. Її можна виразити такою формулою:

$$f(x_1, x_2, x_3) = 2,9x_1 + 0,6x_2x_3, \quad (8.10)$$

де x_1, x_2, x_3 – змінні задачі, які відповідають площі арматури A_s , ширині балки b і висоті балки h відповідно.

У задачі існує два обмеження, які необхідно задовольнити:

$$g_1 = \frac{x_2}{x_3} - 4 \leq 0;$$

$$g_2 = 180 + 7,375 \frac{x_1^2}{x_3} - x_1x_2 \leq 0.$$

Змінна x_1 є дискретною змінною, а x_2 є цілим числом, змінна x_3 – неперервна. Для всіх них області зміни наведені нижче:

$$x_1 \in \{6, 6,16, 6,32, 6,6, 7, 7,11, 7,2, 7,8, 7,9, 8, 8,4\};$$

$$28 \leq x_2 \leq 40; 5 \leq x_3 \leq 10.$$

8.1.11 Трубчаста колона

Однією з найвідоміших інженерних задач є конструювання трубчастої колони, що сприймає вертикальне навантаження P (рис. 8.10) [103].

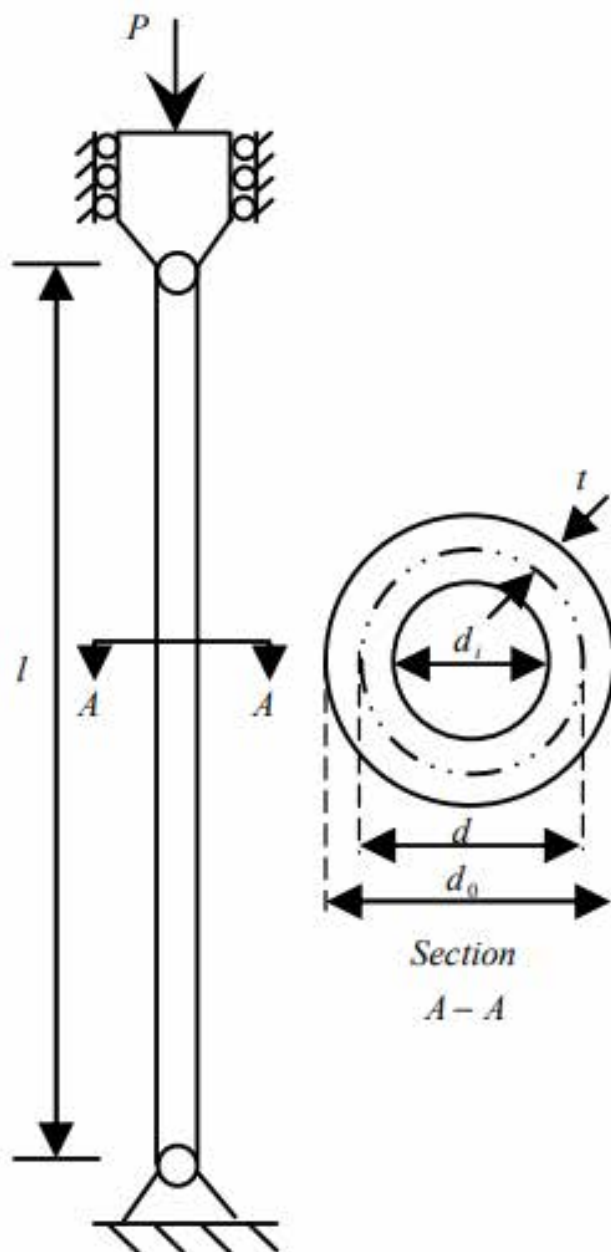


Рисунок 8.10 – Схема трубчастої колони

Задача має лише дві змінні: середній діаметр колони $d(x_1)$ і товщину труби $t(x_2)$. Цільова функція для мінімізації відображає мінімальну вартість колони (включаючи витрати на матеріал і спорудження):

$$f(x_1, x_2) = 9,8x_1x_2 + 2x_1. \quad (8.11)$$

Для даної задачі навантаження $P=25000$ Н. Матеріал колони має межу текучості $\sigma_y=5000$ Н/см², модуль пружності $E=8,5 \times 10^6$ Н/см². Довжина стовпа колони $l=250$ см. Два нелінійні обмеження задовольняють напруження прогину (обмеження g_1) і межу текучості (обмеження g_2) колони:

$$g_1 = \frac{P}{\pi x_1 x_2 \sigma_y} - 1 \leq 0;$$

$$g_2 = \frac{8Pl^2}{\pi^3 E x_1 x_2 (x_1^2 + x_2^2)} - 1 \leq 0.$$

Змінні змінюються у наступних межах:

$$2 \leq x_1 \leq 14; 0,2 \leq x_2 \leq 0,8.$$

8.1.12 Параметри металокопструкції автомобіля при дії бокового удару

Згідно з процедурами Європейського комітету з підвищеної безпеки транспортних засобів (EEVC), автомобіль може піддаватись боковому зіткненню (рис. 8.11) [98].

Youn та інші автори [108] спростили аналітичну постановку цієї задачі оптимізації. Таким чином, мета розв'язання задачі полягає у тому, щоб мінімізувати масу металокопструкції автомобіля шляхом вибору параметрів (змінних): товщини внутрішньої частини середньої стійки (x_1), підсилення середньої стійки (x_2), внутрішньої сторони підлоги (x_3), поперечин (x_4), дверної балки (x_5), посилення лінії дверей (x_6), рейлінгів на даху (x_7), матеріали внутрішньої частини центральної стійки (x_8), внутрішньої сторони підлоги (x_9), висота бар'єру (x_{10}) і місце удару (x_{11}).

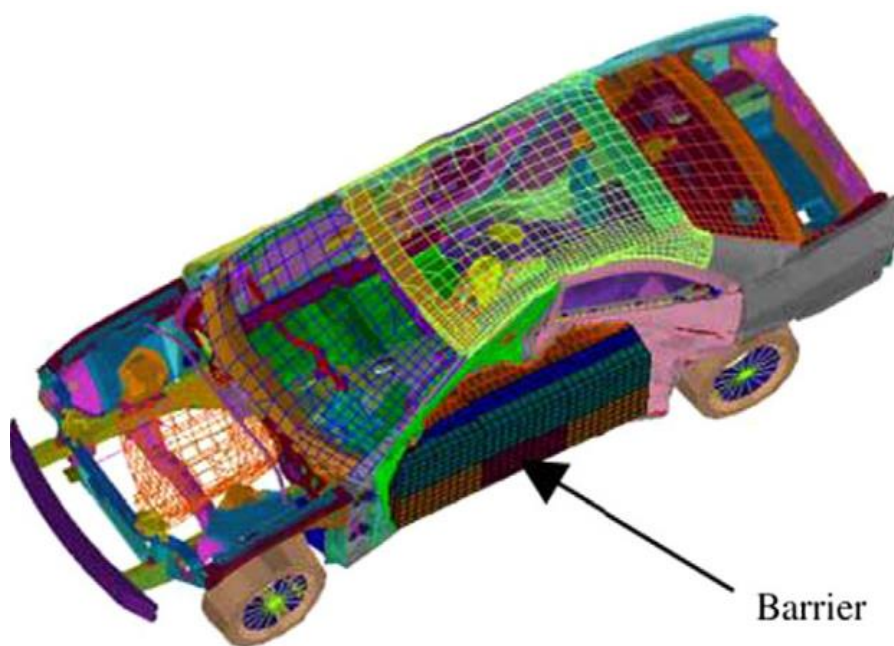


Рисунок 8.11 – Модель кінцевих елементів, використана в задачі бокового удару автомобіля

Вираз цільової функції для мінімізації виглядає наступним чином:

$$f(x_1, x_2, x_3, x_4, x_5, x_7) = 1,98 + 4,9x_1 + 6,67x_2 + 6,98x_3 + 4,01x_4 + 1,78x_5 + 2,37x_7, \quad (8.12)$$

з урахуванням десяти обмежень:

$$g_1 = 1,16 - 0,3717x_2x_4 - 0,00931x_2x_{10} - 0,484x_3x_9 + 0,01343x_6x_{10} - 1 \leq 0;$$

$$g_2 = 46,36 - 9,9x_2 - 12,9x_1x_2 + 0,1107x_3x_{10} - 32 \leq 0;$$

$$g_3 = 33,86 + 2,95x_3 + 0,1792x_{10} - 5,057x_1x_2 - 11,0 \times \\ \times x_2x_8 - 0,0215x_5x_{10} - 9,98x_7x_8 + 22,0x_8x_9 - 32 \leq 0;$$

$$g_4 = 28,98 + 3,818x_3 - 4,2x_1x_2 + 0,0207x_5x_{10} + \\ + 6,63x_6x_9 - 7,7x_7x_8 + 0,32x_9x_{10} - 32 \leq 0;$$

$$g_5 = 0,261 - 0,0159x_1x_2 - 0,188x_1x_8 - 0,019x_2x_7 + 0,0144x_3x_5 + 0,0008757 \times \\ \times x_5x_{10} + 0,08045x_6x_9 + 0,00139x_8x_{11} + 0,00001575x_{10}x_{11} - 0,32;$$

$$g_6 = 0,214 - 0,00817x_5 - 0,131x_1x_8 - 0,0704x_1x_9 + 0,03099x_2x_6 - 0,018x_2x_7 + \\ + 0,0208x_3x_8 + 0,121x_3x_9 - 0,00364x_5x_6 + 0,0007715x_5x_{10} - 0,0005354x_6x_{10} + \\ + 0,00121x_8x_{11} + 0,00184x_9x_{10} - 0,02x_2^2 - 0,32;$$

$$g_7 = 0,74 - 0,61x_2 - 0,163x_3x_8 + 0,001232x_3x_{10} - 0,166x_7x_9 + 0,227x_2^2 - 0,32;$$

$$g_8 = 4,72 - 0,5x_4 - 0,19x_2x_3 - 0,0122x_4x_{10} + 0,009325x_6x_{10} + 0,000191x_{11}^2 - 4;$$

$$g_9 = 10,58 - 0,674x_1x_2 - 1,95x_2x_8 + 0,02054x_3x_{10} - 0,0198x_4x_{10} + 0,028x_6x_{10} - 9,9;$$

$$g_{10} = 16,45 - 0,489x_3x_7 - 0,843x_5x_6 + 0,0432x_9x_{10} - 0,0556x_9x_{11} - 0,000786x_{11}^2 - 15,7.$$

Аргументи змінюються у таких областях:

$$0,5 \leq x_1, x_2, x_3, x_4, x_5, x_6, x_7 \leq 1,5; x_8, x_9 \in \{0,192, 0,345\}; -30 \leq x_{10}; x_{11} \leq 30.$$

8.2 Аналіз результатів застосування PSO-алгоритмів

У таблицях у Додатку Е наведено всі отримані результати розв'язання задач. Позначення NV (not valid) означає, що знайдені розв'язки не задовольняють усім обмеженням, а штрафне значення цільової функції впливає на статистичний показник. Найкращі значення серед порівнянних виділені жирним шрифтом. Крім того, під кожною з таблиць представлені найкращі значення цільової функції, що знайдені у відомих роботах. На основі цих даних можемо порівняти мінімальні значення, які наведені у табл. Е.1.-Е.12 з відомими розв'язками. Для цього враховуємо їх співвідношення (найкращі отримані результати/відоме найменше значення цільової функції). Усі відповідні дані наведені нижче (табл. 8.1).

Аналіз отриманих даних, які наведено у табл. 8.1 показує, що використовувані у даному дослідженні алгоритми є достатньо ефективними у плані пошуку глобальних мінімумів цільових функцій, які відповідають оптимізаційним задачам інженерії. Єдине виключення – алгоритм LDW-PSO, який показав погану пошукову діяльність при вирішенні задачі конструкції зубчастої передачі у порівнянні із іншими алгоритмами. Сума показників, які наведені у табл. 8.1 показує дуже незначну відмінність у ефективності роботи алгоритмів. Все ж найкаршим за цим показником є алгоритм VCT-PSO. Для алгоритмів VCT-PSO, Ring-PSO, R-Ring-PSO, ME-D-PSO, PV-ND-PSO та PSO-Hybrid різниця у цьому показнику з'являється у тисячних долях.

Таблиця 8.1 – Співвідношення між отриманими у даному розділі та відомими у науковій літературі розв’язками дванадцяти інженерних оптимізаційних задач

Задача	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-ND-PSO	PSO-Hybrid
1	1,01332	1,02735	1,01332	1,01332	1,01332	1,01343	1,01402	1,01332
2	1,00011	1,00068	1,00011	1,00011	1,00011	1,00046	1,00019	1,00011
3	1,00002	1,00545	1,00011	1,00002	1,00005	1,00014	1,00117	1,00005
4	1,00082	1,00082	1,00082	1,00082	1,00082	1,00082	1,00082	1,00082
5	1,00001	1,00928	1,00001	1,00001	1,00001	1,00001	1,00075	1,00001
6	8,54544	1,00005	1,00005	1,00005	1,00005	1,00005	1,00005	1,00005
7	1,00001	1,00152	1,00001	1,00001	1,00001	1,00001	1,00003	1,00001
8	1,00000	1,00000	1,00000	1,00000	1,00000	1,00000	1,00001	1,00000
9	1,00000	1,00466	1,00000	1,00000	1,00000	1,00000	1,00025	1,00000
10	1,00000	1,00004	1,00000	1,00000	1,00000	1,00000	1,00000	1,00000
11	1,00050	1,00063	1,00050	1,00050	1,00050	1,00050	1,00053	1,00050
12	0,91110	0,92403	0,91111	0,91136	0,91132	0,91173	0,91205	0,91174
Сума	19,47132	11,97451	11,92603	11,92619	11,92618	11,92716	11,92988	11,92660

8.3 Задача глобальної оптимізації (визначення оптимального за швидкодією керування)

Існує велика потреба в ефективних методах оптимізації у багатьох сферах сучасної науки і виробництва. Для деяких з них оптимізація є «основною», обов'язковою процедурою. Навчання нейронної мережі неможливо уявити без застосування методу оптимізації [1]; оптимально налаштовані контролери дозволяють мінімізувати витрати енергії та матеріалів на багатьох виробництвах [109]; оптимальне планування є важливою проблемою для величезної кількості організацій [110]; оптимальне керування міським транспортом дозволяє покращити транспортне обслуговування, зменшити забруднення та споживання палива [111]. Повний список прикладів величезний.

Ефективність методів оптимізації можна оцінити двома способами: шляхом застосування до класу реальних задач або із залученням деяких еталонних показників (синтетичних цільових функцій). Обидва способи мають недоліки та переваги, і бажано апробувати новий алгоритм оптимізації (або нову модифікацію існуючого) шляхом його застосування до цих двох класів задач оптимізації [112]. Однак більшість реальних задач оптимізації, які представлені в наукових працях, досить прості. В результаті ефективність застосованих алгоритмів оптимізації незначно відрізняється один від одного, тобто практично неможливо виявити пошукові особливості алгоритмів оптимізації.

У цій роботі запропоновано два типи задач оптимізації, які пов'язані із реальними задачами керування та мають практичну цінність. Крім того, як буде показано далі, їх досить складно мінімізувати. Тому, на нашу думку, вони можуть бути широко використаними для оцінки ефективності алгоритмів оптимізації.

Розв'язок оптимального керування за тривалістю/часом можна отримати за допомогою принципу максимуму Понтрягіна [113]. Зазначимо, що

обов'язковим елементом цієї задачі є обмеження на керування. Дійсно, без обмежень на керування задача не має сенсу, оскільки справедливий такий вираз:

$$\lim_{u \rightarrow \infty} t_{ac} = 0, \quad (8.14)$$

де t_{ac} – тривалість керованого періоду; u – функція керування. Вираз (8.14) є досить загальним і має широкий спектр інтерпретацій.

Структура розв'язку задачі керування за оптимальною тривалістю відома: керування u має змінюватись від верхнього обмеження до нижнього і навпаки. Можна уявити найпростіший випадок: прискорення одномасової системи від стану спокою до сталої швидкості v . Для цього прикладу керування u дорівнює лише верхньому обмеженню протягом певного періоду часу (залежно від значення сталої швидкості v). Для більш загальних випадків u кілька разів змінюється між від одного обмеження до іншого. Кількість таких змін (перемикань) залежить від порядку системи лише у випадку, коли система не є коливальною [114]. Таким чином, вихідну задачу можна звести до задачі визначення моментів перемикання функції керування.

Задача оптимального за тривалістю керування краном з вантажем добре відома і присвячена їй велика кількість робіт. У подальшому аналізі літератури зосередимося на методах її розв'язання, а не на результатах та їх практичному значенні.

Існує два загальні класи керування, що застосовуються до цієї задачі: замкнуте (функція змінних стану) і розімкнуте (функція часу). Останній розглядається в цьому дослідженні. Однак варто згадати деякі статті [60, 115-117], де було отримано замкнуте керування. Тут, наприклад, для визначення коефіцієнтів лінійного регулятора системи „кран-вантаж” застосовано модифікацію методу PSO [115]. Цей же метод було застосовано для навчання штучної нейронної мережі (ШНМ) як контролера [60], але з розширеним критерієм (тривалість і потужність RMS) і додатковими обмеженнями (швидкість рушійної сили обмежена). У роботі [116] для розв'язання задачі

використано ШНМ. Він був навчений за допомогою алгоритму Левенберга-Марквардта з байєсівською регуляризацією. Замкнуте оптимальне керування може бути отримано на основі розімкнутого керування [117] за принципом максимуму Понтрягіна [113].

Клас задач, де керування є розімкненим, як було зазначено раніше, вимагає визначення моментів перемикання. Для цього було застосовано PSO [118]. У цій статті було розглянуто два типи обмежень на керування та показано різницю в динаміці системи при дії отриманих керувань. Лабораторні досліди [119] підтвердили отримані теоретичні результати. Автори статті [120] звели задачу планування руху крана із усуненням коливань вантажу до розв'язку послідовності задач лінійного програмування (ЛП) з фіксованим часом. Було також наведено теоретичне підтвердження цього підходу. Автори роботи [121] розглядали задачу оптимального керування за швидкодією як ЛП. Для її розв'язання використовувався пакет CVX [122]. У роботі [123] метод послідовного квадратичного програмування використовувався для розв'язання задачі оптимального за тривалістю керування. Для досягнення цієї мети залучено інструментарій MATLAB GPOPS [124]. У деяких дослідженнях [125-127] для визначення моментів перемикання керування використовувався підхід фазової площини. Моменти перемикання знаходяться як точки, де система змінює рух з однієї фазової траєкторії на іншу. Автори роботи [128] визначили моменти перемикання керування як розв'язок системи трансцендентних рівнянь. Метод моментів також можна застосувати для знаходження оптимального за часом керування [126, 129]. Однак це вимагає відносно складних розрахунків.

Усі розв'язки задач у згаданих статтях відносяться до деякого заздалегідь заданого набору параметрів (маса вантажу, довжина канату тощо). Таким чином, постає наступне питання: як прикладні підходи працюватимуть на різноманітному наборі параметрів? У даному дослідженні використано множину параметрів, що вимагає розв'язання не однієї задачі, а сукупності задач (двох типів з різними параметрами). Такий підхід може гарантувати

(певною мірою) загальність результатів [130], а алгоритм оптимізації, застосований для їх розв'язання, може бути оцінений щодо його ефективності.

8.3.1 Постановка та зведення задачі глобальної оптимізації

Сформулюємо задачу оптимального керування. Об'єкт керування являє собою кран з вантажем на гнучкому підвісі (канаті). Цю динамічну систему можна представити у вигляді математичного маятника з рухомою точкою підвісу (рис. 8.13).

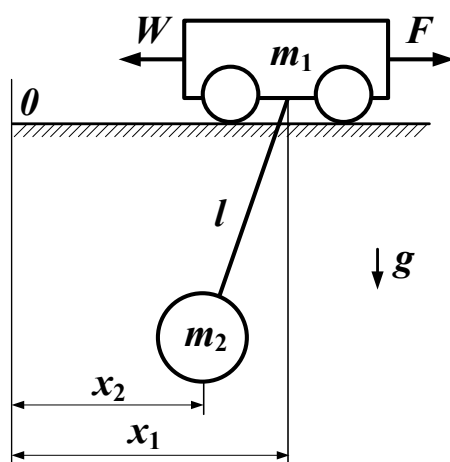


Рисунок 8.13 – Динамічна модель системи „кран-вантаж”

Математична модель, яка відповідає рис. 3.13, має такий вигляд:

$$\begin{cases} m_1 \ddot{x}_1 + m_2 \ddot{x}_2 = F - W \text{sign}(\dot{x}_1); \\ x_1 = x_2 + \frac{l}{g} \ddot{x}_2, \end{cases} \quad (8.15)$$

де m_1 і m_2 – приведені маси крана і вантажу відповідно; x_1 і x_2 – положення мас m_1 і m_2 відповідно; l – довжина каната; g – прискорення вільного падіння; F і W – зведені до маси m_1 рушійна сила та сила опору відповідно.

Систему диференціальних рівнянь (8.15) можна звести до одного диференціального рівняння четвертого порядку:

$${}^{IV}x_2 + \ddot{x}_2 \Omega^2 = \frac{F - W}{m_1} \Omega_0^2, \quad (8.16)$$

де Ω та Ω_0 – частоти власних коливань вантажу з рухомими та нерухомими

точками підвісу відповідно ($\Omega = \sqrt{\frac{m_1 + m_2}{m_1} \frac{g}{l}}$, $\Omega_0 = \sqrt{\frac{g}{l}}$).

Прискорення системи «кран-навантаження» можна описати наступними граничними умовами:

$$\begin{cases} x_1(0) = 0, & x_2(0) = 0, & \dot{x}_1(0) = 0, & \dot{x}_2(0) = 0; \\ x_1(T) = x_2(T), & \dot{x}_1(T) = v, & \dot{x}_2(T) = v, \end{cases} \quad (8.17)$$

де T – тривалість розгону системи до сталої швидкості; v – усталена швидкість системи. З урахуванням другого рядка системи (3.16) можна переписати крайові умови лише для однієї функції x_2 :

$$\begin{cases} x_2(0) = 0, & \dot{x}_2(0) = 0, & \ddot{x}_2(0) = 0, & \ddot{\ddot{x}}_2(0) = 0; \\ \dot{x}_2(T) = v, & \ddot{x}_2(T) = 0, & \ddot{\ddot{x}}_2(T) = 0. \end{cases} \quad (8.18)$$

Критерій задачі оптимального керування такий:

$$\int_0^T dt = T \rightarrow \min . \quad (8.19)$$

Як було зазначено вище, немає сенсу мінімізації (8.19) без обмежень на керування. Для цієї задачі вони такі:

$$F_{\min} \leq F \leq F_{\max} , \quad (8.20)$$

де F_{\max} і F_{\min} – максимальне і мінімальне значення рушійної сили.

Аналіз багатьох робіт [123, 125-129], де задача (8.15)-(8.20) розв'язувалася в подібних постановках, дозволяє отримати структуру її розв'язку: сила F повинна змінюватись від F_{\max} до F_{\min} , і навпаки. Кількість таких змін – два.

Таким чином, весь період прискорення можна розглядати як три підперіоди, де сила F є постійним (F_{\max} або F_{\min}) значенням. Цей факт дозволяє отримати аналітичний розв'язок диференціального рівняння (8.16) для i -го підперіоду:

$$\begin{aligned}
x_2 = & \frac{1}{2m_1\Omega^4} (2m_1\Omega^2 (\ddot{x}_2(\tau_{i-1}) + \ddot{x}_2(\tau_{i-1})(t - \tau_{i-1}) + (\dot{x}_2(\tau_{i-1})(t - \tau_{i-1}) + \\
& + x_2(\tau_{i-1}))\Omega^2) + (F_i - W)(\Omega^2(t - \tau_{i-1})^2 - 2)\Omega_0^2 - 2(\ddot{x}_2(\tau_{i-1})m_1\Omega^2 - \Omega_0^2 \times \\
& \times (F_i - W)) \cos((t - \tau_{i-1})\Omega) - 2m_1\ddot{x}_2(\tau_{i-1})\Omega \sin((t - \tau_{i-1})\Omega)), \quad i \in \overline{(1, 3)},
\end{aligned} \quad (8.21)$$

де F_i – постійне значення рушійної сили на i -му підперіоді (F_{\max} чи F_{\min}); τ_{i-1} – момент початку i -го підперіоду (рис. 8.14).

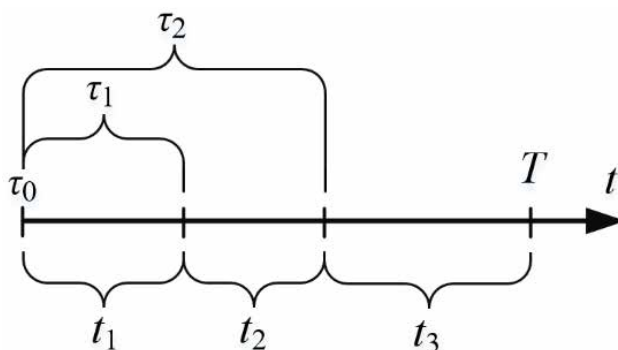


Рисунок 8.14 – Часові координати підперіодів

Наступний крок обчислення – підстановка відповідних виразів для кожного з підперіодів у закон (8.21):

$$\begin{cases}
x_2(\tau_0) \rightarrow 0, \dot{x}_2(\tau_0) \rightarrow 0, \ddot{x}_2(\tau_0) \rightarrow 0, \ddot{\ddot{x}}_2(\tau_0) \rightarrow 0, F = F_{\max}, \tau_0 \leq t \leq \tau_1; \\
x_2(\tau_1) \rightarrow x_{2,t_1}, \dot{x}_2(\tau_1) \rightarrow \dot{x}_{2,t_1}, \ddot{x}_2(\tau_1) \rightarrow \ddot{x}_{2,t_1}, \ddot{\ddot{x}}_2(\tau_1) \rightarrow \ddot{\ddot{x}}_{2,t_1}, F = F_{\min}, \tau_1 < t \leq \tau_2; \\
x_2(t_1 + t_2) \rightarrow x_{2,t_1+t_2}, \dot{x}_2(t_1 + t_2) \rightarrow \dot{x}_{2,t_1+t_2}, \ddot{x}_2(t_1 + t_2) \rightarrow \ddot{x}_{2,t_1+t_2}, \ddot{\ddot{x}}_2(t_1 + t_2) \rightarrow \ddot{\ddot{x}}_{2,t_1+t_2}, \\
F = F_{\max}, \tau_2 < t \leq T,
\end{cases} \quad (8.22)$$

$$\tau_0 = 0, \tau_1 = t_1, \tau_2 = t_1 + t_2, t_1 + t_2 + t_3 = T,$$

де t_1, t_2, t_3 – тривалість першого, другого та третього підперіодів відповідно (рис. 8.14).

У результаті ми отримали три розв'язки, які відповідають трьом підперіодам. Позначимо їх як $x_{2,1}, x_{2,2}, x_{2,3}$ (індекси 1, 2 і 3 позначають підперіод). Маючи всі розв'язки (за три підперіоди) задаємо умови стиків розв'язків у моменти τ_1, τ_2 :

$$\begin{cases}
x_{2,1}(\tau_1) = x_{2,\tau_1}, \dot{x}_{2,1}(\tau_1) = \dot{x}_{2,\tau_1}, \ddot{x}_{2,1}(\tau_1) = \ddot{x}_{2,\tau_1}, \ddot{\ddot{x}}_{2,1}(\tau_1) = \ddot{\ddot{x}}_{2,\tau_1}; \\
x_{2,2}(\tau_2) = x_{2,\tau_2}, \dot{x}_{2,2}(\tau_2) = \dot{x}_{2,\tau_2}, \ddot{x}_{2,2}(\tau_2) = \ddot{x}_{2,\tau_2}, \ddot{\ddot{x}}_{2,2}(\tau_2) = \ddot{\ddot{x}}_{2,\tau_2}.
\end{cases} \quad (8.23)$$

Ці розрахунки приводять рішення $x_{2,3}=x_{2,3}(t)$. Підстановка $t \rightarrow t_1+t_2+t_3$ дає кінцеву координату $x_{2,3}(t_1+t_2+t_3)$ та її старші похідні за часом. Отже, маємо $\dot{x}_{2,3}(t_1+t_2+t_3)$, $\ddot{x}_{2,3}(t_1+t_2+t_3)$, $\ddot{\ddot{x}}_{2,3}(t_1+t_2+t_3)$.

Щоб задовольнити кінцеві умови (8.18), розглянемо «кінцеву енергію» (коротко TE) системи – функцію, яка відображає (в певному сенсі) рівень задоволення кінцевих умов (8.18):

$$\begin{aligned}
 TE &= \frac{m_2}{2} \left((v - \dot{x}_{2,3}(t_1+t_2+t_3))^2 + (\ddot{x}_{2,3}(t_1+t_2+t_3))^2 + \left(\frac{\ddot{\ddot{x}}_{2,3}(t_1+t_2+t_3)}{\Omega_0^2} \right)^2 \right) = \\
 &= \frac{m_2}{2m_1^2\Omega_0^6} (\Omega^2\Omega_0^4(F_{\max} - W - (F_{\max} - F_{\min}))(\cos(t_3\Omega) - \cos((t_2+t_3)\Omega)) + (W - F_{\max}) \times \\
 &\times \cos((t_1+t_2+t_3)\Omega))^2 + \Omega^4((F_{\max} - F_{\min})(\sin(t_3\Omega) - \sin((t_2+t_3)\Omega)) + (F_{\max} - W) \times \\
 &\times \sin((t_1+t_2+t_3)\Omega))^2 + (m_1v\Omega^3 - (F_{\min}t_2 + F_{\max}(t_1+t_3) - (t_1+t_2+t_3)W)\Omega\Omega_0^2 + \Omega_0^2 \times \\
 &\times ((F_{\max} - F_{\min})(\sin(t_3\Omega) - \sin((t_2+t_3)\Omega)) + (F_{\max} - W)\sin((t_1+t_2+t_3)\Omega)))^2). \quad (8.24)
 \end{aligned}$$

Перший член у дужках відповідає кінетичній «надлишковій або недостатній енергії» маси m_2 . Ця величина показує відхилення енергії маси m_2 в момент $t_1+t_2+t_3$ від значення $m_2v^2/2$. Другий і третій доданки – потенціальна і кінетична енергії коливань вантажу (маятника).

Крім того, слід звернути увагу на значення F_{\min} у функції (8.24). Це значення впливає на тривалість T : чим менше F_{\min} , тим менше T .

З іншого боку, перемикання з F_{\max} на F_{\min} є важким режимом для приводу крана та металевої конструкції. Таким чином, для зменшення динамічних впливів встановимо $F_{\min}=0$ [118]. У відомих розв'язках задач [121, 123-129] використовувалися симетричні обмеження $F_{\min} = -F_{\max}$. Ми пропонуємо називати задачу з обмеженнями $F_{\min} = -F_{\max}$ Типом 1, а задачу з $F_{\min}=0$ – Типом 2.

Таким чином, ми звели початкову задачу до наступної:

$$\begin{aligned}
 &\arg \min_{t_1, t_2, t_3} \left(\frac{2E}{m_2v^2} + (t_1+t_2+t_3) \frac{\Omega}{2\pi} \right), \\
 E &= \begin{cases} w \cdot TE, & \text{if } TE \geq \Delta; \\ 0, & \text{if } TE < \Delta, \end{cases} \quad (8.25)
 \end{aligned}$$

де Δ – деякий умовний поріг енергії, де величина TE незначна і тому нею можна знехтувати (в рамках поточного дослідження $\Delta=10^{-2}$ Дж); w – ваговий коефіцієнт, що збільшує значущість мінімізації TE (приймаємо $w=10^6$).

Перший член виразу (8.25) у дужках – питома енергія системи. Другий – це питома тривалість розгону системи. Обидва терміни є безрозмірними.

Точний розв'язок задачі зводить перший член критерію (8.25) до значення, що є меншим за Δ , і всі кінцеві умови (8.18) практично виконуються (відхилення кінцевих значень фазових координат від (3.18) регулюється Δ , чим ближче кінцеві фазові координати до кінцевих умов (3.18)).

Включення значень w і Δ в цільову функцію (8.25) можна пояснити двоетапним процесом пошуку. На першому етапі мінімізується значення E , на другому – алгоритм шукає мінімальне значення T (оскільки $E=0$). Така структура цільової функції (8.25) дозволяє уникнути хибних розв'язків вихідної задачі (8.15)-(8.20). Дійсно, можна уявити принаймні два таких приклади:

- а) $w=1$, тоді $t_1 \rightarrow 0$, $t_2 \rightarrow 0$, $t_3 \rightarrow 0$. Для цього випадку цільова функція (8.25) має глобальний мінімум, який дорівнює 1. Другий і третій члени TE (3.24) у дужках (перший рядок виразу) дорівнюють нулю, лише перший член дорівнює v ;
- б) $\Delta=0$, то на якомусь етапі цільової функції (8.25) добуток $w \cdot TE$ буде зменшено до значення, порівнянного зі значеннями t_1 , t_2 і t_3 . Це означає, що члени $w \cdot TE$ та $(t_1+t_2+t_3)\Omega/2 \pi$ почнуть конкувати у мінімізації. У цьому випадку немає гарантії повної мінімізації критерію (3.19).

8.3.2 Короткий аналіз цільової функції

Для розв'язання задачі (8.25) необхідно задати числові значення системи (табл. 8.2). Цільова функція (8.25) має складну топологію. Щоб це показати, підставимо значення (табл. 8.2) $m_2=10000$ кг і $l=6$ м у вираз (8.25) і прийmemo $t_3=t_1$ (випадок, загальний для більшості розв'язків задач).

Таблиця 8.2 – Числові значення параметрів динамічної системи „кран-вантаж”

Параметр	одиниця	Значення
m_1	кг	42000
v	м/с	0,7
F_{\max}	Н	24107
F_{\min}	Н	-24107 (Тип-1) або 0 (Тип-2)
W	Н	$0,01(m_1+m_2)g$

Далі досліджуємо лише функцію $TE(t_1, t_2)$, оскільки вона сильно впливає на цільову функцію (8.25), принаймні до тих пір, поки не виконується умова $TE \leq \Delta$ (другий член критерію (8.25) лінійно залежить від значень t_1, t_2 і t_3 , таким чином, для алгоритму немає великих труднощів знайти мінімум цієї суми).

Ілюстрації характеристик функції $TE(t_1, t_2)$ для обох задач (Типу 1 і Типу 2) наведено на рис. 8.15 і рис. 8.16 відповідно. На цих графіках червона точка показує розташування глобального мінімуму.

Функція $TE(t_1, t_2)$ для обох випадків є мультимодальною, несепарабельною. Її плато досить пологіе (рис. 8.15, а та рис. 8.16, а), що ускладнює пошук розташування глобального мінімуму.

Поле градієнта (рис. 8.15, б і рис. 8.16, б) має два розділених каньйони. Перший відображає максимум (підйом) функції $TE(t_1, t_2)$, інший – її мінімум (нижнє плато). На вигині останнього розташований глобальний мінімум (для обох випадків задач Типу 1 і Типу 2). Таким чином, градієнтно-орієнтовані методи оптимізації можуть не досягти глобального мінімуму. Атрактор глобального мінімуму досить вузький і алгоритм повинен зробити належну кількість ітерацій, щоб досягти його з необхідною точністю. Слід підкреслити, що тут покладаємо $t_3=t_1$. Наведений короткий аналіз стосується $TE(t_1, t_2, t_3)$ лише певною мірою. Загальний випадок $t_3 \neq t_1$ більш складний.

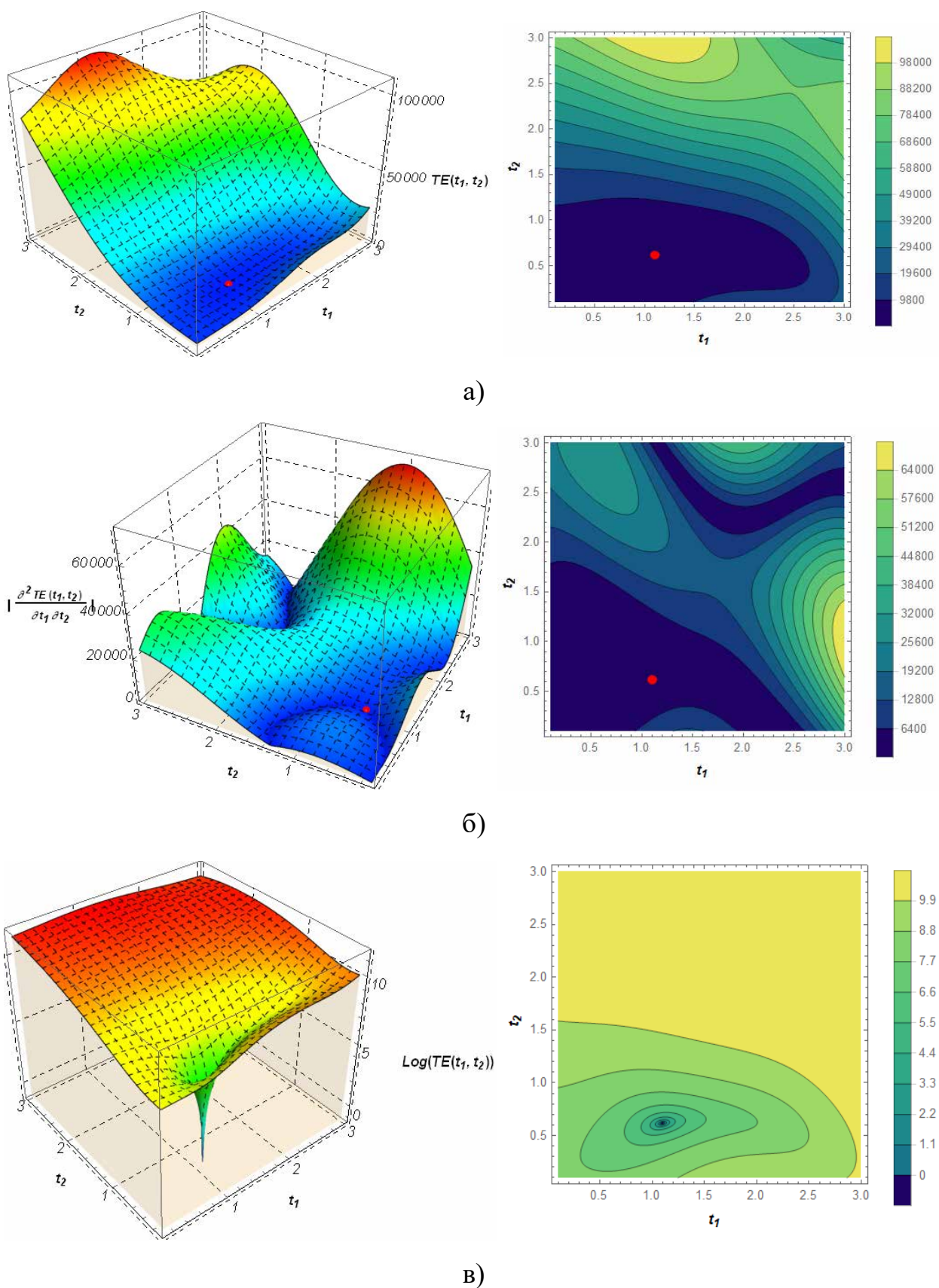
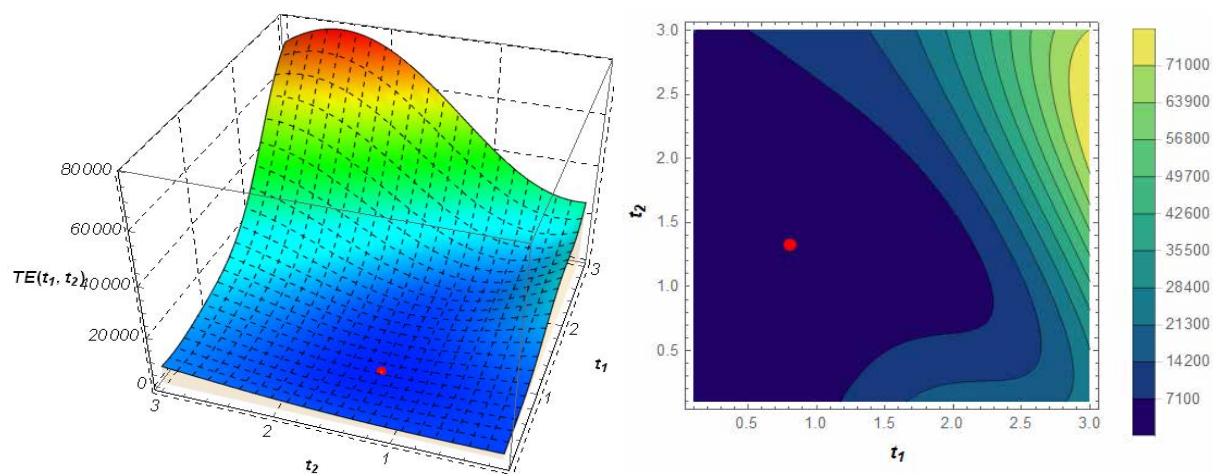
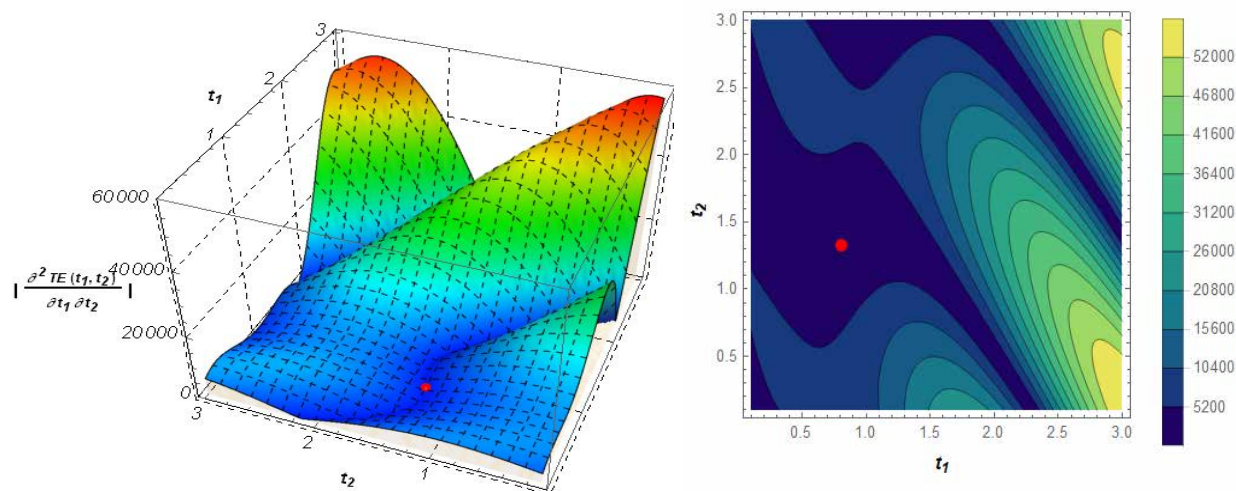


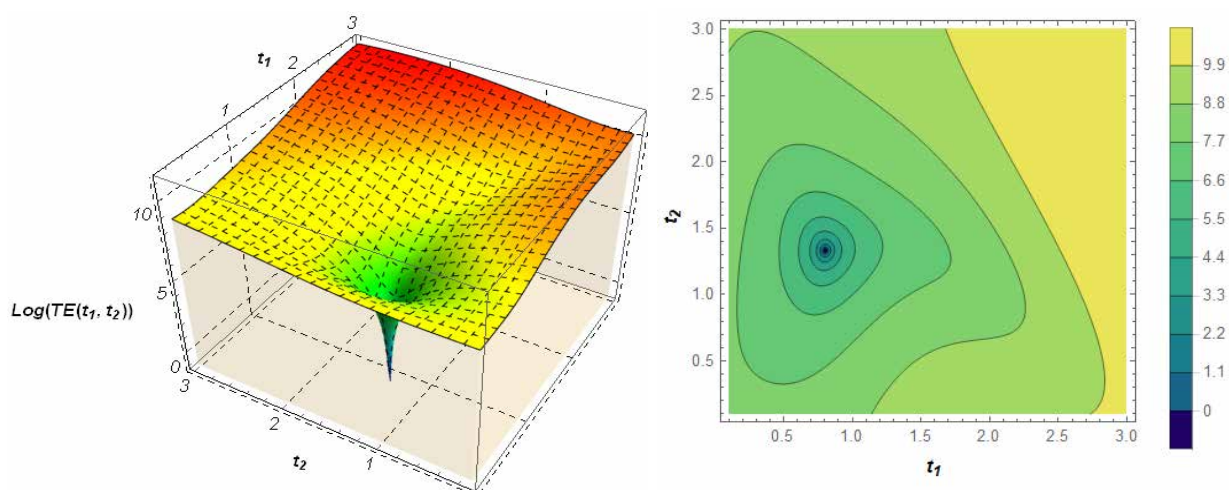
Рисунок 8.15 – Особливості топології функції TE (Тип 1) (тривимірні та контурні графіки): а) функція TE ; б) $\left| \frac{\partial^2 TE(t_1, t_2)}{\partial t_1 \partial t_2} \right|$ функція; в) функція $\log(TE)$



а)



б)



в)

Рисунок 8.16 – Особливості топології функції TE (Тип 2) (тривимірні та контурні графіки): а) функція TE ; б) $\left| \frac{\partial^2 TE(t_1, t_2)}{\partial t_1 \partial t_2} \right|$ функція; в) функція $\log(TE)$

Внизу цільової функції (8.25) є випуклість (рис. 8.17). Червона точка позначає положення глобального мінімуму. Вона на краю нижньої ями (рис. 8.17). Однак, як було зазначено раніше, топологія нижньої ями досить проста. Вона похила без будь-яких нелінійностей (каньйони, мінімуми з вузькими атракторами тощо).

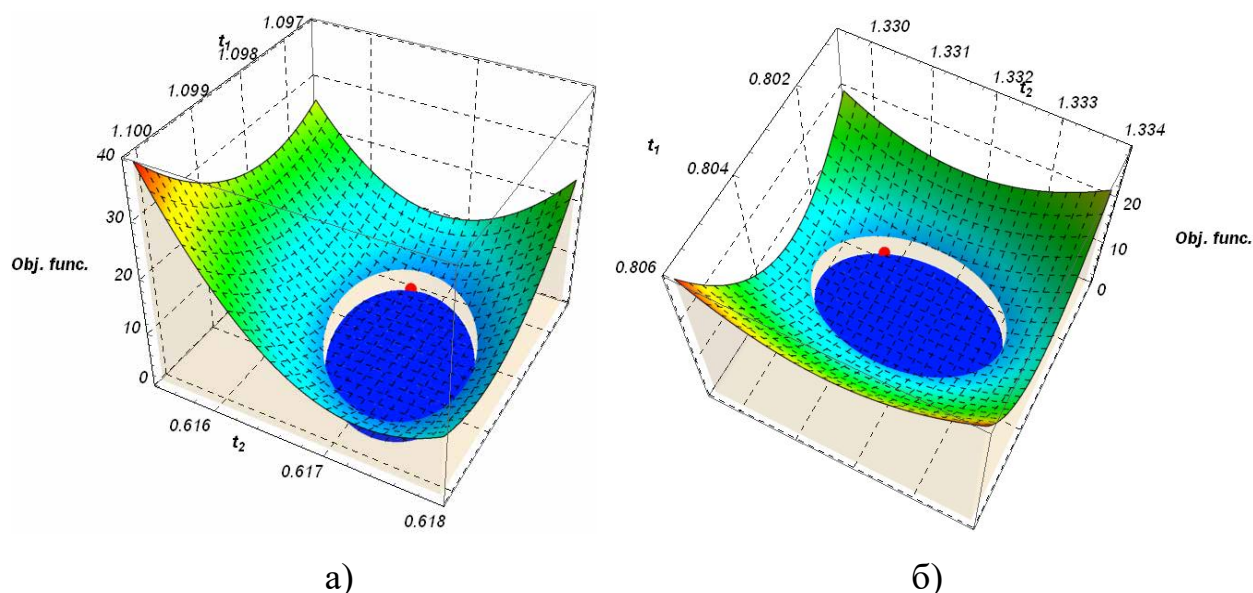


Рисунок 8.17 – Особливості топології цільової функції (3.25) біля її дна (глобальний мінімум) для задач: а) Типу 1; б) Типу 2

Ширина нижньої ями визначається величиною Δ . Як тільки алгоритм оптимізації знаходить нижню яму цільової функції (8.25), він прямує до глобального мінімуму, і задача оптимального керування розв'язана.

8.4 Аналіз результатів розв'язання задачі глобальної оптимізації

Для оцінки ефективності оптимізаційних алгоритмів застосовували їх до задачі (8.25) кілька разів з різними значеннями m_2 та l . У незалежних пусках алгоритмів величини m_2 змінювалися від 500 до 25000 кг з кроком 500 кг; l варіювався від 3 до 10 м з кроком 1 м. Таким чином, задачу розв'язували 400 разів (кожного разу з дещо різними параметрами m_2 або l). Умовно назвемо цю групу розв'язків задачі раундом.

У поточній роботі оцінили ефективність пошуку оптимізаційних метаевристичних алгоритмів. Розглядаються модифікації відомих (батьківських) метаевристичних алгоритмів (PSO, диференціальна еволюція – скорочено DE, оптимізація сірого вовка – GWO, пошук гармонії – HS та пошук зозулі – CS). Для кожного алгоритму кількість його агентів (частинки – для PSO-алгоритмів; вектори – для DE best/1/bin [31], DE rand/1/bin [83, 84] і SADE – для mGWO [85] і GWOEPD [86]; світляки – для FA [87]; гармонії – для ABHS [88] та PAHS [89]; зозулі – для ACS [90]) однакова, вона рівна 25. У табл. 8.3 наведено параметри застосованих алгоритмів оптимізації, окрім PSO-алгоритмів.

Таблиця 8.3 – Параметри оптимізаційних алгоритмів

Батьківський алгоритм	Загальні параметри алгоритму	Алгоритми	Конкретні параметри алгоритмів
DE	$CR=0,5$; $SF=0,6$	best/1/bin	-
		rand/1/bin	-
	-	SADE	$\tau_1=\tau_2=0,1$; $SF_{low}=0,1$; $SF_{up}=0,9$;
GWO	-	mGWO	-
		GWOEPD	-
FA	-	RaFA	$\alpha=1$
HS	-	ABHS	$HMCR_{max}=1,0$; $HMCR_{min}=0,9$; $PAR_{max}=1,0$; $PAR_{min}=0,3$
		PAHS	$HMCR_{max}=0,99$; $HMCR_{min}=0,7$; $PAR_{max}=0,99$; $PAR_{min}=0,01$; $BW_{min}=0,001$
CS	$\alpha=1$, $p_a=0,25$	ACS	-

Для кожного раунду кількість ітерацій алгоритму збільшувалася з 40 до 400. Індикатором для оцінки алгоритмів є певна кількість невдалих запусків алгоритму:

$$FAR = \left(\frac{OR}{SR} - 1 \right) 100\%, \quad (8.26)$$

де OR – загальна кількість пусків алгоритму; SR – кількість успішних пусків алгоритму. Успішний пусків алгоритму повинен відповідати наступній умові $T < 2\pi/\Omega_{\min}$, тобто кінцевий розв’язок задачі має бути нижчим від встановленого заздалегідь порогу (у практичному сенсі це запобігає прискоренню системи протягом періоду вільних коливань вантажу). Поставили жорсткішу умову і вибрати менший період, віднесений до розглянутих значень l і m_2 . У досліджуваному випадку $2\pi/\Omega_{\max} = 5,02$ с. Усі отримані дані, що відображають роботу алгоритмів оптимізації, наведені у табл. 8.4. та табл. 8.5.

Таблиця 8.4 – Чисельні значення оцінки FAR алгоритмів оптимізації окрім PSO-алгоритмів

Алгоритми Ітерації	best/1/bin	rand/1/bin	mGWO	GWOEPD	RaFA	ABHS
Задача оптимізації Тип 1						
40	4,2	*	3,5	37,2	64,2	*
100	3,8	8,0	5,2	53,0	19,2	*
200	3,0	5,0	3,8	*	6,2	77,0
400	4,5	7,5	3,2	69,0	4,8	22,8
Задача оптимізації Тип 2						
40	26,2	*	38,0	37,5	45,8	*
100	20,8	*	33,0	45,2	22,0	*
200	15,2	*	42,2	51,8	21,5	61,5
400	21,8	*	35,5	*	12,8	39,5

* тут і в табл. 8.5 показник FAR більше, ніж 100%. У деяких випадках алгоритм не знаходив усі необхідних розв’язків задач і його пуски переривалися.

Таблиця 8.5 – Чисельні значення оцінки *FAR* PSO-алгоритмів оптимізації

Ітерації \ Алгоритми	LDW-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-ND-PSO		PSO-Hybrid
						$\sigma_{\max}=0,35$	$\sigma_{\max}=0,02$	
Задача оптимізації Типу 1								
40	4,5	4,8	*	*	15,0	*	4,0	*
100	3,5	4,0	1,0	2,8	8,8	*	4,5	4,8
200	3,8	4,0	0,0	2,0	6,0	*	5,2	1,8
400	3,0	3,8	0,0	2,0	5,8	7,8	3,8	2,0
Задача оптимізації Типу 2								
40	12,5	16,8	*	*	24,5	*	14,2	*
100	13,2	20,5	1,0	26,8	23,2	*	8,0	9,0
200	19,2	15,2	1,0	27,2	22,2	82,8	11,5	8,5
400	16,0	14,0	0,0	9,8	21,2	10,5	10,2	6,8

Алгоритми CS, PAHS і SADE, IA-PSO для обох задач оптимізації показали погану пошукову активність. Їх ефективність низька, показник *FAR* для всіх них більше 100% і тому вони не були показані в табл. 8.4 та 8.5.

Успішні запуски різних алгоритмів привели до рішень (значення t_1 , t_2 і t_3), які відповідають глобальному мінімуму цільової функції (8.25). Це підтверджується значеннями t_1 , t_2 і t_3 : різні алгоритми знаходили майже однакові значення. Для ілюстрації рішень були побудовані (Додаток Є, рис. Є.1).

Жоден алгоритм, крім Ring-PSO, не досяг нульового значення індикатора *FAR* (ідеальний випадок, коли кожен запуск алгоритму призводить

до розв'язання проблеми). Таким чином, є можливості вдосконалення алгоритмів, щоб зменшити FAR до нуля.

Найефективнішими алгоритмами для розглянутих задач серед не PSO-алгоритмів є DE/best/1/bin та mGWO. Серед класу PSO-алгоритмів найкращі Ring-PSO, VCT-PSO, LDWPSO, PV-ND-PSO (при $\sigma_{\max}=0,02$) та PSO-Hybrid.

Крім того, варто відмітити сильний вплив параметра σ_{\max} на пошукову діяльність алгоритму PV-ND-PSO. У табл. 8.5 явно видно, що при $\sigma_{\max}=0,02$ пошукова діяльність цього алгоритму значно підвищилась. Наперед вказати раціональне значення для тієї чи іншої цільової функції σ_{\max} неможливо. Тому при розв'язанні тієї чи іншої оптимізаційної задачі необхідно виконувати декілька пробних пусків алгоритму PV-ND-PSO починаючи з налаштування $\sigma_{\max}=0,35$ і зменшуючи величину σ_{\max} до поки не буде досягнуто ефективного способу розв'язування задачі. Інший спосіб підвищення ефективності – змінювати величину σ незалежно від поточної ітерації, а від поточного зменшення цільової функції на певній ітерації.

Збільшення кількості ітерацій позитивно впливає на роботу алгоритмів. Усі вони (для обох випадків задач Типу 1 та Типу 2) за окремими виключеннями, які пов'язані із недостатньою кількістю ітерацій (40 та 100) досягли нижньої ями цільової функції. Єдина відмінність полягає в кількості необхідних для цього ітерацій.

Ці дані приводять нас до висновку: PSO-алгоритми (навіть відносно прості) підходять для розв'язання задач оптимізації Типу 1 і Типу 2.

Висновки до восьмого розділу:

1. Наведено 12 стандартних оптимізаційних задач інженерії, які характеризуються обмеженнями та незначною кількістю аргументів. Вони відносяться до галузі механічної інженерії та будівельних конструкцій. Їх розв'язання із використанням досліджуваних PSO-алгоритмів дає підстави вважати, що найбільш перспективними є VCT-PSO, Ring-PSO, R-Ring-PSO, ME-D-PSO, PV-ND-PSO та PSO-Hybrid.
2. Запропоновано два варіанти задачі глобальної оптимізації, яка ґрунтується на задачі оптимальної за швидкодією керування рухом динамічної системи „кран-вантаж”. Досліджено топологічні характеристики відповідних цільових функцій та встановлено, що їх атрактор є відносно вузьким, а самі функції є несепарабельними, мультимодальними та невивуклими.
3. Для дослідження пошукових особливостей алгоритмів оптимізації (9 не з класу PSO-алгоритмів та 8 PSO-алгоритмів) проведено багатократне розв'язання обох задач. Серед кращих алгоритмів за ефективністю локалізації глобальних мінімумів обох цільових функцій (для задач Типу 1 та Типу 2) є DE/best/1/bin та mGWO, а з класу PSO-алгоритмів Ring-PSO, VCT-PSO, LDWPSO, PV-ND-PSO (при $\sigma_{\max}=0,02$) та PSO-Hybrid.
4. Встановлено значний вплив на пошукову діляльність алгоритму PV-ND-PSO величини σ_{\max} . У рамках виконання досліджень на основі розв'язування задач Типу 1 та Типу 2 встановлено, що величину σ_{\max} необхідно задавати як величину сотих долей одиниці. Загальні рекомендації щодо розв'язання задач за допомогою алгоритму PV-ND-PSO передбачає багаторазовий запуск алгоритму із початковим налаштуванням $\sigma_{\max}=0,35$ з подальшим зменшенням цієї величини.

РОЗДІЛ 9

ЗАСТОСУВАННЯ PSO-АЛГОРИТМІВ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧ КЕРУВАННЯ РУХОМ МЕХАНІЗМІВ ВАНТАЖОПІДЙОМНИХ МАШИН ТА РОБОТОТЕХНІКИ

9.1 Параметрична оптимізація частотно-керованого пуску динамічної системи «візок-вантаж» механізму зміни вильоту баштового крана

Надійність та ефективність експлуатації при роботі динамічної системи «візок-вантаж» механізму зміни вильоту баштового крана є однією із ключових проблем під час будівництва (відбудови) будівельних споруд різного типу й складності.

Покращити величину вказаних вище показників, можна за рахунок мінімізації величини небажаних динамічних та енергетичних навантажень. Для вирішення цієї проблеми необхідно провести параметричну оптимізацію частотно-керованого електроприводу досліджуваної динамічної системи. Актуальною проблемою на сьогоднішній день є питання підвищення продуктивності та надійності при роботі вантажопідіймальних кранів.

Зокрема, авторами статті [131], виконано розробку методу, який дозволяє максимізувати продуктивність крана шляхом планування його оптимальної траєкторії руху. Для створення моделі механізму підйому вантажу використано рівняння Лагранжа. На основі моделі розроблено метод планування траєкторії руху вантажу на гнучкому підвісі.

У праці [132] пропонується зменшувати амплітуду коливання вантажу, закріпленого на гнучкому підвісі, за рахунок регулювання кута повороту баштового крана. Автори приділяють увагу двом підходам при керуванні системою із відкритим контуром на основі формування сигналу керування і закритим контуром керування кутом повороту на основі величини відхилення закріпленого вантажу.

У роботі [133] запропоновано покращувати умови праці кранівника за допомогою розробленої системи керування коливаннями вантажу. В основу системи входить визначення кута відхилення від вертикалі гнучкого підвісу вантажу за допомогою інерційного вимірювального блоку. Робота системи керування є можливою лише тоді, коли відомо в реальному часі положення вантажу на гнучкому підвісі.

У статті [134] наведено оптимальне керування механізмом повороту крана стрілового типу за критерієм швидкодії. Для виконання теоретичних досліджень приймалася нелінійна математична модель руху крана у часі, в якій змінним параметром була використана довжина гнучкого підвісу вантажу.

У праці [135] дослідниками представлено систему нелінійного керування стріловим краном за допомогою контролера із функцією зворотного зв'язку. Дана система керування має можливість мінімізувати розгойдування вантажу, але не враховує силу вітрових поривів.

У статті [136] дослідниками представлено математичну модель лебідки механізму підйому вантажу стрілового крана, яка дозволяє здійснювати рекуперацію електричної енергії. Основна увага приділяється величині втрат енергії у двигуні при перехідних процесах руху. Виконано підтвердження теоретичних розрахунків за допомогою експериментальних досліджень.

У праці [137] пропонується виконувати швидке та точне позиціонування вантажу на гнучкому підвісі для стрілового крана із мінімізацією енергетичних втрат у приводі.

При проведенні теоретичних досліджень використано метод лінійної алгебри, а математична модель описується рівняннями подвійного математичного маятника.

В роботі [138] автори пропонують використовувати алгоритм оптимального керування механізмом вантажопідіймального крана, що дає можливість мінімізувати затрачену електричну енергію.

Робота алгоритму здійснюється за рахунок вимкнення кінематичного зв'язку між двигуном та візком через встановлення електромеханічної муфти у приводі. А це в свою чергу, вимагає додаткових змін до конструкції, що не завжди є бажаним.

В статті [139] представлено метод здійснення керування механізмами щоглового крана по оптимальній траєкторії за енергоефективністю. Пропонована оптимальна траєкторія не значно мінімізує небажані енергетичні навантаження, а її практична реалізація супроводжується значними коливаннями системи.

У статті [140], використовуючи модифікований алгоритм оптимізації роєм часточок (ME-PSO), виконано мінімізацію втрат електроенергії у частотно-керованому електроприводі, під час першого етапу пуску механізму зміни вильоту вантажу баштового крана.

Отримані результати порівняно із результатами при прямому пуску та при пуску за стандартно налаштованими параметрами частотно-керованого електроприводу досліджуваного механізму.

Із аналізу існуючих досліджень видно, що питанням, які присвячені дослідженню оптимальних за динамічними та енергетичними показниками параметрів частотно-керованих приводів вантажопідіймальних кранів приділено не достатньо уваги.

Мета роботи полягає у дослідженні та мінімізації небажаних динамічних та енергетичних показників динамічної системи «візок-вантаж» механізму зміни вильоту баштового крана під час перехідного режиму пуску. Для досягнення поставленої мети необхідно вирішити наступні завдання:

- 1) здійснити постановку задач оптимізації;
- 2) провести мінімізацію оптимізаційних критеріїв;
- 3) виконати аналіз отриманих результатів.

Для виконання досліджень використаємо існуючу чотиримасову динамічну модель руху системи «візок-вантаж» механізму зміни вильоту баштового крана [141-143], яку представлено на (рис. 9.1).

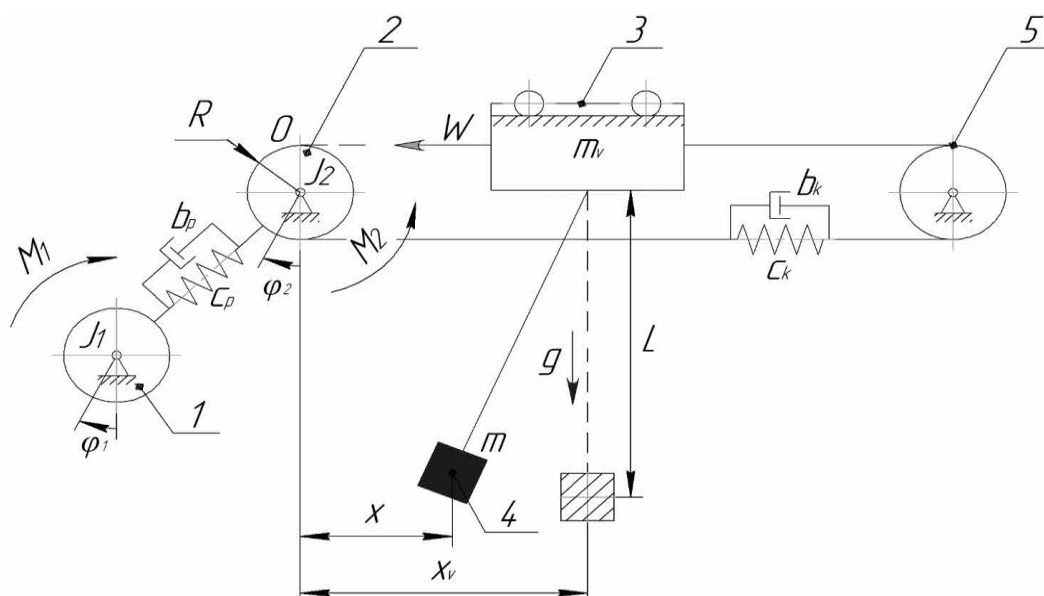


Рисунок 9.1 – Розрахункова чотиримасова модель динамічної системи «візок-вантаж» механізму зміни вильоту баштового крана

На рис. 9.1, використано наступні позначення: 1 – електродвигун приводу канатного барабану; 2 – канатний барабан приводу вантажного візка; 3 – вантажний візок; 4 – вантаж, що закріплений на гнучкому підвісі; 5 – канатний блок; M_1 – зведений до канатного барабану рушійний момент приводу електродвигуна; M_2 – зведений момент опору на приводному барабані, який є змінною величиною; R – радіус канатного барабану; W – зведена сила статичного опору переміщення вантажного візка, що є змінною величиною; J_1 та J_2 – зведені до канатного барабану моменти інерції, що обертаються із кутовою швидкістю двигуна та канатного барабану відповідно; L – довжина гнучкого підвісу закріпленого вантажу; m_v та m – зведені маси вантажного візка та вантажу відповідно; c_p , c_k – відповідно зведені коефіцієнти жорсткості приводу канатного барабана і тягового канату; b_p , b_k – зведені коефіцієнти демпфування приводів канатного барабану та канату вантажного візка відповідно; g – прискорення вільного падіння; φ_1 та φ_2 – зведені узагальнені кутові координати повороту ротора електродвигуна та канатного барабану відповідно; x_v , x – зведені узагальнені лінійні координати переміщення центрів мас вантажного візка та вантажу відповідно [141, 142]. В динамічній моделі (рис. 9.1), прийнято наступні припущення: 1) величиною

моменту інерції канатного блока 5 нехтуємо; 2) закріплений на гнучкому підвісі вантаж виконує маятникові коливання; 3) величиною слабину тягового каната нехтуємо; 4) вітрові навантаження зведено до величини W (сила статичного опору переміщення візка); 5) розглядається плоска задача (всі елементи рухаються у горизонтальній площині); 6) зведені коефіцієнти жорсткості c_p і c_k та зведені коефіцієнти дисипації b_p й b_k , приймаємо незмінними при перехідних режимах руху досліджуваної системи; 7) усі компоненти системи вважаються абсолютно жорсткими тілами, окрім приводу канатного барабану і тягового канату, які мають відповідні пружно-дисипативні властивості [141, 142]. Рух у часі чотиримасової моделі динамічної системи «візок-вантаж» (рис. 9.1) описується за допомогою системи диференціальних рівнянь, які доповнено рівняннями руху асинхронного електродвигуна. У результаті проведених підстановок отримано узагальнену систему диференціальних рівнянь, що дає змогу описати рух динамічної системи «візок-вантаж» механізму зміни вильоту та її асинхронного електродвигуна у часі [140-142]:

$$\left\{ \begin{array}{l} \frac{di_{1\alpha}}{dt} = \frac{1}{\delta \cdot L_S} \cdot (u_{1\alpha} - i_{1\alpha} \cdot R_S + k_r \cdot e_{2\alpha}); \\ \frac{di_{1\beta}}{dt} = \frac{1}{\delta \cdot L_S} \cdot (u_{1\beta} - i_{1\beta} \cdot R_S - k_r \cdot e_{2\beta}); \\ \frac{di_{2\alpha}}{dt} = -\frac{1}{\delta \cdot L_R} \cdot ((u_{1\alpha} - i_{1\alpha} \cdot R_S) \cdot k_s + e_{2\alpha}); \\ \frac{di_{2\beta}}{dt} = -\frac{1}{\delta \cdot L_R} \cdot ((u_{1\beta} - i_{1\beta} \cdot R_S) \cdot k_s - e_{2\beta}); \\ \frac{3}{2} \cdot p \cdot L_W \cdot (i_{1\beta} \cdot i_{2\alpha} - i_{1\alpha} \cdot i_{2\beta}) \cdot U_P \cdot \eta_P = J_1 \cdot \ddot{\varphi}_1 + c_p \cdot (\varphi_1 - \varphi_2) + \\ + b_p \cdot (\dot{\varphi}_1 - \dot{\varphi}_2); \\ M_2 \cdot \text{sign}(\dot{\varphi}_2) + J_2 \cdot \ddot{\varphi}_2 = c_p \cdot (\varphi_1 - \varphi_2) - b_p \cdot (\dot{\varphi}_1 - \dot{\varphi}_2) - c_k \cdot (\varphi_2 \cdot R - x_v) \times \\ \times R - b_k \cdot (\dot{\varphi}_2 \cdot R - \dot{x}_v) \cdot R; \\ W \cdot \text{sign}(\dot{x}_v) + m_v \cdot \ddot{x}_v = c_k \cdot (\varphi_2 \cdot R - x_v) + b_k \cdot (\dot{\varphi}_2 \cdot R - \dot{x}_v) - m \cdot \frac{g}{L} \cdot (x_v - x); \\ \ddot{x} = \frac{g}{L} \cdot (x_v - x), \end{array} \right. \quad (9.1)$$

де $i_{1\alpha}$ і $i_{1\beta}$ – проєкції узагальненого вектору струму статора на нерухомі ортогональні координатні осі α та β ; $u_{1\alpha}$, і $u_{1\beta}$ – проєкції узагальненого вектору напруги статора на координатні осі α й β ($u_{1\alpha} = U_{MAX} \cdot \cos(2 \cdot \pi \cdot \int f dt)$, $u_{1\beta} = U_{MAX} \cdot \sin(2 \cdot \pi \cdot \int f dt)$); $i_{2\alpha}$ та $i_{2\beta}$ – проєкції узагальненого вектору струму ротора на ортогональні осі α і β відповідно; U_{MAX} – амплітуда фазної напруги живлення електродвигуна; f – частота напруги живлення електродвигуна; $e_{2\beta}$, та $e_{2\alpha}$ – ЕРС, котрі індуюються потокозчепленнями ротора по осях α та β відповідно ($e_{2\alpha} = p \cdot \omega_{dv} \cdot (L_R \cdot i_{2\beta} + L_W \cdot i_{1\beta}) + i_{2\alpha} \cdot R_R$; ($e_{2\beta} = p \cdot \omega_{dv} \cdot (L_S \cdot i_{2\alpha} + L_W \cdot i_{1\alpha}) + i_{2\beta} \cdot R_R$); ω_{dv} – кутова швидкість обертання ротора електродвигуна механізму; p – кількість пар полюсів електричної машини; R_S – активний опір статорної обмотки; R_R – зведений до статора активний опір роторної обмотки електродвигуна; δ – коефіцієнт розсіювання $\delta = 1 - (1 + X_1 (2\pi f L_W)^{-1}) \cdot (1 + X_2 \cdot (2\pi f L_W)^{-1})^{-1}$; X_1 – індуктивний опір статорної обмотки; X_2 – зведений до статора індуктивний опір роторної обмотки електродвигуна; L_S і L_R – індуктивності статорної та роторної обмоток відповідно; L_W – взаємодуктивність; k_s та k_r – коефіцієнти магнітного зв'язку статора та ротора відповідно, що визначаються наступним чином: $k_s = L_W \cdot L_S^{-1}$ і $k_r = L_W \cdot L_R^{-1}$; U_P – передаточне число приводу механізму зміни вильоту вантажу; η_P – ККД приводу механізму зміни вильоту вантажу [140-144].

Початкові умови руху для узагальненої системи диференціальних рівнянь (1) є нульовими (динамічна система знаходиться у стані спокою) та мають наступний вигляд [140-145]:

$$\begin{cases} i_{1\alpha}(0) = i_{1\beta}(0) = i_{2\alpha}(0) = i_{2\beta}(0) = 0; \\ \varphi_1(0) = \dot{\varphi}_1(0) = \varphi_2(0) = \dot{\varphi}_2(0) = x_g(0) = \dot{x}_g(0) = x(0) = \dot{x}(0) = 0. \end{cases} \quad (9.2)$$

Чисельне інтегрування системи диференціальних рівнянь (9.1), здійснено за параметрами, що зведено до табл. 9.1.

Таблиця 9.1 – Параметри системи «візок-вантаж» механізму зміни вильоту

Параметр	Розмірність	Значення	Параметр	Розмірність	Значення
M_2	Н	13,4	m_v	кг	250
U_P	-	29,06	m	кг	10000
R	м	0,14	C_p	Нм/рад	$28,5 \cdot 10^3$
W	Н	2011	c_k	Н/м	$3,12 \cdot 10^5$
J_1	кгм ²	96,2	b_p	Нс/рад	27484,3
J_2	кгм ²	160	b_k	Нс/м	17575,2
L	м	10	g	м/с ²	9,81
η_P	-	0,89	P_{NOM}	кВт	7,5
p	-	3	ω_{dv}	рад/с	97,8

Оскільки керування електроприводом значної кількості сучасних механізмів баштового крана здійснюється за допомогою частотного перетворювача, який володіє великою кількістю опцій налаштувань, тому виконаємо налаштування найбільш суттєвих його параметрів [140, 144].

Для частотно-керованого електроприводу динамічної системи «візок-вантаж» механізму зміни вильоту найбільш суттєвими опціями є наступні:

- тривалість наростання частоти напруги живлення до усталеного значення t_1 (знаходиться у межах від 0 до 3600 с) [140, 144];
- величина початкової напруги живлення електродвигуна U_0 (від 0 до 380 В) [140, 145];
- тип характеристики наростання частоти напруги живлення f до відповідного усталеного значення $f_{UST}=50$ Гц (лінійна, S-подібна, 2S-подібна, U-подібна) [140, 144].

Для того, щоб електропривод досліджуваної системи на початку пуску подолав сили статичного опору, необхідно забезпечити потрібну перевантажувальну здатність електроприводу, що залежить від зміни частоти напруги живлення [144]. Залежність напруги живлення електродвигуна

системи від U від частоти напруги живлення f можна визначити за наступною залежністю [140, 144, 145]:

$$U = U_0 + (U_{NOM.} - U_0) \cdot \frac{f}{f_{UST.}}, \quad (9.3)$$

де $U_{NOM.}$ – номінальна напруга живлення електродвигуна системи (380 В).

Вираз, котрий дозволяє визначити величину середньоквадратичного значення крутного моменту у пружній муфті приводу динамічної системи в залежності від величини параметрів t_1 , U_0 та f , має наступний вигляд:

$$M_{MUF.} = f_1(t_1, U_0, f) = \left[\int_0^{t_1} (c_p \cdot (\varphi_1 - \varphi_2))^2 dt \right]^{\frac{1}{2}} \rightarrow \min, \quad (9.4)$$

де f_1 – функція параметрів регулювання; t – час.

Залежність, яка дозволяє визначити середньоквадратичне значення зусилля у тяговому канаті приводу вантажного візка динамічної системи в залежності від t_1 , U_0 та f , має наступний вигляд:

$$F_{KAN.} = f_1(t_1, U_0, f) = \left[\int_0^{t_1} (c_k \cdot (\varphi_2 \cdot R - x_v))^2 dt \right]^{\frac{1}{2}} \rightarrow \min. \quad (9.5)$$

Залежність, яка дозволяє встановити середньоквадратичне значення потужності у електроприводі динамічної системи у залежності від t_1 , U_0 та f , має наступний вигляд:

$$P = f_1(t_1, U_0, f) = \left[\int_0^{t_1} (M_1 \cdot \dot{\varphi}_1)^2 dt \right]^{\frac{1}{2}} \rightarrow \min. \quad (9.6)$$

Вирази (9.4)-(9.6) виступали у якості оптимізаційних критеріїв при подальших дослідженнях. Враховуючи те, що максимальна величина оптимізаційних критеріїв (9.4)-(9.6) для досліджуваної системи напряму залежала від значень t_1 , U_0 та f , то було знайдено такі значення цих параметрів, при яких величини $M_{MUF.}$, $F_{KAN.}$ та P є мінімальними. Для цього використано

модифікований метод оптимізації роєм часточок R-Ring-PSO [71]. При цьому досліджувана система розглядалася як MISO-функція (рис. 9.2), у якій в якості входу виступали величини t_1 , U_0 та f , а виходами були величина критеріїв (M_{MUF} , F_{KAN} та P). При цьому для кожного з критеріїв оптимізації, значення параметрів (t_1 , U_0 та f) встановлювалися індивідуально [140, 144].

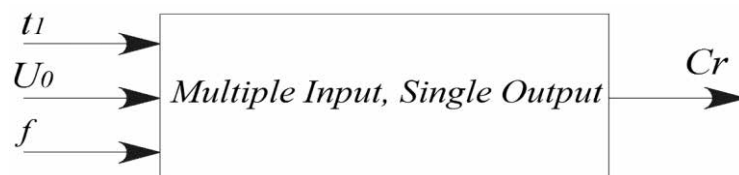


Рисунок 9.2 – Умовне зображення MISO-функції для проведення параметричної оптимізації

Величини параметрів, при яких буде проведена параметрична оптимізація, зведено до табл. 9.2.

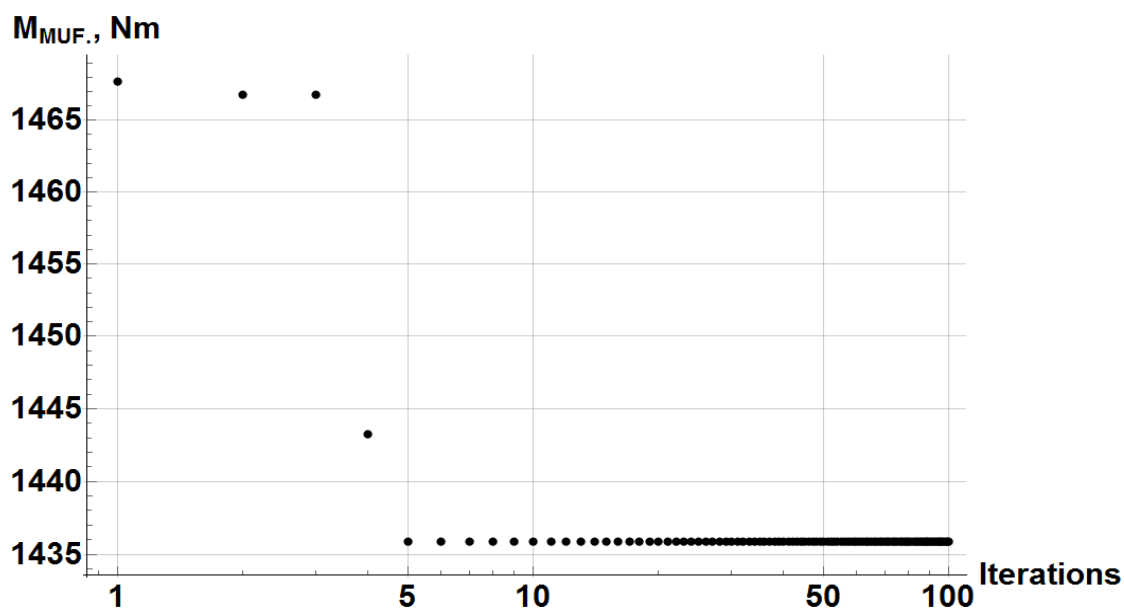
Таблиця 9.2 – Основні значення параметрів для проведення оптимізації

Параметри	Значення або коментар
Загальна кількість частинок у рої	40
Загальна кількість ітерацій	100
t_1 , с	від 0,5 до 4,0
U_0 , В	від 0,01 до 380
f	Для цього аргументу задана відповідність „значення аргументу – характеристика наростання частоти напруги живлення”: 1 – лінійна; 2 – S-подібна; 3 – 2S-подібна; 4 – U-подібна

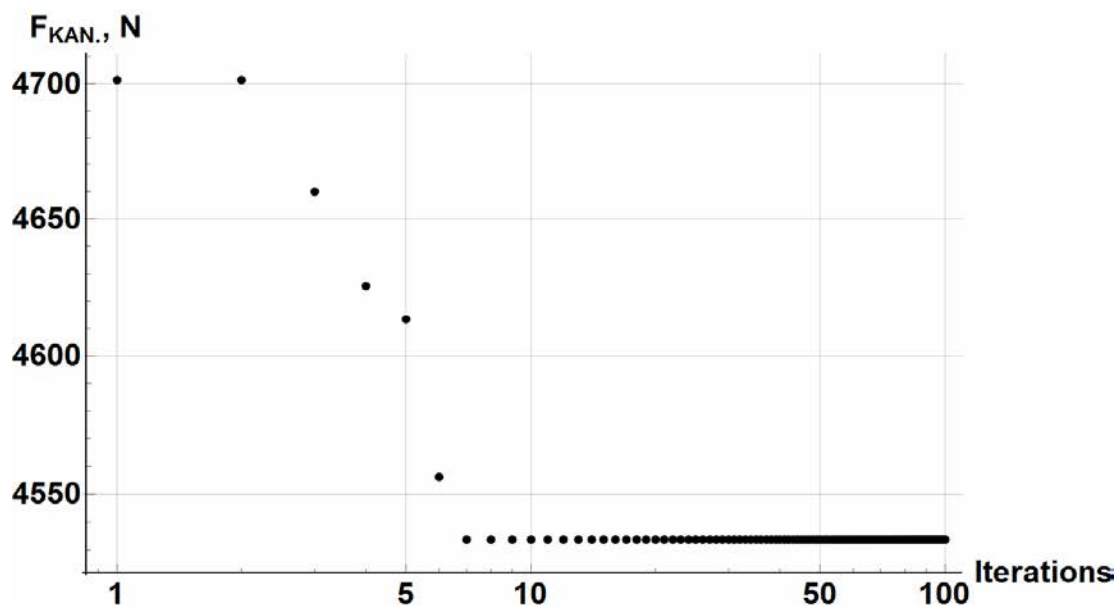
Метод оптимізації роєм часточок R-Ring-PSO [71] дозволив визначити відповідний порядковий номер (від 1 до 4), що відповідає бажаному типу характеристики для певного оптимізаційного критерію.

В результаті використання методу R-Ring-PSO виконано мінімізацію оптимізаційних критеріїв M_{MUF} , F_{KAN} та P . Результати роботи методу оптимізації наведено на рис. 9.3.

Аналізуючи наведені графіки функцій (рис. 9.3), можливо зробити висновок, що глобальні мінімуми оптимізаційних критеріїв (9.4)-(9.6) при використанні методу R-Ring-PSO знаходяться до десятої ітерації.



а)



б)

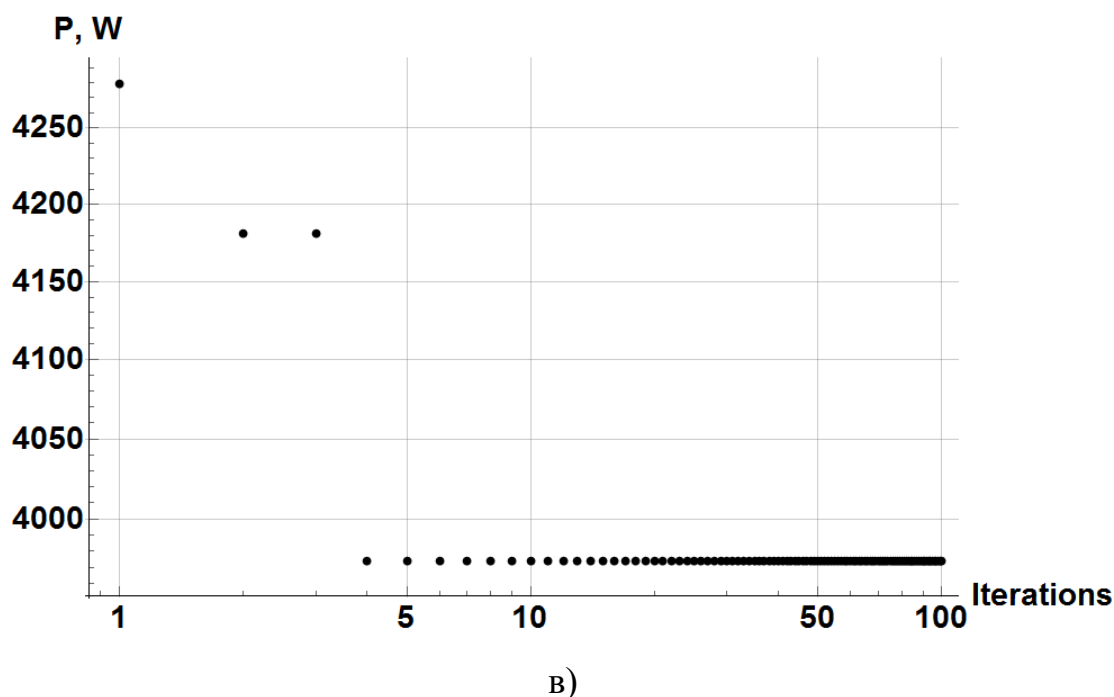


Рисунок 9.3 – Графічні залежності зміни величин при здійсненні параметричної оптимізації: а) крутного моменту в муфті; б) тягового зусилля у канаті; в) потужності у приводі

Оптимальні значення налаштування параметрів t_1 , U_0 та f частотно-керованого електроприводу зведено до табл. 9.3.

Таблиця 9.3 – Оптимальні значення параметрів частотно-керованого приводу

Параметри	Тип оптимізаційного критерію		
	$M_{MUF.}$	$F_{KAN.}$	P
t_1, c	4.0	0.60	4.0
U_0, B	380	0.01	0.01
f	Лінійна	2S-подібна	U-подібна

Окрім того, за результатами, які представлено у табл. 9.3, побудовано відповідні графічні залежності. При оптимізації за критерієм (9.4) наведено графічні залежності, які описують характер зміни крутного моменту у пружній муфті системи (залежність чорного кольору) та кутову швидкість електродвигуна (сірого кольору залежність) (рис. 9.4).

$M_{\text{МУФ.}}, \text{Nm}; \phi_1 \times 5, \text{rad/s}$

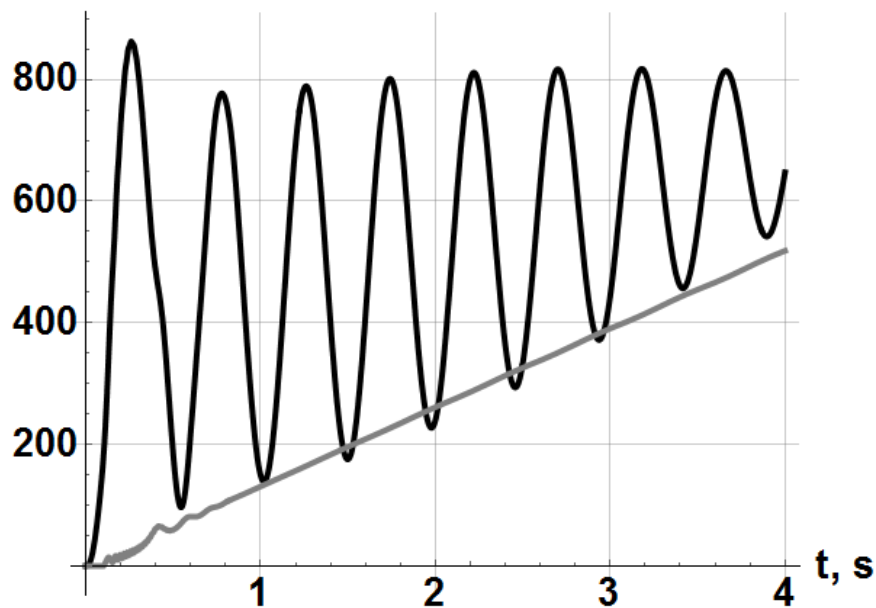


Рисунок 9.4 – Графічні залежності зміни функцій крутного моменту в пружній муфті та кутової швидкості електроприводу для критерію (9.4)

Для критерію оптимізації (9.5) представлено графічні залежності, які описують характер зміни зусилля у тяговому канаті та кутову швидкість електродвигуна (рис 9.5).

$F_{\text{КАН.}}, \text{N}; \phi_1 \times 20, \text{rad/s.}$

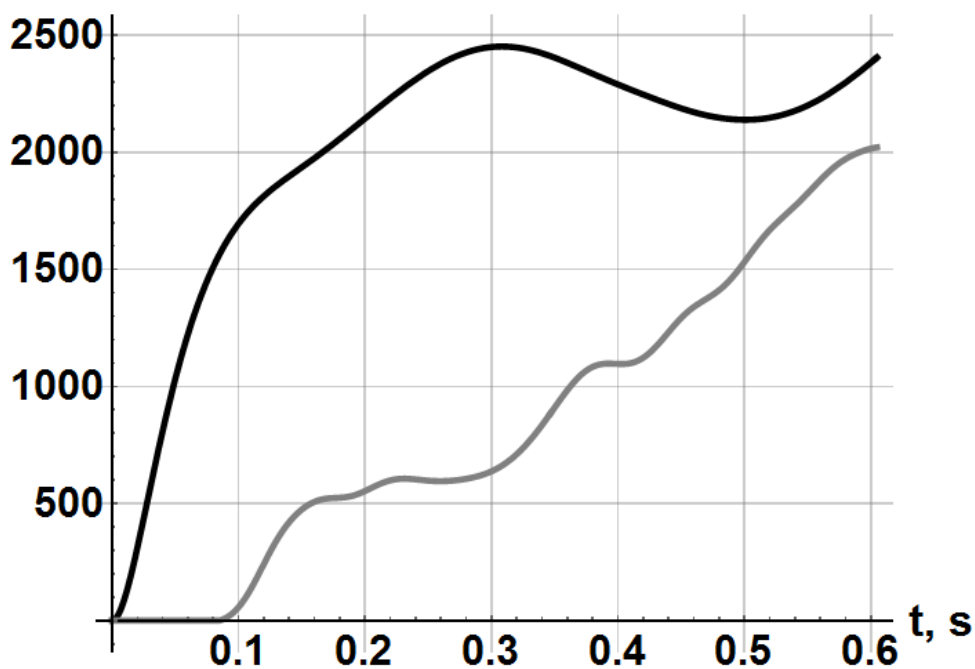


Рисунок 9.5 – Графічні залежності зміни функцій зусилля у тяговому канаті та кутової швидкості електроприводу для критерію (9.5)

При оптимізації за критерієм (9.6) представлено графічні залежності, які описують характер зміни потужності у електроприводі та кутову швидкість ротора електродвигуна (рис. 9.6).

P, W; $\phi_1 \times 15$, rad/s

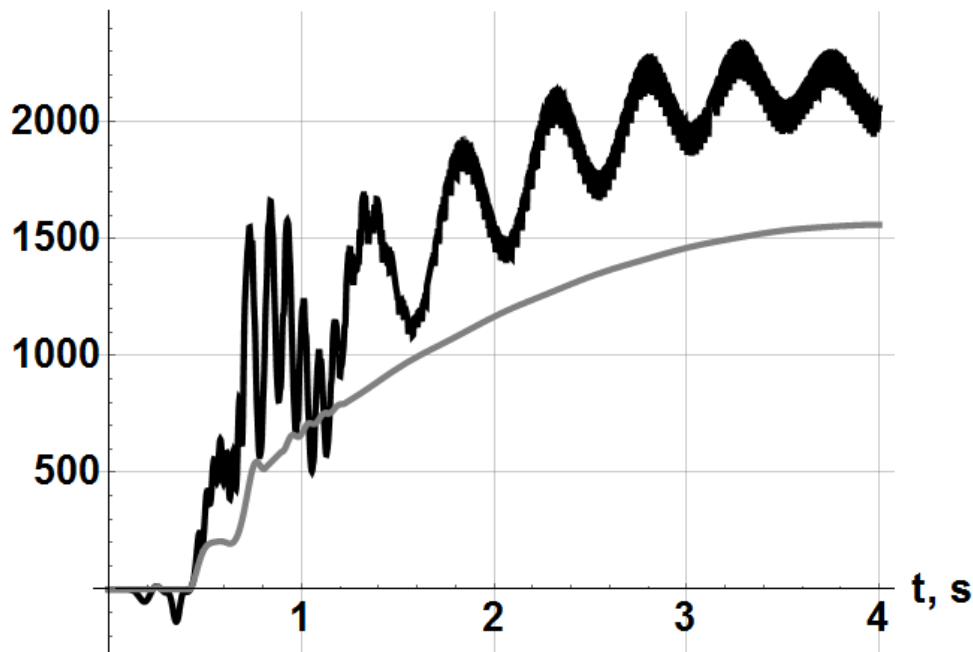


Рисунок 9.6 – Графічні залежності зміни функцій потужності у електродвигуні та кутової швидкості ротора електроприводу для критерію (9.6)

Аналіз графічних залежностей, наведених на рис. 9.4-9.6, дозволив сформулювати наступні висновки. Наростання величини кутової швидкості частотно-керованого електроприводу до усталеного значення для усіх критеріїв оптимізації відбувається плавно (за виключенням початкового пуску). Це дозволяє підвищити ресурс експлуатації електродвигуна досліджуваної системи.

Спостерігається певний коливальний характер крутного моменту в пружній муфті (рис. 9.4), який пов'язаний із виходом системи зі стану спокою, однак величина його є не суттєвою. Значення зусилля у тяговому канаті та потужності у електроприводі (рис. 9.5-9.6) мають плавний характер наростання, що позитивно впливає на надійність та довговічність досліджуваної системи.

Для оцінки повноти картини проведеної параметричної оптимізації в таблиці 4 наведено значення величин оптимізаційних критеріїв (9.4)-(9.6) при

використанні різних типів характеристик. Найнижчі значення показників оптимізаційних критеріїв виділено жирним шрифтом.

Таблиця 9.4 – Значення критеріїв оптимізації за різних типів характеристик

Тип характеристики f	Критерії оптимізації		
	M_{MUF} , Нм	F_{KAN} , Н	P , Вт
Лінійна	1435,9	4685,9	4484,4
S-подібна	1689,1	4874,8	5714,9
2S-подібна	1439,9	4533,7	4996,1
U-подібна	1980,5	4764,1	3974,1

Аналіз наведених даних в табл. 9.4 підтверджує відповідність встановлених типів характеристик наростання частоти напруги живлення (табл. 9.3), оскільки при їх використанні спостерігаються мінімальні значення досліджуваних критеріїв. Так, наприклад, значення критерію M_{MUF} при лінійній характеристиці на 27,9 % менше у порівнянні із аналогічним значенням за U-подібної характеристики. Значення критерію F_{KAN} за використання 2S-подібної характеристики на 6,9 % менше у порівнянні із аналогічним значенням за S-подібної характеристики. Величина критерію P за U-подібної характеристики на 30,4 % менше у порівнянні із тотожним значенням при S -подібній характеристиці.

9.2 Оптимізація за енергетичним критерієм сумісного пуску механізмів зміни вильоту та повороту баштового крану

Одним з шляхів підвищення продуктивності баштових кранів є суміщення роботи механізмів. До таких механізмів, які досить часто здійснюють суміщення в роботі, відносяться механізми зміни вильоту та повороту крана. При суміщенні роботи цих механізмів в елементах конструкції та приводних механізмів виникають динамічні перевантаження. Особливо небезпечними ці перевантаження є тоді, коли механізми зміни

вильоту та повороту одночасно здійснюють перехідні процеси (пуск, гальмування, зміна швидкості руху). В цей час стають некерованими просторові коливання вантажу на гнучкому підвісі, які впливають на стійкість роботи крана. Робота механізмів в такому режимі приводить до збільшення енергетичних витрат та зниження надійності роботи окремих механізмів та крана в цілому. Крім того, це створює небезпеку для обслуговуючого персоналу та ускладнює роботу кранівника.

Тому вибір режимів руху приводів при сумісній роботі механізмів, які мінімізують динамічні навантаження і усувають коливання елементів конструкції крана та вантажу на гнучкому підвісі є досить актуальною задачею, яка відповідає вимогам роботи баштових кранів.

При роботі кранових механізмів виникають значні динамічні навантаження, які мають суттєвий вплив на продуктивність та надійність вантажопідйомних кранів. Питанням дослідження динаміки вантажопідйомних кранів, зокрема баштових, присвячена значна кількість праць, серед яких можна виділити такі з них [146-150]. В наукових працях [148-150] досліджувались динамічні процеси в баштових кранах, а в роботах [147, 150] – мостового типу і, зокрема, контейнерних перевантажувачів з елементами керування рухом. В роботі [151] здійснено моделювання динаміки поворотного крана і керування приводними механізмами.

Останнім часом значна увага приділяється дослідженням динаміки спільного руху кранових механізмів [152-158]. В роботах [155-158] досліджувалась динаміка механізмів підйому та зміни вильоту вантажу для різних типів і конструкцій вантажопідйомних кранів. При цьому встановлювались причини виникнення коливань вантажу на гнучкому підвісі. Значна увага приділяється дослідженням динаміки сумісного руху механізмів зміни вильоту та повороту вантажопідйомних кранів [152-154]. В роботах [152, 153] розглянуто динаміку спільного руху механізмів зміни вильоту вантажу та повороту баштового крана. Для мінімізації коливань вантажу на гнучкому підвісі здійснювалось керування приводним електродвигуном

механізму зміни вильоту при усталеному обертанні механізму повороту крана [154]. В роботі [159] розроблено математичну модель динаміки сумісного руху механізмів зміни вильоту та повороту баштового крана з балочною стрілою. На базі розробленої моделі досліджено динамічні процеси роботи двох механізмів, які дозволили визначити їхні кінематичні, силові та енергетичні характеристики. При цьому також досліджені просторові маятникові коливання вантажу на гнучкому підвісі та виявлені динамічні перевантаження механізмів зміни вильоту і повороту крана [157].

Для зменшення коливань вантажу на гнучкому підвісі розв'язувались оптимізаційні задачі при роботі окремих механізмів [160-162]. Так, наприклад, в дослідженнях [160] вирішувалась оптимізаційна задача мінімізації потужності приводного механізму та усунення коливань вантажу на гнучкому підвісі при роботі механізму повороту крана. Для проведення оптимізації режиму повороту крана використовувався комплексний критерій. В роботі [162] оптимізовано процес пуску механізму зміни вильоту баштового крана при усталеному режимі повороту, де за критерій оптимізації використано середньоквадратичне значення рушійного моменту приводу. В роботі [161] для зменшення високочастотних коливань в приводі механізму переміщення вантажного візка при усталеному повороті крана здійснено оптимізацію режиму руху за критерієм середньоквадратичного значення швидкості зміни рушійного моменту. Така оптимізація приводить до мінімізації динамічних навантажень на елементи приводних механізмів та крана в цілому. Задачі зменшення коливань в баштових кранах вирішуються за рахунок вибору режимів руху приводних механізмів. В роботі [163] така задача вирішувалась для механізму зміни вильоту вантажу при усталеному повороті баштового крана шляхом оптимізації режиму пуску приводного механізму з використанням комплексного критерію. Тому вибір режимів руху приводних механізмів, що зменшують коливання ланок і підвищують ефективність роботи вантажопідійомних кранів, є досить актуальним і потребує більш повних досліджень.

Важливим напрямком досліджень вантажопідйомних кранів є оптимізація планування траєкторій руху кранових механізмів [164-167]. В роботах [164, 165] розв'язувались задачі оптимізації планування траєкторій переміщення контейнерним краном при дії динамічних навантажень з системою усунення розгойдування вантажу на гнучкому підвісі. Авторами роботи [166] вирішувалась оптимізаційна задача стратегії керування для мінімізації коливань вантажу на гнучкому підвісі. В роботі [167] здійснено оптимізацію планування рухом мостових кранів з гнучким підвісом змінної довжини. В роботі [168] розв'язувалась задача оптимального керування крановими системами із вхідним насиченням та обмеженнями стану системи.

Аналіз проведених досліджень динаміки та оптимізації вантажопідйомних кранів показує, що вибір оптимальних траєкторій та режимів руху приводних механізмів, що зменшують коливання ланок і підвищують ефективність роботи вантажопідйомних кранів, є досить актуальним і потребує більш повних досліджень.

9.2.1 Постановка задачі оптимального керування

Стрілову систему баштового крана представлено механічною системою (рис. 4.7), до складу якої входять зведені до осі повороту крана абсолютно тверді маси приводного механізму повороту з моментом інерції I_1 та поворотної частини крана з моментом інерції I_2 , які з'єднані між собою пружним елементом з коефіцієнтом жорсткості C_1 .

Зведена до осі повороту барабана абсолютно жорстка маса приводу механізму зміни вильоту з моментом інерції I_3 з'єднана з візком масою m_3 пружним канатом з коефіцієнтом жорсткості C_3 чи C_3' в залежності від напрямку руху візка. Вантаж масою m з'єднано з центром мас візка за допомогою гнучкого підвісу з незмінною довжиною H під час руху механізмів зміни вильоту та повороту крана.

На стрілову систему крана діють рушійні моменти приводних механізмів повороту крана M_1 та зміни вильоту вантажу M_3 , а також момент сил опору M_2 в поворотній частині крана і сила опору переміщення візка W .

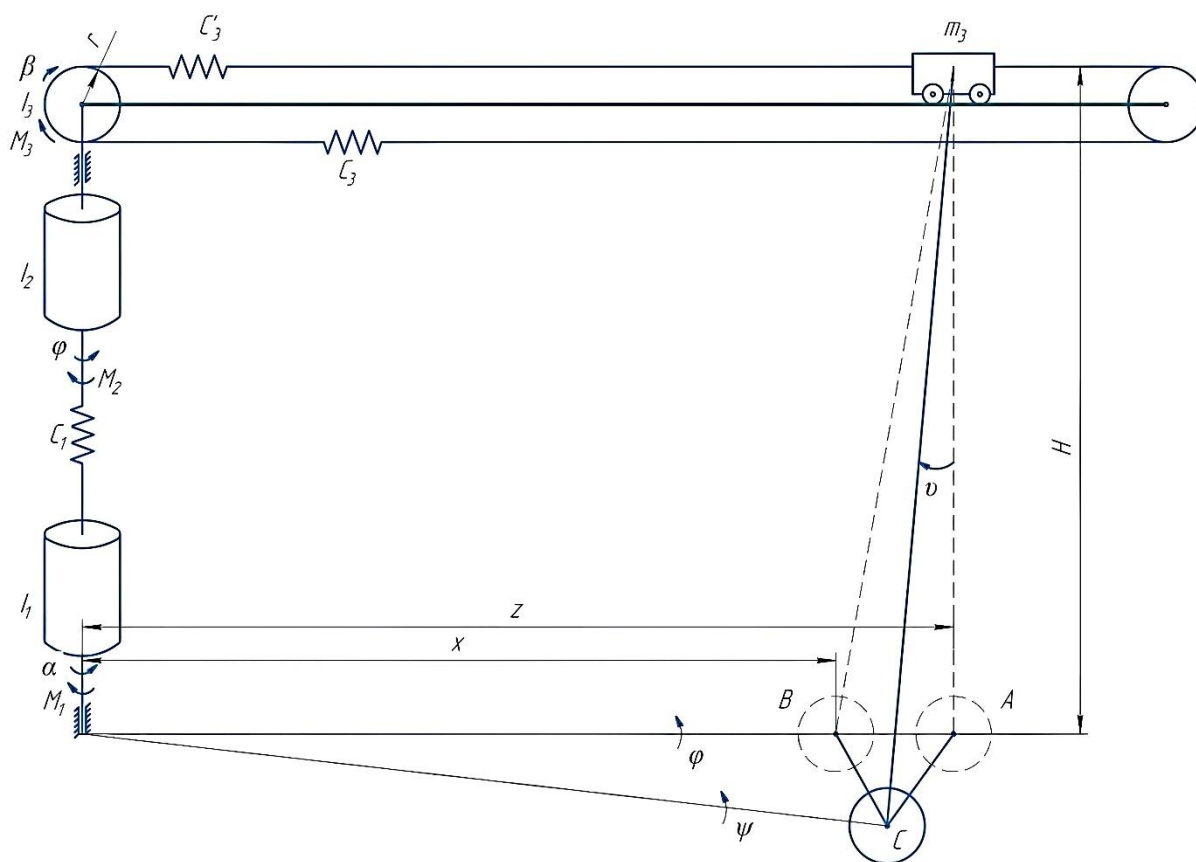


Рисунок 9.7 – Динамічна модель стрілової системи крана при сумісному русі механізмів зміни вильоту та повороту

В наведеній динамічній моделі (рис. 9.7) основний рух приводних механізмів, пружні переміщення поворотного механізму та тягового органу візка, а також коливання вантажу на гнучкому підвісу в площині зміни вильоту та при повороті крана. З наведеного встановлено, що представлена динамічна модель стрілової системи баштового крана має шість ступенів свободи.

В площині повороту крана за узагальнені координати обрані кутові координати зведеної маси приводу механізму повороту α , поворотної частини крана φ та повороту вантажу ψ . В площині зміни вильоту за узагальнені координати обрано лінійні координати центрів мас візка z та вантажу x , а також кутлова координата повороту зведеної маси приводу механізму зміни

вильоту β . Кутове відхилення гнучкого підвісу вантажу від вертикалі визначається кутовою координатою ν , яка визначається залежністю:

$$\nu = \frac{u}{H}, \quad (9.7)$$

де

$$u = \sqrt{(z-x)^2 + x^2(\varphi-\psi)^2 + 2(z-x)x(\varphi-\psi)\sin\left(\frac{\varphi-\psi}{2}\right)}. \quad (9.8)$$

Для отримання диференціальних рівнянь сумісного руху механізмів зміни вильоту та повороту баштового крана, представлених динамічною моделлю, що наведена на рис. 9.7, використаємо рівняння Лагранжа другого роду:

$$\begin{aligned} \frac{d}{dt} \frac{\partial T}{\partial \dot{\alpha}} - \frac{\partial T}{\partial \alpha} &= M_1 - \frac{\partial \Pi}{\partial \alpha}; & \frac{d}{dt} \frac{\partial T}{\partial \dot{\beta}} - \frac{\partial T}{\partial \beta} &= M_3 - \frac{\partial \Pi}{\partial \beta}; & \frac{d}{dt} \frac{\partial T}{\partial \dot{\varphi}} - \frac{\partial T}{\partial \varphi} &= -M_2 - \frac{\partial \Pi}{\partial \varphi}; \\ \frac{d}{dt} \frac{\partial T}{\partial \dot{\psi}} - \frac{\partial T}{\partial \psi} &= -\frac{\partial \Pi}{\partial \psi}; & \frac{d}{dt} \frac{\partial T}{\partial \dot{z}} - \frac{\partial T}{\partial z} &= -\frac{\partial \Pi}{\partial z}; & \frac{d}{dt} \frac{\partial T}{\partial \dot{x}} - \frac{\partial T}{\partial x} &= -\frac{\partial \Pi}{\partial x}, \end{aligned} \quad (9.9)$$

де T , Π – відповідно кінетична та потенціальна енергії, які визначаються наступними виразами:

$$T = \frac{1}{2} I_1 \dot{\alpha}^2 + \frac{1}{2} I_2 \dot{\varphi}^2 + \frac{1}{2} I_3 \dot{\beta}^2 + \frac{1}{2} m_3 (\dot{z}^2 + z^2 \dot{\varphi}^2) + \frac{1}{2} m (\dot{x}^2 + x^2 \dot{\psi}^2); \quad (9.10)$$

$$\Pi = \frac{1}{2} C_1 (\alpha - \varphi)^2 + \frac{1}{2} C_3 (\beta r - z)^2 - mgH \cos \nu. \quad (9.11)$$

Тут r – радіус приводного барабана механізму зміни вильоту; g – прискорення вільного падіння.

Візьмемо похідні від виразів (9.10) і (9.11), згідно з рівняннями (9.9), в результаті чого отримаємо:

$$\begin{aligned} \frac{\partial T}{\partial \alpha} = \frac{\partial T}{\partial \beta} = \frac{\partial T}{\partial \varphi} = \frac{\partial T}{\partial \psi} &= 0; & \frac{\partial T}{\partial z} &= m_3 z \dot{\varphi}^2; & \frac{\partial T}{\partial x} &= mx \dot{\psi}^2; & \frac{\partial T}{\partial \dot{\alpha}} &= I_1 \dot{\alpha}; \\ \frac{\partial T}{\partial \dot{\beta}} &= I_3 \dot{\beta}; & \frac{\partial T}{\partial \dot{\varphi}} &= (I_2 + m_3 z^2) \dot{\varphi}; & \frac{\partial T}{\partial \dot{\psi}} &= mx^2 \dot{\psi}; & \frac{\partial T}{\partial \dot{z}} &= m_3 \dot{z}; & \frac{\partial T}{\partial \dot{x}} &= mx; \end{aligned} \quad (9.12)$$

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{\alpha}} = I_1 \ddot{\alpha}; \quad \frac{d}{dt} \frac{\partial T}{\partial \dot{\beta}} = I_3 \ddot{\beta}; \quad \frac{d}{dt} \frac{\partial T}{\partial \dot{\phi}} = (I_2 + m_3 z^2) \ddot{\phi} + 2m_3 z \dot{z} \dot{\phi}; \quad (9.13)$$

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{\psi}} = mx^2 \ddot{\psi} + 2m x \dot{x} \dot{\psi}; \quad \frac{d}{dt} \frac{\partial T}{\partial \dot{z}} = m_3 \ddot{z}; \quad \frac{d}{dt} \frac{\partial T}{\partial \dot{x}} = m \ddot{x}; \quad (9.14)$$

$$\frac{\partial \Pi}{\partial \alpha} = C_1 (\alpha - \varphi); \quad \frac{\partial \Pi}{\partial \beta} = C_3 r (\beta r - z); \quad (9.15)$$

$$\frac{\partial \Pi}{\partial \phi} = -C_1 (\alpha - \varphi) + mgH \sin \nu \cdot \frac{\partial \nu}{\partial \phi} = -C_1 (\alpha - \varphi) + \frac{mg}{H} xz (\varphi - \psi); \quad (9.16)$$

$$\frac{\partial \Pi}{\partial \psi} = mgH \sin \nu \cdot \frac{\partial \nu}{\partial \psi} = -\frac{mg}{H} xz (\varphi - \psi); \quad (9.17)$$

$$\frac{\partial \Pi}{\partial z} = -C_3 (\beta r - z) + mgH \sin \nu \cdot \frac{\partial \nu}{\partial z} = -C_3 (\beta r - z) + \frac{mg}{H} (z - x); \quad (9.18)$$

$$\frac{\partial \Pi}{\partial x} = mgH \sin \nu \cdot \frac{\partial \nu}{\partial x} = \frac{mg}{H} (x - z). \quad (4.19)$$

В результаті підстановки виразів (9.6)-(9.19) в систему рівнянь (9.9) будемо мати диференціальні рівняння сумісного руху механізмів зміни вильоту та повороту баштового крана:

$$\begin{aligned} I_1 \ddot{\alpha} &= M_1 - C_1 (\alpha - \varphi); \quad I_3 \ddot{\beta} = M_3 - C_3 r (\beta r - z); \\ (I_2 + m_3 r^2) \ddot{\phi} + 2m_3 z \dot{z} \dot{\phi} &= -M_2 + C_1 (\alpha - \varphi) - \frac{mg}{H} xz (\varphi - \psi); \\ mx^2 \ddot{\psi} + 2m x \dot{x} \dot{\psi} &= \frac{mg}{H} xz (\varphi - \psi); \quad m_3 \ddot{z} - m_3 z \dot{\phi}^2 = C_3 (\beta r - z) - \frac{mg}{H} (z - x) - W; \\ m \ddot{x} - m x \dot{\psi}^2 &= -\frac{mg}{H} (x - z). \end{aligned} \quad (9.20)$$

Оскільки під час сумісного руху механізмів зміни вильоту та повороту крана виникає потреба в зменшенні енергетичних витрат, тому за критерій оптимізації процесу сумісного пуску механізмів візьмемо середньоквадратичне значення суми потужностей приводних механізмів

$$P_{\text{ск}} = \left(t_1^{-1} \int_0^{t_1} \left((M_1 \dot{\alpha})^2 + (M_3 \dot{\beta})^2 \right) dt \right)^{1/2} \rightarrow \min \quad (9.21)$$

при виконанні наступних крайових умов руху:

$$t = 0: \alpha = \alpha_0, \dot{\alpha} = 0, \beta = \beta_0, \dot{\beta} = 0, z = z_0, \dot{z} = 0, x = z_0, \dot{x} = 0, \varphi = 0,$$

$$\dot{\varphi} = 0, \psi = 0, \dot{\psi} = 0;$$

$$t = t_1: \alpha = \alpha_0 + \frac{\omega_1 t_1}{2}, \dot{\alpha} = \omega_1, \beta = \beta_0 + \frac{\omega_3 t_1}{2}, \dot{\beta} = \omega_3, z = z_0 + \frac{v_3 t_1}{2}, \dot{z} = v_3,$$

$$x = z_0 + \frac{v_3 t_1}{2}, \dot{x} = v_3, \varphi = \frac{\omega_1 t_1}{2}, \dot{\varphi} = \omega_1, \psi = \frac{\omega_1 t_1}{2}, \dot{\psi} = \omega_1.$$

(9.22)

Тут t – час; t_1 – тривалість процесу пуску механізмів зміни вильоту та повороту крана; α_0, β_0 – початкові положення кутових координат приводних механізмів повороту та зміни вильоту, відповідно; z_0 – початкове положення візка та вантажу; ω_1, ω_3 – усталені кутові швидкості повороту крана та приводного барабана механізму зміни вильоту; v_3 – усталена швидкість переміщення візка та вантажу.

9.2.2 Розв'язання задачі

В поставленій варіаційній задачі (9.21), (9.22) необхідно знайти режими руху приводних механізмів, які мінімізують критерій (9.21) і задовольняють крайові умови (9.22).

Для формування критерію (9.21) виразимо рушійний момент M_1 та M_3 , а також координати ланок через узагальнені координати вантажу x і ψ та їхні похідні за часом.

З першого та другого рівнянь системи (9.20) знайдемо вирази рушійних моментів:

$$M_1 = I_1 \ddot{\alpha} + C_1 (\alpha - \varphi); \quad (9.23)$$

$$M_3 = I_3 \ddot{\beta} + C_3 r (\beta r - z). \quad (9.24)$$

З третього та четвертого рівнянь системи (9.20) знайдемо вираз, що входить в залежність (9.23), в результаті чого отримаємо

$$C_1 (\alpha - \varphi) = (I_2 + m_3 z^2) \ddot{\varphi} + 2m_3 z \dot{z} \dot{\varphi} + m x (x \ddot{\psi} + 2\dot{x} \dot{\psi}) + M_2. \quad (9.25)$$

Аналогічно з п'ятого та шостого рівнянь системи (9.20) знайдемо вираз, що входить в залежність (9.24)

$$C_3(\beta r - z) = m_3(\ddot{z} - z\dot{\phi}^2) + m(\ddot{x} - x\dot{\psi}^2) + W. \quad (9.26)$$

В залежності (9.25) та (9.26) входить координата візка та її похідні за часом. Виразимо її через координати вантажу. Для цього скористаємось останнім рівнянням системи (9.20) і отримаємо координату візка, яка має вигляд

$$z = x + \frac{H}{g}(\ddot{x} - x\dot{\psi}^2). \quad (9.27)$$

Візьмемо похідні за часом від виразу (9.27):

$$\dot{z} = \dot{x} + \frac{H}{g}(\ddot{x} - 2x\dot{\psi}\dot{\psi} - \dot{x}\dot{\psi}^2); \quad (9.28)$$

$$\ddot{z} = \ddot{x} + \frac{H}{g}\left(x - 4\dot{x}\dot{\psi}\dot{\psi} - 2x(\dot{\psi}\ddot{\psi} + \dot{\psi}^2)\right); \quad (9.29)$$

$$\ddot{\ddot{z}} = \ddot{\ddot{x}} + \frac{H}{g}\left(x - 4\dot{x}\dot{\psi}\dot{\psi} - 6\dot{x}(\dot{\psi}\ddot{\psi} + \dot{\psi}^2) - 2x(\dot{\psi}^IV\dot{\psi} + 3\dot{\psi}\ddot{\psi})\right). \quad (9.30)$$

Із залежності (9.25) виразимо координату приводу механізму повороту

$$\alpha = \varphi + \frac{1}{C_1}\left((I_2 + m_3z^2)\ddot{\phi} + 2m_3z\dot{z}\dot{\phi} + mx(x\dot{\psi} + \dot{x}\dot{\psi}) + M_2\right). \quad (9.31)$$

Двічі продиференціюємо за часом вираз (9.31), в результаті чого отримаємо

$$\dot{\alpha} = \dot{\phi} + \frac{1}{C_1}\left((I_2 + m_3z^2)\ddot{\phi} + 2m_3(z(2\dot{z}\dot{\phi} + \dot{z}\dot{\phi}) + \dot{z}^2\dot{\phi}) + m(x^2\ddot{\psi} + 2x(2\dot{x}\dot{\psi} + \dot{x}\dot{\psi}) + 2\dot{x}^2\dot{\psi})\right); \quad (9.32)$$

$$\ddot{\alpha} = \ddot{\phi} + \frac{1}{C_1}\left((I_2 + m_3z^2)^IV\dot{\phi} + 2m_3(3\dot{z}(\dot{z}\dot{\phi} + \dot{z}\dot{\phi}) + z(3\ddot{z}\dot{\phi} + 3\dot{z}\ddot{\phi} + \ddot{z}\dot{\phi})) + m\left(x^2\dot{\psi}^IV + 6\dot{x}(\dot{x}\dot{\psi} + \dot{x}\dot{\psi}) + 2x(3\ddot{x}\dot{\psi} + 3\dot{x}\ddot{\psi} + \ddot{x}\dot{\psi})\right)\right). \quad (9.33)$$

З рівняння (9.26) знайдемо кутову координату приводного барабана механізму зміни вильоту

$$\beta = \frac{z}{r} + \frac{1}{C_3 r} \left(m_3 (\ddot{z} - z\dot{\phi}^2) + m (\ddot{x} - x\dot{\psi}^2) + W \right) \quad (9.34)$$

Двічі продиференціюємо за часом вираз (4.34), після чого отримаємо:

$$\dot{\beta} = \frac{\dot{z}}{r} + \frac{1}{C_3 r} \left(m_3 (\ddot{z} - \dot{z}\dot{\phi}^2 - 2z\dot{\phi}\ddot{\phi}) + m (\ddot{x} - \dot{x}\dot{\psi}^2 - 2x\dot{\psi}\ddot{\psi}) \right); \quad (9.35)$$

$$\begin{aligned} \ddot{\beta} = \frac{\ddot{z}}{r} + \frac{1}{C_3 r} \left(m_3 \left(z - \ddot{z}\dot{\phi}^2 - 4\dot{z}\dot{\phi}\ddot{\phi} - 2z(\ddot{\phi}^2 + \dot{\phi}\ddot{\phi}) \right) + \right. \\ \left. + m \left(x - \ddot{x}\dot{\psi}^2 - 4\dot{x}\dot{\psi}\ddot{\psi} - 2x(\ddot{\psi}^2 + \dot{\psi}\ddot{\psi}) \right) \right). \end{aligned} \quad (9.36)$$

У вирази рушійних моментів приводних механізмів та їх кінематичних характеристик входять координата поворотної частини крана та її похідні за часом, тому виразимо їх через координати вантажу. Для цього використаємо четверте рівняння системи (9.20), з якого маємо

$$\varphi = \psi + \frac{H}{gz} (x\ddot{\psi} + 2\dot{x}\dot{\psi}) \quad (9.37)$$

Візьмемо похідні за часом від виразу (9.37) включно до четвертого порядку:

$$\dot{\phi} = \dot{\psi} + \frac{H}{gz^2} \left((x\ddot{\psi} + 3\dot{x}\dot{\psi} + 2\ddot{x}\dot{\psi})z - \dot{z}(x\ddot{\psi} + 2\dot{x}\dot{\psi}) \right); \quad (9.38)$$

$$\begin{aligned} \ddot{\phi} = \ddot{\psi} + \frac{H}{gz^3} \left(z \left(z \left(x\ddot{\psi} + 4\dot{x}\ddot{\psi} + 5\ddot{x}\dot{\psi} + 2\ddot{\ddot{x}}\dot{\psi} \right) - \dot{z}(x\ddot{\psi} + 2\dot{x}\dot{\psi}) \right) - \right. \\ \left. - 2\dot{z}(z(x\ddot{\psi} + 3\dot{x}\dot{\psi} + 2\ddot{x}\dot{\psi}) - \dot{z}(x\ddot{\psi} + 2\dot{x}\dot{\psi})) \right); \end{aligned} \quad (9.39)$$

$$\begin{aligned} \ddot{\phi} = \ddot{\psi} + \frac{H}{gz^4} \left(z^2 \left(z \left(x\ddot{\psi} + 5\dot{x}\ddot{\psi} + 9\ddot{x}\dot{\psi} + 7\ddot{\ddot{x}}\dot{\psi} + 2\ddot{\ddot{\ddot{x}}}\dot{\psi} \right) - \right. \right. \\ \left. \left. - \dot{z}(x\ddot{\psi} + 2\dot{x}\dot{\psi}) \right) - 3z\dot{z} \left(z \left(x\ddot{\psi} + 4\dot{x}\ddot{\psi} + 5\ddot{x}\dot{\psi} + 2\ddot{\ddot{x}}\dot{\psi} \right) - 2\dot{z}(x\ddot{\psi} + 2\dot{x}\dot{\psi}) \right) + \right. \\ \left. + 6\dot{z}^2 (z(x\ddot{\psi} + 3\dot{x}\dot{\psi} + 2\ddot{x}\dot{\psi}) - \dot{z}(x\ddot{\psi} + 2\dot{x}\dot{\psi})) \right); \end{aligned} \quad (9.40)$$

$$\begin{aligned}
\varphi = \psi + \frac{H}{gz^5} & \left(z \left(z^2 \left(z \left(x \overset{VI}{\psi} + 6 \dot{x} \overset{V}{\psi} + 14 \ddot{x} \overset{IV}{\psi} + 16 \ddot{\ddot{x}} \overset{IV}{\psi} + 9 \overset{IV}{x} \overset{IV}{\psi} + 2 \overset{V}{x} \overset{IV}{\psi} \right) - \right. \right. \\
& \left. \left. - 3 \ddot{z} \left(x \overset{IV}{\psi} + 4 \dot{x} \overset{IV}{\psi} + 5 \ddot{x} \overset{IV}{\psi} + 2 \ddot{\ddot{x}} \overset{IV}{\psi} \right) - \right. \right. \\
& \left. \left. - 4 \dot{z} \left(x \overset{IV}{\psi} + 3 \dot{x} \overset{IV}{\psi} + 2 \ddot{x} \overset{IV}{\psi} \right) - z \left(x \overset{IV}{\psi} + 2 \dot{x} \overset{IV}{\psi} \right) \right) + \right. \\
& + 4 z \dot{z} \ddot{z} \left(x \overset{IV}{\psi} + 2 \dot{x} \overset{IV}{\psi} \right) + 12 \dot{z} \ddot{z} \left(z \left(x \overset{IV}{\psi} + 3 \dot{x} \overset{IV}{\psi} + 2 \ddot{x} \overset{IV}{\psi} \right) - \dot{z} \left(x \overset{IV}{\psi} + 2 \dot{x} \overset{IV}{\psi} \right) \right) - \\
& - 4 \dot{z} \left(z^2 \left(z \left(x \overset{V}{\psi} + 4 \dot{x} \overset{IV}{\psi} + 9 \ddot{x} \overset{IV}{\psi} + 7 \ddot{\ddot{x}} \overset{IV}{\psi} + 2 \overset{IV}{x} \overset{IV}{\psi} \right) - 2 \ddot{z} \left(x \overset{IV}{\psi} + 2 \dot{x} \overset{IV}{\psi} \right) \right) + \right. \\
& + 6 \dot{z}^2 \left(z \left(x \overset{IV}{\psi} + 3 \dot{x} \overset{IV}{\psi} + 2 \ddot{x} \overset{IV}{\psi} \right) - \dot{z} \left(x \overset{IV}{\psi} + 3 \dot{x} \overset{IV}{\psi} + 2 \ddot{x} \overset{IV}{\psi} \right) - \ddot{z} \left(x \overset{IV}{\psi} + 2 \dot{x} \overset{IV}{\psi} \right) \right) - \\
& - 3 z \dot{z} \left(z \left(x \overset{IV}{\psi} + 4 \dot{x} \overset{IV}{\psi} + 5 \ddot{x} \overset{IV}{\psi} + 2 \ddot{\ddot{x}} \overset{IV}{\psi} \right) - 2 \ddot{z} \left(x \overset{IV}{\psi} + 2 \dot{x} \overset{IV}{\psi} \right) \right) + \\
& \left. + 6 \dot{z}^2 \left(z \left(x \overset{IV}{\psi} + 3 \dot{x} \overset{IV}{\psi} + 2 \ddot{x} \overset{IV}{\psi} \right) - \dot{z} \left(x \overset{IV}{\psi} + 2 \dot{x} \overset{IV}{\psi} \right) \right) \right). \tag{9.41}
\end{aligned}$$

Виразимо крайові умови (9.22) через координати вантажу x і ψ та їхні похідні за часом. Розглянемо крайові умови на початку пуску, коли $t=0$. Тут відомими є координати $x(0)=z_0$; $\dot{x}(0)=0$; $\psi(0)=0$; $\dot{\psi}(0)=0$. З залежності (9.27) маємо, що умові $z=z_0$ відповідає співвідношення

$$\ddot{x}(0) = 0 \tag{9.42}$$

Також із виразу (9.28) знаходимо, що умові $\dot{z}=0$ відповідає залежність

$$\ddot{x}(0) = 0 \tag{9.43}$$

Далі, виходячи з залежності (9.31) і умови $\alpha=0$, маємо

$$\ddot{\psi}(0) = 0 \tag{9.44}$$

З умови $\dot{\alpha}=0$ за даною залежності (9.32) знаходимо

$$\ddot{\psi}(0) = 0 \tag{9.45}$$

Використовуючи умову $\beta = (W/(C_3 r)) + z_0/r$, за допомогою залежностей (9.43) та (9.44) знаходимо

$$\overset{IV}{x}(0) = 0 \tag{9.46}$$

Також, використовуючи умову $\dot{\beta} = 0$ і залежність (9.45) та третю похідну за часом від координати візка, отримуємо

$${}^V x(0) = 0 \quad (9.47)$$

З умови $\alpha = M_2 / C_1$ та виразів (9.30) і (9.39) маємо

$${}^{IV} \psi(0) = 0 \quad (9.48)$$

Використавши умову $\dot{\alpha} = 0$ та залежності (4.32) і (4.40), знаходимо

$${}^V \psi(0) = 0 \quad (9.49)$$

Тепер розглянемо кінцеві умови пуску механізмів зміни вильоту та повороту крана, коли $t = t_1$. Позначимо $v_3 = v$, $\omega_1 = \omega$, тоді $\omega_3 = v/r$. Тут відомими є такі умови: $x(t_1) = z_1 = z_0 + vt_1/2$; $\dot{x}(t_1) = v$; $\psi(t_1) = \alpha t_1/2$; $\dot{\psi}(t_1) = \omega$.

З умови $z(t_1) = z_1 = z_0 + vt_1/2$ за допомогою залежності (9.27) знаходимо, що

$$\ddot{x}(t_1) = z_1 \omega^2 = \left(z_0 + \frac{vt_1}{2} \right) \omega^2 \quad (9.50)$$

Використовуючи залежність (9.28) і умову $\dot{z}(t_1) = v$, маємо

$$\ddot{x}(t_1) = -3v\omega^2 \quad (9.51)$$

За допомогою умови $\varphi(t_1) = \alpha t_1/2$ та залежності (9.37) отримуємо

$$\ddot{\psi}(t_1) = -\frac{2v\omega}{z_1} \quad (9.52)$$

З умови $\dot{\varphi}(t_1) = \omega$ та залежності (9.38) отримуємо співвідношення

$$\ddot{\psi}(t_1) = \frac{6v^2\omega}{z_1^2} - 2\omega^3 \quad (9.53)$$

Використавши умову $\beta(t_1) = (W/(C_3 r)) + z_1/r$ і залежності (9.29) і (9.34), маємо

$${}^{IV} x(t_1) = \frac{12v^2\omega^2}{z_1} - 4\omega^4 z_1 \quad (9.54)$$

Введемо деякі позначення: $I_{23} = I_2 + m_3 z_1^2$; $k_3 = m_3 z_1^2 / I_{23}$; $k = (m - 2m_3) z_1^2 / I_{23}$.

Виходячи з умови $\dot{\beta}(t_1) = v/r$, виразів (9.35) і (9.30) та наведених позначень, отримаємо

$${}^V x(t_1) = v\omega^2 \left(24 \left(\frac{5v^2}{z_1^2} - 3\omega^2 \right) - k \frac{g}{H} \right). \quad (9.55)$$

З умови $\alpha(t_1) = M_2/C_1 + \omega t_1/2$ та виразів (9.31) і (9.39) знаходимо

$${}^{IV} \psi(t_1) = \frac{v\omega}{z_1} \left((k+2) \frac{g}{H} + 24 \left(\omega^2 - \frac{v^2}{z_1^2} \right) \right). \quad (9.56)$$

В результаті використання виразів (9.54) і (9.62) та умови $\dot{\alpha}(t_1) = \omega$ маємо

$${}^V \psi(t_1) = 2\omega \left(60 \frac{v^4}{t_1^4} - 13\omega^4 + \frac{g}{H} \left(\omega^2(1-k_3) - \frac{v^2}{z_1^2} (11+2k(k_3+2)+k_3) \right) \right). \quad (9.57)$$

Виходячи з виразів (9.42)-(9.57), крайові умови (9.22) представимо в наступному вигляді:

$$\begin{aligned} x(0) &= z_0, \dot{x}(0) = 0, \ddot{x}(0) = 0, \ddot{\ddot{x}}(0) = 0, {}^{IV} x(0) = 0, {}^V x(0) = 0; \\ \psi(0) &= 0, \dot{\psi}(0) = 0, \ddot{\psi}(0) = 0, \ddot{\ddot{\psi}}(0) = 0, {}^{IV} \psi(0) = 0, {}^V \psi(0) = 0; \\ x(t_1) &= z_1 = z_0 + \frac{vt_1}{2}, \dot{x}(t_1) = v, \ddot{x}(t_1) = z_1\omega^2, \ddot{\ddot{x}}(t_1) = -3v\omega^2; \\ {}^{IV} x(t_1) &= \frac{12v^2\omega^2}{z_1} - 4\omega^4 z_1, {}^V x(t_1) = v\omega^2 \left(24 \left(5 \frac{v^2}{z_1^2} - 3\omega^2 \right) - kg/H \right); \\ \psi(t_1) &= \frac{\omega t_1}{2}, \dot{\psi}(t_1) = \omega, \ddot{\psi}(t_1) = \frac{-2v\omega}{z_1}, \ddot{\ddot{\psi}}(t_1) = \frac{6v^2\omega}{z_1^2} - 2\omega^3; \end{aligned} \quad (9.58)$$

$${}^{IV} \psi(t_1) = \frac{v\omega}{z_1} \left((k+2) \frac{g}{H} + 24 \left(\omega^2 - \frac{v^2}{z_1^2} \right) \right);$$

$${}^V \psi(t_1) = 2\omega \left(60 \frac{v^4}{z_1^4} - 13\omega^4 + \frac{g}{H} \left(\omega^2(1-k_3) - \frac{v^2}{z_1^2} (11+2k(k_3+2)+k_3) \right) \right).$$

Інтегральний функціонал (9.21) з виразами (9.23)-(9.40) і крайовими умовами (9.58) представляють собою варіаційну задачу, в якій необхідно знайти закони зміни координат вантажу $x(t)$ та $\psi(t)$, які мінімізують значення критерію (9.21) та забезпечують крайові умови (9.58).

Представлена варіаційна задача є нелінійною, тому для її розв'язку використаємо наближений метод. Для розв'язку поставленої варіаційної задачі представимо шукані функції $x(t)$ та $\psi(t)$ у вигляді поліномів з двома доданками

$$x(t) = x_1(t) + x_2(t); \quad \psi(t) = \psi_1(t) + \psi_2(t). \quad (9.59)$$

У виразах (9.59) перші доданки $x_1(t)$ та $\psi_1(t)$ – це вибрані поліноми, що задовольняють крайові умови, а другі – $x_2(t)$ та $\psi_2(t)$ – поліноми, в які входять вільні коефіцієнти і задовольняють нульовим крайовим умовам:

$$\begin{aligned} x_2(0) = 0, \quad \dot{x}_2(0) = 0, \quad \ddot{x}_2(0) = 0, \quad \overset{IV}{x}_2(0) = 0, \quad \overset{V}{x}_2(0) = 0; \\ x_2(t_1) = z_0, \quad \dot{x}_2(t_1) = v, \quad \ddot{x}_2(t_1) = 0, \quad \overset{IV}{x}_2(t_1) = 0, \quad \overset{V}{x}_2(t_1) = 0; \\ \psi_2(0) = 0, \quad \dot{\psi}_2(0) = 0, \quad \ddot{\psi}_2(0) = 0, \quad \overset{IV}{\psi}_2(0) = 0, \quad \overset{V}{\psi}_2(0) = 0; \\ \psi_2(t_1) = \frac{\omega t_1}{2}, \quad \dot{\psi}_2(t_1) = \omega, \quad \ddot{\psi}_2(t_1) = 0, \quad \overset{IV}{\psi}_2(t_1) = 0, \quad \overset{V}{\psi}_2(t_1) = 0. \end{aligned} \quad (9.60)$$

$$(9.61)$$

Функції $x_1(t)$ та $\psi_1(t)$ виберемо у вигляді поліномів одинадцятого порядку

$$x_1 = \sum_{i=0}^{11} C_i t^i, \quad 0 \leq t \leq t_1; \quad (9.62)$$

$$\psi_1 = \sum_{i=0}^{11} D_i t^i, \quad 0 \leq t \leq t_1. \quad (9.63)$$

Оскільки в крайових умовах (9.58) функції $x(t)$ та $\psi(t)$ входять до п'ятого порядку включно, тому візьмемо їх

$$\begin{aligned} \dot{x}_1 &= C_1 + 2C_2 t + 3C_3 t^2 + 4C_4 t^3 + 5C_5 t^4 + 6C_6 t^5 + 7C_7 t^6 + 8C_8 t^7 + 9C_9 t^8 + 10C_{10} t^9 + 11C_{11} t^{10}; \\ \ddot{x}_1 &= 2C_2 + 6C_3 t + 12C_4 t^2 + 20C_5 t^3 + 30C_6 t^4 + 42C_7 t^5 + 56C_8 t^6 + 72C_9 t^7 + 90C_{10} t^8 + 110C_{11} t^9; \\ \overset{IV}{x}_1 &= 6C_3 + 24C_4 t + 60C_5 t^2 + 120C_6 t^3 + 210C_7 t^4 + 336C_8 t^5 + 504C_9 t^6 + 720C_{10} t^7 + 990C_{11} t^8; \end{aligned} \quad (9.64)$$

$$\overset{IV}{x}_1 = 24C_4 + 120C_5 t + 360C_6 t^2 + 840C_7 t^3 + 1680C_8 t^4 + 3024C_9 t^5 + 5040C_{10} t^6 + 7920C_{11} t^7;$$

$$\overset{V}{x}_1 = 120C_5 + 720C_6 t + 2520C_7 t^2 + 6720C_8 t^3 + 15120C_9 t^4 + 30240C_{10} t^5 + 55440C_{11} t^6.$$

Похідні від функції (9.63) будуть мати вигляд, аналогічний (9.64), де постійні C_i замінюються на відповідні D_i .

Підставивши в залежності (9.62) і (9.64) крайові умови (9.58), отримаємо:

$$C_0 = z_0; C_1 = C_2 = C_3 = C_4 = C_5 = 0. \quad (9.65)$$

Інші постійні $C_6 \dots C_{11}$ визначаються з системи рівнянь

$$C_6 + C_7 t_1 + C_8 t_1^2 + C_9 t_1^3 + C_{10} t_1^4 + C_{11} t_1^5 = \frac{(z_1 - z_0)}{t_1^6};$$

$$6C_6 + 7C_7 t_1 + 8C_8 t_1^2 + 9C_9 t_1^3 + 10C_{10} t_1^4 + 11C_{11} t_1^5 = \frac{V}{t_1^6};$$

$$30C_6 + 42C_7 t_1 + 56C_8 t_1^2 + 72C_9 t_1^3 + 90C_{10} t_1^4 + 110C_{11} t_1^5 = \frac{\ddot{x}(t_1)}{t_1^4};$$

$$120C_6 + 210C_7 t_1 + 336C_8 t_1^2 + 504C_9 t_1^3 + 720C_{10} t_1^4 + 990C_{11} t_1^5 = \frac{\ddot{x}(t_1)}{t_1^3}; \quad (9.66)$$

$$360C_6 + 840C_7 t_1 + 1680C_8 t_1^2 + 3024C_9 t_1^3 + 5040C_{10} t_1^4 + 7920C_{11} t_1^5 = \frac{IV}{t_1^2} x(t_1);$$

$$720C_6 + 2520C_7 t_1 + 6720C_8 t_1^2 + 15120C_9 t_1^3 + 30240C_{10} t_1^4 + 55440C_{11} t_1^5 = \frac{V}{t_1} x(t_1).$$

Підставивши вирази (9.65), а також знайдені в результаті розв'язку системи (9.66) в залежність (9.62), знайдемо функцію, яка забезпечує крайові умови по координаті x та її похідним за часом. Аналогічно знаходиться функція, яка забезпечує крайові умови по координаті ψ та її похідним за часом.

Поліноми $x_2(t)$ і $\psi_2(t)$ представимо у наступному вигляді:

$$x_2(t) = t^6 (t - t_1)^6 \sum_{i=0}^n A_i t^i, \quad 0 \leq t \leq t_1; \quad (9.67)$$

$$\psi_2(t) = t^6 (t - t_1)^6 \sum_{i=0}^n B_i t^i, \quad 0 \leq t \leq t_1, \quad (9.68)$$

де $A_0 \dots A_n, B_0 \dots B_n$ – вільні коефіцієнти, від яких залежить значення критерію оптимізації (1.15); $t^6(t-t_1)^6$ – множник, який забезпечує виконання нульових крайових умов (9.60) і (9.61) при будь-яких значеннях коефіцієнтів $A_0 \dots A_n$,

$B_0 \dots B_n$. Ці коефіцієнти є вільними і використовуються для знаходження мінімального значення критерію (9.21).

Підставивши вирази (9.62)-(9.68) з урахуванням крайових умов (9.60), (9.61) в залежності (9.59), знайдемо вирази функцій $x(t)$ та $\psi(t)$, які включають вільні коефіцієнти $A_0 \dots A_n, B_0 \dots B_n$.

Знаючи вирази функцій $x(t)$ та $\psi(t)$, знайдемо залежності $z(t), \alpha(t), \beta(t)$ і $\varphi(t)$. Ці функції також залежать від вільних коефіцієнтів. В результаті проведення інтегрування критерій (9.21) також стає функцією вільних коефіцієнтів $A_0 \dots A_n, B_0 \dots B_n$. Таким чином, наближений розв'язок варіаційної задачі (9.21), (9.58) зводиться до знаходження мінімуму функції багатьох змінних.

Розв'яжемо цю задачу шляхом використання метаевристичного методу R-Ring-PSO [71].

При цьому критерій (9.21) представимо функцією, що залежить від 10 вільних коефіцієнтів

$$C_r = P(A_0, A_1, \dots, A_4, B_0, B_1, \dots, B_4). \quad (9.69)$$

Розрахунки оптимального режиму сумісного руху механізмів зміни вильоту та повороту баштового крана за критерієм середньоквадратичного значення потужності приводів виконаємо за таких значень параметрів стрілової системи: $m=5000$ кг; $m_3=300$ кг; $I_1=5,51 \cdot 10^5$ кгм²; $I_2=4,92 \cdot 10^6$ кгм²; $I_3=30,0$ кгм²; $M_2=39890$ Нм; $W=5500$ Нм; $C_1=6,63 \cdot 10^6$ Нм/рад; $C_3=1,36 \cdot 10^6$ Нм/рад; $r=0,15$ м; $z_0=10$ м; $v=0,85$ м/с; $\omega=0,075$ рад/с; $H=5$ м; $t_1=5$ с.

В результаті розв'язування оптимізаційної задачі отримано наступні значення вільних коефіцієнтів: $A_0=1 \cdot 10^{-7}$; $A_1=1 \cdot 10^{-7}$; $A_2=1 \cdot 10^{-7}$; $A_3=4,115 \cdot 10^{-8}$; $A_4=7,378 \cdot 10^{-9}$; $B_0=4,378 \cdot 10^{-8}$; $B_1=6,649 \cdot 10^{-9}$; $B_2=1,018 \cdot 10^{-9}$; $B_3=-6,319 \cdot 10^{-11}$; $B_4=5,261 \cdot 10^{-11}$.

На основі отриманого розв'язку оптимізаційної задачі сумісного пуску механізмів зміни вильоту та повороту баштового крана побудовано графічні залежності кінематичних (рис. 9.8-9.11), силових (рис. 9.12, 9.13) та енергетичних (рис. 9.14, 9.15) характеристик для базового розв'язку, коли всі

вільні коефіцієнти дорівнюють нулю (графіки чорного кольору), а також для оптимального розв'язку (графіки сірого кольору).

$\dot{\alpha}-\dot{\phi}$, rad/s

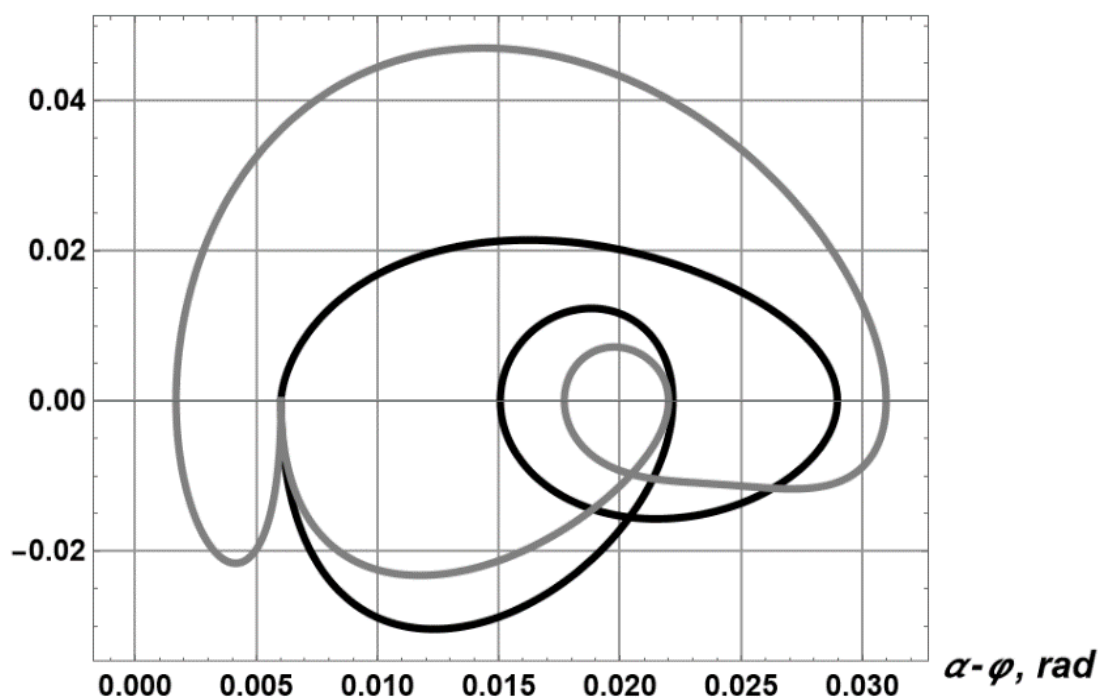


Рисунок 9.8 – Фазові траєкторії пружних коливань колони крана

$\dot{z}-\dot{x}$, m/s

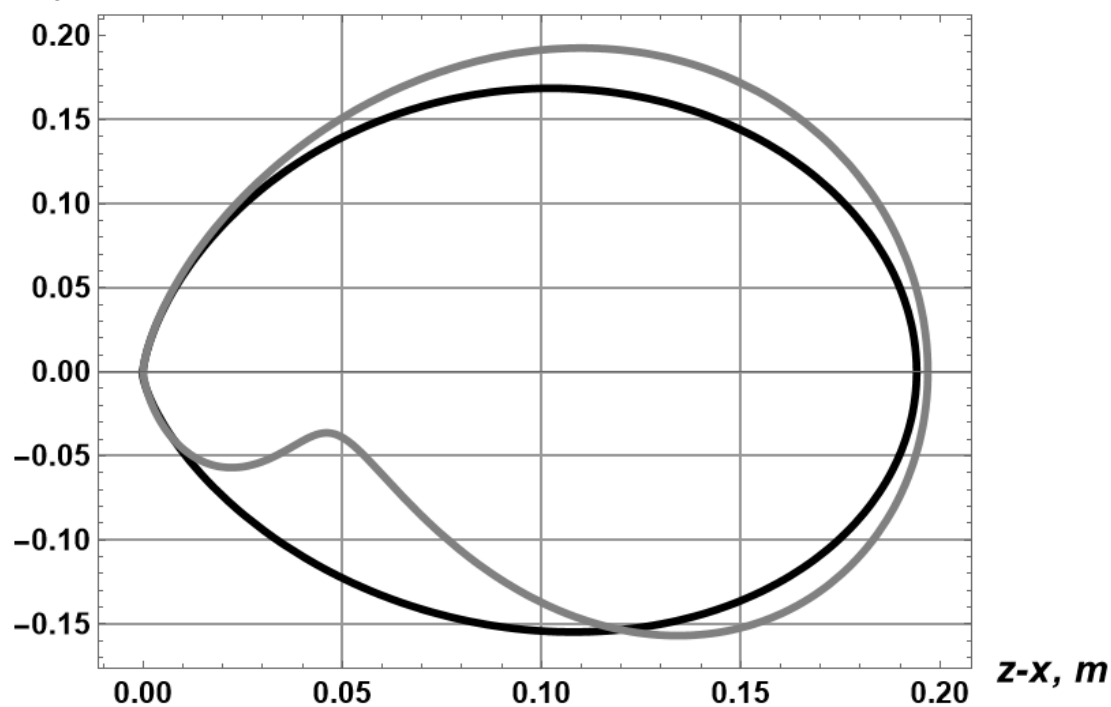


Рисунок 9.9 – Фазові траєкторії пружних коливань візка

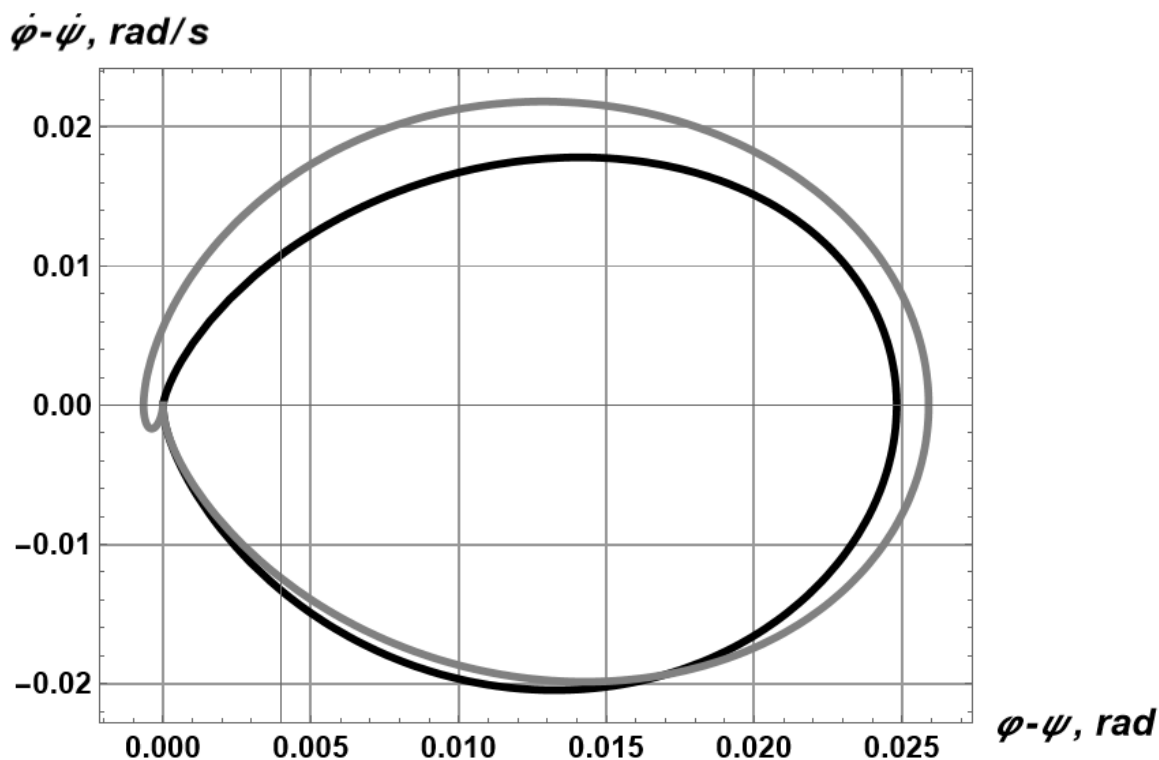


Рисунок 9.10 – Фазові траєкторії маятникових коливань вантажу

9.2.3 Аналіз результатів розв’язання задачі

З наведених фазових портретів деформації колони крана (рис. 9.8), відхилення канату від вертикалі в площині руху візка (рис. 9.9) та в площині, перпендикулярній до руху візка (рис. 9.10) можна зробити висновок, що за час пуску як для базового, так і оптимального розв’язку усі ці коливальні процеси є затухаючими. Це вказує на те, що при виході на усталений режим руху в елементах стрілової системи відсутні коливальні процеси. Разом з тим, оптимальний режим руху має переваги перед базовим режимом. Так, наприклад, максимальне значення деформації колони при оптимальному режимі в порівнянні з базовим режимом зменшилось на 23%, а максимальна швидкість деформації відповідно зменшилась на 26%. Аналогічна картина спостерігається і при коливаннях вантажу у взаємно перпендикулярних площинах, де покращені показники спостерігаються у оптимального режиму руху в порівнянні з базовим.

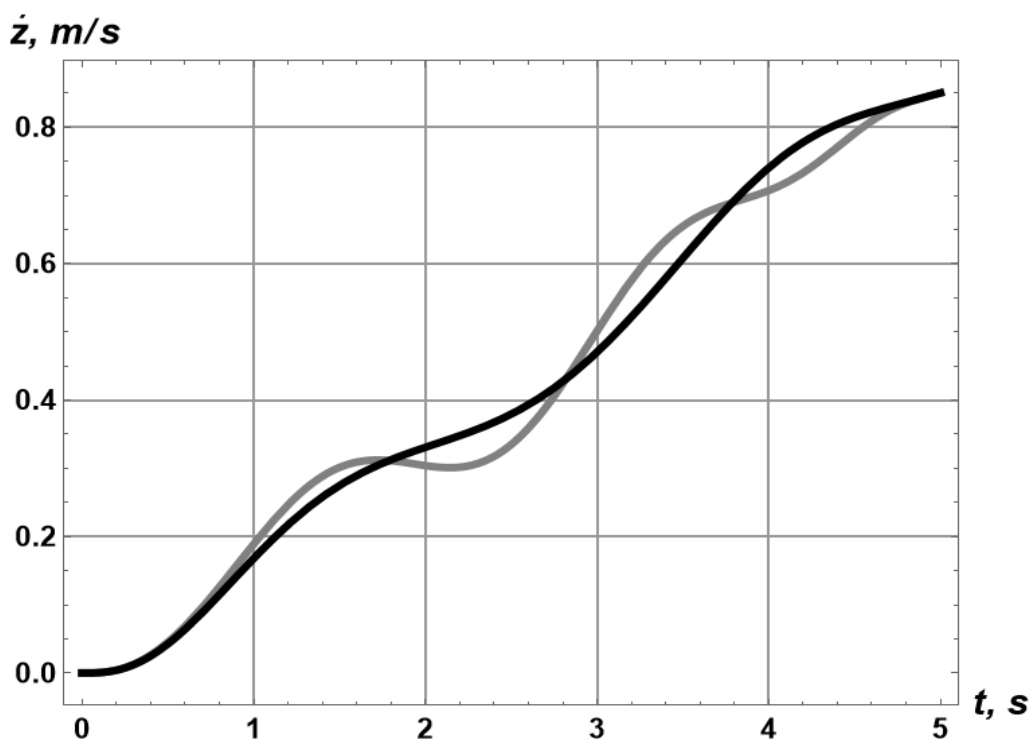


Рисунок 9.11 – Графіки швидкості руху візка

Порівнюючи графіки швидкості переміщення візка при оптимальному та базовому режимах руху, можна бачити незначні відхилення між ними, де базовий режим руху має більш плавний характер зміни.

З графічних залежностей рушійних моментів приводів повороту крана (рис. 9.12) та переміщення візка (рис. 9.13) можна бачити, що вони є досить близькими по характеру зміни як при базовому, так і при оптимальному режимах руху, де оптимальний режим руху має незначне зменшення максимальних значень моментів.

Аналіз графічних залежностей потужності приводу механізму повороту (рис. 9.14) показує, що характер зміни при базовому та оптимальному режимах в цілому є досить близьким з певними відхиленнями. При цьому максимальне значення потужності при базовому режимі на 33% перевищує відповідне значення при оптимальному режимі руху. Зміни потужності приводу механізму переміщення візка (рис. 9.15) при базовому і оптимальному режимах є досить близькими і їхні максимальні значення співпадають.

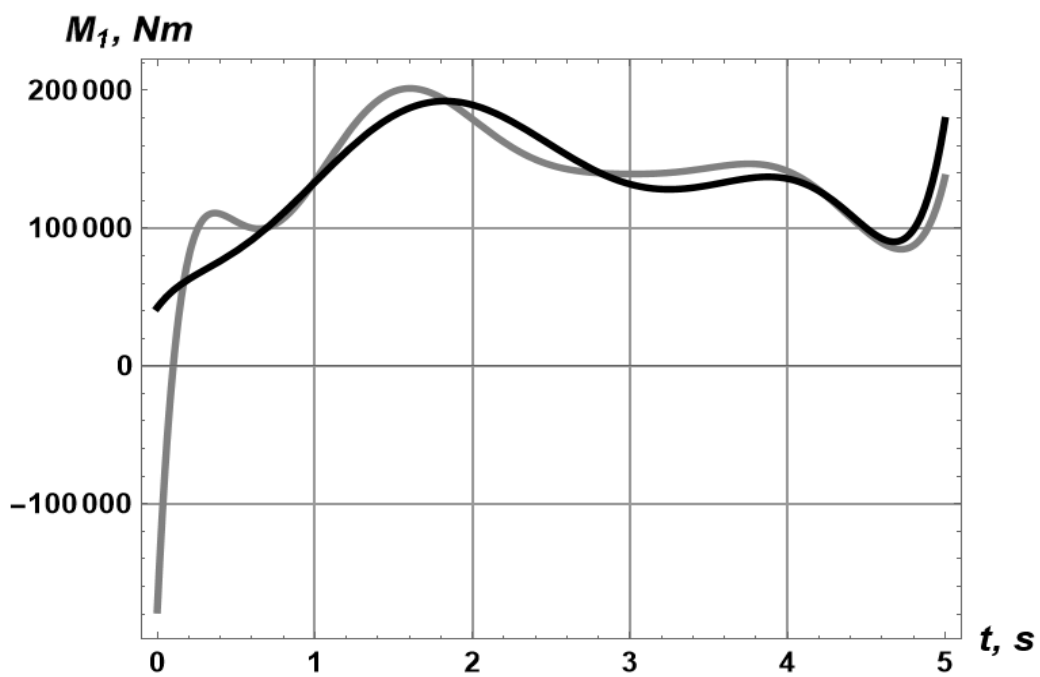


Рисунок 9.12 – Графіки рушійних моментів приводу механізму повороту крана

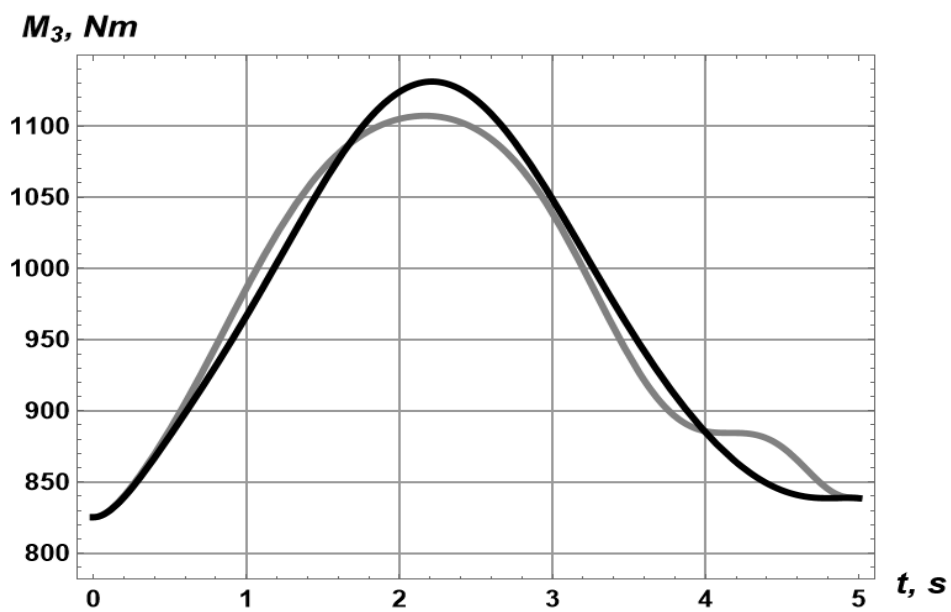


Рисунок 9.13 – Графіки рушійних моментів приводу механізму переміщення візка

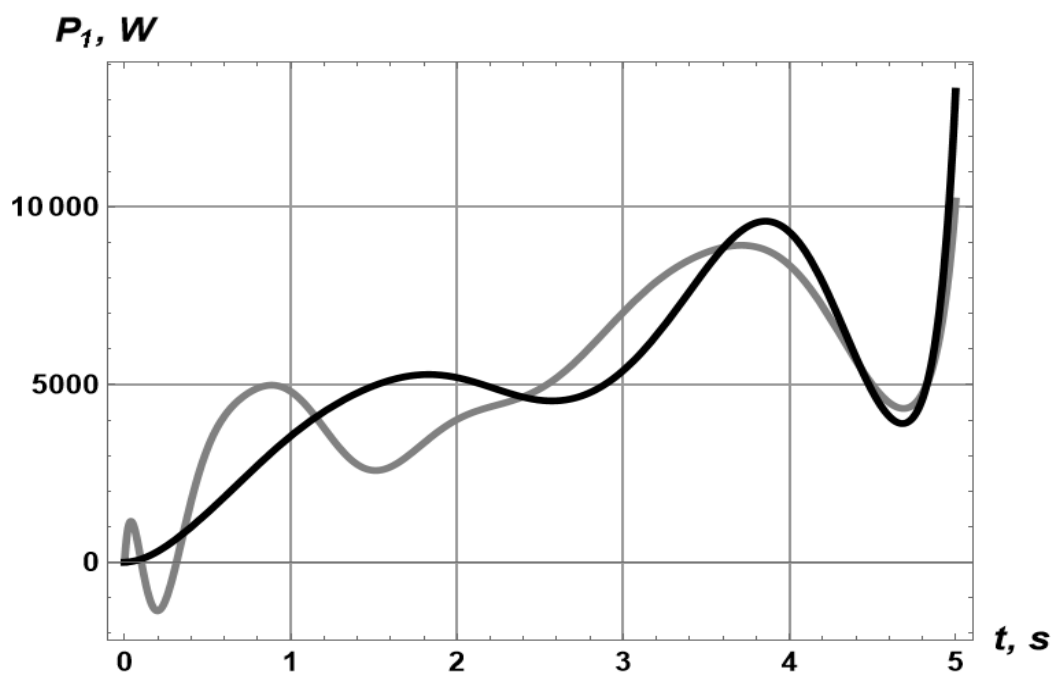


Рисунок 9.14 – Графіки потужностей, які споживані приводом механізму повороту крана

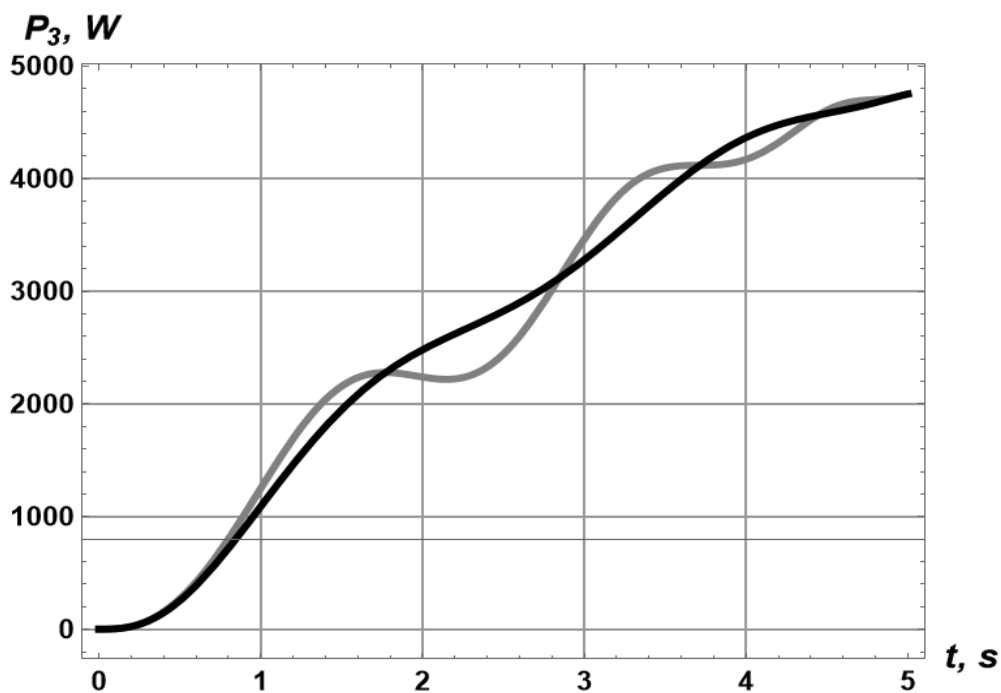


Рисунок 9.15 – Графіки потужностей, які споживані приводом механізму переміщення візка

Для отриманих розв'язків задачі розраховано чисельні значення кінематичних, динамічних та енергетичних показників (табл. 9.5).

Таблиця 9.5 – Чисельні значення кінематичних, динамічних та енергетичних показників

Показник	Од-ця вимір.	Розв'язок	
		Базовий (нульові значення коефіцієнтів)	Оптимальний (оптимальні значення коефіцієнтів)
1	2	3	4
Максимальні значення			
Деформація колони	рад	0,02897	0,03098
Деформація каната	м	0,005461	0,005454
Відхилення вантажу від вертикалі у площині руху візка	м	0,1942	0,1970
Відхилення вантажу від вертикалі у площині, що перпендикулярна площині руху візка	рад	0,02483	0,02591
Рушійний момент приводу механізму повороту крана	Нм	192204	201402
Рушійний момент приводу механізму переміщення візка	Нм	1131	1107
Потужність приводу механізму повороту крана	Вт	13257	10166

1	2	3	4
Потужність приводу механізму переміщення візка	Вт	4752	4752
Сумарна потужність приводу механізмів повороту крана та переміщення візка	Вт	18009	14918
Середньоквадратичні значення			
Деформація колони	рад	0,019856	0,020258
Деформація каната	м	0,0046264	0,0046204
Відхилення вантажу від вертикалі у площині руху візка	м	0,10161	0,09843
Відхилення вантажу від вертикалі у площині, що перпендикулярна площині руху візка	рад	0,013447	0,013868
Рушійний момент приводу механізму повороту крана	Нм	13847	141275
Рушійний момент приводу механізму переміщення візка	Нм	978	976
Потужність приводу механізму повороту крана	Вт	5671	5662

1	2	3	4
Потужність приводу механізму переміщення візка	Вт	3101	3095
Сумарна потужність приводу механізмів повороту крана та переміщення візка	Вт	8690	8653

Аналіз графічних залежностей (рис. 9.8-9.15), а також чисельних даних, які наведено у табл. 9.5, показує, що обидва розв'язки задачі є доволі близькими. Це означає, що базовий розв'язок, який відповідає лише забезпеченню крайових умов руху системи (9.22), задано вдало у сенсі мінімізації критерію (9.21). Тобто, суттєвий вплив на мінімізацію критерію (9.21) мають задані крайові умови руху (9.22).

Величини критерію (9.21), які відповідають базовому розв'язку та розв'язку, що отриманий з застосуванням оптимізаційного підходу, близькі між собою. Повторні запуски алгоритму PSO-Rot-Ring не дали змоги знайти кращий результат.

Можна висунути гіпотезу, що знайдений оптимальний розв'язок відповідає глобальному мінімуму критерію (9.21) на множині коефіцієнтів $A_0 \dots A_4, B_0 \dots B_4$, що обмежені значеннями $-10^{-7} \dots 10^{-7}$.

9.3 Планування оптимальної траєкторії руху дволанкового маніпулятора

Однією з головних важливих особливостей сучасного виробництва є широке використання роботів. Випадок, коли рухаються лише дві ланки роботів, відомий як маніпулятор з двома степенями свободи (2 DOF) [169].

Для керування роботами з 2 DOF використовується широкий спектр підходів. Модифіковане керування режимом ковзання та керування зворотним кроком для цієї невизначеної нелінійної системи, чітко демонструють свою високу ефективність [170]. Мета керування у цій роботі була досягнута, але ці два контролери показують на різну поведінку під час процесу керування.

Добре відомий H_∞ -оптимальний регулятор можна успішно застосувати для подібних задач керування, навіть для маніпулятора міліметрового масштабу 2 DOF з приводами на основі м'якого гідрогелю [171].

Автори в дослідженні [172] вирішили задачу відстеження, пов'язану з траєкторією, залучаючи нелінійну модель прогнозного керування. Шляхом математичної трансформації задачі вони звели її до задачі квадратичного програмування та підкреслили важливість вибору оптимізаційного алгоритму для її вирішення.

У роботі [173] лінійно-квадратичний регулятор використано для проектування оптимальних елементів керування роботом для слідування по бажаній траєкторії руху. Остання була представлена як поліном 5-го порядку.

Метаевристичні прийоми також широко використовуються для задач оптимізації руху роботів. Наприклад, у роботі [174] алгоритми BA, PSO та ABC були застосовані для налаштування ПД-регулятора робота з 2 DOF. Результати свідчать про ефективність пошуку рішень таких задач.

Для вирішення задач планування шляху дволанкової руки робота використовується модифікований алгоритм lbest-PSO [175]. Це дозволило згенерувати найкоротший шлях із уникненням перешкод.

Таким чином, застосування метаевристичних методів для проектування оптимального керування роботом може принести ефективні рішення лінійних проблем. У поточному дослідженні ця теза доведена для випадку, коли критерієм мінімізації є нелінійний функціонал.

9.3.1 Постановка задачі

Динамічна модель маніпулятора для керування проілюстрована на рис. 9.16.

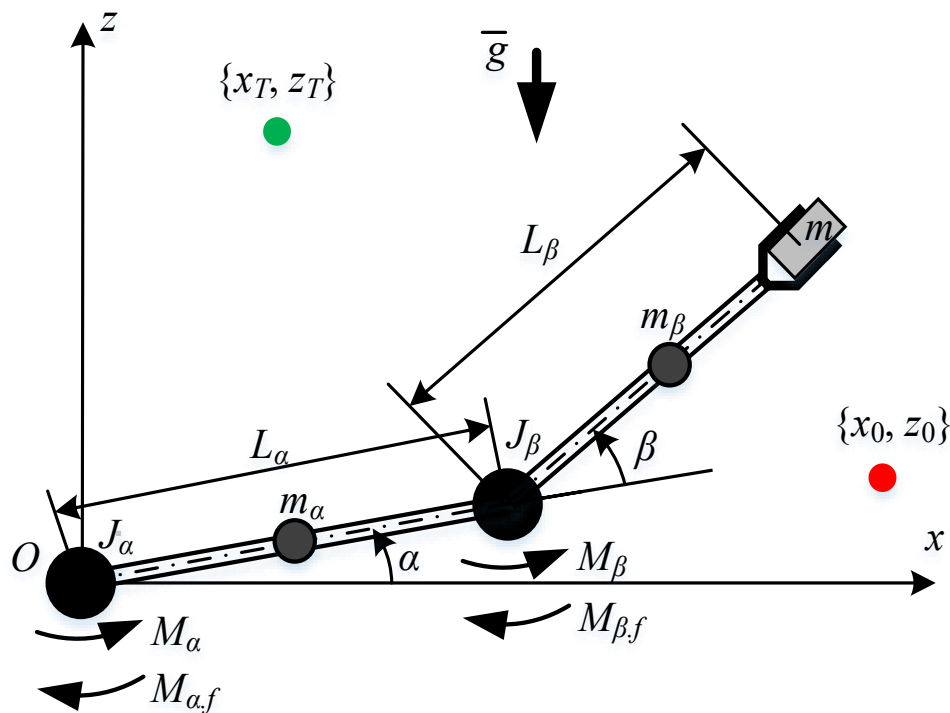


Рисунок 9.16 Динамічна модель робота-маніпулятора

У представленій моделі (рис. 9.16) на кінці другої ланки знаходиться хватний пристрій робота, який несе вантаж масою m . Обидві ланки, довжиною L_α і L_β відповідно, здійснюють обертальний рух, який зумовлений приводними моментами M_α і M_β , моментами тертя $M_{\alpha,f}$ і $M_{\beta,f}$ та інерційними особливостями елементів маніпулятора.

Початкова точка описується координатами x_0 і y_0 , кінцева – координатами x_T і y_T . Оскільки обидві ланки обертаються, проекції ланок і ваги маніпулюваного тіла змінюються. Змінюється і загальний момент інерції системи.

Таким чином, математична модель описується системою нелінійних диференціальних рівнянь:

$$\left\{ \begin{array}{l} M_{\alpha} - M_{\alpha,f} \text{sign}(\dot{\alpha}) = (m_{\alpha} \frac{\partial y_1}{\partial \alpha} + m_{\beta} \frac{\partial y_c}{\partial \alpha} + m \frac{\partial y}{\partial \alpha})g + \\ + J_{\alpha} \ddot{\alpha} + m_{\beta} (\ddot{x}_c \frac{\partial x_c}{\partial \alpha} + \ddot{y}_c \frac{\partial y_c}{\partial \alpha}) + m (\ddot{x} \frac{\partial x}{\partial \alpha} + \ddot{y} \frac{\partial y}{\partial \alpha}); \\ M_{\beta} - M_{\beta,f} \text{sign}(\dot{\beta}) = (m_{\beta} \frac{\partial y_c}{\partial \beta} + m \frac{\partial y}{\partial \beta})g + J_{\beta} \ddot{\beta} + \\ + m_{\beta} (\ddot{x}_c \frac{\partial x_c}{\partial \beta} + \ddot{y}_c \frac{\partial y_c}{\partial \beta}) + m (\ddot{x} \frac{\partial x}{\partial \beta} + \ddot{y} \frac{\partial y}{\partial \beta}), \end{array} \right. \quad (9.70)$$

де α і β узагальнені координати динамічної системи, які є кутовими координатами першої та другої ланок відповідно; J_{α} і J_{β} – зведені моменти інерції ланок маніпулятора робота; m – зведена маса маніпуляційного пристрою та тіла, яким маніпулюють; m_{α} і m_{β} – зведені маси першої та другої ланок маніпулятора відповідно; M_{α} і M_{β} – зведені крутні моменти приводів робота; $M_{\alpha,f}$ і $M_{\beta,f}$ – крутні моменти, які викликані силами тертя в шарнірних

з'єднаннях $\frac{\partial y_1}{\partial \alpha}$, $\frac{\partial y_c}{\partial \alpha}$, $\frac{\partial y}{\partial \alpha}$, $\frac{\partial x_c}{\partial \alpha}$, $\frac{\partial x}{\partial \alpha}$, $\frac{\partial y_c}{\partial \beta}$, $\frac{\partial y}{\partial \beta}$, $\frac{\partial x_c}{\partial \beta}$, $\frac{\partial y_c}{\partial \beta}$, $\frac{\partial x}{\partial \beta}$, $\frac{\partial y}{\partial \beta}$, \ddot{x}_c , \ddot{y}_c , \ddot{x} , \ddot{y} – оператори, які можна описати таким чином:

$$\left\{ \begin{array}{l} \frac{\partial y_1}{\partial \alpha} = \frac{L_{\alpha} \cos(\alpha)}{2}; \\ \frac{\partial y_c}{\partial \alpha} = L_{\alpha} \cos(\alpha) + \frac{L_{\beta} \cos(\alpha + \beta)}{2}; \\ \frac{\partial y}{\partial \alpha} = L_{\alpha} \cos(\alpha) + L_{\beta} \cos(\alpha + \beta); \\ \frac{\partial x_c}{\partial \alpha} = -L_{\alpha} \sin(\alpha) + L_{\beta} \sin(\alpha + \beta); \\ \frac{\partial x}{\partial \alpha} = -L_{\alpha} \sin(\alpha) - L_{\beta} \sin(\alpha + \beta); \\ \frac{\partial y}{\partial \alpha} = L_{\alpha} \cos(\alpha) + L_{\beta} \cos(\alpha + \beta); \\ \frac{\partial y_c}{\partial \beta} = \frac{L_{\beta} \cos(\alpha + \beta)}{2}; \\ \frac{\partial y}{\partial \beta} = L_{\beta} \cos(\alpha + \beta); \\ \frac{\partial x_c}{\partial \beta} = -\frac{L_{\beta} \sin(\alpha + \beta)}{2}; \\ \frac{\partial x}{\partial \beta} = -L_{\beta} \sin(\alpha + \beta); \end{array} \right.$$

$$\left\{ \begin{array}{l}
\ddot{x}_c = -\ddot{\alpha}L_\alpha \sin(\alpha) - \dot{\alpha}^2 L_\alpha \cos(\alpha) - \frac{\ddot{\alpha} + \ddot{\beta}}{2} L_\beta \sin(\alpha + \beta) - \\
-\frac{(\dot{\alpha} + \dot{\beta})^2}{2} L_\beta \cos(\alpha + \beta); \\
\ddot{y}_c = -\ddot{\alpha}L_\alpha \cos(\alpha) - \dot{\alpha}^2 L_\alpha \sin(\alpha) - \frac{\ddot{\alpha} + \ddot{\beta}}{2} L_\beta \cos(\alpha + \beta) - \\
-\frac{(\dot{\alpha} + \dot{\beta})^2}{2} L_\beta \sin(\alpha + \beta); \\
\ddot{x} = -\ddot{\alpha}L_\alpha \sin(\alpha) - \dot{\alpha}^2 L_\alpha \cos(\alpha) - (\ddot{\alpha} + \ddot{\beta})L_\beta \sin(\alpha + \beta) - \\
-(\dot{\alpha} + \dot{\beta})^2 L_\beta \cos(\alpha + \beta); \\
\ddot{y} = -\ddot{\alpha}L_\alpha \cos(\alpha) - \dot{\alpha}^2 L_\alpha \sin(\alpha) - (\ddot{\alpha} + \ddot{\beta})L_\beta \cos(\alpha + \beta) - \\
-(\dot{\alpha} + \dot{\beta})^2 L_\beta \sin(\alpha + \beta).
\end{array} \right. \quad (9.71)$$

У рамках поточного дослідження розглядаємо стан спокою на початку та в кінці руху. Крім того, потрібні нульові кутові прискорення ланок. Таким чином, крайові умови для першої ланки такі:

$$\left\{ \begin{array}{l}
\alpha(0) = \alpha_0, \dot{\alpha}(0) = 0, \ddot{\alpha}(0) = 0; \\
\alpha(T) = \alpha_T, \dot{\alpha}(T) = 0, \ddot{\alpha}(T) = 0,
\end{array} \right. \quad (9.72)$$

а для другої:

$$\left\{ \begin{array}{l}
\beta(0) = \beta_0, \dot{\beta}(0) = 0, \ddot{\beta}(0) = 0; \\
\beta(T) = \beta_T, \dot{\beta}(T) = 0, \ddot{\beta}(T) = 0,
\end{array} \right. \quad (9.73)$$

де α_0 та β_0 , α_T та β_T – початкове та кінцеве кутові положення першої та другої ланок відповідно. Для розглянутої задачі існує два набори значень α_0 , β_0 , α_T і β_T , які задовольняють координати початкової $\{x_0, y_0\}$ та кінцевої $\{x_T, y_T\}$ точок. Ми знайдемо їх у наступному викладі матеріалу.

Переміщення маніпулюваного тіла від початкової до кінцевої точок повинен забезпечувати мінімізацію комплексного критерію:

$$Cr = \delta_1 \sqrt{T^{-1} \int_0^T (M_\alpha \dot{\alpha})^2 + (M_\beta \dot{\beta})^2 dt} + \delta_2 \max(M_\alpha \dot{\alpha}) + (1 - \delta_1 - \delta_2) \max(M_\beta \dot{\beta}) \rightarrow \min, \quad (9.74)$$

де δ_1 та δ_2 – вагові коефіцієнти, що відповідають мінімізації середньоквадратичної загальної потужності робота та мінімізації максимального значення потужності приводу, що обертає першу ланку робота-маніпулятора. Розробка критерію (9.74) є компромісом між інтегральним і максимальним критеріями.

9.3.2 Розв’язання задачі

Для знаходження виразів для визначення величин α_0 і β_0 необхідно розв’язати систему рівнянь:

$$\begin{cases} L_\alpha \cos(\alpha_0) + L_\beta \cos(\beta_0) = x_0; \\ L_\alpha \sin(\alpha_0) + L_\beta \sin(\beta_0) = y_0. \end{cases} \quad (9.75)$$

Зазначимо, що система рівнянь (9.75) має два розв’язки:

$$\begin{aligned} \alpha_{0.1,2} &= \arctg \left(\frac{L_\alpha^3 y_0^2 + L_\alpha y_0^2 (x_0^2 + y_0^2 - L_\beta^2) \pm x_0 q}{L_\alpha x_0 (L_\alpha^2 - L_\beta^2 + x_0^2) y_0 + L_\alpha x_0 y_0^3 \mp y_0 q} \right); \\ \beta_{0.1,2} &= \arctg \left(\frac{-L_\alpha^3 y_0^2 + L_\alpha y_0^2 (x_0^2 + y_0^2 - L_\beta^2) \mp x_0 q}{\mp L_\alpha x_0 (L_\alpha^2 - L_\beta^2 + x_0^2) y_0 + L_\alpha x_0 y_0^3 + y_0 q} \right); \\ q &= \sqrt{L_\alpha^2 y_0^2 (2L_\alpha^2 (x_0^2 + y_0^2 + L_\beta^2) - L_\alpha^4 - (x_0^2 + y_0^2 - L_\beta^2)^2)}. \end{aligned} \quad (9.76)$$

Надалі розглянемо обидва рішення (9.76) і виберемо те, яке краще мінімізує критерій (9.74). Подібні розрахунки справедливі для визначення α_T і β_T . Для цього випадку індекси „0” у формулах (9.76) слід замінити на „T”.

Для виконання початкових умов (9.72), (9.73) та мінімізації критерію (9.74) розглянемо розв’язки задачі (закони $\alpha(t)$ та $\beta(t)$) як суперпозиції функцій:

$$\begin{aligned} \alpha(t) &= \alpha_{b.c.}(t) + \alpha_{Cr.}(t); \\ \beta(t) &= \beta_{b.c.}(t) + \beta_{Cr.}(t), \end{aligned} \quad (9.77)$$

де $\alpha_{b.c.}$ і $\beta_{b.c.}$ – закони руху, які задовольняють граничні умови (9.72) та (9.73) відповідно; α_{Cr} і β_{Cr} – закони руху, які задовольняють критерій мінімізації Cr . Для закону руху $\alpha(t)$ можна записати такі вирази:

$$\begin{aligned}\alpha_{b.c.}(t) &= \alpha_0 + (\alpha_T - \alpha_0)(10\tau^3 - 15\tau^4 + 6\tau^5); \\ \alpha_{Cr}(t) &= (1-\tau)^3 \tau^3 (A_0 + A_1\tau + A_2\tau^2 + A_3\tau^3),\end{aligned}\quad (9.78)$$

де τ – відносний час ($\tau=t/T$); T – тривалість керованого руху; A_0, \dots, A_3 – коефіцієнти, що мінімізують критерій Cr . Для випадку координати β всі коефіцієнти A_0, \dots, A_3 повинні бути замінені на B_0, \dots, B_3 .

Таким чином, вихідна задача (9.70)-(9.74) зводиться до мінімізації цільової функції:

$$Cr(A_0, A_1, A_2, A_3, B_0, B_1, B_2, B_3) \rightarrow \min. \quad (9.79)$$

Оскільки функція (9.79) є нелінійною, нам потрібно вибрати оптимізатор із сильними пошуковими властивостями. У рамках поточного дослідження обрали R-Ring-PSO [71] – модифікацію PSO, яка поєднує в собі добрі пошукові характеристики з досить швидкою збіжністю [176, 177].

Усі розрахунки проводились для випадків початкової та кінцевої кутових координат ланок маніпулятора (9.76). Наведений розв'язок задачі відображає найкращий (з точки зору мінімізації значення критерію Cr) варіант. Усі розрахунки проводились для числових значень системи, наведених у табл. 9.6. Область пошуку для кожного з коефіцієнтів A_0, \dots, A_3 та B_0, \dots, B_3 була встановлена від -50 до 50.

Таблиця 9.6. Чисельні параметри, які використані у розрахунках

Позначення параметра	Одиниця вимірювання	Значення
1	2	3
$M_{\alpha,f}$	Нм	0,20
$M_{\beta,f}$	Нм	0,10
J_α	кгм ²	0,04

Продовження табл. 9.6

1	2	3
J_β	кгм ²	0,03
L_α	м	0,30
L_β	м	0,25
m_α	кг	0,60
m_β	кг	0,50
m	кг	5,00
T	с	1,00
δ_1	-	0,60
δ_2	-	0,20
x_0	м	0,40
y_0	м	0,20
x_T	м	0,10
y_T	м	0,45

Графік, який показує динаміку мінімізації Cr , наведено на рис. 9.17. Червона лінія на рис. 9.17 показує базовий розв'язок задачі – випадок нульових значень коефіцієнтів A_0, \dots, A_3 та B_0, \dots, B_3 .

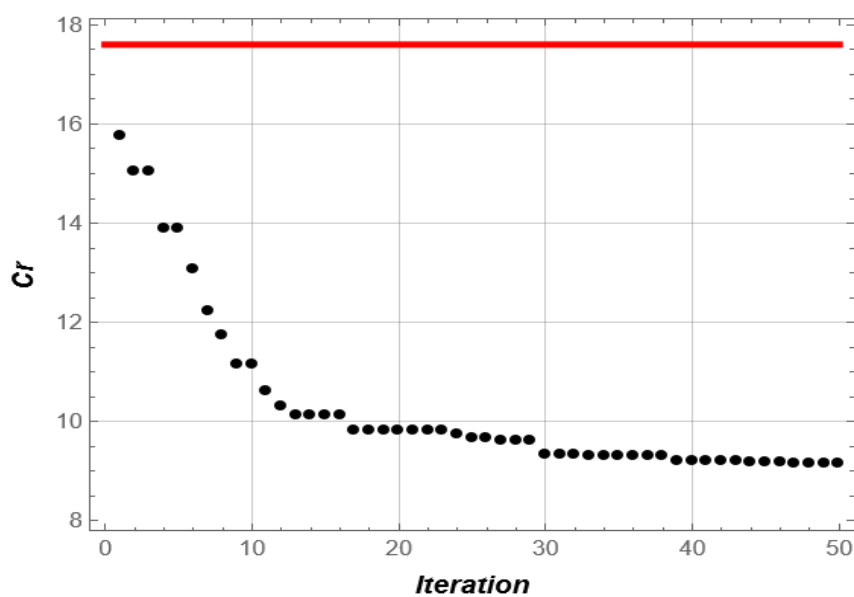


Рисунок. 9.17 – Зменшення критерію Cr при застосуванні алгоритму R-Ring-PSO

Аналіз графіка (рис. 9.17) чітко свідчить, що на першій же ітерації R-Ring-PSO знаходить розв'язок, який є кращим, ніж базовий. На подальших ітераціях значення цільової функції продовжує зменшуватися. Приблизно після сорока ітерацій алгоритм показує стадію збіжності. Ми можемо зробити висновок, що R-Ring-PSO був розумним вибором для мінімізації Cr . Знайдений розв'язок наведено в табл. 9.7.

Таблиця 9.7 – Числові значення знайдених коефіцієнтів

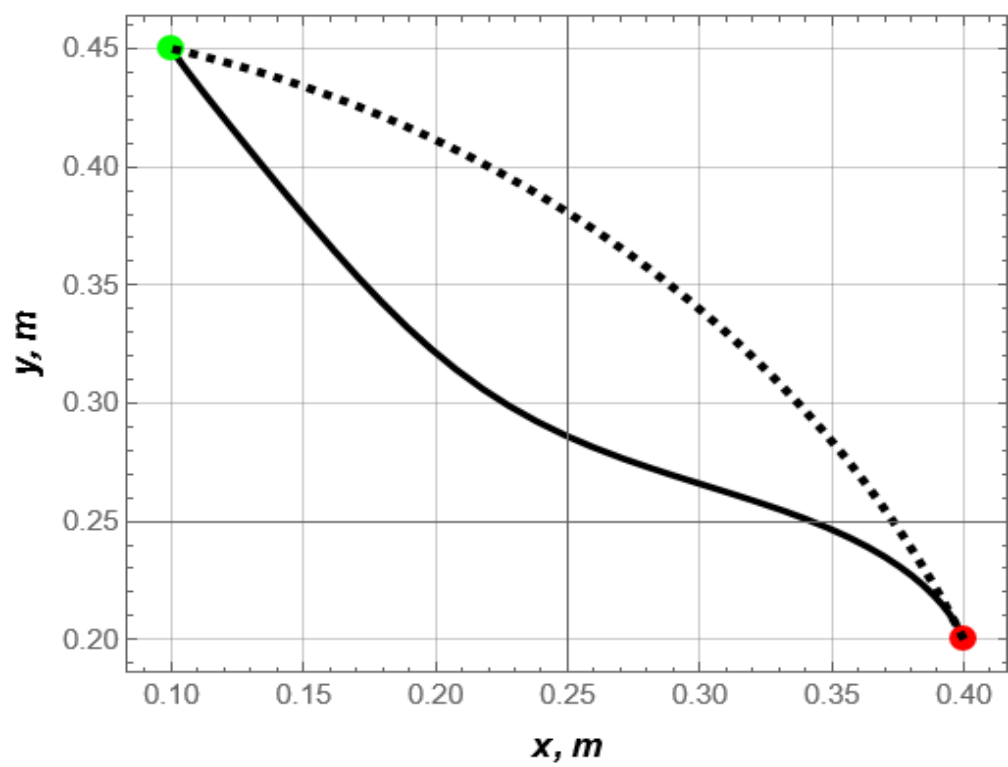
Коефіцієнт	Значення
A_0	$6,256 \cdot 10^0$
A_1	$-3,877 \cdot 10^1$
A_2	$-2,115 \cdot 10^1$
A_3	$2,896 \cdot 10^0$
B_0	$-7,782 \cdot 10^{-2}$
B_1	$1,260 \cdot 10^0$
B_2	$9,313 \cdot 10^0$
B_3	$2,414 \cdot 10^1$

9.3.3 Короткий аналіз оптимального керування

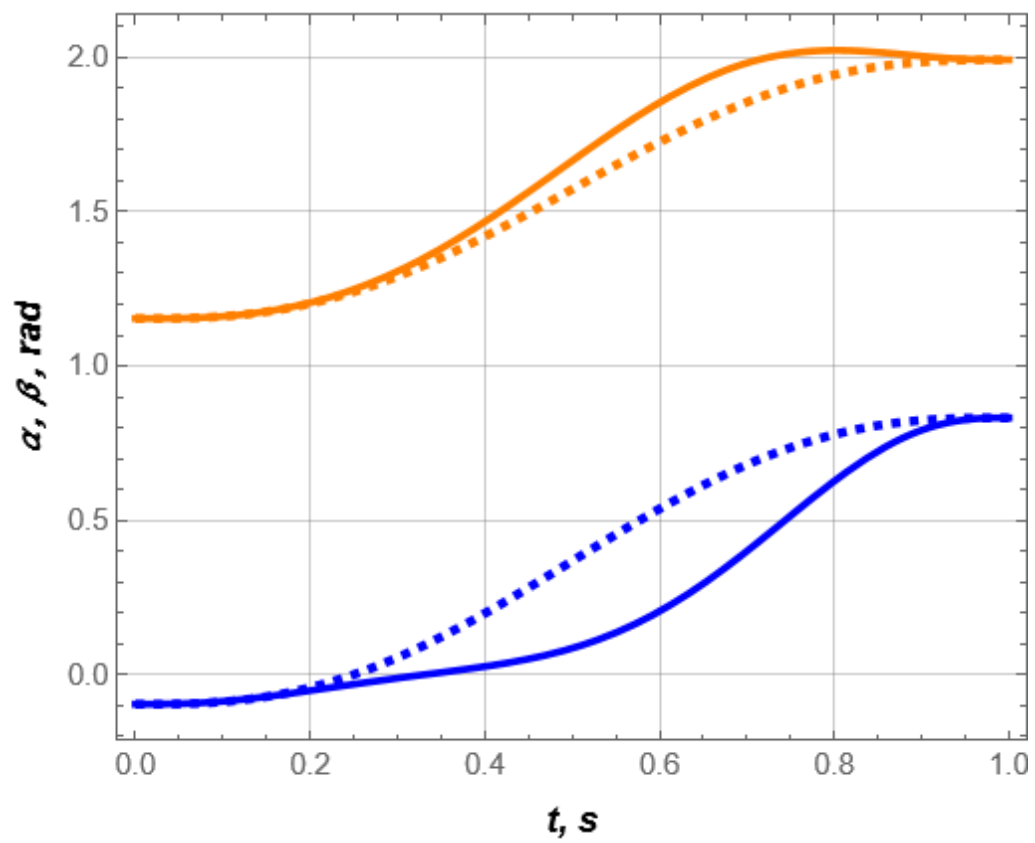
Для аналізу режимів руху за знайденими оптимальними керуваннями було побудовано графіки (рис. 9.18). Штрихові криві відносяться до базисного розв'язку задачі, суцільні – до функцій (9.78) з коефіцієнтами з табл. 9.7.

На рис. 9.18 сині криві відносяться до руху першої ланки, помаранчеві – до руху другої ланки. Чорні криві на рис. 9.18, г показують споживану потужність приводів обох ланок.

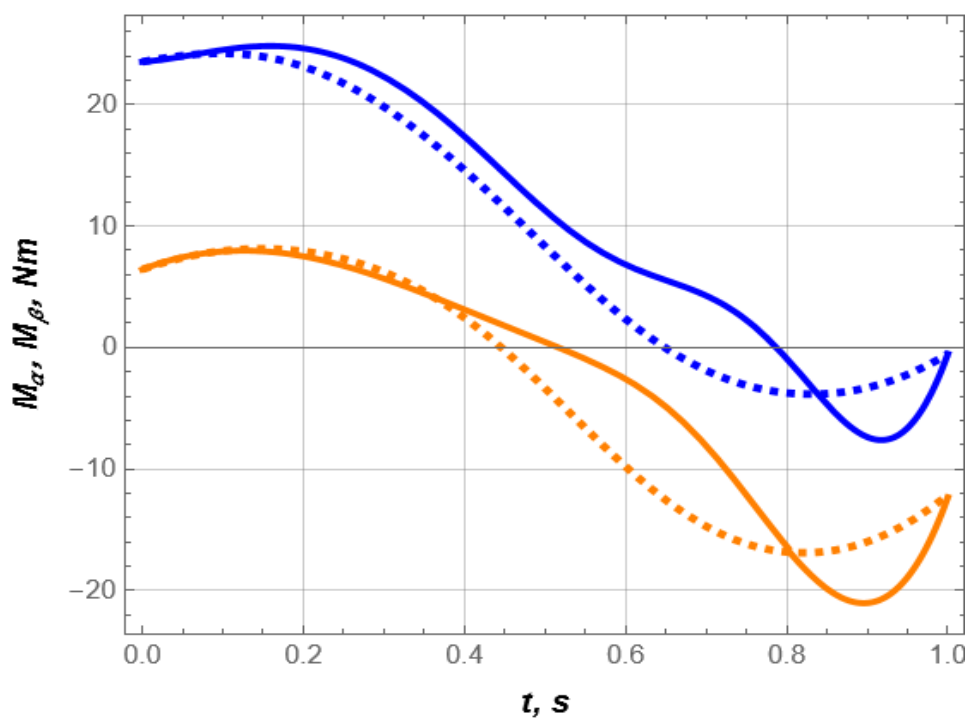
Для повного аналізу було розраховано набір оціночних показників, які ілюструють ключові динамічні та енергетичні особливості руху. Вони наведені в табл. 9.8.



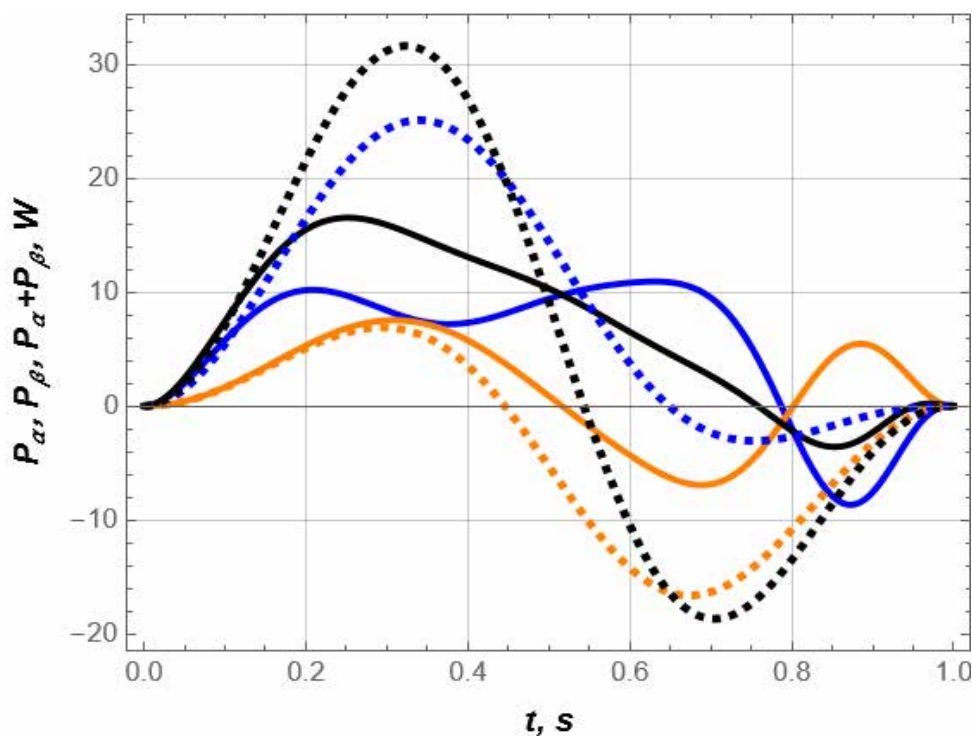
a)



б)



б)



г)

Рисунок 9.18 – Графіки руху ланок при оптимальному керуванні: а) траєкторія руху керованого тіла у площині „z-y”; б) кутові положення ланок маніпулятора; в) приводні моменти, що діють на першу та другу ланки маніпулятора; г) потужність, яку споживають приводи ланок, і загальна споживана потужність

Найважливішою особливістю, про яку свідчать графіки (рис. 9.18), є їх плавність. Практична реалізація таких законів руху не потребує швидкої зміни моментів. Таким чином, приводи не експлуатуються в жорсткому режимі, що зумовлює низький рівень динамічних навантажень в елементах маніпулятора робота та зменшує втрати енергії в приводах робота.

Таблиця 9.8 – Числові значення оціночних показників

Оціночний показник	Одиниця вимірювання	Значення для розв'язку	
		базового	оптимального
Клас динамічних показників			
M_α RMS*	Нм	14,5	15,9
M_β RMS		10,7	10,4
M_α максимум		24,2	24,8
M_α мінімум		-3,84	-7,62
M_β максимум		8,12	7,96
M_β мінімум		-16,9	-21,0
Довжина траєкторії	м	0,48	0,46
Клас енергетичних показників			
P_α RMS	Вт	12,9	7,85
P_β RMS		8,44	4,61
$P_\alpha+P_\beta$ RMS		15,4	9,08
P_α максимум		25,1	10,9
P_α мінімум		-3,05	-8,65
P_β максимум		6,89	7,53
P_β мінімум		-16,6	-6,92
$P_\alpha+P_\beta$ максимум		31,6	16,5
$P_\alpha+P_\beta$ мінімум		-18,7	-3,57
Cr			17,6

* RMS – середньоквадратичне значення

Зазначена особливість зумовлена класом функцій, які використовуються для апроксимації оптимального керування. Вони є поліномами (9.78).

Можна зробити висновок, що за оптимального керування тіло, яким маніпулюють, знаходиться ближче до початку координат. Таким чином, вимога до еквівалентного крутного моменту для переміщення тіла нижча. Довжини обох траєкторій (табл. 9.8) майже однакові.

Спостерігаючи за координатами кутових координат (рис. 9.18, б), можна стверджувати, що базисні закони не вимагають зміни швидкостей (справді, кутові координати завжди зростають). Оптимальний закон координати β означає зміну знаку її похідної за часом.

Порівняння функцій M_α і M_β свідчить про незначну перевагу базисних функцій (у якісному та кількісному відношеннях). Найбільш істотна різниця – мінімальне значення M_α (49,6%). Однак інші показники мають незначні відмінності.

Оскільки критерій Cr (9.79) містить функції потужності, найбільш суттєві відмінності є у функціях P_α , P_β та $P_\alpha+P_\beta$. Перш за все, застосування R-Ring-PSO дозволило знизити значення Cr на 47,9%. Для функцій P_α , P_β та $P_\alpha+P_\beta$ середньоквадратичні значення зменшились у порівнянні з базисним законом на 39,1...41,0%. Максимальні показники зменшились на 47,7...56,6%. Аналогічний діапазон для мінімальних значень становить 58,3...80,9%. Винятками є лише мінімальне значення P_α (зросло на 64,7 %) та максимальне значення P_β (зросло на 8,5 %).

Аналіз графічних залежностей, які показані на рис. 9.18, г показує перевагу оптимального керування над базисним. Знайдені коефіцієнти (табл. 9.7) роблять криві потужності P_α і P_β більш узгодженими для того, щоб мінімізувати критерій Cr . Це означає, що їхні екстремуми трапляються в різні моменти часу. В момент часу 0,7 с функція P_α має максимум, який „компенсує” мінімум функції P_β .

Висновки до дев'ятого розділу:

1. Виконано параметричну оптимізацію частотно-керованого електроприводу, яка дозволила мінімізувати величину небажаних динамічних (крутний момент у пружній муфті та зусилля у тяговому канаті приводу вантажного візка) та енергетичних (потужність у приводі) характеристик динамічної системи «візок-вантаж» механізму зміни вильоту баштового крана. Для проведення оптимізації використано метод R-Ring-PSO. Встановлено, що мінімальні значення динамічних критеріїв при використанні оптимальних характеристик (лінійної та 2S-подібної) у діапазоні 6,9... 27,9 % є меншими ніж при використанні не оптимальних (U-подібної та S-подібної) характеристик. Мінімальне значення енергетичного критерію при використанні U-подібної характеристики на 30,4 % є меншим ніж при S-подібній. Тому для мінімізації величини небажаних динамічних показників при частотно-керованому пуску динамічної системи «візок-вантаж» механізму зміни вильоту бажано використовувати лінійну та 2S-подібну характеристики, а для мінімізації величини небажаного енергетичного показника U-подібну. При цьому необхідно також раціонально встановлювати параметри t_1 та U_0 .
2. Представлено результати досліджень оптимізації режимів пуску при сумісному русі механізмів зміни вильоту вантажу та повороту баштового крана з балочною стрілою. Стрілова система баштового крана представлена механічною системою з 6 ступенями свободи, для якої складено систему диференціальних рівнянь руху. Для мінімізації енергетичних витрат в роботі поставлено варіаційну задачу, в якій необхідно визначити закони руху механізмів зміни вильоту та повороту крана при мінімізації інтегрального критерію та забезпеченні необхідних крайових умов переміщень та швидкостей ланок стрілової системи. Шукані функції координат зміни вильоту та повороту вантажу

представлено у вигляді поліномів з двома доданками. Перші доданки – це поліноми, що задовольняють крайові умови руху стрілової системи, а другі – поліноми, в які входять вільні коефіцієнти і задовольняють нульовим крайовим умовам. Для знаходження вільних коефіцієнтів в варіаційній задачі використано метод R-Ring-PSO. Отримано режими руху, які мінімізують потужність приводних механізмів, зменшують динамічні навантаження і усувають коливання ланок та вантажу при виході на усталений режим руху механізмів.

3. У дослідженні поставлено та розв'язано задачу оптимального керування роботом-маніпулятором. Вона зведена до задачі мінімізації нелінійної цільової функції. Її нелінійність зумовлена нелінійністю математичної моделі системи та нелінійним комплексним критерієм оптимізації. Для пошуку розв'язку задачі застосовано метаевристичний метод R-Ring-PSO. Аналіз графіку зменшення значень критерію Cr під час застосування алгоритму свідчить про його збіжність та локалізацію розв'язку задачі. Графіки функцій P_α і P_β (рис. 9.18, г), а також розраховані значення оціночних показників (табл. 9.8) підтверджують покращення енергетичних властивостей руху системи за оптимального керування. Один із недоліків знайденого розв'язку пов'язаний зі зміною знаку кутової швидкості. Іншими незначними недоліками є гірші значення показників небажаного крутного моменту (середньоквадратичного та екстремального). Щоб уникнути їх, слід розробити критерій, у якому ці небажані функції штрафуються, що представляє предмет подальшого дослідження.

ВИСНОВКИ

1. Аналіз кількості проіндексованих у наукометричних базах праць за тематикою PSO дає підстави стверджувати, що наукові дослідження у галузі модифікації, гібридизації та застосування алгоритмів, що отримані на основі PSO, активно проводяться у більшості розвинених країн світу. Більше того, спостерігається збільшення кількості наукових праць за роками, що вказує на те, що резерви удосконалень методу PSO наразі не вичерпані та існує стійка потреба до продовження досліджень як у частині вдосконалення алгоритмів, так і їх застосування у різноманітних галузях прикладної науки, зокрема, інженерії.
2. У роботі із застосуванням різних підходів проведено подальші модифікації PSO, які дали змогу отримати алгоритми із покращеними пошуковими характеристиками. Вони ґрунтуються на ефекті коливань вектору положення частинки при його оновленні (PV-IA-PSO-1 та PV-IA-PSO-2), м'якій переініціалізації рою (SR-D-PSO), гібридизації алгоритмів (M-PSO-Hybrid) та синтезі локальної та глобальної топологій зв'язків між частинками (LG-Ring-PSO). Всі ці алгоритми мають параметри, які впливають на їхні пошукові особливості.
3. Розроблено методику визначення оптимальних значень параметрів розроблених модифікацій PSO, яка ґрунтується на багатократному запуску алгоритмів для мінімізації тестових функцій. У цих експериментах налаштуванні параметри розглядалися як незалежні фактори і на основі діаграм Бокса-Віскера знайдено оптимальні діапазони їхніх значень. Це дало змогу отримати рекомендації щодо налаштувань параметрів ($\sigma_{\max}=0,35\dots0,4$ для PV-IA-PSO-1, $B=1\dots1,1$ для PV-IA-PSO-2, $\psi=7$ для M-PSO-Hybrid, $a=1\dots2$ для LG-Ring-PSO).
4. Для того, щоб довести ефективність модифікацій PSO обрано двадцять загальновідомих тестових функцій, які мають різні особливості, що впливають на відшукування їхніх глобальних мінімумів. Така різноманітність характеристик тестових функцій зумовлена вимогами

застосовності і високої ефективності роботи розроблених модифікацій PSO для якомога ширшого класу задач оптимізації. Опис тестових функцій та їхніх характеристик можуть бути використані для потреб оцінки інших оптимізаційних алгоритмів.

5. Розроблено оцінки та інструменти, які дають змогу вивчати еволюцію руху рою, виконувати динамічне (за ітераціями) оцінювання статистичних показників алгоритму, визначати успішність його роботи та отримувати інформацію стосовно залученості частинок у пошук розв'язку задачі. Цей комплекс результатів дає змогу цілеспрямовано знаходити сильні і слабкі сторони модифікацій PSO, порівнювати алгоритми між собою та виробляти подальші стратегії їх удосконалень.
6. Аналіз статистичних даних щодо результатів чисельних експериментів по дослідженню ефективності застосування PSO-алгоритмів для задач мінімізації синтетичних тестових функцій показав, що для функцій f_5 , f_9 , f_{18} , f_{19} та f_{20} алгоритми PV-IA-PSO та PSO-Hybrid виявились неефективними. Для решти функцій їхнє застосування асоціювалось із суттєвою ефективністю локалізації мінімумів. Наголосимо, що відмінність між PV-IA-PSO та IA-PSO є незначною, що дає підстави вважати алгоритм IA-PSO гарною платформою для подальшого вдосконалення пошукових властивостей алгоритму.
7. Розроблено загальний l-best алгоритм RSTC_{J/K}-PSO та доведено, що він є узагальненням добре відомих алгоритмів Ring-PSO і фон Неймана-PSO. Аналіз його пошукових результатів дає змогу стверджувати, що існує певний клас функцій, мінімуми яких досить ефективно можуть бути знайдені за допомогою цього алгоритму. При цьому рекомендується виконати кілька пробних запусків алгоритму з налаштуванням $J=0,04K$, $J=0,15K$, $J=0,25K$, $J=0,50K$ і $J=0,7K$ та встановити таку величину J , при якій робота алгоритму буде найефективнішою у сенсі „витрачені обчислювальні ресурси – досягнутий результат”.

8. Аналіз розв'язання задач оптимального керування рухом нелінійних динамічних систем маятникового типу показав сильні пошукові властивості алгоритмів PV-IA-PSO та PSO-Hybrid. Ці алгоритми показали високу ефективність при розв'язанні задач оптимального налаштування ПІ-регуляторів об'єктів 2-го та 3-го порядків. Наявність складних топологічних особливостей цільових функцій задач оптимального керування, що отримані із застосуванням декількох штрафних функцій, не створює принципових складнощів при застосуванні вказаних алгоритмів.
9. Застосування PSO-алгоритмів для розв'язання 12 задач інженерії показує ефективність алгоритмів VCT-PSO, Ring-PSO, R-Ring-PSO, MED-PSO, PV-ND-PSO та PSO-Hybrid. На основі задачі оптимального за швидкодією керування розроблено задачу глобальної оптимізації та детально досліджено особливості її цільової функції. Чисельні експерименти, які проведені на основі її мінімізації, показали перспективність алгоритмів DE/best/1/bin та mGWO, Ring-PSO, VCT-PSO, LDWPSO, PV-ND-PSO (при $\sigma_{\max}=0,02$) та PSO-Hybrid. Крім того, на основі проведення чисельних експериментів встановлено, що застосування алгоритму PV-ND-PSO бажано проводити у форматі багаторазового запуску із початковим налаштуванням $\sigma_{\max}=0,35$ з подальшим зменшенням цієї величини.
10. За допомогою методу R-Ring-PSO розв'язано три задачі інженерії: параметрична оптимізація частотно-керованого електроприводу механізму зміни вильоту баштового крана, оптимізація режимів пуску при сумісному русі механізмів зміни вильоту вантажу та повороту баштового крану з балочною стрілою, оптимізація керування рухом робота-маніпулятора з вибором оптимальної траєкторії переміщення його робочого органу. Всі задачі розв'язані із використанням єдиної методології, яка передбачає зведення вихідної задачі до задачі безумовної оптимізації із заданими топологічними властивостями.

ДОДАТКИ

Додаток А

Таблиця А.1 – Кількість проіндексованих у наукометричних базах даних праць за роками

Запити	Роки										Всього за всіма роками
	2023	2022	2021	2020	2019	2018	2017	2016	2015	2014	
Scopus											
PSO	11862	27593	22916	19812	17976	14517	11644	10414	8834	8252	189856
PSO global optimization	4285	9329	7618	6429	5456	4502	3579	3230	2870	2676	61340
PSO modification	648	1269	1057	854	684	536	405	341	311	280	7529
PSO hybridization	472	1076	908	847	633	509	426	383	362	272	6782
PSO application	10670	24277	20060	17153	15262	12282	9807	8682	7396	6766	158977
PSO survey	4084	9040	7370	6258	5450	4182	3259	2774	2311	1894	52658
Web of Science											
PSO	2204	5118	5034	4212	4364	3936	3296	3352	2507	2443	49024
PSO global optimization	281	674	691	573	621	524	488	506	422	376	7622
PSO modification	36	72	75	64	69	54	56	82	38	38	791
PSO hybridization	26	69	82	76	51	34	45	45	36	22	600
PSO application	438	1049	985	780	802	724	594	608	440	442	9385
PSO survey	31	81	76	78	59	63	47	31	34	30	622
Всього проіндексованих робіт											
Scopus	1626586	4003528	3933720	3680572	3534339	3327421	3186354	3091858	2974908	2958650	-
Web of Science	258237	600805	711447	635454	611706	623414	653675	668821	631732	581928	-

Таблиця А.2 – Кількість опублікованих робіт за категоріями Scopus

Запити	Категорії							
	Інженерія	Комп'ютерні науки	Математика	Енергія	Фізика та астрономія	Матеріалознавство	Науки про навколишнє середовище	Науки про прийняття рішень
PSO	105113	96663	46603	27980	19635	15799	10918	10164
PSO global optimization	31981	34713	15897	8438	5055	4628	4018	3432
PSO modification	3628	3503	1730	907	782	889	691	287
PSO hybridization	3067	4334	1971	662	477	536	349	434
PSO application	89205	83567	37307	23613	16336	13519	9388	8775
PSO survey	26983	32663	12770	5745	4394	3735	2568	3606

Таблиця А.3 – Кількість опублікованих робіт за категоріями Web of Science

Запити	Категорії									
	Інженерія / Електротехніка / Електроніка	Комп'ютерні науки / Штучний інтелект	Комп'ютерні науки / Теоретичні методи	Комп'ютерні науки / Інформаційні системи	Автоматизація / Системи керування	Телекомунікації	Енергетика / Паливо	Комп'ютерні науки / Міждисциплінарні застосування	Міждисциплінарна інженерія	Матеріалознавство
PSO	15091	11022	6068	5119	4437	4171	4125	4016	2971	1806
PSO global optimization	2318	2371	1126	912	795	623	520	810	479	243
PSO modification	210	227	105	69	49	47	51	79	48	35
PSO hybridization	137	220	114	100	26	50	35	84	50	15
PSO application	2976	2264	1242	1075	844	875	673	850	573	398
PSO survey	100	100	62	59	19	39	37	38	34	24

Таблиця А.4 – Кількість опублікованих праць за країнами

Запити	Країни											
	Китай	Індія	Іран	Сполучені Штати Америки	Великобританія	Малайзія	Тайвань	Австралія	Саудівська Аравія	Єгипет	Україна	Ізраїль
Scopus												
PSO	77245	31604	13296	12213	6066	6001	4741	4428	4272	4214	188	179
PSO global optimization	24954	10576	4346	3891	2223	2245	1617	1833	1772	1772	38	40
PSO modification	2339	1272	772	592	313	396	186	245	314	213	22	11
PSO hybridization	1785	1745	528	458	225	383	157	220	317	277	6	1
PSO application	64167	27891	11981	9439	4969	5446	4083	3927	4030	3837	144	123
PSO survey	17020	11090	3662	3906	2352	2008	1195	1715	1758	1453	55	50
Web of Science												
PSO	18360	7687	4126	3525	1331	1543	1828	1205	1130	1232	24	60
PSO global optimization	3709	1045	537	449	212	225	304	204	183	154	2	5
PSO modification	153	143	79	78	19	29	17	17	29	23	4	1
PSO hybridization	77	187	48	37	11	54	18	13	26	31	-	-
PSO application	3575	1537	732	645	252	314	323	247	220	246	1	9
PSO survey	171	109	42	99	44	20	7	20	11	10	-	1

Додаток Б

Оптимізаційні алгоритми, що розроблені і використані у дослідженні

PSO

PSO[function_, Arguments_, DomainArgumentList_, SwarmPopulation_, Iterations_] :=

(

(*

function - is cost function;

Arguments - are all the independent variables of the cost function. It is array of arguments;

DomainArgumentList - is the search domain. It represents the borders of multivariable domain. It is the array of the borders;

InitialArgumentValues - is an initial values for each argument;

SwarmPopulation - is a number of the particles in the swarm;

Iterations - is a number of the iterations of the main cycle;

CostFunctionValues - is a array of the cost function values during the optimization procedure.

*)

(*ЗАДАННЯ ФУНКЦІЇ ЦІНИ*)

func=function;

CostFunctionValues=List[];

(*ІНІЦІАЛІЗАЦІЯ МАСИВУ ЗНАЧЕНЬ ФУНКЦІЇ ЦІНИ*)

(*ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ*)

w=0.72;c2=1.19; c1=c2;

(*ІНІЦІАЛІЗАЦІЯ РОЮ*)

```

Swarm=List[];
Do[
  (*ПОЧАТКОВІ ПОЛОЖЕННЯ ЧАСТИНОК*)
  ParticlePosition=Table[RandomReal[DomainArgumentList[[i]], {i,
1,Length[Arguments], 1}];
  (*ПОЧАТКОВІ ШВИДКОСТІ ЧАСТИНОК*)
  ParticleVelocity=Table[0, {i, 1,Length[Arguments], 1}];
  ParticleCost=Apply[func,ParticlePosition];
  (*ВИЗНАЧЕННЯ НАЙКРАЩИХ ПОЛОЖЕНЬ ДЛЯ КОЖНОЇ
ЧАСТИНКИ
Найкращі положення спочатку відповідають початковим положенням*)
  ParticleBest=ParticlePosition;
  ParticleBestCost=ParticleCost;
  (*k-та частинка*)
  Particle=List[ParticlePosition, ParticleVelocity, ParticleCost, ParticleBest,
ParticleBestCost];
  AppendTo[Swarm, Particle], SwarmPopulation];

  (*знаходження найкращої частинки у рої*)
  GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];
  BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]];

  (*заповнили значенням величини функції ціни масив*)
  AppendTo[CostFunctionValues, GlobalBestCost];

  (*ЦИКЛ РОЗРАХУНКУ*)
  Monitor[
  Do[
    (*оновлення положень та швидкостей для кожної частинки*)

```

```

Do[
  r1=Table[RandomReal[{0.,1.}], Length[Arguments]];
  r2=Table[RandomReal[{0.,1.}], Length[Arguments]];

  Swarm[[k]][[2]]=w*Swarm[[k]][[2]]+c1*r1*(Swarm[[k]][[4]]-
Swarm[[k]][[1]])+
  c2*r2*(Swarm[[BestParticleNumber]][[4]]-Swarm[[k]][[1]]);

  Swarm[[k]][[1]]=Swarm[[k]][[1]]+Swarm[[k]][[2]];

(*перевірка виходу за межі верхньої та нижньої границі*)
Do[

Piecewise[{{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[1]],Swarm[[k]][[1]][[j]]<DomainArgumentList[[j]][[1]]},
{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[2]],Swarm[[k]][[1]][[j]]>DomainArgumentList[[j]][[2]]}}]

, {j, 1, Length[Arguments]}];

(*оновлення значення функції в новому положенні частинки*)
Swarm[[k]][[3]]=Apply[func, Swarm[[k]][[1]]]
,{k, 1, SwarmPopulation, 1}];

(*оновлення PersonalBest*)
Do[If[Swarm[[k]][[3]]<Swarm[[k]][[5]], Swarm[[k]][[5]]=Swarm[[k]][[3]];
Swarm[[k]][[4]]=Swarm[[k]][[1]], {k, 1, SwarmPopulation, 1}];

(*оновлення GlobalBest*)
GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];
BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,

```

```
1}],GlobalBestCost][[1]][[1]]];
```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost]
```

```
,{step, 2, Iterations, 1}]
```

(*візуалізація даних ІТЕРАЦІЯ-ЗНАЧЕННЯ ФУНКЦІЇ*)

```
,{Round[step/Iterations*100., 0.01] "% " ,GlobalBestCost}];
```

(*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2) значення аргументів при яких вона досягається; 3) дискретні значення функції ціни протягом процедури мінімізації*)

```
Return[{GlobalBestCost, Swarm[[BestParticleNumber]][[4]],
```

```
CostFunctionValues}]
```

```
);
```

SRD-PSO

```
SRDPSO[function_, Arguments_, DomainArgumentList_,SwarmPopulation_,
```

```
Iterations_, $\lambda$ _]:=
```

```
(
```

(*ЗАДАННЯ ФУНКЦІЇ ЦІНИ*)

```
func=function;
```

```
CostFunctionValues=List[];
```

(*ІНІЦІАЛІЗАЦІЯ МАСИВУ ЗНАЧЕНЬ ФУНКЦІЇ ЦІНИ*)

(*ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ*)

(*ІНІЦІАЛІЗАЦІЯ РОЮ*)

```
Swarm=List[];
```

Do[

(*ПОЧАТКОВІ ПОЛОЖЕННЯ ЧАСТИНОК*)

ParticlePosition=Table[RandomReal[DomainArgumentList[[i]], {i,
1,Length[Arguments], 1}];

(*ПОЧАТКОВІ ШВИДКОСТІ ЧАСТИНОК*)

ParticleVelocity=Table[0, {i, 1,Length[Arguments], 1}];

ParticleCost=Apply[func,ParticlePosition];

(*ВИЗНАЧЕННЯ НАЙКРАЩИХ ПОЛОЖЕНЬ ДЛЯ КОЖНОЇ
ЧАСТИНКИ

Найкращі положення спочатку відповідають початковим положенням*)

ParticleBest=ParticlePosition;

ParticleBestCost=ParticleCost;

(*k-та частинка*)

Particle=List[ParticlePosition, ParticleVelocity, ParticleCost, ParticleBest,
ParticleBestCost];

AppendTo[Swarm, Particle], SwarmPopulation];

(*знаходження найкращої частинки у рої*)

GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}];

BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]]];

(*заповнили значенням величини функції ціни масив*)

AppendTo[CostFunctionValues, GlobalBestCost];

(*ЦИКЛ РОЗРАХУНКУ*)

Monitor[

Do[

(*оновлення положень та швидкостей для кожної частинки*)

```

Do[
  r1=Table[RandomReal[{0.,1.}], Length[Arguments]];
  r2=Table[RandomReal[{0.,1.}], Length[Arguments]];

  w=Table[RandomReal[{0.0,1.}], Length[Arguments]];
  c1=Table[RandomReal[{0.7,2}], Length[Arguments]];
  c2=Table[RandomReal[{0.7,2}], Length[Arguments]];

  Swarm[[k]][[2]]=w*Swarm[[k]][[2]]+c1*r1*(Swarm[[k]][[4]]-
Swarm[[k]][[1]])+
  c2*r2*(Swarm[[BestParticleNumber]][[4]]-Swarm[[k]][[1]]);

  Swarm[[k]][[1]]=Swarm[[k]][[1]]+Swarm[[k]][[2]];

(*перевірка виходу за межі верхньої та нижньої границі*)
Do[

Piecewise[{{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[1]],Swarm[[k]][[1]][[j]]<DomainArgumentList[[j]][[1]]},
{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[2]],Swarm[[k]][[1]][[j]]>DomainArgumentList[[j]][[2]]}}]

, {j, 1, Length[Arguments]}];

(*оновлення значення функції в новому положенні частинки*)
Swarm[[k]][[3]]=Apply[func, Swarm[[k]][[1]]]

,{k, 1, SwarmPopulation, 1}];

(*оновлення PersonalBest*)
Do[If[Swarm[[k]][[3]]<Swarm[[k]][[5]], Swarm[[k]][[5]]=Swarm[[k]][[3]];

```

```
Swarm[[k]][[4]]=Swarm[[k]][[1]],{k, 1, SwarmPopulation, 1}];
```

(*оновлення GlobalBest*)

```
GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}];
```

```
BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}],GlobalBestCost][[1]][[1]];
```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost];
```

```
Do[
```

(*вибір випадкової частинки - не найкращої*)

```
RandomParticle=RandomChoice[Cases[Table[j, {j, 1, SwarmPopulation}],Except[BestParticleNumber]]];
```

```
Swarm[[RandomParticle]][[1]]=Table[RandomReal[DomainArgumentList[[i]], {i, 1,Length[Arguments], 1}];
```

```
Swarm[[RandomParticle]][[2]]=Table[0, {i, 1,Length[Arguments], 1}];
```

```
Swarm[[RandomParticle]][[3]]=Apply[func, Swarm[[RandomParticle]][[1]]];
```

```
Swarm[[RandomParticle]][[4]]=Swarm[[RandomParticle]][[1]];
```

```
Swarm[[RandomParticle]][[5]]=Swarm[[RandomParticle]][[3]]
```

```
,λ]
```

```
,{step, 2, Iterations, 1}]
```

(*візуалізація даних ІТЕРАЦІЯ-ЗНАЧЕННЯ ФУНКЦІЇ*)

```
,{Round[step/Iterations*100., 0.01] "% " ,GlobalBestCost}];
```

(*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2) значення аргументів при яких вона досягається; 3) дискретні значення функції ціни протягом процедури мінімізації*)

```
Return[{GlobalBestCost, Swarm[[BestParticleNumber]][[4]],
CostFunctionValues}]
);
```

VCT-PSO

```
VCTPSO[function_, Arguments_, DomainArgumentList_, SwarmPopulation_,
Iterations_, RC_] :=
```

```
(
(*ЗАДАННЯ ФУНКЦІЇ ЦІНИ*)
```

```
func=function;
```

```
CostFunctionValues=List[];
```

```
(*ІНІЦІАЛІЗАЦІЯ МАСИВУ ЗНАЧЕНЬ ФУНКЦІЇ ЦІНИ*)
```

```
(*ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ*)
```

```
w=0.72;c2=1.19; c1=c2;
```

```
(*ІНІЦІАЛІЗАЦІЯ РОЮ*)
```

```
Swarm=List[];
```

```
Do[
```

```
(*ПОЧАТКОВІ ПОЛОЖЕННЯ ЧАСТИНОК*)
```

```
ParticlePosition=Table[RandomReal[DomainArgumentList[[i]], {i,
1,Length[Arguments], 1}];
```

```
(*ПОЧАТКОВІ ШВИДКОСТІ ЧАСТИНОК*)
```

```
ParticleVelocity=Table[0, {i, 1,Length[Arguments], 1}];
```

```
ParticleCost=Apply[func,ParticlePosition];
```

```
(*ВИЗНАЧЕННЯ НАЙКРАЩИХ ПОЛОЖЕНЬ ДЛЯ КОЖНОЇ
ЧАСТИНКИ
```

```
Найкращі положення спочатку відповідають початковим положенням*)
```

```
ParticleBest=ParticlePosition;
```

```
ParticleBestCost=ParticleCost;
```

```
(*k-та частинка*)
```

```
Particle=List[ParticlePosition, ParticleVelocity, ParticleCost, ParticleBest,
ParticleBestCost];
```

```
AppendTo[Swarm, Particle], SwarmPopulation];
```

(*знаходження найкращої частинки у рої*)

```
GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];
```

```
BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]];
```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost];
```

(*ЦИКЛ РОЗРАХУНКУ*)

```
Monitor[
```

```
Do[
```

(*оновлення положень та швидкостей для кожної частинки*)

```
Do[
```

```
  r1=Table[RandomReal[{0.,1.}], Length[Arguments]];
```

```
  r2=Table[RandomReal[{0.,1.}], Length[Arguments]];
```

(*ця умова - перевірка, що даний номер ітерації є кратним RC. Якщо так, то виконується заміна номера частинки у когнітивній складовій на випадковий, якщо ні - то номер частинки не змінюється*)

```
If[Element[step/RC,Integers],
```

```
CognitiveNumber=RandomInteger[{1,SwarmPopulation}], CognitiveNumber=k];
```

```
Swarm[[k]][[2]]=w*Swarm[[k]][[2]]+c1*r1*(Swarm[[CognitiveNumber]][[4]]-
Swarm[[k]][[1]])+
```

```
  c2*r2*(Swarm[[BestParticleNumber]][[4]]-Swarm[[k]][[1]]);
```

```
Swarm[[k]][[1]]=Swarm[[k]][[1]]+Swarm[[k]][[2]];
```

(*перевірка виходу за межі верхньої та нижньої границі*)

Do[

Piecewise[{{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[1]],Swarm[[k]][[1]][[j]]<DomainArgumentList[[j]][[1]]},
 {Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[2]],Swarm[[k]][[1]][[j]]>DomainArgumentList[[j]][[2]]}}

, {j, 1, Length[Arguments]}];

(*оновлення значення функції в новому положенні частинки*)

Swarm[[k]][[3]]=Apply[func, Swarm[[k]][[1]]]

, {k, 1, SwarmPopulation, 1}];

(*оновлення PersonalBest*)

Do[If[Swarm[[k]][[3]]<Swarm[[k]][[5]], Swarm[[k]][[5]]=Swarm[[k]][[3]];

Swarm[[k]][[4]]=Swarm[[k]][[1]], {k, 1, SwarmPopulation, 1}];

(*оновлення GlobalBest*)

GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}];

BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}],GlobalBestCost][[1]][[1]];

(*заповнили значенням величини функції ціни масив*)

AppendTo[CostFunctionValues, GlobalBestCost]

, {step, 2, Iterations, 1}]

(*візуалізація даних ІТЕРАЦІЯ-ЗНАЧЕННЯ ФУНКЦІЇ*)

, {Round[step/Iterations*100., 0.01] "% " ,GlobalBestCost}];

(*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2)

значення аргументів при яких вона досягається; 3) дискретні значення функції ціни протягом процедури мінімізації*)

```
Return[{GlobalBestCost, Swarm[[BestParticleNumber]][[4]],
CostFunctionValues}]
);
```

PV-IA-PSO-1

```
PVIAPSO1[function_, Arguments_, DomainArgumentList_, SwarmPopulation_,
Iterations_, δMax_] :=
```

```
(
(*ЗАДАННЯ ФУНКЦІЇ ЦІНИ*)
func=function;
```

```
CostFunctionValues=List[];
```

```
(*ІНІЦІАЛІЗАЦІЯ МАСИВУ ЗНАЧЕНЬ ФУНКЦІЇ ЦІНИ*)
```

```
(*ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ*)
```

```
c2=1.19; c1=c2;
```

```
(*ІНІЦІАЛІЗАЦІЯ РОЮ*)
```

```
Swarm=List[];
```

```
Do[
```

```
(*ПОЧАТКОВІ ПОЛОЖЕННЯ ЧАСТИНОК*)
```

```
ParticlePosition=Table[RandomReal[DomainArgumentList[[i]]], {i,
1,Length[Arguments], 1}];
```

```
(*ПОЧАТКОВІ ШВИДКОСТІ ЧАСТИНОК*)
```

```
ParticleVelocity=Table[0, {i, 1,Length[Arguments], 1}];
```

```
ParticleCost=Apply[func,ParticlePosition];
```

```
(*ВИЗНАЧЕННЯ НАЙКРАЩИХ ПОЛОЖЕНЬ ДЛЯ КОЖНОЇ
ЧАСТИНКИ
```

Найкращі положення спочатку відповідають початковим положенням*)

```
ParticleBest=ParticlePosition;
```

```
ParticleBestCost=ParticleCost;
```

(*k-та частинка*)

```
Particle=List[ParticlePosition, ParticleVelocity, ParticleCost, ParticleBest, ParticleBestCost];
```

```
AppendTo[Swarm, Particle], SwarmPopulation];
```

(*знаходження найкращої частинки у рої*)

```
GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];
```

```
BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}],GlobalBestCost][[1]][[1]]];
```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost];
```

(*заповнення масиву відстаней від частинок до глобального мінімуму, який знайшов увесь рій*)

```
DistList=Table[\[Sqrt](Total[(Swarm[[BestParticleNumber]][[4]]- Swarm[[k]][[1]]]^2)), {k, 1, SwarmPopulation, 1}];
```

```
 $\delta\text{Min}=10^{-20}$ ;
```

(*ЦИКЛ РОЗРАХУНКУ*)

```
Monitor[
```

```
Do[
```

(*оновлення положень та швидкостей для кожної частинки*)

```
Do[
```

```
w0=RandomReal[{0.5,1.}]; w=w0*(1-DistList[[k]]/Max[DistList]);
```

```
r1=Table[RandomReal[{0.,1.}], Length[Arguments]];
```

```
r2=Table[RandomReal[{0.,1.}], Length[Arguments]];
```

$$\delta = (\delta_{\text{Max}} - \delta_{\text{Min}}) * (\text{Iterations} - \text{step}) / \text{Iterations} + \delta_{\text{Min}};$$

(*показує нормальний/Гаусівський розподіл з середнім μ та стандартним відхиленням σ *)

$$\rho = \text{RandomVariate}[\text{NormalDistribution}[0, \delta]];$$

$$\begin{aligned} \text{Swarm}[[k]][[2]] &= w * \text{Swarm}[[k]][[2]] + c1 * r1 * (\text{Swarm}[[k]][[4]] - \\ &\text{Swarm}[[k]][[1]]) + \\ &c2 * r2 * (\text{Swarm}[[\text{BestParticleNumber}]][[4]] - \text{Swarm}[[k]][[1]]); \end{aligned}$$

$$\text{Swarm}[[k]][[1]] = \text{Swarm}[[k]][[1]] * (1 - \rho) + \text{Swarm}[[k]][[2]];$$

(*перевірка виходу за межі верхньої та нижньої границі*)

Do[

Piecewise[{ {Swarm[[k]][[2]][[j]] = 0; Swarm[[k]][[1]][[j]] = DomainArgumentList[[j]][[1]], Swarm[[k]][[1]][[j]] < DomainArgumentList[[j]][[1]]},
 {Swarm[[k]][[2]][[j]] = 0; Swarm[[k]][[1]][[j]] = DomainArgumentList[[j]][[2]], Swarm[[k]][[1]][[j]] > DomainArgumentList[[j]][[2]]} }]
 , {j, 1, Length[Arguments]}];

(*оновлення значення функції в новому положенні частинки*)

$$\text{Swarm}[[k]][[3]] = \text{Apply}[\text{func}, \text{Swarm}[[k]][[1]]]$$

, {k, 1, SwarmPopulation, 1}];

(*оновлення PersonalBest*)

$$\text{Do}[\text{If}[\text{Swarm}[[k]][[3]] < \text{Swarm}[[k]][[5]], \text{Swarm}[[k]][[5]] = \text{Swarm}[[k]][[3]]];$$

Swarm[[k]][[4]] = Swarm[[k]][[1]], {k, 1, SwarmPopulation, 1}];

(*оновлення GlobalBest*)

$$\text{GlobalBestCost} = \text{Min}[\text{Table}[\text{Swarm}[[k]][[5]], \{k, 1, \text{SwarmPopulation}, 1\}];$$

```
BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]];
```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost];
```

(*заповнення масиву відстаней від частинок до глобального мінімуму, який знайшов увесь рій*)

```
DistList=Table[\[Sqrt](Total[(Swarm[[BestParticleNumber]][[4]]-
Swarm[[k]][[1]])^2]), {k, 1, SwarmPopulation, 1}];
,{step, 2, Iterations, 1}]
```

(*візуалізація даних ІТЕРАЦІЯ-ЗНАЧЕННЯ ФУНКЦІЇ*)

```
,{Round[step/Iterations*100., 0.01] "% " ,GlobalBestCost}];
```

(*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2) значення аргументів при яких вона досягається; 3) дискретні значення функції ціни протягом процедури мінімізації*)

```
Return[{GlobalBestCost, Swarm[[BestParticleNumber]][[4]],
CostFunctionValues}]
);
```

PV-IA-PSO-2

```
PVIAPSO2[function_, Arguments_, DomainArgumentList_,SwarmPopulation_,
Iterations_, Base_]:=
```

(

(*ЗАДАННЯ ФУНКЦІЇ ЦІНИ*)

```
func=function;
```

```
CostFunctionValues=List[];
```

(*ІНІЦІАЛІЗАЦІЯ МАСИВУ ЗНАЧЕНЬ ФУНКЦІЇ ЦІНИ*)

(*ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ*)

```
c2=1.19; c1=c2;
```

(*ІНІЦІАЛІЗАЦІЯ РОЮ*)

```
Swarm=List[];
```

```
Do[
```

(*ПОЧАТКОВІ ПОЛОЖЕННЯ ЧАСТИНОК*)

```
ParticlePosition=Table[RandomReal[DomainArgumentList[[i]], {i,
1,Length[Arguments], 1}];
```

(*ПОЧАТКОВІ ШВИДКОСТІ ЧАСТИНОК*)

```
ParticleVelocity=Table[0, {i, 1,Length[Arguments], 1}];
```

```
ParticleCost=Apply[func,ParticlePosition];
```

(*ВИЗНАЧЕННЯ НАЙКРАЩИХ ПОЛОЖЕНЬ ДЛЯ КОЖНОЇ
ЧАСТИНКИ

Найкращі положення спочатку відповідають початковим положенням*)

```
ParticleBest=ParticlePosition;
```

```
ParticleBestCost=ParticleCost;
```

(*k-та частинка*)

```
Particle=List[ParticlePosition, ParticleVelocity, ParticleCost, ParticleBest,
ParticleBestCost];
```

```
AppendTo[Swarm, Particle], SwarmPopulation];
```

(*знаходження найкращої частинки у рої*)

```
GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];
```

```
BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]];
```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost];
```

(*заповнення масиву відстаней від частинок до глобального мінімуму, який
знайшов увесь рій*)

```
DistList=Table[\[Sqrt](Total[(Swarm[[BestParticleNumber]][[4]]-
Swarm[[k]][[1]])^2)], {k, 1, SwarmPopulation, 1}];
```

(*ЦИКЛ РОЗРАХУНКУ*)

Monitor[

Do[

(*оновлення положень та швидкостей для кожної частинки*)

Do[

w0=RandomReal[{0.5,1.}]; w=w0*(1-DistList[[k]]/Max[DistList]);

r1=Table[RandomReal[{0.,1.}], Length[Arguments]];

r2=Table[RandomReal[{0.,1.}], Length[Arguments]];

δ =Base^(Swarm[[k]][[1]][[-1]]);

(*показує нормальний/Гаусівський розподіл з середнім μ та стандартним відхиленням σ *)

ρ =RandomVariate[NormalDistribution[0, δ]];

Swarm[[k]][[2]]=w*Swarm[[k]][[2]]+c1*r1*(Swarm[[k]][[4]]-Swarm[[k]][[1]])+

c2*r2*(Swarm[[BestParticleNumber]][[4]]-Swarm[[k]][[1]]);

Swarm[[k]][[1]]=Swarm[[k]][[1]]*(1- ρ)+Swarm[[k]][[2]];

(*перевірка виходу за межі верхньої та нижньої границі*)

Do[

Piecewise[{{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[1]],Swarm[[k]][[1]][[j]]<DomainArgumentList[[j]][[1]]},

{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[2]],Swarm[[k]][[1]][[j]]>DomainArgumentList[[j]][[2]]} }

, {j, 1, Length[Arguments]}];

(*оновлення значення функції в новому положенні частинки*)

```
Swarm[[k]][[3]]=Apply[func, Swarm[[k]][[1]]]
,{k, 1, SwarmPopulation, 1}];
```

(*оновлення PersonalBest*)

```
Do[If[Swarm[[k]][[3]]<Swarm[[k]][[5]], Swarm[[k]][[5]]=Swarm[[k]][[3]];
Swarm[[k]][[4]]=Swarm[[k]][[1]]], {k, 1, SwarmPopulation, 1}];
```

(*оновлення GlobalBest*)

```
GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];
BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]];
```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost];
```

(*заповнення масиву відстаней від частинок до глобального мінімуму, який знайшов увесь рій*)

```
DistList=Table[\[Sqrt](Total[(Swarm[[BestParticleNumber]][[4]]-
Swarm[[k]][[1]]^2)), {k, 1, SwarmPopulation, 1}];
,{step, 2, Iterations, 1}];
```

(*візуалізація даних ІТЕРАЦІЯ-ЗНАЧЕННЯ ФУНКЦІЇ*)

```
,{Round[step/Iterations*100., 0.01] "% " ,GlobalBestCost}];
```

(*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2) значення аргументів при яких вона досягається; 3) дискретні значення функції ціни протягом процедури мінімізації*)

```
Return[{GlobalBestCost, Swarm[[BestParticleNumber]][[4]],
CostFunctionValues}]
);
```

IA-PSO

```

IAPSO[function_, Arguments_, DomainArgumentList_, SwarmPopulation_,
Iterations_] :=
(
(*ЗАДАННЯ ФУНКЦІЇ ЦІНИ*)
func=function;

CostFunctionValues=List[];
(*ІНІЦІАЛІЗАЦІЯ МАСИВУ ЗНАЧЕНЬ ФУНКЦІЇ ЦІНИ*)
(*ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ*)
c2=1.19; c1=c2;

(*ІНІЦІАЛІЗАЦІЯ РОЮ*)
Swarm=List[];
Do[
(*ПОЧАТКОВІ ПОЛОЖЕННЯ ЧАСТИНОК*)
ParticlePosition=Table[RandomReal[DomainArgumentList[[i]]], {i,
1,Length[Arguments], 1}];
(*ПОЧАТКОВІ ШВИДКОСТІ ЧАСТИНОК*)
ParticleVelocity=Table[0, {i, 1,Length[Arguments], 1}];
ParticleCost=Apply[func,ParticlePosition];
(*ВИЗНАЧЕННЯ НАЙКРАЩИХ ПОЛОЖЕНЬ ДЛЯ КОЖНОЇ
ЧАСТИНКИ
Найкращі положення спочатку відповідають початковим положенням*)
ParticleBest=ParticlePosition;
ParticleBestCost=ParticleCost;
(*k-та частинка*)
Particle=List[ParticlePosition, ParticleVelocity, ParticleCost, ParticleBest,
ParticleBestCost];
AppendTo[Swarm, Particle], SwarmPopulation];

```

(*знаходження найкращої частинки у рої*)

```
GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];

```

```
BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]];

```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost];

```

(*заповнення масиву відстаней від частинок до глобального мінімуму, який знайшов увесь рій*)

```
DistList=Table[\[Sqrt](Total[(Swarm[[BestParticleNumber]][[4]]-
Swarm[[k]][[1]])^2)], {k, 1, SwarmPopulation, 1}];

```

(*ЦИКЛ РОЗРАХУНКУ*)

```
Monitor[

```

```
Do[

```

```
    (*оновлення положень та швидкостей для кожної частинки*)

```

```
Do[

```

```
    w0=RandomReal[{0.5,1.}]; w=w0*(1-DistList[[k]]/Max[DistList]);

```

```
    c2=1.19; c1=c2;

```

```
    r1=Table[RandomReal[{0.,1.}], Length[Arguments]];

```

```
    r2=Table[RandomReal[{0.,1.}], Length[Arguments]];

```

```
    ρ=Table[RandomReal[{-0.25,0.25}], Length[Arguments]];

```

```
    Swarm[[k]][[2]]=w*Swarm[[k]][[2]]+c1*r1*(Swarm[[k]][[4]]-
Swarm[[k]][[1]])+

```

```
    c2*r2*(Swarm[[BestParticleNumber]][[4]]-Swarm[[k]][[1]]);

```

Swarm[[k]][[1]]=Swarm[[k]][[1]]*(1-ρ)+Swarm[[k]][[2]];

(*перевірка виходу за межі верхньої та нижньої границі*)

Do[

Piecewise[{{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[1]],Swarm[[k]][[1]][[j]]<DomainArgumentList[[j]][[1]]},
 {Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[2]],Swarm[[k]][[1]][[j]]>DomainArgumentList[[j]][[2]]}}]
 , {j, 1, Length[Arguments]}];

(*оновлення значення функції в новому положенні частинки*)

Swarm[[k]][[3]]=Apply[func, Swarm[[k]][[1]]]
 ,{k, 1, SwarmPopulation, 1}];

(*оновлення PersonalBest*)

Do[If[Swarm[[k]][[3]]<Swarm[[k]][[5]], Swarm[[k]][[5]]=Swarm[[k]][[3]];
 Swarm[[k]][[4]]=Swarm[[k]][[1]],{k, 1, SwarmPopulation, 1}];

(*оновлення GlobalBest*)

GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}];
 BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,
 1}],GlobalBestCost][[1]][[1]];

(*заповнили значенням величини функції ціни масив*)

AppendTo[CostFunctionValues, GlobalBestCost];

(*заповнення масиву відстаней від частинок до глобального мінімуму,
 який знайшов увесь рій*)

DistList=Table[\[Sqrt](Total[(Swarm[[BestParticleNumber]][[4]]-

```
Swarm[[k]][[1]]^2)), {k, 1, SwarmPopulation, 1}];
,{step, 2, Iterations, 1}]
```

```
(*Візуалізація даних ІТЕРАЦІЯ-ЗНАЧЕННЯ ФУНКЦІЇ*)
,{Round[step/Iterations*100., 0.01] "% " ,GlobalBestCost}];
```

(*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2) значення аргументів при яких вона досягається; 3) дискретні значення функції ціни протягом процедури мінімізації*)

```
Return[{GlobalBestCost, Swarm[[BestParticleNumber]][[4]],
CostFunctionValues}]
);
```

LDW-PSO

```
LDWPSO[function_, Arguments_, DomainArgumentList_,SwarmPopulation_,
Iterations_] :=
```

```
(
(*ЗАДАННЯ ФУНКЦІЇ ЦІНИ*)
func=function;
```

```
CostFunctionValues=List[];
```

```
(*ІНІЦІАЛІЗАЦІЯ МАСИВУ ЗНАЧЕНЬ ФУНКЦІЇ ЦІНИ*)
```

```
(*ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ*)
```

```
c2=1.19; c1=c2;
```

```
(*ІНІЦІАЛІЗАЦІЯ РОЮ*)
```

```
Swarm=List[];
```

```
Do[
```

```
(*ПОЧАТКОВІ ПОЛОЖЕННЯ ЧАСТИНОК*)
```

```
ParticlePosition=Table[RandomReal[DomainArgumentList[[i]]], {i,
1,Length[Arguments], 1}];
```

(*ПОЧАТКОВІ ШВИДКОСТІ ЧАСТИНОК*)

```
ParticleVelocity=Table[0, {i, 1,Length[Arguments], 1}];
```

```
ParticleCost=Apply[func,ParticlePosition];
```

(*ВИЗНАЧЕННЯ НАЙКРАЩИХ ПОЛОЖЕНЬ ДЛЯ КОЖНОЇ
ЧАСТИНКИ

Найкращі положення спочатку відповідають початковим положенням*)

```
ParticleBest=ParticlePosition;
```

```
ParticleBestCost=ParticleCost;
```

(*k-та частинка*)

```
Particle=List[ParticlePosition, ParticleVelocity, ParticleCost, ParticleBest,
ParticleBestCost];
```

```
AppendTo[Swarm, Particle], SwarmPopulation];
```

(*знаходження найкращої частинки у рої*)

```
GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];
```

```
BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]]];
```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost];
```

(*ЦИКЛ РОЗРАХУНКУ*)

```
Monitor[
```

```
Do[
```

```
WMax=0.9; WMin=0.4;
```

```
w=(WMax-WMin)*(Iterations-step)/Iterations+WMin;
```

(*оновлення положень та швидкостей для кожної частинки*)

```

Do[
  r1=Table[RandomReal[{0.,1.}], Length[Arguments]];
  r2=Table[RandomReal[{0.,1.}], Length[Arguments]];

  Swarm[[k]][[2]]=w*Swarm[[k]][[2]]+c1*r1*(Swarm[[k]][[4]]-
Swarm[[k]][[1]])+
  c2*r2*(Swarm[[BestParticleNumber]][[4]]-Swarm[[k]][[1]]);

  Swarm[[k]][[1]]=Swarm[[k]][[1]]+Swarm[[k]][[2]];

(*перевірка виходу за межі верхньої та нижньої границі*)
Do[

Piecewise[{{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[1]],Swarm[[k]][[1]][[j]]<DomainArgumentList[[j]][[1]]},
{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[2]],Swarm[[k]][[1]][[j]]>DomainArgumentList[[j]][[2]]}}]

, {j, 1, Length[Arguments]}];

(*оновлення значення функції в новому положенні частинки*)
Swarm[[k]][[3]]=Apply[func, Swarm[[k]][[1]]]
,{k, 1, SwarmPopulation, 1};

(*оновлення PersonalBest*)
Do[If[Swarm[[k]][[3]]<Swarm[[k]][[5]], Swarm[[k]][[5]]=Swarm[[k]][[3]];
Swarm[[k]][[4]]=Swarm[[k]][[1]],{k, 1, SwarmPopulation, 1}];

(*оновлення GlobalBest*)
GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];

```

```
BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]]];
```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost]
```

```
,{step, 2, Iterations, 1}]
```

(*візуалізація даних ІТЕРАЦІЯ-ЗНАЧЕННЯ ФУНКЦІЇ*)

```
,{Round[step/Iterations*100., 0.01] "% " ,GlobalBestCost}];
```

(*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2) значення аргументів при яких вона досягається; 3) дискретні значення функції ціни протягом процедури мінімізації*)

```
Return[{GlobalBestCost, Swarm[[BestParticleNumber]][[4]],
CostFunctionValues}]
);
```

Rot-Ring-PSO

```
RotRingPSO[function_, Arguments_, DomainArgumentList_,SwarmPopulation_,
Iterations_] :=
```

```
(
(*ЗАДАННЯ ФУНКЦІЇ ЦІНИ*)
func=function;
```

```
CostFunctionValues=List[];
```

(*ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ*)

```
w=0.72;c2=1.19; c1=c2;
```

(*ІНІЦІАЛІЗАЦІЯ РОЮ*)

```

Swarm=List[];
Do[
  (*ПОЧАТКОВІ ПОЛОЖЕННЯ ЧАСТИНОК*)
  ParticlePosition=Table[RandomReal[DomainArgumentList[[i]]], {i,
1,Length[Arguments], 1}];

  (*ПОЧАТКОВІ ШВИДКОСТІ ЧАСТИНОК*)
  ParticleVelocity=Table[0, {i, 1,Length[Arguments], 1}];
  ParticleCost=Apply[func,ParticlePosition];
  (*ВИЗНАЧЕННЯ НАЙКРАЩИХ ПОЛОЖЕНЬ ДЛЯ КОЖНОЇ
ЧАСТИНКИ
Найкращі положення спочатку відповідають початковим положенням*)
  ParticleBest=ParticlePosition;
  ParticleBestCost=ParticleCost;
  (*k-та частинка*)
  Particle=List[ParticlePosition, ParticleVelocity, ParticleCost, ParticleBest,
ParticleBestCost];
  AppendTo[Swarm, Particle], SwarmPopulation];

  (*знаходження найкращої частинки у рої*)
  GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];
  BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]];

  (*заповнили значенням величини функції ціни масив*)
  AppendTo[CostFunctionValues, GlobalBestCost];

  Speed=1;
  (*ЦИКЛ РОЗРАХУНКУ*)
  Monitor[

```

```

z=0;
Do[
  (*оновлення положень та швидкостей для кожної частинки*)
Do[
  UpperNeighbour=k+1+z*Speed;
  LowerNeighbour=k-1+z*Speed;

If[LowerNeighbour<1, LowerNeighbour=-1];

  If[Mod[UpperNeighbour,SwarmPopulation]>0,
UpperNeighbour=Mod[UpperNeighbour,SwarmPopulation],
UpperNeighbour=SwarmPopulation];
  If[Mod[LowerNeighbour,SwarmPopulation]>0,
LowerNeighbour=Mod[LowerNeighbour,SwarmPopulation],
LowerNeighbour=SwarmPopulation];

LocalBestCost=Min[Swarm[[k]][[5]],Swarm[[LowerNeighbour]][[5]],Swarm[[Upper
Neighbour]][[5]]];
(*знаходження номера локально найкращої частинки*)
LocalBestParticleNumber=Position[Transpose[Swarm][[5]],LocalBestCost][[1]][[
1]];

r1=Table[RandomReal[{0.,1.}], Length[Arguments]];
r2=Table[RandomReal[{0.,1.}], Length[Arguments]];

Swarm[[k]][[2]]=w*Swarm[[k]][[2]]+c1*r1*(Swarm[[k]][[4]]-
Swarm[[k]][[1]])+
c2*r2*(Swarm[[LocalBestParticleNumber]][[4]]-Swarm[[k]][[1]]);

```

```
Swarm[[k]][[1]]=Swarm[[k]][[1]]+Swarm[[k]][[2]];
```

(*перевірка виходу за межі верхньої та нижньої границі*)

```
Do[
```

```
Piecewise[{{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[1]],Swarm[[k]][[1]][[j]]<DomainArgumentList[[j]][[1]]},
```

```
{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[2]],Swarm[[k]][[1]][[j]]>DomainArgumentList[[j]][[2]]} }
```

```
, {j, 1, Length[Arguments]}];
```

(*оновлення значення функції в новому положенні частинки*)

```
Swarm[[k]][[3]]=Apply[func, Swarm[[k]][[1]]]
```

```
,{k, 1, SwarmPopulation, 1}];
```

```
z++;
```

(*оновлення PersonalBest*)

```
Do[If[Swarm[[k]][[3]]<Swarm[[k]][[5]], Swarm[[k]][[5]]=Swarm[[k]][[3]];
```

```
Swarm[[k]][[4]]=Swarm[[k]][[1]],{k, 1, SwarmPopulation, 1}];
```

(*оновлення GlobalBest*)

```
GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];
```

```
BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}],GlobalBestCost][[1]][[1]];
```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost]
```

```
,{step, 2, Iterations, 1}]
```

(*візуалізація даних ІТЕРАЦІЯ-ЗНАЧЕННЯ ФУНКЦІЇ*)

```
,{Round[step/Iterations*100., 0.01] "% " ,GlobalBestCost}];
(*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2)
значення аргументів при яких вона досягається; 3) дискретні значення
функції ціни протягом процедури мінімізації*)
Return[{GlobalBestCost, Swarm[[BestParticleNumber]][[4]],
CostFunctionValues}]
);
```

Ring-PSO

```
PSORing[function_, Arguments_, DomainArgumentList_, SwarmPopulation_,
Iterations_] :=
```

```
(
(*ЗАДАННЯ ФУНКЦІЇ ЦІНИ*)
func=function;

CostFunctionValues=List[];
(*ІНІЦІАЛІЗАЦІЯ МАСИВУ ЗНАЧЕНЬ ФУНКЦІЇ ЦІНИ*)
(*ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ*)
w=0.72;c2=1.19; c1=c2;

(*ІНІЦІАЛІЗАЦІЯ РОЮ*)
Swarm=List[];
Do[
(*ПОЧАТКОВІ ПОЛОЖЕННЯ ЧАСТИНОК*)
ParticlePosition=Table[RandomReal[DomainArgumentList[[i]], {i,
1,Length[Arguments], 1}];
(*ПОЧАТКОВІ ШВИДКОСТІ ЧАСТИНОК*)
ParticleVelocity=Table[0, {i, 1,Length[Arguments], 1}];
ParticleCost=Apply[func,ParticlePosition];
(*ВИЗНАЧЕННЯ НАЙКРАЩИХ ПОЛОЖЕНЬ ДЛЯ КОЖНОЇ
```

ЧАСТИНКИ

Найкращі положення спочатку відповідають початковим положенням*)

ParticleBest=ParticlePosition;

ParticleBestCost=ParticleCost;

(*k-та частинка*)

Particle=List[ParticlePosition, ParticleVelocity, ParticleCost, ParticleBest, ParticleBestCost];

AppendTo[Swarm, Particle], SwarmPopulation];

(*знаходження найкращої частинки у рої*)

GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];

BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}],GlobalBestCost][[1]][[1]]];

(*заповнили значенням величини функції ціни масив*)

AppendTo[CostFunctionValues, GlobalBestCost];

(*ЦИКЛ РОЗРАХУНКУ*)

Monitor[

Do[

(*оновлення положень та швидкостей для кожної частинки*)

Do[

(*знаходження найкращої частинки у локальному рої*)

(*перша частинка*)

If[k==1, LocalBestCost=Min[Swarm[[-1]][[5]],Swarm[[k+1]][[5]]];

(*остання частинка*)

If[k==SwarmPopulation, LocalBestCost=Min[Swarm[[k-1]][[5]],Swarm[[1]][[5]]];

If[SwarmPopulation>k>1, LocalBestCost=Min[Swarm[[k-1]][[5]],Swarm[[k+1]][[5]]];

(*знаходження номера локально найкращої частинки*)

```
LocalBestParticleNumber=Position[Transpose[Swarm][[5],LocalBestCost][[1]][[1]]];
```

```
r1=Table[RandomReal[{0.,1.}], Length[Arguments]];
```

```
r2=Table[RandomReal[{0.,1.}], Length[Arguments]];
```

```
Swarm[[k]][[2]]=w*Swarm[[k]][[2]]+c1*r1*(Swarm[[k]][[4]]-Swarm[[k]][[1]])+c2*r2*(Swarm[[LocalBestParticleNumber]][[4]]-Swarm[[k]][[1]]);
```

```
Swarm[[k]][[1]]=Swarm[[k]][[1]]+Swarm[[k]][[2]];
```

(*перевірка виходу за межі верхньої та нижньої границі*)

```
Do[
```

```
Piecewise[{{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[1]],Swarm[[k]][[1]][[j]]<DomainArgumentList[[j]][[1]]},
```

```
{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[2]],Swarm[[k]][[1]][[j]]>DomainArgumentList[[j]][[2]]} }
```

```
, {j, 1, Length[Arguments]}];
```

(*оновлення значення функції в новому положенні частинки*)

```
Swarm[[k]][[3]]=Apply[func, Swarm[[k]][[1]]]
```

```
, {k, 1, SwarmPopulation, 1}];
```

(*оновлення PersonalBest*)

```
Do[If[Swarm[[k]][[3]]<Swarm[[k]][[5]], Swarm[[k]][[5]]=Swarm[[k]][[3]];
```

```
Swarm[[k]][[4]]=Swarm[[k]][[1]], {k, 1, SwarmPopulation, 1}];
```

(*оновлення GlobalBest*)

GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];

BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}],GlobalBestCost][[1]][[1]];

(*заповнили значенням величини функції ціни масив*)

AppendTo[CostFunctionValues, GlobalBestCost]

,{step, 2, Iterations, 1}]

(*візуалізація даних ІТЕРАЦІЯ-ЗНАЧЕННЯ ФУНКЦІЇ*)

,{Round[step/Iterations*100., 0.01] "% " ,GlobalBestCost}];

(*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2) значення аргументів при яких вона досягається; 3) дискретні значення функції ціни протягом процедури мінімізації*)

Return[{GlobalBestCost, Swarm[[BestParticleNumber]][[4]],

CostFunctionValues}]

);

LG-PSO-Ring

LGPSORing[function_, Arguments_, DomainArgumentList_,SwarmPopulation_, Iterations_, a _]:=

(

(*ЗАДАННЯ ФУНКЦІЇ ЦІНИ*)

func=function;

CostFunctionValues=List[];

(*ІНІЦІАЛІЗАЦІЯ МАСИВУ ЗНАЧЕНЬ ФУНКЦІЇ ЦІНИ*)

(*ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ*)

w=0.72;c2=1.19; c1=c2;

(*ІНІЦІАЛІЗАЦІЯ РОЮ*)

```
Swarm=List[];
```

```
Do[
```

(*ПОЧАТКОВІ ПОЛОЖЕННЯ ЧАСТИНОК*)

```
ParticlePosition=Table[RandomReal[DomainArgumentList[[i]], {i,
1,Length[Arguments], 1}];
```

(*ПОЧАТКОВІ ШВИДКОСТІ ЧАСТИНОК*)

```
ParticleVelocity=Table[0, {i, 1,Length[Arguments], 1}];
```

```
ParticleCost=Apply[func,ParticlePosition];
```

(*ВИЗНАЧЕННЯ НАЙКРАЩИХ ПОЛОЖЕНЬ ДЛЯ КОЖНОЇ
ЧАСТИНКИ

Найкращі положення спочатку відповідають початковим положенням*)

```
ParticleBest=ParticlePosition;
```

```
ParticleBestCost=ParticleCost;
```

(*k-та частинка*)

```
Particle=List[ParticlePosition, ParticleVelocity, ParticleCost, ParticleBest,
ParticleBestCost];
```

```
AppendTo[Swarm, Particle], SwarmPopulation];
```

(*знаходження найкращої частинки у рої*)

```
GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];
```

```
BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]]];
```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost];
```

(*ЦИКЛ РОЗРАХУНКУ*)

```
Monitor[
```

```

Do[
  (*оновлення положень та швидкостей для кожної частинки*)
Do[
  (*знаходження найкращої частинки у локальному рої*)
(*перша частинка*)
If[k==1 , LocalBestCost=Min[Swarm[[1]][[5]],Swarm[[k]][[5]],Swarm[[k+1]][[5]]];
  (*остання частинка*)
If[k==SwarmPopulation, LocalBestCost=Min[Swarm[[k-1]][[5]],Swarm[[k]][[5]],Swarm[[1]][[5]]];
  If[SwarmPopulation>k>1, LocalBestCost=Min[Swarm[[k-1]][[5]],Swarm[[k]][[5]],Swarm[[k+1]][[5]]];

(*знаходження номера локально найкращої частинки*)
LocalBestParticleNumber=Position[Transpose[Swarm][[5]],LocalBestCost][[1]][[1]];

  ggl={};
  RandNum=RandomReal[];
  Do[If[(step/Iterations)^a>RandNum, AppendTo[ggl,
Swarm[[BestParticleNumber]][[4]][[j]], AppendTo[ggl,
Swarm[[LocalBestParticleNumber]][[4]][[j]]], {j, 1, Length[Arguments]}];

  r1=Table[RandomReal[{0.,1.}], Length[Arguments]];
  r2=Table[RandomReal[{0.,1.}], Length[Arguments]];

  Swarm[[k]][[2]]=w*Swarm[[k]][[2]]+c1*r1*(Swarm[[k]][[4]]-
Swarm[[k]][[1]])+
  c2*r2*(ggl-Swarm[[k]][[1]]);

```

```
Swarm[[k]][[1]]=Swarm[[k]][[1]]+Swarm[[k]][[2]];
```

(*перевірка виходу за межі верхньої та нижньої границі*)

```
Do[
```

```
Piecewise[{{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[1]],Swarm[[k]][[1]][[j]]<DomainArgumentList[[j]][[1]]},
```

```
{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[2]],Swarm[[k]][[1]][[j]]>DomainArgumentList[[j]][[2]]} }
```

```
, {j, 1, Length[Arguments]}];
```

(*оновлення значення функції в новому положенні частинки*)

```
Swarm[[k]][[3]]=Apply[func, Swarm[[k]][[1]]]
```

```
, {k, 1, SwarmPopulation, 1}];
```

(*оновлення PersonalBest*)

```
Do[If[Swarm[[k]][[3]]<Swarm[[k]][[5]], Swarm[[k]][[5]]=Swarm[[k]][[3]];
```

```
Swarm[[k]][[4]]=Swarm[[k]][[1]], {k, 1, SwarmPopulation, 1}];
```

(*оновлення GlobalBest*)

```
GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}];
```

```
BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}],GlobalBestCost][[1]][[1]];
```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost]
```

```
, {step, 2, Iterations, 1} ]
```

(*візуалізація даних ІТЕРАЦІЯ-ЗНАЧЕННЯ ФУНКЦІЇ*)

```
,{Round[step/Iterations*100., 0.01] "% " ,GlobalBestCost}}];
```

(*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2) значення аргументів при яких вона досягається; 3) дискретні значення функції ціни протягом процедури мінімізації*)

```
Return[{GlobalBestCost, Swarm[[BestParticleNumber]][[4]],
CostFunctionValues}]
);
```

ME-D-PSO

```
BreakPSO[function_, Arguments_, DomainArgumentList_,SwarmPopulation_,
Iterations_, AcceptableRate_, CostFuncVal_, InitialArgumentValues_]:=
```

```
(
(*ЗАДАННЯ ФУНКЦІЇ ЦІНИ*)
func=function;
CostFunctionValues=CostFuncVal;
```

```
(*ІНІЦІАЛІЗАЦІЯ МАСИВУ ЗНАЧЕНЬ ФУНКЦІЇ ЦІНИ*)
(*ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ*)
```

```
(*ІНІЦІАЛІЗАЦІЯ РОЮ*)
Swarm=List[];
```

```
Do[
(*ПОЧАТКОВІ ПОЛОЖЕННЯ ЧАСТИНОК*)
ParticlePosition=Table[RandomReal[DomainArgumentList[[i]]], {i,
1,Length[Arguments], 1}];
```

```
(*ПОЧАТКОВІ ШВИДКОСТІ ЧАСТИНОК*)
ParticleVelocity=Table[0, {i, 1,Length[Arguments], 1}];
ParticleCost=Apply[func,ParticlePosition];
```

```
(*ВИЗНАЧЕННЯ НАЙКРАЩИХ ПОЛОЖЕНЬ ДЛЯ КОЖНОЇ
```

ЧАСТИНКИ

Найкращі положення спочатку відповідають початковим положенням*)

```
ParticleBestPosition=ParticlePosition;
```

```
ParticleBestCost=ParticleCost;
```

(*k-та частинка*)

```
Particle=List[ParticlePosition, ParticleVelocity, ParticleCost,  
ParticleBestPosition, ParticleBestCost];
```

```
AppendTo[Swarm, Particle], SwarmPopulation];
```

(*знаходження найкращої частинки у рої*)

```
GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];
```

```
BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,  
1}],GlobalBestCost][[1]][[1]]];
```

(* якщо вже була переініціалізація, то присвоїти найкращій частинці
GlobalBest *)

```
If[Length[InitialArgumentValues]>0,Swarm[[BestParticleNumber]][[1]]=InitialAr  
gumentValues;
```

```
Swarm[[BestParticleNumber]][[3]]=Apply[func,  
InitialArgumentValues];Swarm[[BestParticleNumber]][[4]]=InitialArgumentValue  
s;
```

```
Swarm[[BestParticleNumber]][[5]]=Swarm[[BestParticleNumber]][[3]],(*заповни  
ли значенням величини функції ціни масив*)
```

```
AppendTo[CostFunctionValues, GlobalBestCost]]];
```

(*ЦИКЛ РОЗРАХУНКУ*)

```
Monitor[step=Length[CostFunctionValues];
```

```
While[step<=Iterations-1.,
```

(*оновлення положень та швидкостей для кожної частинки*)

Do[

r1=Table[RandomReal[{0.,1.}], Length[Arguments]];

r2=Table[RandomReal[{0.,1.}], Length[Arguments]];

w=Table[RandomReal[{0.0,1.}], Length[Arguments]];

c1=Table[RandomReal[{0,2.}], Length[Arguments]];

c2=Table[RandomReal[{0,2.}], Length[Arguments]];

Swarm[[k]][[2]]=w*Swarm[[k]][[2]]+c1*r1*(Swarm[[k]][[4]]-
Swarm[[k]][[1]])+

c2*r2*(Swarm[[BestParticleNumber]][[4]]-Swarm[[k]][[1]]);

Swarm[[k]][[1]]=Swarm[[k]][[1]]+Swarm[[k]][[2]];

(*перевірка виходу за межі верхньої та нижньої границі*)

Do[

Piecewise[{{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[1]],Swarm[[k]][[1]][[j]]<DomainArgumentList[[j]][[1]]},

{Swarm[[k]][[2]][[j]]=0;Swarm[[k]][[1]][[j]]=DomainArgumentList[[j]][[2]],Swarm[[k]][[1]][[j]]>DomainArgumentList[[j]][[2]]} }

, {j, 1, Length[Arguments]}];

(*оновлення значення функції в новому положенні частинки*)

Swarm[[k]][[3]]=Apply[func, Swarm[[k]][[1]]]

, {k, 1, SwarmPopulation, 1}];

(*оновлення PersonalBest*)

```
Do[If[Swarm[[k]][[3]]<Swarm[[k]][[5]], Swarm[[k]][[5]]=Swarm[[k]][[3]];
Swarm[[k]][[4]]=Swarm[[k]][[1]], {k, 1, SwarmPopulation, 1}];
```

(*оновлення GlobalBest*)

```
GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}];
```

```
BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]];
```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost];
```

(*умова закінчення однієї епохи*)

```
If[CostFunctionValues[[-1]]!=CostFunctionValues[[-2]],
```

```
If[Abs[(CostFunctionValues[[-1]]-CostFunctionValues[[-
2]])/CostFunctionValues[[-1]]]<=AcceptableRate, Break[]];
```

```
step++]
```

(*візуалізація даних ІТЕРАЦІЯ-ЗНАЧЕННЯ ФУНКЦІЇ*)

```
,{Round[step/Iterations*100., 0.01] "% " ,GlobalBestCost}];
```

(*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2) значення аргументів при яких вона досягається; 3) дискретні значення функції ціни протягом процедури мінімізації*)

```
Return[{GlobalBestCost, Swarm[[BestParticleNumber]][[4]],
CostFunctionValues}]
```

```
);
```

```
listGlobPos=List[];
```

```
listCoFuVal=List[];
```

```
MEPSODiversity[function_, Arguments_,
```

```
DomainArgumentList_,SwarmPopulation_, Iterations_, AcceptableRate_,
```

```
CostFuncVal_, InitialArgumentValues_] :=
```

```

(
  step=Length[CostFuncVal];
  IterationNumber=Iterations;
  ArgumentsList=Arguments;

  While[step<=IterationNumber-1.,
    Br=BreakPSO[function,
  ArgumentsList,DomainArgumentList,SwarmPopulation,
  IterationNumber,AcceptableRate,listCoFuVal,listGlobPos];
    listGlobPos=Br[[2]];
    listCoFuVal=Br[[3]];
  ];
  listGlobPos=List[];
  listCoFuVal=List[];
  Return[{GlobalBestCost,Swarm[[BestParticleNumber]][[4]],
  CostFunctionValues}]
);

```

M-PSO-Hybrid

```

MPSOHybrid[function_, Arguments_, DomainArgumentList_,SwarmPopulation_,
  Iterations_, IterationsPerAlgorithm_]:=

```

(*ЗАДАННЯ ФУНКЦІЇ ЦІНИ*)

```
func=function;
```

```
CostFunctionValues=List[];
```

(*ІНІЦІАЛІЗАЦІЯ МАСИВУ ЗНАЧЕНЬ ФУНКЦІЇ ЦІНИ*)

(*ІНІЦІАЛІЗАЦІЯ ПОЧАТКОВИХ ДАНИХ*)

```
c2=1.19; c1=c2;
```

MachineEpsilon=10⁻¹⁰⁰;

(*ІНІЦІАЛІЗАЦІЯ РОЮ*)

Swarm=List[];

Do[

(*ПОЧАТКОВІ ПОЛОЖЕННЯ ЧАСТИНОК*)

ParticlePosition=Table[RandomReal[DomainArgumentList[[i]], {i,
1,Length[Arguments], 1}];

(*ПОЧАТКОВІ ШВИДКОСТІ ЧАСТИНОК*)

ParticleVelocity=Table[0, {i, 1,Length[Arguments], 1}];

ParticleCost=Apply[func,ParticlePosition];

(*ВИЗНАЧЕННЯ НАЙКРАЩИХ ПОЛОЖЕНЬ ДЛЯ КОЖНОЇ
ЧАСТИНКИ

Найкращі положення спочатку відповідають початковим положенням*)

ParticleBest=ParticlePosition;

ParticleBestCost=ParticleCost;

(*k-та частинка*)

Particle=List[ParticlePosition, ParticleVelocity, ParticleCost, ParticleBest,
ParticleBestCost];

AppendTo[Swarm, Particle], SwarmPopulation];

(*знаходження найкращої частинки у рої*)

GlobalBestCost=Min[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation, 1}]];

BestParticleNumber=Position[Table[Swarm[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]]];

(*заповнили значенням величини функції ціни масив*)

AppendTo[CostFunctionValues, GlobalBestCost];

(*ЦИКЛ РОЗРАХУНКУ*)

IncRatio=0.05;

(*Перший прогон алгоритмів - для визначення їхніх ефективностей*)

```
algorithm1=LDWPSOInto[func, Arguments,
DomainArgumentList,SwarmPopulation, Iterations, 1, IterationsPerAlgorithm,
Swarm];
```

```
algorithm1Efficiency=(algorithm1[[3]][[1]]-algorithm1[[3]][[-
1]])/(algorithm1[[3]][[1]]+MachineEpsilon);
```

```
algorithm2=IANDPSOInto[func, Arguments,
DomainArgumentList,SwarmPopulation, Iterations, IterationsPerAlgorithm+1,
IterationsPerAlgorithm, algorithm1[[1]]];
```

```
algorithm2Efficiency=(algorithm2[[3]][[-IterationsPerAlgorithm+1]]-
algorithm2[[3]][[-1]])/(algorithm2[[3]][[-
IterationsPerAlgorithm+1]]+MachineEpsilon);
```

```
algorithm3=PSORingRotInto[func, Arguments,
DomainArgumentList,SwarmPopulation, Iterations, 2*IterationsPerAlgorithm+1,
IterationsPerAlgorithm, algorithm2[[1]]];
```

```
algorithm3Efficiency=(algorithm3[[3]][[-IterationsPerAlgorithm+1]]-
algorithm3[[3]][[-1]])/(algorithm3[[3]][[-
IterationsPerAlgorithm+1]]+MachineEpsilon);
```

```
algorithm4=RandCognPSOInto[func, Arguments,
DomainArgumentList,SwarmPopulation, Iterations, 3*IterationsPerAlgorithm+1,
IterationsPerAlgorithm, algorithm3[[1]]];
```

```
algorithm4Efficiency=(algorithm4[[3]][[-IterationsPerAlgorithm+1]]-
algorithm4[[3]][[-1]])/(algorithm4[[3]][[-
IterationsPerAlgorithm+1]]+MachineEpsilon);
```

```
swarm=algorithm4[[1]];
```

```
(*Цикл використання алгоритмів*)
```

```
Do[
```

```
If[algorithm1Efficiency==0 && algorithm2Efficiency==0 &&
```

```
algorithm3Efficiency==0 && algorithm4Efficiency==0,
```

```
algorithmIndex=RandomChoice[{1,2,3,4}],
```

```
algorithmIndex=RandomChoice[{algorithm1Efficiency,algorithm2Efficiency,algo-  
rithm3Efficiency,algorithm4Efficiency}->{1,2,3,4}]]];
```

```
If[algorithmIndex== 1, algorithm=LDWPSOInto[func, Arguments,
```

```
DomainArgumentList,SwarmPopulation, Iterations, i*IterationsPerAlgorithm+1,
```

```
IterationsPerAlgorithm, swarm];
```

```
algorithm1Efficiency=(algorithm[[3]][[-IterationsPerAlgorithm+1]]-  
algorithm[[3]][[-1]])/(algorithm[[3]][[-  
IterationsPerAlgorithm+1]]+MachineEpsilon)];
```

```
If[algorithmIndex== 2, algorithm=IANDPSOInto[func, Arguments,
```

```
DomainArgumentList,SwarmPopulation, Iterations,i*IterationsPerAlgorithm+1,
```

```
IterationsPerAlgorithm, swarm];
```

```
algorithm2Efficiency=(algorithm[[3]][[-IterationsPerAlgorithm+1]]-  
algorithm[[3]][[-1]])/(algorithm[[3]][[-  
IterationsPerAlgorithm+1]]+MachineEpsilon)];
```

```
If[algorithmIndex== 3, algorithm=PSORingRotInto[func, Arguments,
```

```
DomainArgumentList,SwarmPopulation, Iterations,i*IterationsPerAlgorithm+1,
```

```
IterationsPerAlgorithm,swarm];
```

```
algorithm3Efficiency=(algorithm[[3]][[-IterationsPerAlgorithm+1]]-
```

```
algorithm[[3]][[-1]])/(algorithm[[3]][[-IterationsPerAlgorithm+1]]+MachineEpsilon)];
```

```
If[algorithmIndex== 4, algorithm=RandCognPSOInto[func, Arguments,
DomainArgumentList,SwarmPopulation, Iterations,i*IterationsPerAlgorithm+1,
IterationsPerAlgorithm,swarm];
```

```
algorithm4Efficiency=(algorithm[[3]][[-IterationsPerAlgorithm+1]]-
algorithm[[3]][[-1]])/(algorithm[[3]][[-IterationsPerAlgorithm+1]]+MachineEpsilon)];
```

```
swarm=algorithm[[1]]
```

```
,{i, 4, Iterations/IterationsPerAlgorithm-1}];
```

(*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2) значення аргументів при яких вона досягається; 3) дискретні значення функції ціни протягом процедури мінімізації*)

```
Return[{GlobalBestCost, swarm[[BestParticleNumber]][[4]],
CostFunctionValues}]
);
```

```
LDWPSOInto[function_, Arguments_, DomainArgumentList_,SwarmPopulation_,
Iterations_, InitialStep_, IterationsPerAlgorithm_, swarm1 _]:=
```

```
(
Swarm1=swarm1;
```

```
If[algorithm1Efficiency==0,
```

```
RandomParticle=RandomInteger[{ 1, SwarmPopulation}];
```

```
Swarm1 [[RandomParticle]][[1]]=Table[RandomReal[DomainArgumentList[[i]],
```

```
{i, 1, Length[Arguments], 1}]; Swarm1[[RandomParticle]][[2]]=Table[0, {i,
1, Length[Arguments], 1}]]];
```

```
If[algorithm1Efficiency==0,
```

```
    RandomParticle=RandomInteger[{ 1, SwarmPopulation}];
```

```
Swarm1[[RandomParticle]][[1]]=Table[RandomReal[DomainArgumentList[[i]]],
{i, 1, Length[Arguments], 1}]; Swarm1[[RandomParticle]][[2]]=Table[0, {i,
1, Length[Arguments], 1}]]];
```

(*ЦИКЛ РОЗРАХУНКУ*)

```
Monitor[
```

```
Do[
```

```
    WMax=0.9; WMin=0.4;
```

```
    w=(WMax-WMin)*(Iterations-step)/Iterations+WMin;
```

(*оновлення положень та швидкостей для кожної частинки*)

```
Do[
```

```
    r1=Table[RandomReal[{0.,1.}], Length[Arguments]];
    r2=Table[RandomReal[{0.,1.}], Length[Arguments]];

```

```
    Swarm1[[k]][[2]]=w*Swarm1[[k]][[2]]+c1*r1*(Swarm1[[k]][[4]]-
Swarm1[[k]][[1]])+

```

```
    c2*r2*(Swarm1[[BestParticleNumber]][[4]]-Swarm1[[k]][[1]]);
```

```
    Swarm1[[k]][[1]]=Swarm1[[k]][[1]]+Swarm1[[k]][[2]];

```

(*перевірка виходу за межі верхньої та нижньої границі*)

```
Do[
```

```
Piecewise[{{Swarm1[[k]][[2]][[j]]=0;Swarm1[[k]][[1]][[j]]=DomainArgumentList
[[j]][[1]],Swarm1[[k]][[1]][[j]]<DomainArgumentList[[j]][[1]]},
```

```
{Swarm1[[k]][[2]][[j]]=0;Swarm1[[k]][[1]][[j]]=DomainArgumentList[[j]][[2]],Swarm1[[k]][[1]][[j]]>DomainArgumentList[[j]][[2]]}}
```

```
, {j, 1, Length[Arguments]}];
```

(*оновлення значення функції в новому положенні частинки*)

```
Swarm1[[k]][[3]]=Apply[func, Swarm1[[k]][[1]]]
```

```
, {k, 1, SwarmPopulation, 1}];
```

(*оновлення PersonalBest*)

```
Do[If[Swarm1[[k]][[3]]<Swarm1[[k]][[5]],
```

```
Swarm1[[k]][[5]]=Swarm1[[k]][[3]]; Swarm1[[k]][[4]]=Swarm1[[k]][[1]], {k, 1, SwarmPopulation, 1}];
```

(*оновлення GlobalBest*)

```
GlobalBestCost=Min[Table[Swarm1[[k]][[5]], {k,1, SwarmPopulation, 1}]];
BestParticleNumber=Position[Table[Swarm1[[k]][[5]], {k,1, SwarmPopulation,
```

```
1}],GlobalBestCost][[1]][[1]];

```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost]
```

```
, {step, InitialStep, (InitialStep+IterationsPerAlgorithm)-1}]
```

(*візуалізація даних ІТЕРАЦІЯ-ЗНАЧЕННЯ ФУНКЦІЇ*)

```
, {Round[step/Iterations*100., 0.01] "% " ,GlobalBestCost}];
```

(*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2) значення аргументів при яких вона досягається; 3) дискретні значення функції ціни протягом процедури мінімізації*)

```
Return[{Swarm1, Swarm1[[BestParticleNumber]][[4]], CostFunctionValues}]
```

);

IANDPSOInto[function_, Arguments_,

DomainArgumentList_, SwarmPopulation_, Iterations_, InitialStep_,

IterationsPerAlgorithm_, swarm2_] :=

(

Swarm2=swarm2;

If[algorithm2Efficiency==0,

RandomParticle=RandomInteger[{1, SwarmPopulation}];

Swarm2[[RandomParticle]][[1]]=Table[RandomReal[DomainArgumentList[[i]]],

{i, 1, Length[Arguments], 1}]; Swarm2[[RandomParticle]][[2]]=Table[0, {i,

1, Length[Arguments], 1}];

If[algorithm2Efficiency==0,

RandomParticle=RandomInteger[{1, SwarmPopulation}];

Swarm2[[RandomParticle]][[1]]=Table[RandomReal[DomainArgumentList[[i]]],

{i, 1, Length[Arguments], 1}]; Swarm2[[RandomParticle]][[2]]=Table[0, {i,

1, Length[Arguments], 1}];

(*заповнення масиву відстаней від частинок до глобального мінімуму, який
знайшов увесь рій*)

DistList=Table[\[Sqrt](Total[(Swarm2[[BestParticleNumber]][[4]]-

Swarm2[[k]][[1]])^2]), {k, 1, SwarmPopulation, 1}];

 $\delta_{Min}=10^{-20}$; $\delta_{Max}=0.35$;

(*ЦИКЛ РОЗРАХУНКУ*)

Monitor[

```

Do[
  (*оновлення положень та швидкостей для кожної частинки*)
Do[
  w0=RandomReal[{0.5,1.}]; w=w0*(1-DistList[[k]]/Max[DistList]);
  c2=1.19; c1=c2;

  r1=Table[RandomReal[{0.,1.}], Length[Arguments]];
  r2=Table[RandomReal[{0.,1.}], Length[Arguments]];

   $\delta=(\delta_{\text{Max}}-\delta_{\text{Min}})*(Iterations-\text{step})/Iterations+\delta_{\text{Min}};$ 

   $\rho=\text{RandomVariate}[\text{NormalDistribution}[0, \delta]];$ 
  (*represents a normal (Gaussian) distribution with mean  $\mu$  and standard
  deviation  $\sigma$ *)

  Swarm2[[k]][[2]]=w*Swarm2[[k]][[2]]+c1*r1*(Swarm2[[k]][[4]]-
  Swarm2[[k]][[1]])+
  c2*r2*(Swarm2[[BestParticleNumber]][[4]]-Swarm2[[k]][[1]]);

  Swarm2[[k]][[1]]=Swarm2[[k]][[1]]*(1- $\rho$ )+Swarm2[[k]][[2]];

  (*перевірка виходу за межі верхньої та нижньої границі*)
Do[
  Piecewise[{{Swarm2[[k]][[2]][[j]]=0;Swarm2[[k]][[1]][[j]]=DomainArgumentList
  [[j]][[1]],Swarm2[[k]][[1]][[j]]<DomainArgumentList[[j]][[1]]},
  {Swarm2[[k]][[2]][[j]]=0;Swarm2[[k]][[1]][[j]]=DomainArgumentList[[j]][[2]],Sw
  arm2[[k]][[1]][[j]]>DomainArgumentList[[j]][[2]]} } ]
, {j, 1, Length[Arguments]}}];

```

(*оновлення значення функції в новому положенні частинки*)

```
Swarm2[[k]][[3]]=Apply[func, Swarm2[[k]][[1]]]
,{k, 1, SwarmPopulation, 1}];
```

(*оновлення PersonalBest*)

```
Do[If[Swarm2[[k]][[3]]<Swarm2[[k]][[5]],
Swarm2[[k]][[5]]=Swarm2[[k]][[3]]; Swarm2[[k]][[4]]=Swarm2[[k]][[1]],{k, 1,
SwarmPopulation, 1}];
```

(*оновлення GlobalBest*)

```
GlobalBestCost=Min[Table[Swarm2[[k]][[5]], {k,1, SwarmPopulation, 1}]];
BestParticleNumber=Position[Table[Swarm2[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]];
```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost];
```

(*заповнення масиву відстаней від частинок до глобального мінімуму, який знайшов увесь рій*)

```
DistList=Table[\[Sqrt](Total[(Swarm2[[BestParticleNumber]][[4]]-
Swarm2[[k]][[1]]^2)), {k, 1, SwarmPopulation, 1}]
,{step, InitialStep, (InitialStep+IterationsPerAlgorithm)-1}]
```

(*візуалізація даних ІТЕРАЦІЯ-ЗНАЧЕННЯ ФУНКЦІЇ*)

```
,{Round[step/Iterations*100., 0.01] "% " ,GlobalBestCost}];
```

(*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2) значення аргументів при яких вона досягається; 3) дискретні значення функції ціни протягом процедури мінімізації*)

```
Return[{Swarm2, Swarm2[[BestParticleNumber]][[4]], CostFunctionValues}
];
```

PSORingRotInto[function_, Arguments_,

```

DomainArgumentList_,SwarmPopulation_, Iterations_, InitialStep_,
IterationsPerAlgorithm_, swarm3_]:=
(
  Swarm3=swarm3;
If[algorithm3Efficiency==0,
  RandomParticle=RandomInteger[{ 1, SwarmPopulation}];

Swarm3[[RandomParticle]][[1]]=Table[RandomReal[DomainArgumentList[[i]]],
{i, 1,Length[Arguments], 1}]; Swarm3[[RandomParticle]][[2]]=Table[0, {i,
1,Length[Arguments], 1}]];
If[algorithm3Efficiency==0,
  RandomParticle=RandomInteger[{ 1, SwarmPopulation}];

Swarm3[[RandomParticle]][[1]]=Table[RandomReal[DomainArgumentList[[i]]],
{i, 1,Length[Arguments], 1}]; Swarm3[[RandomParticle]][[2]]=Table[0, {i,
1,Length[Arguments], 1}]];

  Speed=1;
  (*ЦИКЛ РОЗРАХУНКУ*)
Monitor[
  z=0;
Do[
  (*оновлення положень та швидкостей для кожної частинки*)
Do[

  UpperNeibour=k+1+z*Speed;
  LowerNeibour=k-1+z*Speed;

If[LowerNeibour<1, LowerNeibour=-1];

```

```
If[Mod[UpperNeighbour,SwarmPopulation]>0,
UpperNeighbour=Mod[UpperNeighbour,SwarmPopulation],
UpperNeighbour=SwarmPopulation];
```

```
If[Mod[LowerNeighbour,SwarmPopulation]>0,
LowerNeighbour=Mod[LowerNeighbour,SwarmPopulation],
LowerNeighbour=SwarmPopulation];
```

```
LocalBestCost=Min[Swarm3[[k]][[5]],Swarm3[[LowerNeighbour]][[5]],Swarm3[[U
pperNeighbour]][[5]]];
```

(*знаходження номера локально найкращої частинки*)

```
LocalBestParticleNumber=Position[Transpose[Swarm3][[5]],LocalBestCost][[1]][
[1]];
```

```
r1=Table[RandomReal[{0.,1.}], Length[Arguments]];
```

```
r2=Table[RandomReal[{0.,1.}], Length[Arguments]];
```

```
Swarm3[[k]][[2]]=w*Swarm3[[k]][[2]]+c1*r1*(Swarm3[[k]][[4]]-
Swarm3[[k]][[1]])+
c2*r2*(Swarm3[[LocalBestParticleNumber]][[4]]-Swarm3[[k]][[1]]);
```

```
Swarm3[[k]][[1]]=Swarm3[[k]][[1]]+Swarm3[[k]][[2]];
```

(*перевірка виходу за межі верхньої та нижньої границі*)

```
Do[
```

```
Piecewise[{{Swarm3[[k]][[2]][[j]]=0;Swarm3[[k]][[1]][[j]]=DomainArgumentList
[[j]][[1]],Swarm3[[k]][[1]][[j]]<DomainArgumentList[[j]][[1]]},
{Swarm3[[k]][[2]][[j]]=0;Swarm3[[k]][[1]][[j]]=DomainArgumentList[[j]][[2]],Sw
arm3[[k]][[1]][[j]]>DomainArgumentList[[j]][[2]]} }]
```

```

, {j, 1, Length[Arguments]}}];
  (*оновлення значення функції в новому положенні частинки*)
Swarm3[[k]][[3]]=Apply[func, Swarm3[[k]][[1]]]
, {k, 1, SwarmPopulation, 1}];
  z++;

  (*оновлення PersonalBest*)
  Do[If[Swarm3[[k]][[3]]<Swarm3[[k]][[5]],
Swarm3[[k]][[5]]=Swarm3[[k]][[3]]; Swarm3[[k]][[4]]=Swarm3[[k]][[1]], {k, 1,
SwarmPopulation, 1}];

  (*оновлення GlobalBest*)
  GlobalBestCost=Min[Table[Swarm3[[k]][[5]], {k,1, SwarmPopulation, 1}]];
  BestParticleNumber=Position[Table[Swarm3[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]];
  (*заповнили значенням величини функції ціни масив*)
  AppendTo[CostFunctionValues, GlobalBestCost]

, {step, InitialStep, (InitialStep+IterationsPerAlgorithm)-1}]

  (*візуалізація даних ІТЕРАЦІЯ-ЗНАЧЕННЯ ФУНКЦІЇ*)
, {Round[step/Iterations*100., 0.01] "% " ,GlobalBestCost}];
  (*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2)
значення аргументів при яких вона досягається; 3) дискретні значення
функції ціни протягом процедури мінімізації*)
Return[{Swarm3, Swarm3[[BestParticleNumber]][[4]], CostFunctionValues}
];

RandCognPSOInto[function_, Arguments_,
DomainArgumentList_,SwarmPopulation_, Iterations_, InitialStep_,

```

```
IterationsPerAlgorithm_, swarm4_] :=
```

```
(
```

```
Swarm4=swarm4;
```

```
If[algorithm4Efficiency==0,
```

```
RandomParticle=RandomInteger[{1, SwarmPopulation}];
```

```
Swarm4[[RandomParticle]][[1]]=Table[RandomReal[DomainArgumentList[[i]]],
```

```
{i, 1, Length[Arguments], 1}]; Swarm4[[RandomParticle]][[2]]=Table[0, {i,
```

```
1, Length[Arguments], 1}];
```

```
If[algorithm4Efficiency==0,
```

```
RandomParticle=RandomInteger[{1, SwarmPopulation}];
```

```
Swarm4[[RandomParticle]][[1]]=Table[RandomReal[DomainArgumentList[[i]]],
```

```
{i, 1, Length[Arguments], 1}]; Swarm4[[RandomParticle]][[2]]=Table[0, {i,
```

```
1, Length[Arguments], 1}];
```

```
RC=5;
```

```
(*ЦИКЛ РОЗРАХУНКУ*)
```

```
Monitor[
```

```
Do[
```

```
(*оновлення положень та швидкостей для кожної частинки*)
```

```
Do[
```

```
r1=Table[RandomReal[{0.,1.}], Length[Arguments]];

```

```
r2=Table[RandomReal[{0.,1.}], Length[Arguments]];

```

```
(*ця умова - перевірка, що даний номер ітерації є кратним RC. Якщо так, то виконується заміна номера частинки у когнітивній складовій на випадковий, якщо ні - то номер частинки не змінюється*)
```

```
If[Element[step/RC, Integers],
```

```
CognitiveNumber=RandomInteger[{1,SwarmPopulation}], CognitiveNumber=k];
```

```
Swarm4[[k]][[2]]=w*Swarm4[[k]][[2]]+c1*r1*(Swarm4[[CognitiveNumber]][[4]]
-Swarm4[[k]][[1]])+
  c2*r2*(Swarm4[[BestParticleNumber]][[4]]-Swarm4[[k]][[1]]);
```

```
Swarm4[[k]][[1]]=Swarm4[[k]][[1]]+Swarm4[[k]][[2]];
```

(*перевірка виходу за межі верхньої та нижньої границі*)

```
Do[
```

```
Piecewise[{{Swarm4[[k]][[2]][[j]]=0;Swarm4[[k]][[1]][[j]]=DomainArgumentList
[[j]][[1]],Swarm4[[k]][[1]][[j]]<DomainArgumentList[[j]][[1]]},
{Swarm4[[k]][[2]][[j]]=0;Swarm4[[k]][[1]][[j]]=DomainArgumentList[[j]][[2]],Sw
arm4[[k]][[1]][[j]]>DomainArgumentList[[j]][[2]]} }]
```

```
, {j, 1, Length[Arguments]}];
```

(*оновлення значення функції в новому положенні частинки*)

```
Swarm4[[k]][[3]]=Apply[func, Swarm4[[k]][[1]]]
```

```
, {k, 1, SwarmPopulation, 1}];
```

(*оновлення PersonalBest*)

```
Do[If[Swarm4[[k]][[3]]<Swarm4[[k]][[5]],
```

```
Swarm4[[k]][[5]]=Swarm4[[k]][[3]]; Swarm4[[k]][[4]]=Swarm4[[k]][[1]], {k, 1,
SwarmPopulation, 1}];
```

(*оновлення GlobalBest*)

```
GlobalBestCost=Min[Table[Swarm4[[k]][[5]], {k,1, SwarmPopulation, 1}]];
```

```
BestParticleNumber=Position[Table[Swarm4[[k]][[5]], {k,1, SwarmPopulation,
1}],GlobalBestCost][[1]][[1]];
```

(*заповнили значенням величини функції ціни масив*)

```
AppendTo[CostFunctionValues, GlobalBestCost]
```

```
, {step, InitialStep, (InitialStep+IterationsPerAlgorithm)-1}]
```

(*візуалізація даних ІТЕРАЦІЯ-ЗНАЧЕННЯ ФУНКЦІЇ*)

```
, {Round[step/Iterations*100., 0.01] "% " , GlobalBestCost}];
```

(*КІНЦЕВИЙ РЕЗУЛЬТАТ - функція повертає: 1) значення функції ціни; 2) значення аргументів при яких вона досягається; 3) дискретні значення функції ціни протягом процедури мінімізації*)

```
Return[{Swarm4, Swarm4[[BestParticleNumber]][[4]], CostFunctionValues}]
```

```
);
```

Додаток В

Визначення оптимального значення ζ для алгоритму SR-D-PSO

Для того, щоб визначити оптимальне значення ζ для алгоритму SR-D-PSO було проведено чисельні експерименти. Для них розмірність тестових функцій дорівнювала 50 ($D=50$).

Для всіх експериментів популяція рою N включала 30 частинок.

Щоб отримати статистично достовірні результати, кожен експеримент проводили 50 разів із незалежними і випадковими умовами ініціалізації рою.

Незалежний фактор ζ змінювався у діапазоні від 1 до 5 з кроком рівним одиниці.

Для оцінки зібраних даних були побудовані діаграми Бокса-Віскера (рис. В.1). Аргументом цих діаграм був десятковий логарифм від отриманих даних чисельного експерименту (скорочено *NEO*). Це необхідно для того, щоб можна було порівняти дані, які відрізняються на порядки.

Для аналізу роботи SR-D-PSO при різних чисельних значеннях параметра ζ було обрано клас тестових синтетичних функцій: f_1, f_4, f_5, f_6 . Вони мають різні топологічні властивості, що і вплинуло на їх вибір.

Розглядаючи діаграми, які показані на рис. В.1, можна зробити висновок, що оптимальне значення ζ дорівнює 1. Дійсно, більші значення ζ погіршують ефективність роботи SR-D-PSO для всіх варіантів тестових цільових функцій f_1, f_4, f_5, f_6 .

Порівнюючи діаграми (рис. В.1), слід підкреслити значний вплив значення ζ на ефективність алгоритму SR-D-PSO при оптимізації тестової функції f_1 , яка, нагадаємо, є унімодальною сепарабельною функцією, і яка не досить складна для проведення оптимізації.

На решту тестових функцій (f_4, f_5, f_6) вплив значення ζ значно слабший.

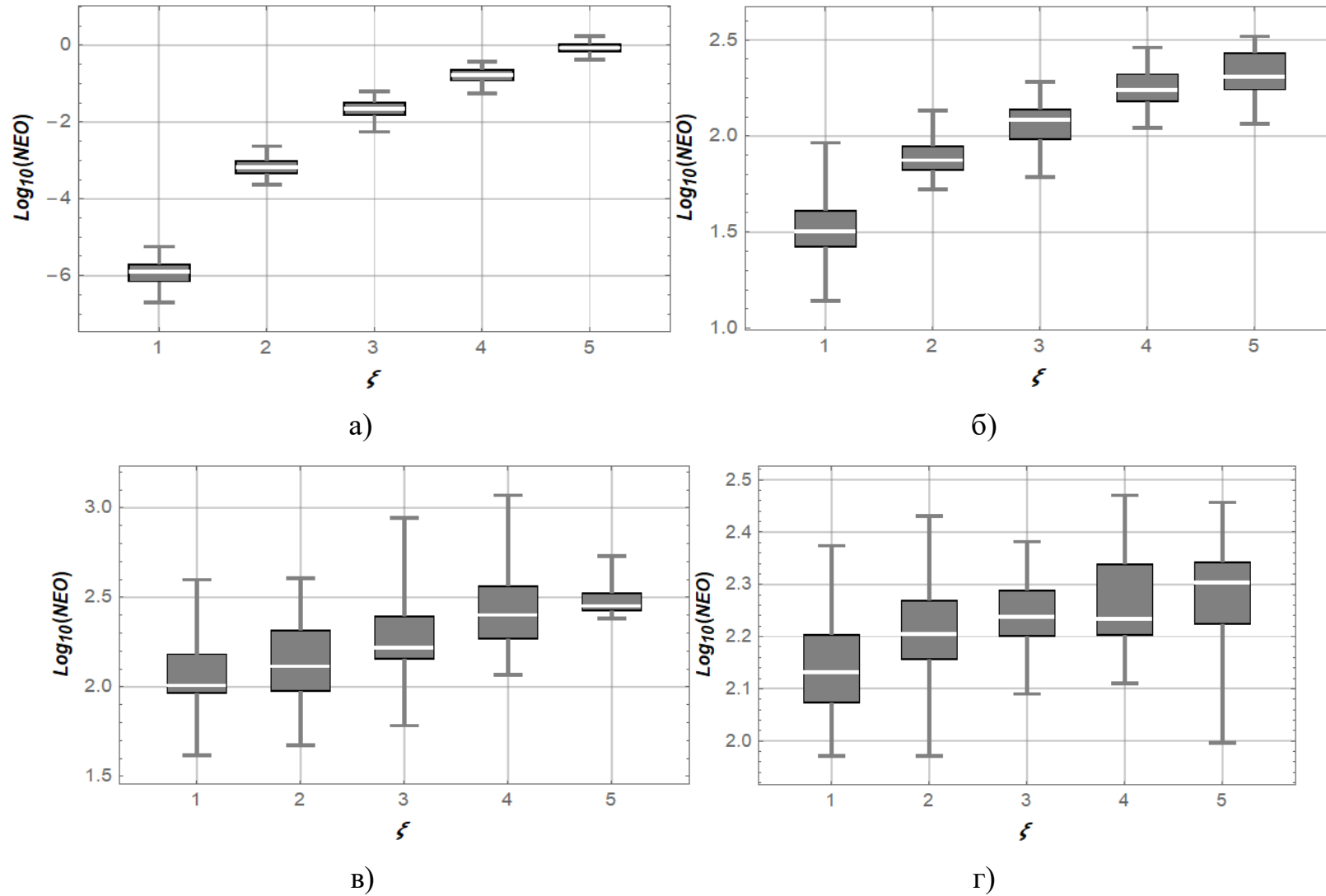


Рисунок В.1 – Діаграми Бокса-Віскера для представлення результатів чисельних експериментів (NEO) по визначенню ξ , які відповідають тестовим функціям: а) f_1 ; б) f_4 ; в) f_5 ; г) f_6

Продовження табл. Г.1

1	2	3	4	5	6	7	8	9	10
	Медіана	-1,00E+00	-1,00E+00	-1,00E+00	-1,00E+00	-1,00E+00	-1,00E+00	-1,00E+00	-1,00E+00
	СВ	2,65E-03	2,35E-15	6,81E-14	1,45E-10	1,03E-11	1,40E-01	1,11E-15	3,66E-07
<i>f4</i>	Max	3,44E+02	2,38E-01	4,35E+02	1,68E+02	8,81E+01	5,34E+03	2,79E-78	5,46E-01
	Min	6,03E+01	3,51E-06	1,85E+02	4,60E+01	2,50E+01	6,19E+01	2,25E-90	3,77E-80
	Середнє	1,82E+02	2,76E-02	2,78E+02	1,01E+02	5,66E+01	3,68E+02	8,41E-80	1,93E-02
	Медіана	1,79E+02	6,08E-03	2,65E+02	1,02E+02	5,57E+01	1,29E+02	1,48E-83	2,55E-51
	СВ	6,74E+01	4,83E-02	5,50E+01	2,65E+01	1,56E+01	9,92E+02	4,00E-79	9,95E-02
<i>f5</i>	Max	1,02E+04	5,10E+01	1,92E+02	1,72E+02	1,92E+02	6,82E+06	4,86E+01	4,73E+01
	Min	1,04E+02	4,58E+01	2,52E+01	3,32E+01	3,41E+01	1,09E+02	4,40E+01	3,99E+01
	Середнє	4,98E+02	4,74E+01	6,74E+01	7,79E+01	8,51E+01	1,37E+05	4,62E+01	4,45E+01
	Медіана	2,75E+02	4,71E+01	4,54E+01	7,85E+01	9,06E+01	2,72E+02	4,60E+01	4,46E+01
	СВ	1,42E+03	1,22E+00	3,80E+01	3,53E+01	4,12E+01	9,64E+05	1,39E+00	1,53E+00
<i>f6</i>	Max	3,05E+02	3,52E+02	4,51E+02	1,99E+02	2,85E+02	7,99E+02	4,13E-158	9,98E-01
	Min	1,37E+02	1,18E+02	2,09E+02	1,03E+02	5,59E+01	1,47E+02	2,08E-166	2,45E-157
	Середнє	2,11E+02	2,18E+02	3,78E+02	1,53E+02	1,86E+02	2,13E+02	1,06E-159	3,33E-02
	Медіана	2,11E+02	2,17E+02	3,87E+02	1,52E+02	2,16E+02	1,94E+02	7,80E-163	4,95E-109
	СВ	3,74E+01	4,71E+01	4,01E+01	2,39E+01	7,60E+01	1,19E+02	6,52E-159	1,82E-01

Продовження табл. Г.1

1	2	3	4	5	6	7	8	9	10
<i>f7</i>	Max	8,99E-01	3,43E-02	1,04E-01	7,40E-03	3,19E-02	3,37E+01	5,00E-162	1,86E-80
	Min	1,68E-02	3,00E-60	1,85E-12	1,35E-07	8,14E-11	9,87E-01	2,30E-169	1,39E-159
	Середнє	3,22E-01	1,05E-02	1,11E-02	4,58E-04	3,46E-03	2,94E+00	1,59E-163	6,18E-82
	Медіана	2,53E-01	7,54E-03	7,39E-03	5,79E-07	9,09E-10	1,05E+00	3,16E-165	3,87E-112
	СВ	2,37E-01	1,11E-02	1,85E-02	1,62E-03	8,05E-03	2,06E+04	7,09E-163	3,39E-81
<i>f8</i>	Max	1,59E+01	4,41E+01	1,30E+00	6,86E+00	2,56E-03	1,07E+02	2,57E-78	5,12E-44
	Min	3,56E-01	2,00E+00	1,03E-05	7,01E-03	3,35E-06	7,14E-01	1,12E-81	3,16E-79
	Середнє	4,92E+00	2,34E+01	9,96E-02	2,10E+00	2,81E-04	6,11E+00	2,37E-79	1,71E-45
	Медіана	4,14E+00	1,40E+01	1,14E-03	1,44E+00	2,77E-05	3,55E+00	2,61E-80	3,46E-64
	СВ	3,61E+00	9,82E+00	2,72E-01	1,85E+00	6,18E-04	1,47E+01	5,68E-79	9,34E-45
<i>f9</i>	Max	1,56E+02	3,26E+02	1,30E+02	1,87E+02	2,41E+02	3,73E+02	2,43E+02	2,05E+02
	Min	7,08E+01	2,75E+02	6,65E+01	1,22E+02	1,04E+02	8,04E+01	1,48E+02	1,02E+02
	Середнє	1,15E+02	2,97E+02	9,46E+01	1,57E+02	2,11E+02	1,33E+02	1,94E+02	1,49E+02
	Медіана	1,17E+02	2,97E+02	9,19E+01	1,56E+02	2,17E+02	1,08E+02	1,95E+02	1,45E+02
	СВ	1,64E+01	1,60E+04	1,47E+01	4,41E+04	2,60E+01	7,95E+01	2,41E+01	2,62E+01
<i>f10</i>	Max	4,41E+04	2,72E+00	3,22E+00	1,73E-02	1,47E+00	2,06E+01	3,55E-15	3,55E-15
	Min	2,23E-01	1,42E-14	2,90E-06	3,89E-04	6,67E-06	3,62E+00	0,00E+00	0,00E+00

Продовження табл. Г.1

1	2	3	4	5	6	7	8	9	10
	Середнє	4,83E+00	1,55E-01	1,03E+00	1,41E-03	2,58E-01	4,97E+00	1,42E-15	3,19E-15
	Медіана	4,30E+00	2,13E-14	1,16E+00	8,58E-04	4,40E-05	4,34E+00	1,41E-15	3,55E-15
	СВ	1,95E+00	6,19E-01	8,40E-01	2,40E-03	4,73E-01	3,25E+00	1,76E-15	1,08E-15
<i>f</i> 11	Max	6,37E-01	6,63E-16	3,34E-01	1,12E-02	8,06E-02	1,78E+00	6,63E-16	6,63E-16
	Min	3,25E-01	6,63E-16	3,38E-02	1,08E-03	2,75E-04	2,51E-01	6,63E-16	6,63E-16
	Середнє	4,55E-01	6,63E-16	1,40E-01	2,69E-03	2,40E-02	4,42E-01	6,63E-16	6,63E-16
	Медіана	4,61E-01	6,63E-16	1,36E-01	2,04E-03	1,43E-02	3,87E-01	6,63E-16	6,63E-16
	СВ	7,70E-02	0,00E+00	6,63E-02	1,98E-03	2,38E-02	2,81E-01	0,00E+00	0,00E+00
<i>f</i> 12	Max	1,69E+01	1,60E-34	9,39E-02	2,73E-04	3,68E-05	9,20E+00	1,14E-77	4,75E-44
	Min	1,24E+00	1,38E-36	1,78E-06	1,05E-04	4,20E-07	3,40E+00	7,10E-81	7,60E-77
	Середнє	7,02E+00	1,94E-35	2,36E-03	1,72E-04	6,09E-06	4,69E+04	8,28E-79	1,58E-45
	Медіана	6,57E+00	1,07E-35	5,42E-05	1,77E-04	3,28E-06	6,19E+00	1,63E-79	2,41E-63
	СВ	4,27E+00	2,66E-35	1,33E-02	3,68E-05	7,01E-06	1,28E+00	2,12E-78	8,67E-45
<i>f</i> 13	Max	8,06E+05	1,27E+04	1,45E+02	1,35E+02	5,13E-01	3,16E+11	1,08E+04	3,76E+03
	Min	1,14E+03	2,70E+03	5,27E-05	3,50E-01	2,41E-04	3,98E+03	6,68E+02	2,32E-08
	Середнє	4,58E+04	6,18E+03	3,89E+00	2,20E+01	3,91E-02	1,22E+10	5,25E+03	4,43E+02
	Медіана	1,02E+04	5,93E+03	5,21E-02	1,40E+01	1,03E-02	1,38E+04	5,01E+03	5,05E+01

Продовження табл. Г.1

1	2	3	4	5	6	7	8	9	10
	СВ	1,21E+05	1,99E+03	2,05E+01	2,68E+01	8,51E-02	6,05E+10	2,56E+03	9,99E+02
<i>f</i> 14	Max	4,40E+04	7,00E-01	8,00E-01	1,60E+00	1,30E+00	3,74E+01	9,99E-02	9,99E-02
	Min	1,90E+00	4,00E-01	5,00E-01	8,00E-01	5,00E-01	4,39E-01	9,99E-02	9,99E-02
	Середнє	3,00E+00	5,32E-01	5,90E-01	1,14E+00	7,82E-01	4,83E+00	9,99E-02	9,99E-02
	Медіана	1,99E+00	5,00E-01	6,00E-01	1,21E+00	7,00E-01	2,10E+00	9,99E-02	9,99E-02
	СВ	8,36E-01	6,53E-02	7,48E-02	1,57E-01	1,87E-01	9,35E+00	1,12E-10	2,11E-10
<i>f</i> 15	Max	1,43E-19	9,94E-11	4,89E-13	1,99E-18	1,67E-19	8,29E-05	2,24E-17	1,26E-19
	Min	4,29E-20	1,77E-16	6,68E-20	3,39E-20	1,21E-19	3,14E-20	1,58E-20	1,75E-20
	Середнє	8,15E-20	2,57E-12	1,04E-14	1,23E-19	1,55E-19	3,95E-06	6,56E-19	7,38E-20
	Медіана	7,83E-20	6,15E-14	1,76E-19	6,84E-20	1,56E-19	4,46E-20	8,50E-20	6,72E-20
	СВ	2,49E-20	1,41E-11	6,92E-14	2,78E-19	8,48E-21	1,54E-05	3,52E-18	3,32E-20
<i>f</i> 16	Max	-2,96E+02	-1,61E+02	-1,42E+02	-2,86E+02	-2,08E+02	-8,56E+01	-2,71E+02	-2,77E+02
	Min	-4,42E+02	-2,50E+02	-1,89E+02	-3,88E+02	-2,52E+02	-4,41E+02	-4,20E+02	-4,28E+02
	Середнє	-3,59E+02	-2,04E+02	-1,61E+02	-3,41E+02	-2,28E+02	-3,68E+02	-3,67E+02	-3,48E+02
	Медіана	-3,58E+02	-2,01E+02	-1,61E+02	-3,43E+02	-2,27E+02	-3,76E+02	-3,67E+02	-3,47E+02
	СВ	3,22E+01	2,37E+01	1,22E+01	4,39E+04	1,09E+01	6,36E+01	3,17E+01	3,46E+01

Продовження табл. Г.1

1	2	3	4	5	6	7	8	9	10
<i>f</i> 17	Max	6,12E-03	-5,00E+00	-5,00E+00	-5,00E+00	-5,00E+00	1,39E+01	-5,00E+00	-5,00E+00
	Min	-4,95E+00	-5,00E+00	-5,00E+00	-5,00E+00	-5,00E+00	-4,64E+00	-5,00E+00	-5,00E+00
	Середнє	-4,68E+00	-5,00E+00	-5,00E+00	-5,00E+00	-5,00E+00	-3,39E+00	-5,00E+00	-5,00E+00
	Медіана	-4,81E+00	-5,00E+00	-5,00E+00	-5,00E+00	-5,00E+00	-4,40E+00	-5,00E+00	-5,00E+00
	СВ	6,89E-01	1,82E-14	3,27E-06	3,23E-05	8,20E-06	4,17E+00	3,22E-14	1,43E-13
<i>f</i> 18	Max	7,96E-01	1,17E+00	1,06E+00	7,41E-01	3,91E-01	9,98E-01	1,24E+00	9,04E-01
	Min	3,69E-01	5,51E-01	5,32E-01	4,58E-01	1,70E-01	4,35E-01	6,76E-01	4,38E-01
	Середнє	5,87E-01	9,05E-01	7,44E-01	6,17E-01	3,01E-01	7,38E-01	9,22E-01	6,58E-01
	Медіана	5,64E-01	8,97E-01	7,33E-01	6,19E-01	3,02E-01	7,34E-01	9,29E-01	6,76E-01
	СВ	1,17E-01	1,32E-01	1,37E-01	6,83E-02	5,60E-02	1,35E-01	1,11E-01	1,33E-01
<i>f</i> 19	Max	-3,18E+03	-2,45E+03	-3,21E+03	-3,27E+03	-3,24E+03	1,57E+03	-2,66E+03	-2,64E+03
	Min	-3,60E+03	-3,04E+03	-3,72E+03	-3,58E+03	-3,69E+03	-3,67E+03	-3,30E+03	-3,52E+03
	Середнє	-3,39E+03	-2,73E+03	-3,43E+03	-3,44E+03	-3,52E+03	-3,26E+03	-3,08E+03	-3,28E+03
	Медіана	-3,41E+03	-2,74E+03	-3,42E+03	-3,44E+03	-3,52E+03	-3,43E+03	-3,11E+03	-3,32E+03
	СВ	8,76E+01	1,55E+02	1,02E+02	6,66E+01	9,20E+01	9,06E+02	1,46E+02	1,96E+02
<i>f</i> 20	Max	-2,25E+01	-1,34E+01	-1,43E+01	-2,94E+01	-1,90E+01	-9,22E+00	-1,55E+01	-2,26E+01
	Min	-3,55E+01	-1,80E+01	-1,75E+01	-3,97E+01	-2,25E+01	-3,40E+01	-2,28E+01	-3,87E+01

Продовження табл. Г.1

1	2	3	4	5	6	7	8	9	10
	Середнє	-3,07E+01	-1,55E+01	-1,55E+01	-3,30E+01	-2,05E+01	-2,94E+01	-1,92E+01	-3,02E+01
	Медіана	-3,09E+01	-1,54E+01	-1,53E+01	-3,26E+01	-2,04E+01	-2,97E+01	-1,89E+01	-3,05E+01
	СВ	2,30E+00	8,07E-01	8,83E-01	2,36E+00	7,43E-01	3,39E+00	2,09E+00	3,59E+00

* СВ – тут і далі цей показник означає стандартне відхилення.

Таблиця Г.2 – Статистичні показники при оптимізації тестових функцій за пізньої зупинки алгоритмів ($J=10000$)

Тестова функція	Показник	Алгоритми							
		LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-IA-PSO	PSO-Hybrid
1	2	3	4	5	6	7	8	9	10
f_5	Max	1,39E+02	4,87E+01	1,52E+02	8,63E+01	1,52E+02	6,38E+02	4,86E+01	4,69E+01
	Min	2,37E+00	4,50E+01	1,91E+01	1,47E-03	8,93E-02	3,30E+01	4,29E+01	2,82E+01
	Середнє	6,51E+01	4,69E+01	5,83E+01	3,93E+01	4,22E+01	1,28E+05	4,53E+01	3,60E+01
	Медіана	8,12E+01	4,70E+01	3,99E+01	3,66E+01	2,88E+01	4,83E+01	4,50E+01	3,51E+01
	СВ	3,09E+01	1,20E+00	3,08E+01	1,92E+01	3,85E+01	9,03E+05	1,69E+00	5,73E+00
f_9	Max	1,16E+02	3,06E+02	1,18E+02	1,79E+02	1,61E+02	3,56E+02	2,50E+02	2,13E+02
	Min	6,53E+01	2,59E+02	6,88E+01	9,14E+01	7,38E+01	3,80E+01	1,43E+02	1,05E+02
	Середнє	8,83E+01	2,82E+02	9,49E+01	1,46E+02	1,13E+02	8,20E+01	1,86E+02	1,53E+02

Продовження табл. Г.2

1	2	3	4	5	6	7	8	9	10
	Медіана	8,76E+01	2,83E+02	9,26E+01	1,49E+02	1,09E+02	7,37E+01	1,86E+02	1,51E+02
	СВ	2,30E+04	1,08E+01	1,46E+01	1,68E+01	2,19E+01	5,73E+01	2,49E+01	2,63E+01
<i>f</i> 13	Мах	2,98E-01	1,21E+04	1,50E-11	1,26E-10	1,91E-11	3,80E+11	9,16E+03	9,73E-11
	Мін	8,20E-09	2,15E+03	9,09E-13	3,18E-12	9,09E-12	7,54E-03	4,74E+02	9,09E-13
	Середнє	7,63E-03	5,15E+03	7,91E-12	2,32E-11	1,42E-11	1,27E+10	3,40E+03	2,54E-11
	Медіана	6,17E-06	4,53E+03	7,28E-12	1,45E-11	1,45E-11	4,27E-02	2,97E+03	2,02E-11
	СВ	4,72E-02	2,59E+03	2,74E-12	2,92E-11	2,46E-12	6,94E+10	2,19E+03	2,23E-11
<i>f</i> 18	Мах	9,27E-01	1,10E+00	6,05E-01	7,28E-01	1,95E-01	1,11E+00	1,16E+00	6,96E-01
	Мін	2,84E-01	4,34E-01	3,21E-01	3,53E-01	8,08E-02	3,94E-01	5,19E-01	3,41E-01
	Середнє	5,54E-01	7,87E-01	4,46E-01	5,32E-01	1,43E-01	7,04E-01	8,53E-01	4,93E-01
	Медіана	5,54E-01	8,00E-01	4,46E-01	5,41E-01	1,39E-01	6,84E-01	8,75E-01	4,76E-01
	СВ	1,32E-01	1,39E-01	7,77E-02	8,50E-02	2,95E-02	1,33E-01	1,44E-01	8,92E-02
<i>f</i> 19	Мах	-3,32E+02	-2,95E+03	-3,21E+03	-3,29E+03	-3,35E+03	9,46E+02	-2,47E+03	-3,05E+03
	Мін	-3,69E+03	-3,53E+03	-3,61E+03	-3,72E+03	-3,77E+03	-3,92E+03	-3,19E+03	-3,52E+03
	Середнє	-3,49E+03	-3,19E+03	-3,40E+03	-3,44E+03	-3,49E+03	-3,75E+03	-2,85E+03	-3,29E+03
	Медіана	-3,51E+03	-3,19E+03	-3,41E+03	-3,41E+03	-3,48E+03	-3,92E+03	-2,87E+03	-3,31E+03
	СВ	9,25E+01	1,39E+02	8,76E+01	9,33E+01	9,04E+01	8,87E+02	1,47E+02	1,18E+02

Продовження табл. Г.2

1	2	3	4	5	6	7	8	9	10
f_{20}	Max	-3,10E+01	-2,41E+01	-1,49E+01	-3,60E+01	-2,59E+01	-8,97E+00	-2,77E+01	-3,21E+01
	Min	-3,65E+01	-3,84E+01	-1,85E+01	-4,35E+01	-3,15E+01	-3,96E+01	-3,94E+01	-4,27E+01
	Середнє	-3,37E+01	-3,17E+01	-1,70E+01	-3,98E+01	-2,86E+01	-3,61E+01	-3,21E+01	-3,77E+01
	Медіана	-3,36E+01	-3,17E+01	-1,70E+01	-3,98E+01	-2,83E+01	-3,74E+01	-3,17E+01	-3,79E+01
	СВ	1,45E+00	2,75E+00	8,74E-01	1,78E+00	1,29E+00	5,49E+00	2,69E+00	3,01E+00

Таблиця Г.3 – Ранги алгоритмів (на основі медіани)

Тестова функція	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-IA-PSO	PSO-Hybrid
1	2	3	4	5	6	7	8	9
f_1	2	6	5	3	4	1	8	7
f_2	2	6	4	3	5	1	8	7
f_{3^*}	2	2	2	2	2	1	2	2
f_4	2	6	1	4	5	3	8	7
f_5	1	5	7	4	3	2	6	8
f_6	4	2	1	6	3	5	8	7

Продовження табл. Г.3

1	2	3	4	5	6	7	8	9
<i>f7</i>	2	3	4	5	6	1	8	7
<i>f8</i>	2	1	5	4	6	3	8	7
<i>f9</i>	6	1	8	4	2	7	3	5
<i>f10</i>	2	6	3	4	5	1	8	7
<i>f11*</i>	1	6	3	5	4	2	6	6
<i>f12</i>	1	6	4	3	5	2	8	7
<i>f13</i>	2	3	7	6	8	1	4	5
<i>f14*</i>	2	6	5	3	4	1	7	7
<i>f15</i>	5	1	2	6	3	8	4	7
<i>f16</i>	6	2	1	4	3	8	7	5
<i>f17*</i>	2	3	3	3	3	1	3	3
<i>f18</i>	7	2	4	6	8	3	1	5
<i>f19</i>	4	1	5	7	8	6	2	3
<i>f20</i>	7	2	1	8	4	5	3	6
Сума рангів	62	70	75	90	91	62	112	118

* оскільки деякі з алгоритмів відповідають однаковим медіанним значенням, вони мають однакові значення показників.

Таблиця Г.4 – Показник $LogCr$ (на основі медіани)

Тестова функція	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-IA-PSO	PSO-Hybrid
1	2	3	4	5	6	7	8	9
f_1	0,03	0,35	0,09	0,06	0,08	0,02	1,00	0,84
f_2	0,03	0,49	0,09	0,08	0,10	0,02	1,00	0,92
f_3^*	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
f_4	0,03	0,08	0,03	0,03	0,03	0,03	1,00	0,63
f_5	0,85	1,00	1,00	0,95	0,94	0,85	1,00	1,00
f_6	0,00	0,00	0,00	0,00	0,00	0,00	1,00	0,67
f_7	0,01	0,02	0,02	0,05	0,06	0,01	1,00	0,68
f_8	0,02	0,01	0,06	0,02	0,08	0,02	1,00	0,80
f_9	0,84	0,23	1,00	0,65	0,43	0,89	0,50	0,70
f_{10}	0,04	0,93	0,08	0,27	0,35	0,04	1,00	0,98
f_{11}	0,04	1,00	0,08	0,19	0,14	0,05	1,00	1,00
f_{12}	0,02	0,46	0,08	0,08	0,10	0,02	1,00	0,80
f_{13}	0,56	0,58	0,95	0,77	1,00	0,56	0,59	0,73

Продовження табл. Г.4

1	2	3	4	5	6	7	8	9
f_{14}	0,50	0,73	0,70	0,59	0,68	0,50	1,00	1,00
f_{15}	0,99	0,68	0,97	0,99	0,97	1,00	0,99	0,99
f_{16}^{**}	-							
f_{17}^{***}	-							
f_{18}	0,73	0,53	0,62	0,69	1,00	0,62	0,52	0,66
f_{19}^{***}	-							
f_{20}^*	0,97	0,57	0,57	1,00	0,73	0,95	0,69	0,96
Сума значень критерію $LogCr$	6,67	8,67	7,34	7,44	7,71	6,58	15,28	14,36

* оскільки логарифм виразу є від'ємним числом, знаменник замінено на функцію „min”;

** не існує стабільного значення $f_{16,j=1}$; визначення показника неможливе;

*** чисельник виразу не можливо обчислити; визначення показника неможливе.

Таблиця Г.5 – Ранги алгоритмів (на основі мінімуму)

Тестова функція	Алгоритми							
	LDW- PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-IA-PSO	PSO- Hybrid
1	2	3	4	5	6	7	8	9
f_1	2	6	5	3	4	1	8	7
f_2	2	6	5	3	4	1	8	7
f_3^*	1	1	1	1	1	1	1	1
f_4	3	6	1	4	5	2	8	7
f_5	2	3	8	7	6	1	4	5
f_6	3	4	1	5	6	2	8	7
f_7	2	6	5	3	4	1	8	7
f_8	3	1	5	4	6	2	8	7
f_9	7	1	8	3	4	6	2	5
f_{10}	2	6	5	3	4	1	7	7
f_{11}^*	1	6	3	4	5	2	6	6
f_{12}	2	6	4	3	5	1	8	7
f_{13}	3	2	7	5	6	1	4	8

Продовження табл. Г.5

1	2	3	4	5	6	7	8	9
f_{14}^*	1	6	3	2	3	5	7	7
f_{15}	4	1	3	5	2	6	8	7
f_{16}	8	2	1	4	3	7	5	6
f_{17}^*	2	3	3	3	3	1	3	3
f_{18}	7	2	3	4	8	6	1	5
f_{19}	5	1	8	4	7	6	2	3
f_{20}	6	2	1	8	3	5	4	7
Сума рангів	66	71	80	78	89	58	110	119

* оскільки деякі з алгоритмів відповідають однаковим мінімальним значенням, вони мають однакові значення показників.

Таблиця Г.6 – Показник $LogCr$ (на основі мінімуму)

Тестова функція	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-IA-PSO	PSO-Hybrid
1	2	3	4	5	6	7	8	9
f_1	0,03	0,36	0,09	0,06	0,08	0,02	1,00	0,94
f_2	0,03	0,49	0,11	0,08	0,10	0,03	1,00	0,96

Продовження табл. Г.6

1	2	3	4	5	6	7	8	9
f_3^*	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
f_4	0,03	0,11	0,03	0,03	0,03	0,03	1,00	0,89
f_5	0,89	0,95	1,00	0,98	0,98	0,89	0,96	0,96
f_6	0,00	0,01	0,00	0,01	0,01	0,00	1,00	0,95
f_7	0,02	0,36	0,08	0,05	0,07	0,01	1,00	0,94
f_8	0,03	0,02	0,09	0,05	0,09	0,03	1,00	0,97
f_9	0,97	0,23	1,00	0,67	0,76	0,90	0,57	0,77
f_{10}^{**}	-							
f_{11}	0,05	1,00	0,11	0,21	0,25	0,06	1,00	1,00
f_{12}	0,03	0,46	0,10	0,08	0,11	0,02	1,00	0,95
f_{13}	0,45	0,43	0,83	0,63	0,79	0,42	0,46	1,00
f_{14}	0,51	0,77	0,73	0,66	0,73	0,75	1,00	1,00
f_{15}	0,98	0,79	0,97	0,98	0,96	0,98	1,00	1,00
f_{16}^{***}	-							
f_{17}^{****}	-							
f_{18}	0,73	0,60	0,61	0,66	1,00	0,68	0,53	0,67

Продовження табл. Г.6

1	2	3	4	5	6	7	8	9
$f19^{****}$	-							
$f20^*$	0,94	0,60	0,58	1,00	0,71	0,92	0,72	0,99
Сума величин критерію $LogCr$	6,70	8,17	7,33	7,15	7,67	6,75	14,23	14,99

* оскільки логарифм у виразі є від'ємним числом, знаменник замінено на функцію „min”;

** для алгоритмів PV-IA-PSO та PSO-Hybrid мінімальні значення дорівнюють нулю (знаменник аргументу функції „log”); визначення показника неможливе;

*** немає стабільного значення $f_{16,j=1}$; визначення показника неможливе;

**** чисельник виразу неможливо обчислити; визначення показника неможливе.

Таблиця Г.7 – Показники алгоритмів *ind*

Тестова функція	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-IA-PSO	PSO-Hybrid
1	2	3	4	5	6	7	8	9
$f1$	1,09	3,35	2,72	1,63	2,17	0,54	4,35	3,72
$f2$	1,09	3,26	2,23	1,72	2,56	0,54	4,35	3,80

Продовження табл. Г.7

1	2	3	4	5	6	7	8	9
f_3	0,55	0,74	1,12	0,68	0,69	0,65	0,76	0,67
f_4	1,36	3,27	0,66	2,17	2,72	1,36	4,35	3,79
f_5	0,84	2,55	2,81	2,78	2,25	0,88	3,07	3,91
f_6	2,03	1,51	0,57	3,10	2,40	2,17	4,35	3,80
f_7	1,09	2,32	2,02	2,54	2,89	0,54	4,35	3,80
f_8	1,36	0,56	2,17	2,17	3,26	1,40	4,35	3,80
f_9	3,52	0,56	3,26	2,03	1,53	3,54	1,51	2,71
f_{10}	1,08	3,17	2,02	2,23	2,49	0,55	4,18	3,81
f_{11}	0,56	3,26	1,63	2,56	2,35	1,08	3,26	3,26
f_{12}	0,83	3,26	2,10	1,72	2,72	0,89	4,35	3,80
f_{13}	1,28	1,49	3,18	3,11	4,04	0,54	2,17	3,30
f_{14}	0,95	3,26	2,44	1,49	2,02	1,57	3,81	3,80
f_{15}	2,76	0,66	1,36	3,09	1,75	3,68	3,01	3,88
f_{16}	3,63	1,13	1,12	2,18	1,68	4,15	3,49	2,89
f_{17}	1,09	1,67	1,64	1,63	1,63	0,54	1,66	1,65
f_{18}	3,79	1,10	1,94	2,98	4,35	2,32	0,62	2,71

Продовження табл. Г.7

1	2	3	4	5	6	7	8	9
f_{19}	2,45	0,57	2,84	3,40	4,17	3,06	1,11	1,71
f_{20}	3,63	1,08	1,13	4,33	2,06	2,69	1,83	3,43
Сума показників <i>ind</i>	34,95	38,76	38,96	47,56	49,73	32,71	60,88	64,25

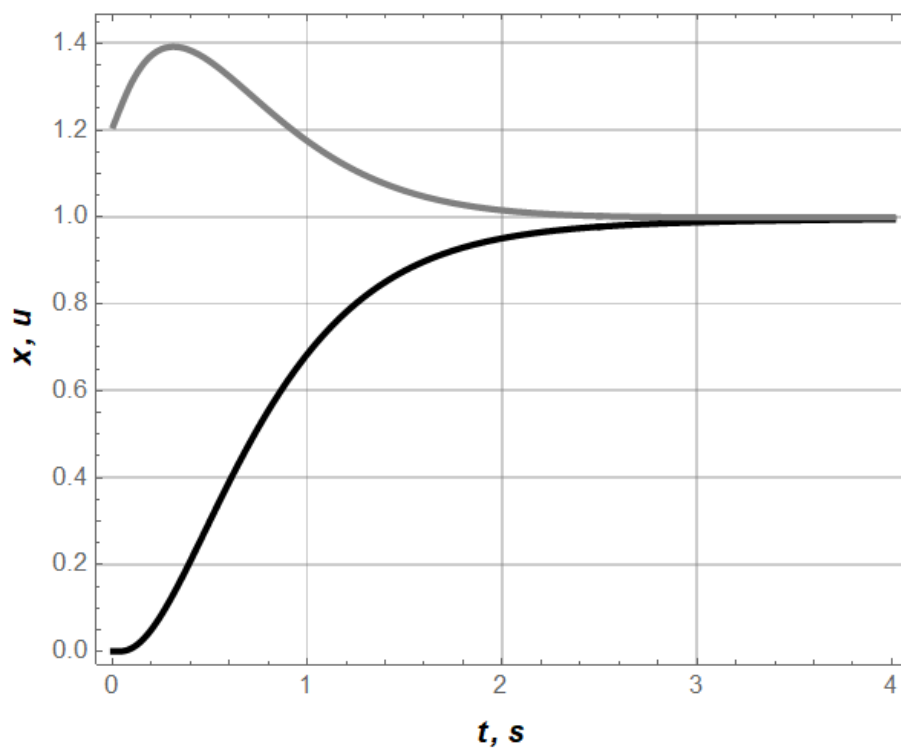
Таблиця Г.8 – Середня тривалість виконання алгоритмів

Тестова функція	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-IA-PSO	PSO-Hybrid
1	2	3	4	5	6	7	8	9
f_1	16,56	21,00	17,73	18,47	18,28	21,77	20,52	24,60
f_2	17,07	21,30	18,35	18,86	18,54	21,82	19,79	26,21
f_3	16,82	20,65	17,62	18,39	18,33	21,23	19,47	23,02
f_4	18,05	22,09	18,67	19,15	19,26	22,03	21,22	24,91
f_5	17,47	21,69	17,89	18,46	18,48	21,92	20,85	23,68
f_6	17,54	21,64	18,14	18,54	18,84	22,67	20,06	25,10
f_7	17,56	22,12	17,95	18,86	18,69	28,94	19,71	25,33
f_8	17,65	22,25	17,82	18,77	18,72	26,65	19,71	25,76

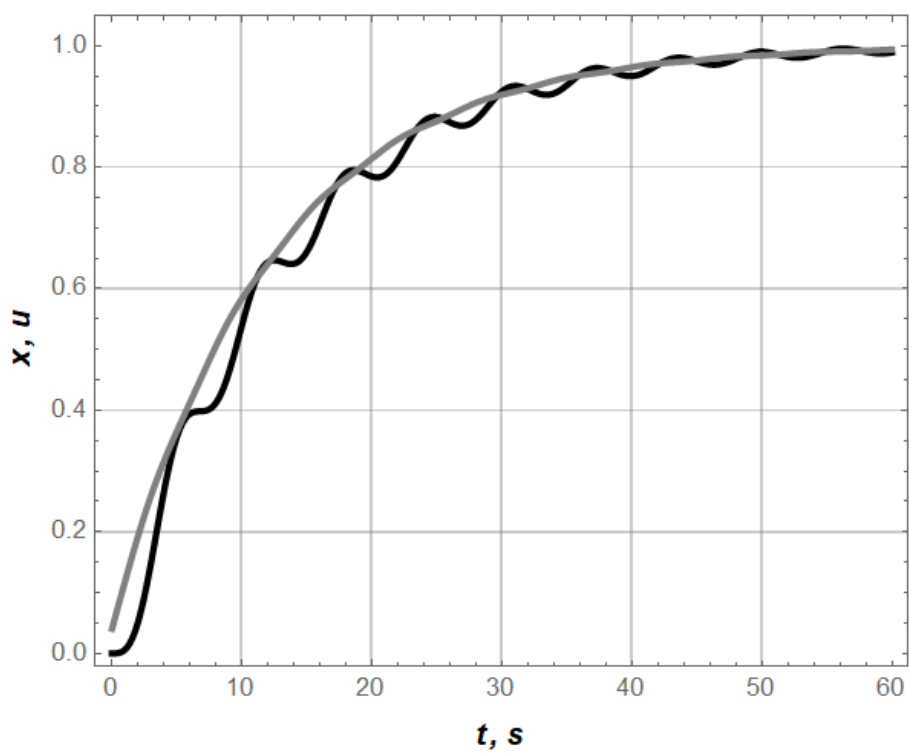
Продовження табл. Г.8

1	2	3	4	5	6	7	8	9
<i>f9</i>	18,07	22,36	18,05	19,33	19,81	28,76	20,78	23,99
<i>f10</i>	17,56	22,00	17,89	18,92	18,54	27,76	19,78	27,95
<i>f11</i>	19,77	23,89	20,37	21,00	20,98	30,95	22,85	29,00
<i>f12</i>	17,39	21,75	17,56	18,68	18,45	28,04	20,27	26,85
<i>f13</i>	17,55	21,77	17,70	18,75	18,51	27,39	20,20	23,91
<i>f14</i>	17,50	21,56	17,75	18,69	18,51	28,21	19,71	24,32
<i>f15</i>	20,63	27,41	26,31	22,24	26,07	33,37	27,05	29,54
<i>f16</i>	18,24	22,64	18,64	19,27	19,64	25,03	20,73	24,97
<i>f17</i>	17,55	21,57	17,88	18,59	18,33	28,80	19,71	25,32
<i>f18</i>	17,71	21,50	18,06	18,65	18,64	19,48	20,37	24,90
<i>f19</i>	17,55	21,62	17,90	18,85	18,41	23,13	19,88	25,16
<i>f20</i>	18,67	23,51	19,70	20,20	20,68	22,69	22,15	26,34
Загальна тривалість виконання алгоритму	356,91	444,32	371,98	382,67	385,71	510,64	414,81	510,86

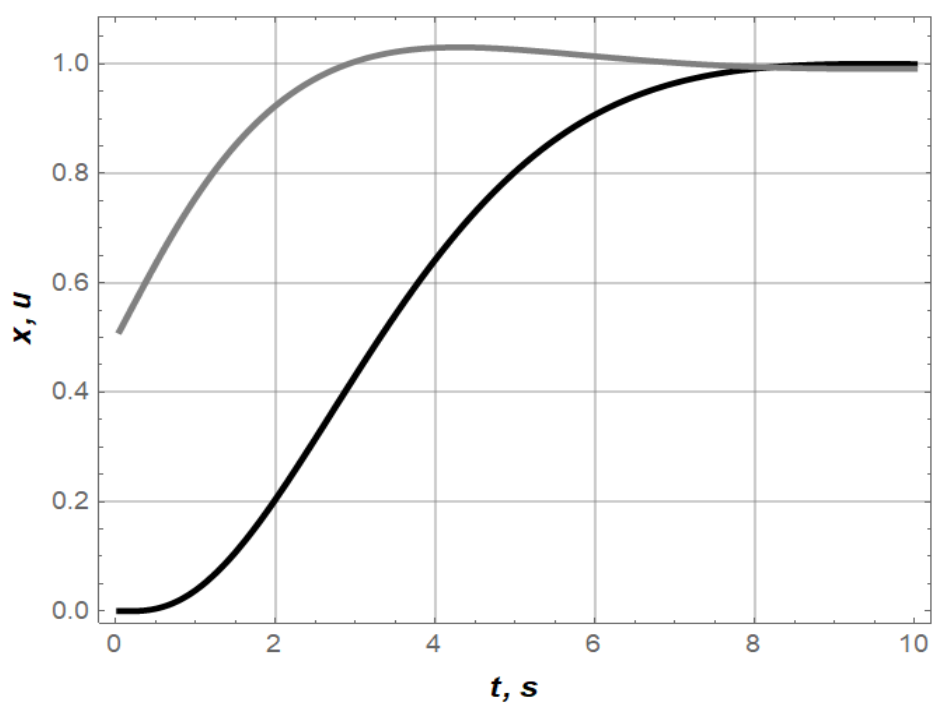
Додаток Д



а)

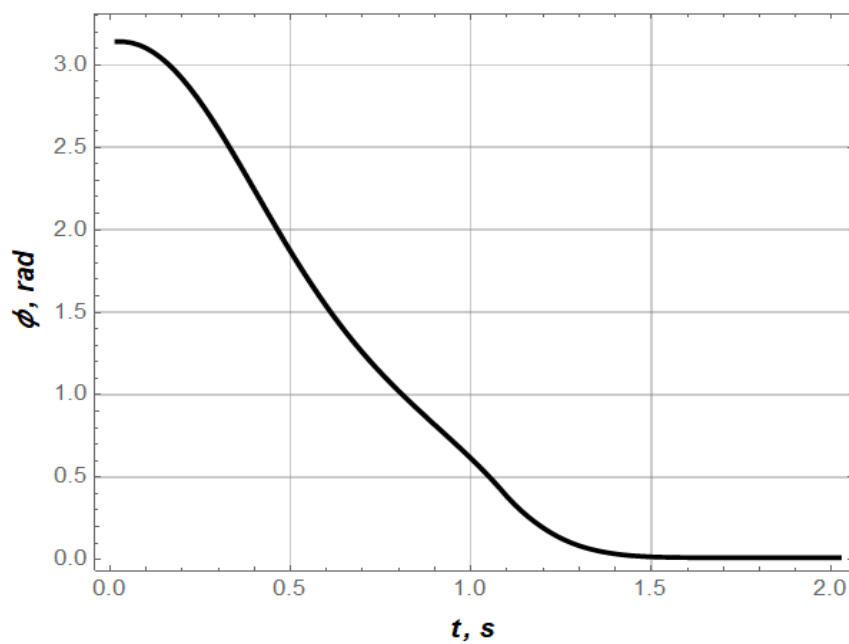


б)

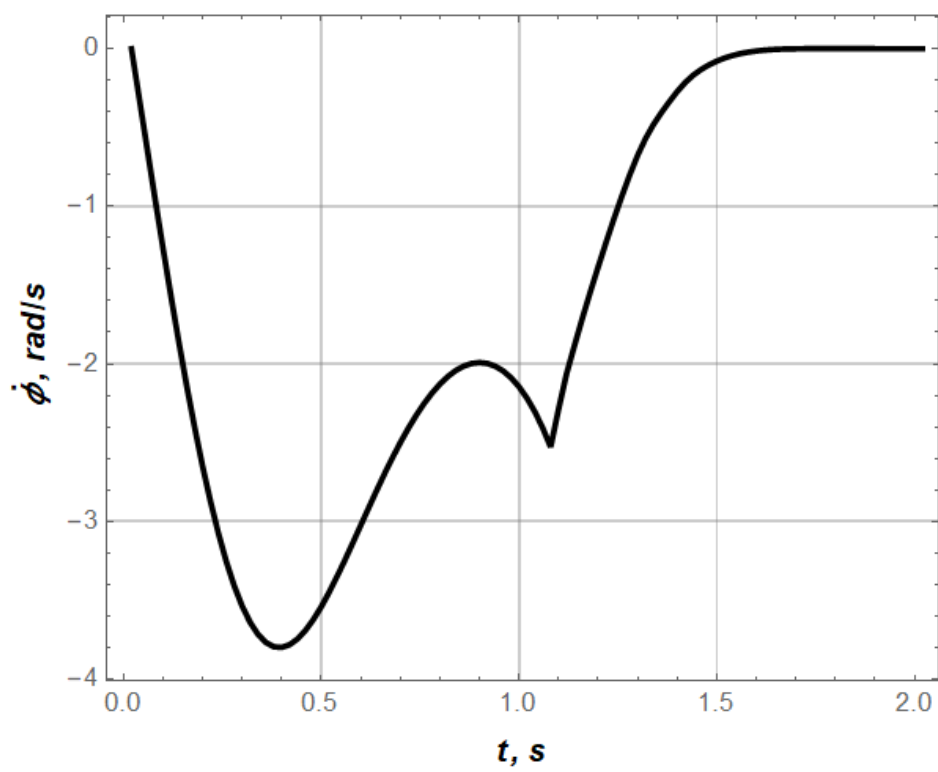


в)

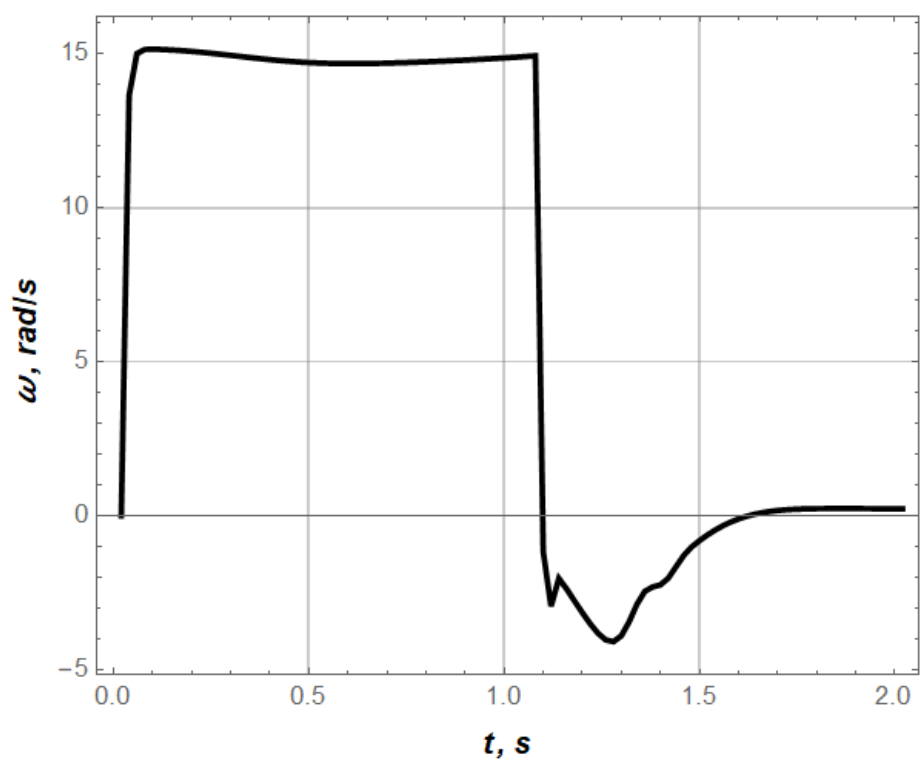
Рисунок Д.1 – Графіки зміни регульованої величини x (чорна крива) та функції регулювання u (сіра крива) для об'єктів регулювання, в контур керування яких включено оптимально налаштований ПІ-регулятор і які описуються передавальними функціями: а) $G_1(s)$; б) $G_2(s)$; в) $G_3(s)$



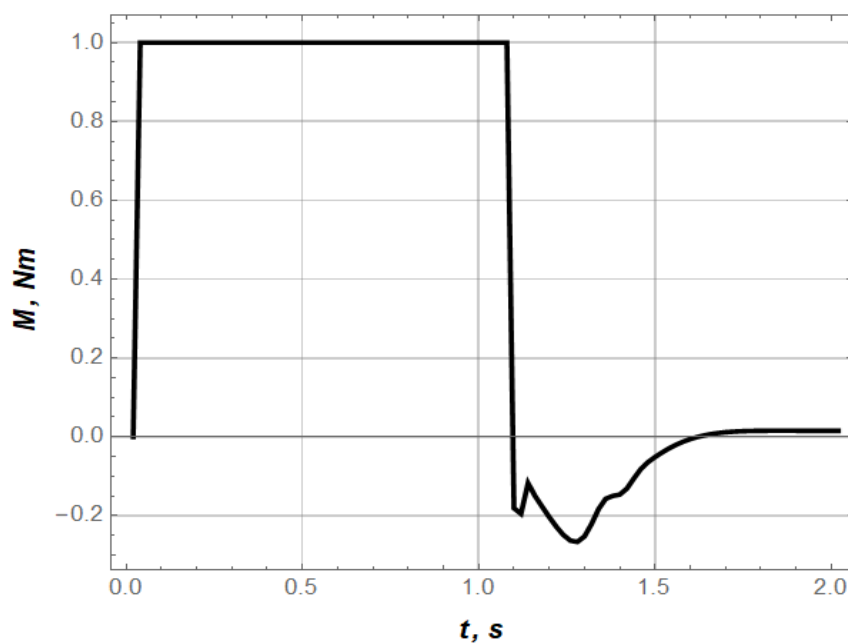
а)



б)

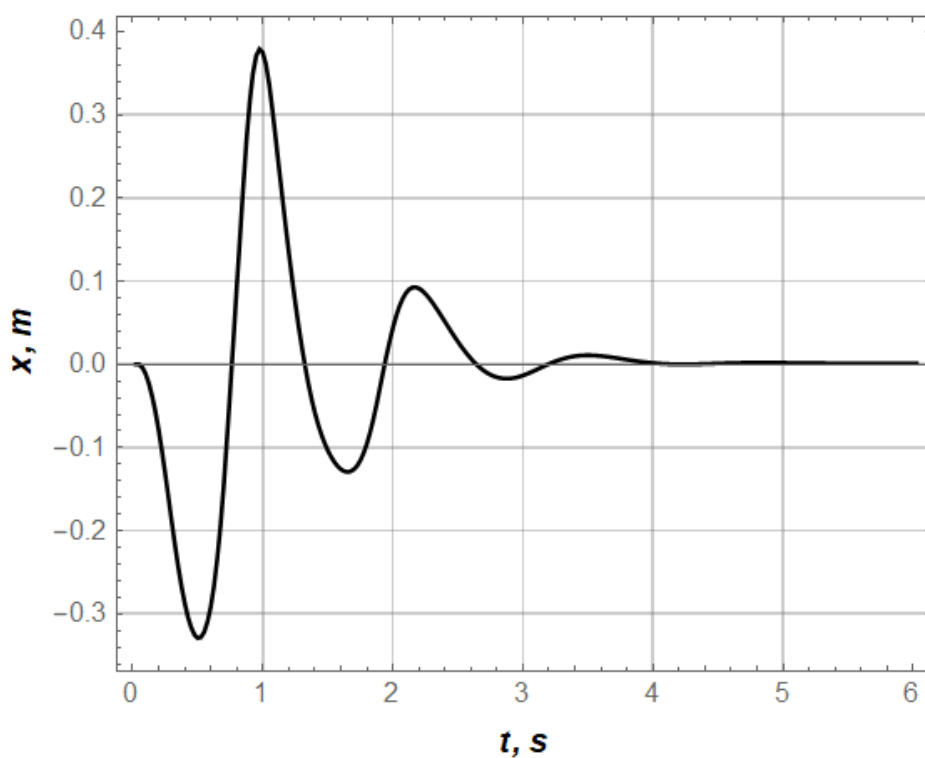


B)

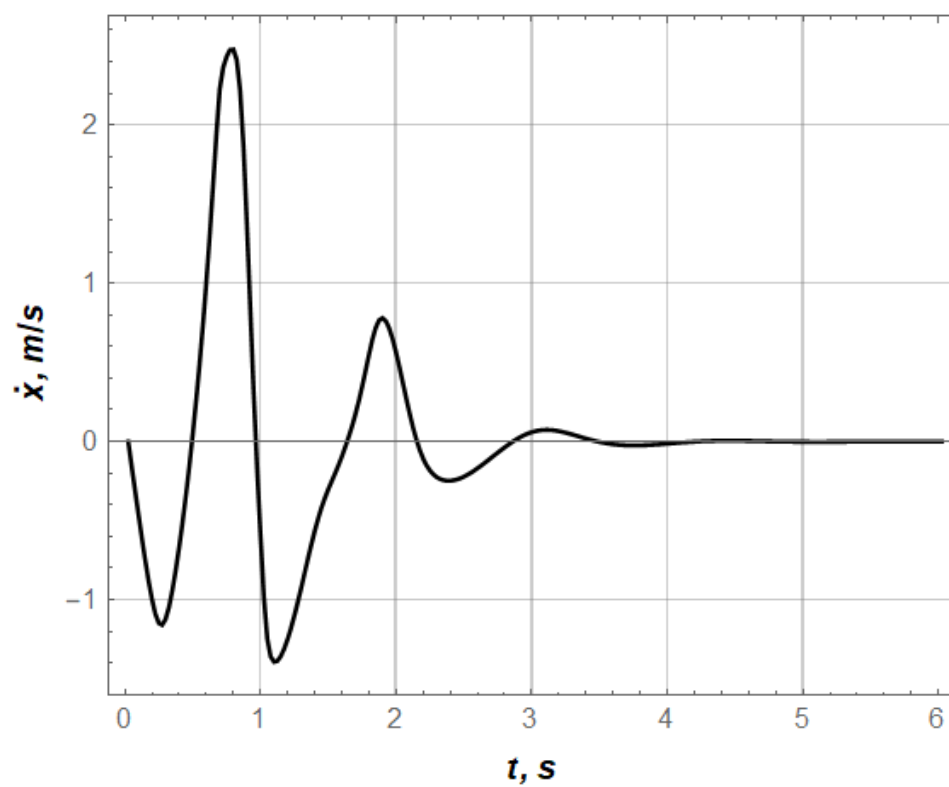


г)

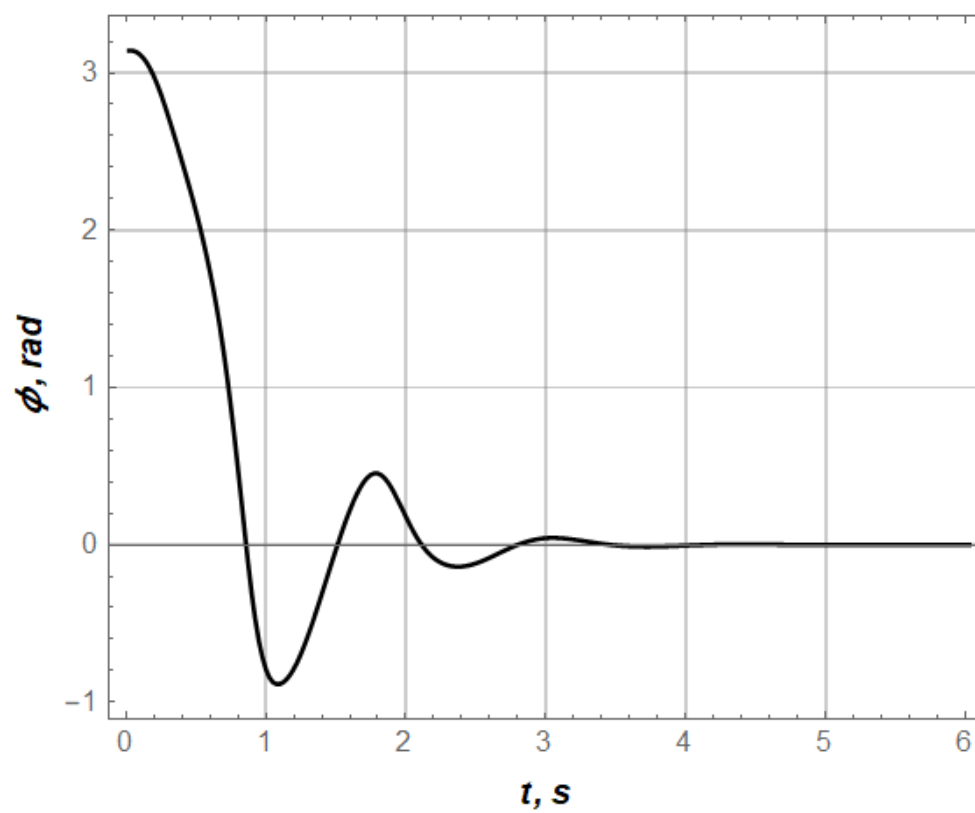
Рисунок Д.2 – Графіки регулювання руху перевернутого маятника з маховиком з використанням ШНМ: а) кут маятникової ланки ; б) кутова швидкість ланки маятника; в) кутова швидкість маховика маятника ; г) приводний момент



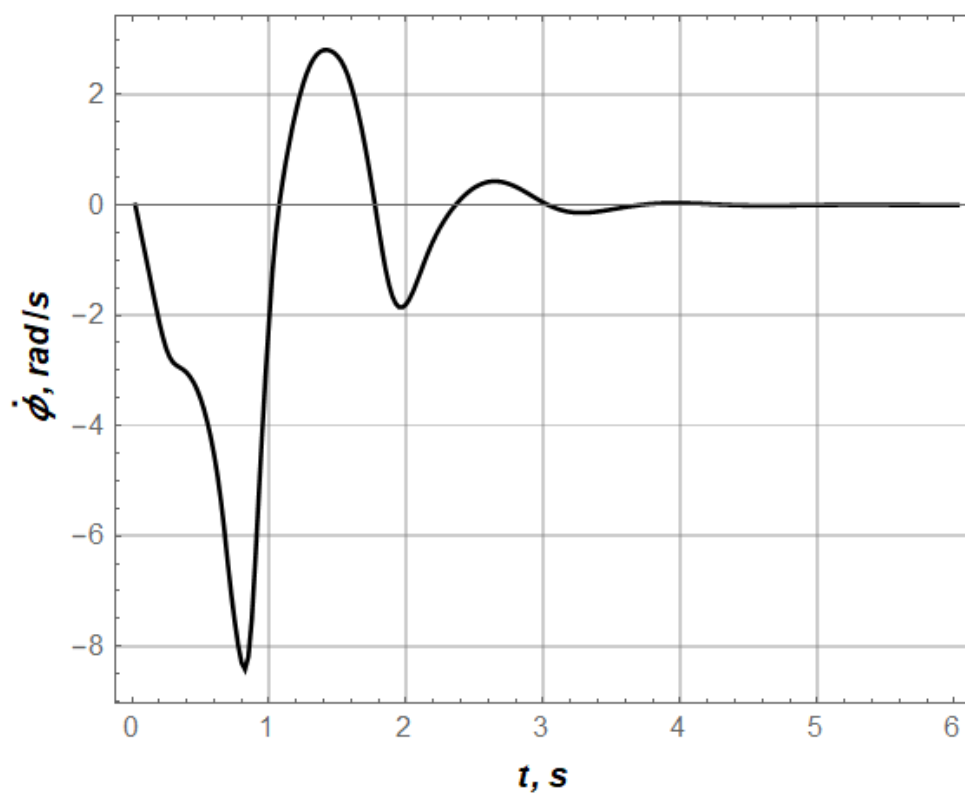
а)



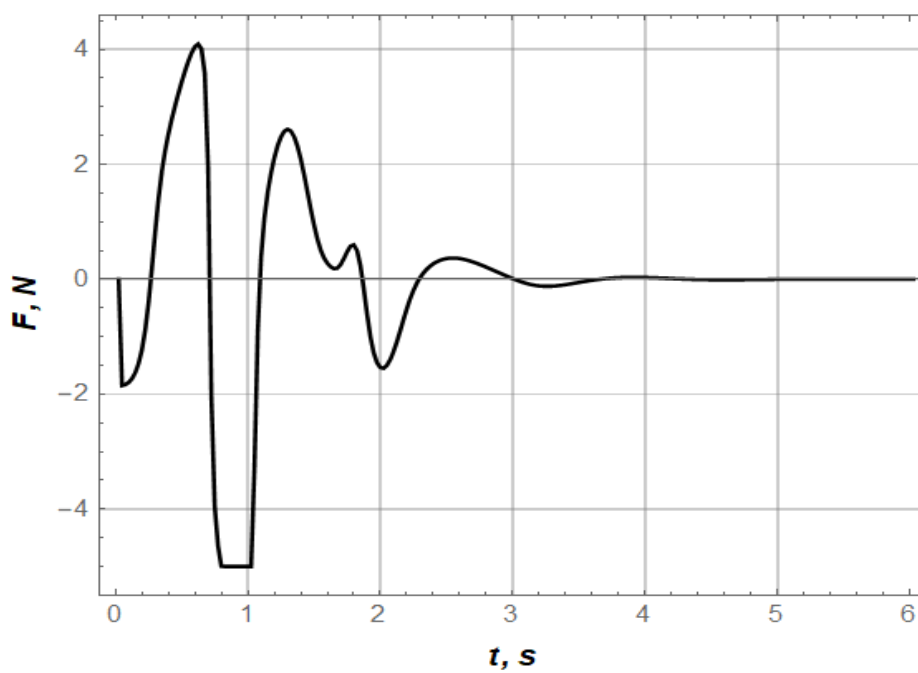
б)



в)

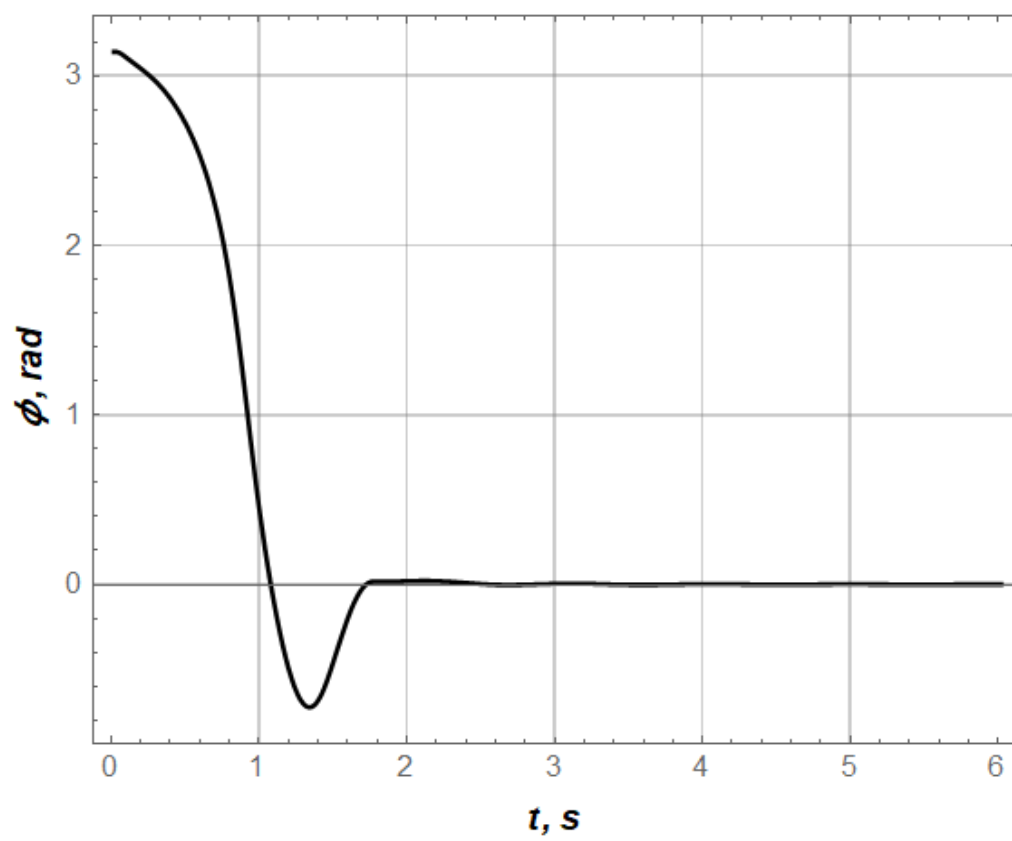


г)

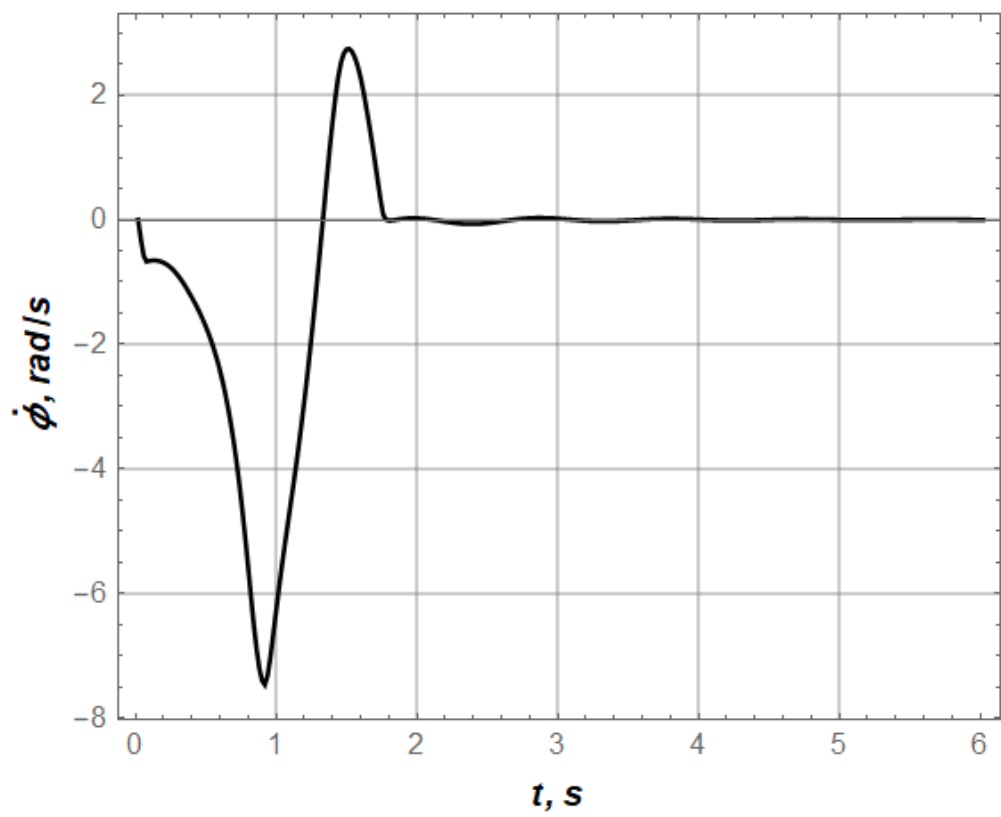


д)

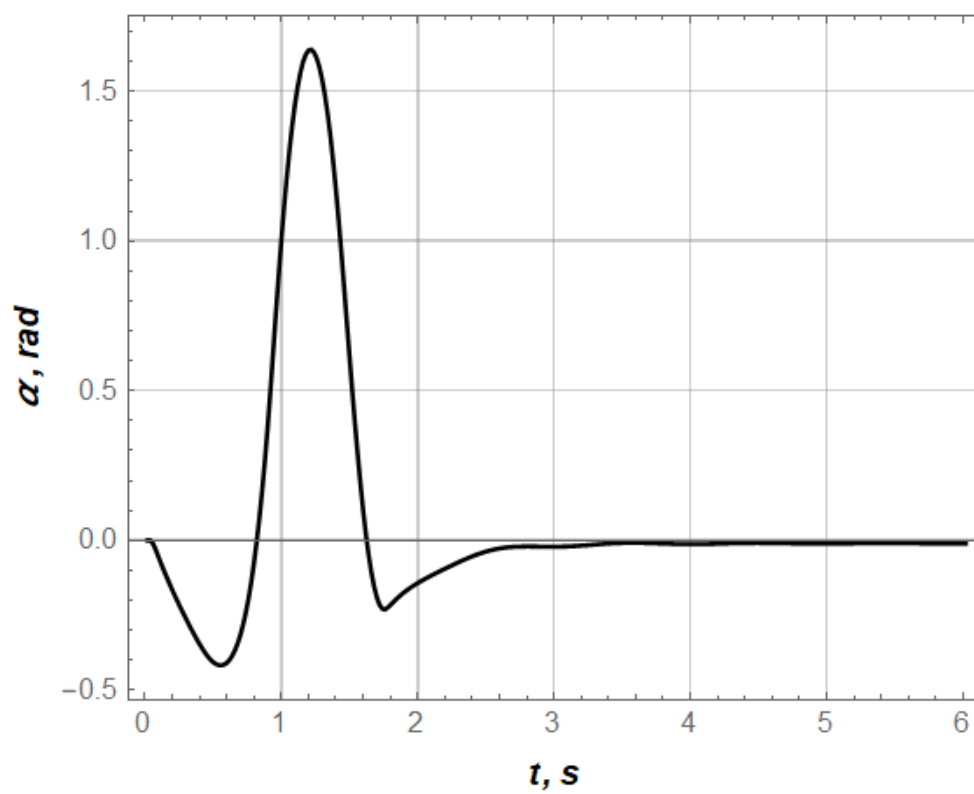
Рисунок Д.3 – Графіки регулювання руху перевернутого маятника з використанням ШНМ: а) координата візка; б) швидкість візка; в) кут нахилу ланки маятника; г) кутова швидкість ланки маятника; д) рушійна сила



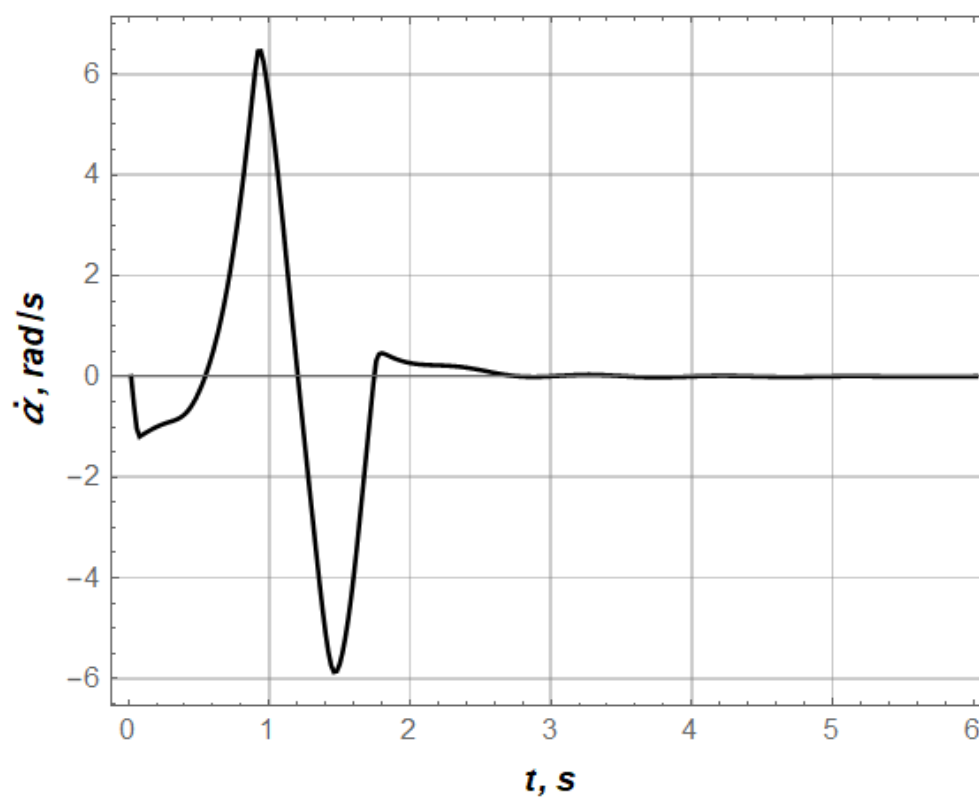
a)



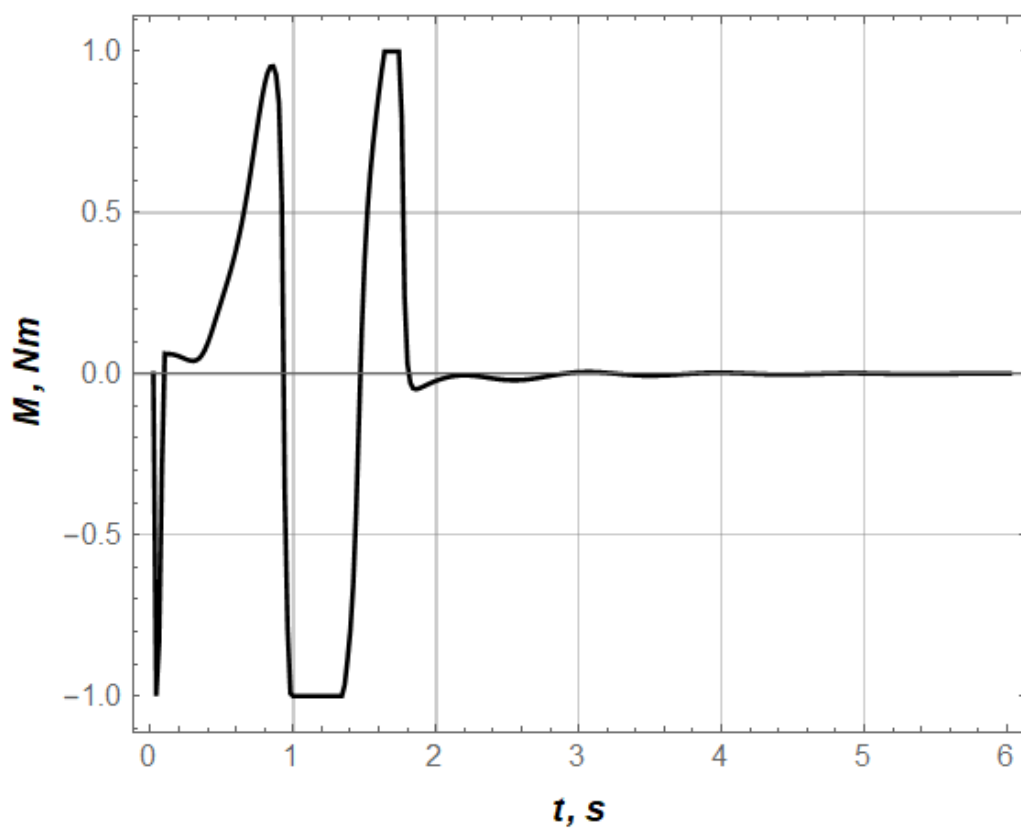
б)



B)



Г)



д)

Рисунок Д.4 – Графіки регулювання руху маятника Фурути з використанням ШНМ: а) кут нахилу ланки маятника; б) кутова швидкість ланки маятника; в) кут нахилу стріли; г) кутова швидкість стріли; д) приводний момент

Додаток Е

Таблиця Е.1 – Статистичні результати розв’язання першої задачі (конструкція зварної балки)

Результати	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-ND-PSO	PSO-Hybrid
Статистичні показники								
Мах	1,82916	1,76404	1,69525	1,82573	1,69525	4,2416	1,70895	1,87048
Min	1,69525	1,71873	1,69525	1,69525	1,69525	1,69544	1,69642	1,69525
Середнє	1,70093	1,74027	1,69525	1,70325	1,69525	1,75407	1,70077	1,70193
Медіана	1,69525	1,74013	1,69525	1,69525	1,69525	1,69705	1,70032	1,69525
СВ*	2,119E-02	1,139E-02	1,168E-07	2,552E-02	1,551E-14	3,598E-01	3,0206E-03	2,788E-02
Значення аргументів, які відповідають кращому розв’язку задачі								
x_1	0,20573	0,19048	0,20573	0,20573	0,20573	0,20553	0,20516	0,20573
x_2	3,25312	3,57614	3,25312	3,25312	3,25312	3,2567	3,26411	3,25312
x_3	9,03662	9,02828	9,03662	9,03662	9,03662	9,03663	9,03446	9,03662
x_4	0,20573	0,20636	0,20573	0,20573	0,20573	0,20573	0,20585	0,20573

Значення цільової функції на найкращому розв’язку цієї задачі [178] рівне 1,672966512E+00.

Таблиця Е.2 – Статистичні результати розв’язання другої задачі (конструкція ємності під тиском)

Результати	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-ND-PSO	PSO-Hybrid
Статистичні показники								
Max	7544,49	7559,08	7332,84	6771,6	7273,51	251991,0	7544,72	7544,49
Min	5850,38	5853,72	5850,38	5850,38	5850,38	5852,46	5850,89	5850,38
Середнє	6416,96	6193,01	6194,41	6031,04	6204,48	11486,30	6361,47	6441,29
Медіана	6222,63	6031,42	6090,53	6059,71	6090,53	6419,11	6234,61	6370,78
СВ*	5,021E+02	4,213E+02	3,580E+02	1,932E+02	3,244E+02	3,471E+04	4,989E+02	5,199E+02
Значення аргументів, які відповідають кращому розв’язку задачі								
x_1	12,0687	12,2556	12,3776	12,0265	12,3433	12,3161	12,2165	12,093
x_2	6,48249	6,28339	6,10892	5,67205	6,32045	5,58208	5,91768	6,41103
x_3	38,8601	38,8522	38,8601	38,8601	38,8601	38,8458	38,8601	38,8601
x_4	221,365	221,599	221,365	221,365	221,365	221,589	221,392	221,365

Значення цільової функції на найкращому розв’язку цієї задачі [179] рівне 5,8497617E+03.

Таблиця Е.3 – Статистичні результати розв’язання третьої задачі (конструкція пружини)

Результати	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-ND-PSO	PSO-Hybrid
Статистичні показники								
Max	0,0170995	0,0133127	0,0129688	0,0127874	0,0132100	0,0513344	0,0142682	0,0145066
Min	0,0126653	0,012734	0,0126664	0,0126652	0,0126656	0,0126668	0,0126798	0,0126656
Середнє	0,0136885	0,0128911	0,0127524	0,0126877	0,012731	0,0137169	0,0128927	0,0129391
Медіана	0,0130559	0,0128227	0,0127296	0,0126737	0,0126765	0,0127889	0,0127671	0,0127558
СВ*	1,306E-03	1,466E-04	6,677E-05	2,702E-05	1,129E-04	5,443E-03	2,860E-04	4,038E-04
Значення аргументів, які відповідають кращому розв’язку задачі								
x_1	0,05177	0,05160	0,05169	0,05167	0,05166	0,05140	0,05185	0,05183
x_2	0,35861	0,35455	0,35681	0,35620	0,35608	0,34986	0,36068	0,36016
x_3	11,1789	11,4876	11,2844	11,3194	11,3269	11,7029	11,0742	11,0902

Значення цільової функції на найкращому розв’язку цієї задачі [99, 101, 180-188] рівне 1,26650E-02.

Таблиця Е.4 – Статистичні результати розв’язання четвертої задачі (конструкція дерев’яної ферми)

Результати	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-ND-PSO	PSO-Hybrid
Статистичні показники								
Max	263,900	263,986	263,896	263,899	263,900	733,469	263,920	263,901
Min	263,896	263,897	263,896	263,896	263,896	263,896	263,896	263,896
Середнє	263,897	263,931	263,896	263,896	263,896	273,288	263,903	263,896
Медіана	263,896	263,927	263,896	263,896	263,896	263,896	263,901	263,896
СВ*	1,264E-03	2,202E-02	8,123E-05	7,051E-04	5,578E-04	6,641E+01	5,656E-03	8,518E-04
Значення аргументів, які відповідають кращому розв’язку задачі								
x_1	0,78870	0,78883	0,78868	0,78866	0,78867	0,78871	0,78938	0,78865
x_2	0,40818	0,40784	0,40825	0,40829	0,40825	0,40814	0,40626	0,40832

Значення цільової функції на найкращому розв’язку цієї задачі [187] рівне 2,6368E+02.

Таблиця Е.5 – Статистичні результати розв’язання п’ятої задачі (конструкція редуктора)

Результати	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-ND-PSO	PSO-Hybrid
Статистичні показники								
Мах	3033,76	3083,49	3033,75	2994,47	2994,47	NA	3014,53	3046,71
Min	2994,47	3022,23	2994,47	2994,47	2994,47	2994,47	2996,69	2994,47
Середнє	3003,22	3060,19	2998,59	2994,47	2994,47	NA	3003,13	2997,15
Медіана	2994,47	3061,77	2994,47	2994,47	2994,47	2994,54	3002,59	2994,47
СВ*	1,572E+01	1,157E+01	1,191E+01	1,539E-11	1,258E-11	NA	3,817E+00	9,353E+00
Значення аргументів, які відповідають кращому розв’язку задачі								
x_1	3,50000	3,51619	3,50000	3,50000	3,50000	3,50000	3,50075	3,50000
x_2	0,70000	0,70000	0,70000	0,70000	0,70000	0,70000	0,70000	0,70000
x_3	17,0000	17,0000	17,0000	17,0000	17,0000	17,0000	17,0000	17,0000
x_4	7,30000	8,30000	7,30000	7,30000	7,30000	7,30006	7,31568	7,30000
x_5	7,71532	7,7526	7,71532	7,71532	7,71532	7,71541	7,76319	7,71532
x_6	3,35021	3,39378	3,35021	3,35021	3,35021	3,35022	3,35079	3,35021
x_7	5,28665	5,28712	5,28665	5,28665	5,28665	5,28666	5,28758	5,28665

Значення цільової функції на найкращому розв’язку цієї задачі [188] рівне 2,99444E+03.

Таблиця Е.6 – Статистичні результати розв’язання шостої задачі (конструкція зубчастої передачі)

Результати	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-ND-PSO	PSO-Hybrid
Статистичні показники								
Max	2,726E-08	1,263E-09	2,726E-08	1,362E-09	9,746E-10	7,802E-08	9,922E-10	1,362E-09
Min	2,308E-11	2,701E-12	2,701E-12	2,701E-12	2,701E-12	2,701E-12	2,701E-12	2,701E-12
Середнє	2,445E-09	2,155E-10	1,406E-09	2,736E-10	5,314E-11	5,376E-09	2,237E-10	1,915E-10
Медіана	9,922E-10	2,308E-11	3,068E-10	2,308E-11	2,308E-11	2,358E-09	2,308E-11	2,308E-11
СВ*	5,360E-09	3,833E-10	4,017E-09	4,185E-10	1,459E-10	1,237E-08	3,894E-10	3,661E-10
Значення аргументів, які відповідають кращому розв’язку задачі								
x_1	53	49	43	49	43	49	43	49
x_2	13	16	19	16	16	19	19	19
x_3	30	19	16	19	19	16	16	16
x_4	51	43	49	43	49	43	49	43

Значення цільової функції на найкращому розв’язку цієї задачі [103, 189-197] рівне 2,700857E-12.

Таблиця Е.7 – Статистичні результати розв’язання сьомої задачі (конструкція консольної балки)

Результати	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-ND-PSO	PSO-Hybrid
Статистичні показники								
Мах	1,34007	1,34858	1,33996	1,34013	1,33999	8,92039	1,34034	1,34005
Min	1,33996	1,34199	1,33996	1,33996	1,33996	1,33996	1,33999	1,33996
Середнє	1,33998	1,34488	1,33996	1,34000	1,33996	1,57591	1,34011	1,33997
Медіана	1,33997	1,34476	1,33996	1,33999	1,33996	1,34002	1,34009	1,33996
СВ*	2,191E-05	1,673E-03	1,678E-06	3,129E-05	4,263E-06	1,216E+00	9,570E-05	1,558E-05
Значення аргументів, які відповідають кращому розв’язку задачі								
x_1	6,01372	5,97875	6,01701	6,01825	6,01614	6,02063	6,02572	6,01524
x_2	5,31241	5,28874	5,3073	5,29647	5,30838	5,29961	5,31047	5,30854
x_3	4,49181	4,50194	4,49636	4,49466	4,49379	4,49756	4,49796	4,49542
x_4	3,50396	3,55799	3,50227	3,50594	3,50252	3,5078	3,49984	3,4988
x_5	2,15179	2,17889	2,15073	2,15845	2,15283	2,14817	2,14017	2,15567

Значення цільової функції на найкращому розв’язку цієї задачі [99, 198, 199] рівне 1,33995E+00.

Таблиця Е.8 – Статистичні результати розв’язання восьмої задачі (конструкція двотаврової балки)

Результати	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-ND-PSO	PSO-Hybrid
Статистичні показники								
Max	0,0136885	0,0130813	0,0130741	0,0135486	0,0132365	NA	0,0130775	0,0130757
Min	0,0130741	0,0130741	0,0130741	0,0130741	0,0130741	0,0130741	0,0130742	0,0130741
Середнє	0,0130866	0,0130758	0,0130741	0,0131111	0,0130792	NA	0,0130752	0,0130742
Медіана	0,0130741	0,0130753	0,0130741	0,0130749	0,0130741	0,0130741	0,0130749	0,0130741
СВ*	8,687E-05	1,747E-06	1,030E-15	8,034E-05	2,480E-05	NA	8,429E-07	2,215E-07
Значення аргументів, які відповідають кращому розв’язку задачі								
x_1	50	50	50	50	50	50	50	50
x_2	80	80	80	80	80	80	80	80
x_3	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9
x_4	2,32179	2,32179	2,32179	2,32179	2,32179	2,32179	2,32178	2,32179

Значення цільової функції на найкращому розв’язку цієї задачі [99, 178] рівне 1,30741E-02.

Таблиця Е.9 – Статистичні результати розв’язання дев’ятої задачі (конструкція гофрованої перегородки)

Результати	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-ND-PSO	PSO-Hybrid
Статистичні показники								
Max	6,84296	7,01644	6,84296	6,84296	6,84296	9,88733	6,86387	6,85285
Min	6,84296	6,87482	6,84296	6,84296	6,84296	6,84298	6,84467	6,84296
Середнє	6,84296	6,94990	6,84296	6,84296	6,84296	6,90411	6,85013	6,84316
Медіана	6,84296	6,94710	6,84296	6,84296	6,84296	6,84312	6,84920	6,84296
СВ*	2,787E-10	3,241E-02	1,117E-14	1,404E-14	7,305E-15	4,305E-01	3,962E-03	1,398E-03
Значення аргументів, які відповідають кращому розв’язку задачі								
x_1	57,6923	57,7526	57,6923	57,6923	57,6923	57,6915	57,6887	57,6923
x_2	34,1476	34,3847	34,1476	34,1476	34,1476	34,1475	34,1344	34,1476
x_3	57,6923	57,2218	57,6923	57,6923	57,6923	57,6917	57,6047	57,6923
x_4	1,05	1,05152	1,05	1,05	1,05	1,05	1,05006	1,05

Значення цільової функції на найкращому розв’язку цієї задачі [99] рівне 6,84295801E+00.

Таблиця Е.10 – Статистичні результати розв’язання десятої задачі (конструкція залізобетонної балки)

Результати	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-ND-PSO	PSO-Hybrid
Статистичні показники								
Max	362,634	362,171	362,634	359,208	359,208	362,634	359,234	359,208
Min	359,208	359,221	359,208	359,208	359,208	359,208	359,208	359,208
Середнє	359,681	359,706	359,474	359,208	359,208	359,277	359,211	359,208
Медіана	359,208	359,511	359,208	359,208	359,208	359,208	359,210	359,208
СВ*	1,186E+00	5,382E-01	9,138E-01	9,043E-14	9,223-14	4,845E-01	4,288E-03	8,285E-12
Значення аргументів, які відповідають кращому розв’язку задачі								
x_1	6,00	6,32	6,32	6,32	6,32	6,32	6,32	6,32
x_2	34	34	34	34	34	34	34	34
x_3	8,50000	8,50063	8,50000	8,50000	8,50000	8,50000	8,50000	8,50000

Значення цільової функції на найкращому розв’язку цієї задачі [98, 103, 189] рівне 3,59208E+02.

Таблиця Е.11 – Статистичні результати розв’язання одинадцятої задачі (конструкція трубчастої колони)

Результати	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-ND-PSO	PSO-Hybrid
Статистичні показники								
Мах	26,4995	26,6281	26,4995	26,4995	26,4995	30,9694	26,5091	26,4995
Min	26,4995	26,5030	26,4995	26,4995	26,4995	26,4995	26,5004	26,4995
Середнє	26,4995	26,5407	26,4995	26,4995	26,4995	26,5889	26,5039	26,4995
Медіана	26,4995	26,5379	26,4995	26,4995	26,4995	26,4995	26,5036	26,4995
СВ*	1,522E-14	2,469E-02	1,295E-14	1,344E-14	1,586E-14	6,321E-01	2,128E-03	5,812E-09
Значення аргументів, які відповідають кращому розв’язку задачі								
x_1	5,45116	5,45286	5,45116	5,45116	5,45116	5,45116	5,45125	5,45116
x_2	0,29197	0,29188	0,29197	0,29197	0,29197	0,29197	0,29197	0,29197

Значення цільової функції на найкращому розв’язку цієї задачі [103, 189] рівне 2,648636147E+01.

Таблиця Е.12 – Статистичні результати розв’язання дванадцятої задачі (параметри металоконструкції автомобіля при дії бокового удару)

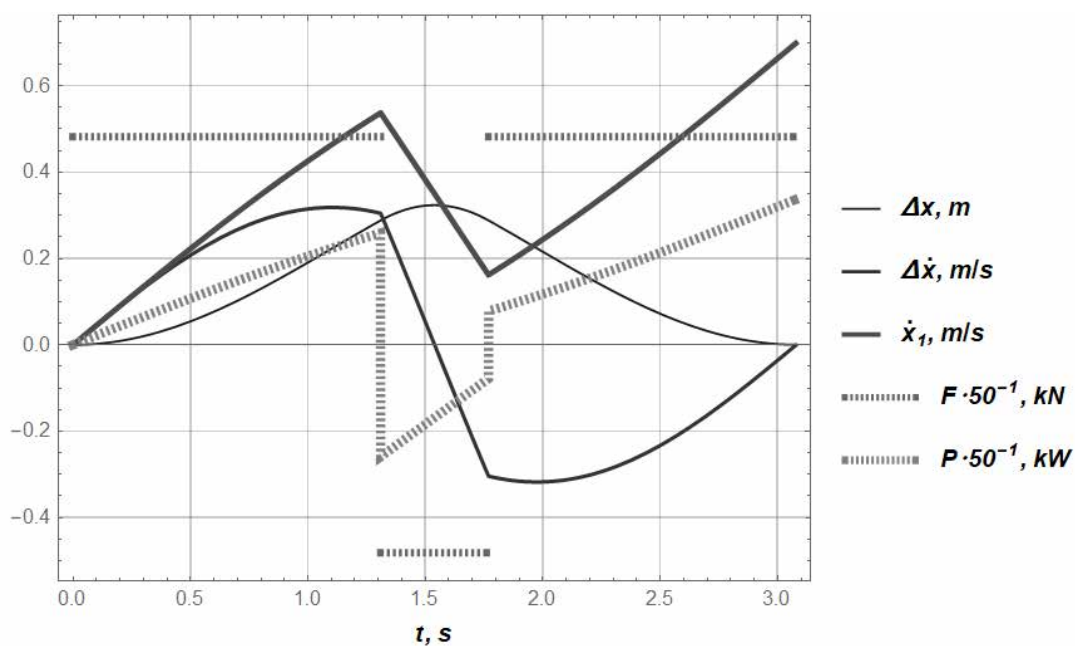
Результати	Алгоритми							
	LDW-PSO	IA-PSO	VCT-PSO	Ring-PSO	R-Ring-PSO	ME-D-PSO	PV-ND-PSO	PSO-Hybrid
1	2	3	4	5	6	7	8	9
Статистичні показники								
Max	22,2849	22,5352	22,8108	21,6704	21,5684	28,5447	22,3966	23,8118
Min	20,8123	21,1075	20,8124	20,8181	20,8172	20,8266	20,8340	20,8268
Середнє	21,2710	21,4407	21,2083	21,1260	21,1132	21,2552	21,3747	21,3307
Медіана	21,1989	21,3954	21,1968	21,1956	21,1963	21,1967	21,2567	21,2035
СВ*	0,35726	0,264911	0,395774	0,182804	0,184625	1,06937	0,346739	0,53139
Значення аргументів, які відповідають кращому розв’язку задачі								
x_1	0,50000	0,50000	0,50002	0,50000	0,50000	0,50000	0,50000	0,50000
x_2	0,98329	0,95025	0,9836	0,97838	0,97785	0,97187	0,97411	0,97772
x_3	0,50000	0,50000	0,50000	0,50000	0,50000	0,50000	0,50000	0,50177
x_4	1,01715	1,11861	1,01644	1,01978	1,02741	1,03946	1,03702	1,02357
x_5	0,50000	0,50964	0,50001	0,50006	0,50003	0,50000	0,50022	0,50056
x_6	0,50000	0,50000	0,50000	0,50009	0,50000	0,50149	0,50000	0,50000

Продовження табл. Е.12

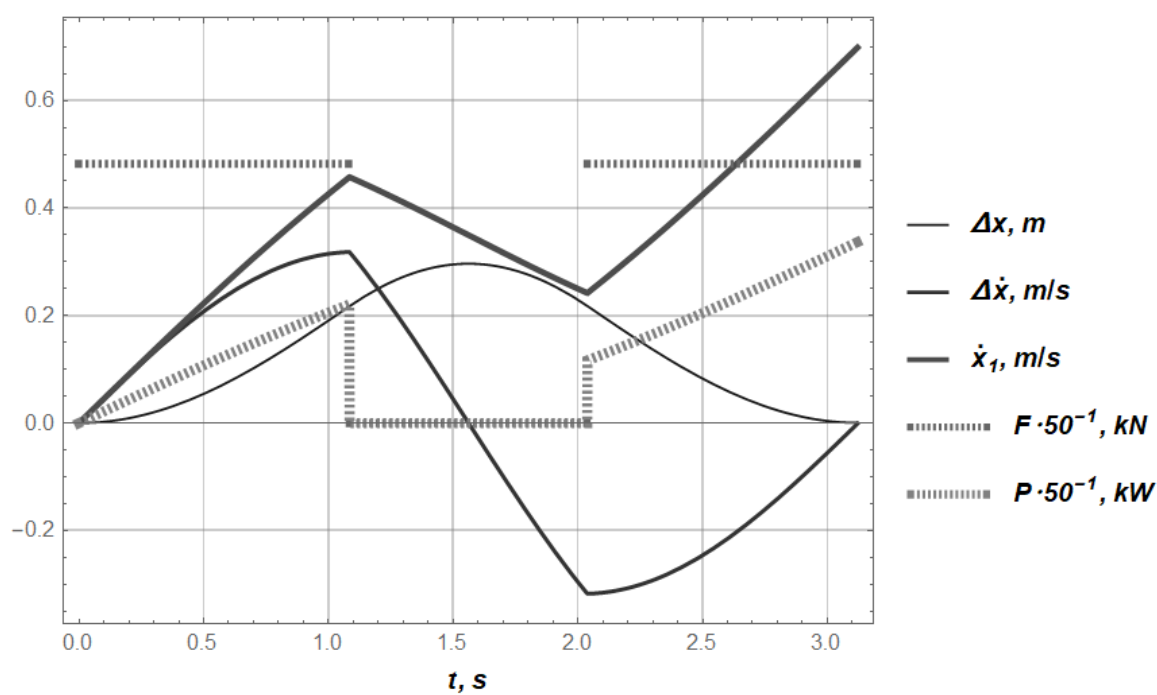
1	2	3	4	5	6	7	8	9
x_7	0,50000	0,53354	0,50030	0,51020	0,50000	0,50036	0,50101	0,50460
x_8	0,34500	0,34500	0,34500	0,34500	0,34500	0,34500	0,34500	0,34500
x_9	0,34500	0,34500	0,34500	0,34500	0,34500	0,34500	0,34500	0,34500
x_{10}	27,7324	20,3615	27,8402	27,3657	26,2747	24,6794	25,1504	26,7352
x_{11}	22,5149	21,0502	22,5437	22,3810	22,1157	21,6603	21,9113	22,2084

Значення цільової функції на найкращому розв'язку цієї задачі [99] рівне $2,284294E+01$.

Додаток Є



а)



б)

Рисунок Є.1 – Графіки характеристик системи «кран-вантаж» при оптимальному керуванні: а) розв'язок задачі типу 1; б) розв'язання задачі типу 2

Порівняльний аналіз графіків, наведених на рис. Є.1, показує розривну форму оптимального керування. Однак для розв'язання задачі Типу 2 відсутня зміна знака рушійної сили F . На практиці це бажана властивість, оскільки не вимагає зміни знака електромагнітного крутного моменту, тобто динамічні впливи, викликані розв'язком задачі Типу 2, набагато нижчі, ніж у випадку розв'язку задачі Типу 1. Максимальний розрив рушійної сили F для керування у задачі Типу 2 менший у два рази.

Постійний знак споживаної потужності – ще одна перевага розв'язку задачі Типу 2: це означає, що немає необхідності вбудовувати в привід крана додаткове силове обладнання – інверсний інвертор.

Для обох випадків досягається усталена швидкість v та маятникові коливання вантажу усуваються. Існує лише незначне (на 1,4%) збільшення T для розв'язку задачі Типу 2 порівняно з розв'язком задачі Типу 1 (рис. Є.1). Величини коливань вантажу для обох типів задач майже однакові. Однак для решти значень m_2 і l тривалість T і величина коливань вантажу змінюється. У загальному випадку це деякі функції від m_1 , m_2 , F_{\max} , F_{\min} , W , v і l .

Підводячи підсумок, можна констатувати: керування типу 2 дозволяє знизити силові та динамічні навантаження на крановий привід і металоконструкцію за рахунок незначного збільшення тривалості розгону T .

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ:

1. Goodfellow I., Bengio Yo., Courville Aa. Deep Learning. MIT Press. 2016. 800 p.
2. Zhang Y., Wang S., Ji G. A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. Mathematical Problems in Engineering. 2015. P.1–38. DOI: 10.1155/2015/931256.
3. Sibalija T. V. Particle swarm optimisation in designing parameters of manufacturing processes: A review (2008–2018). Applied Soft Computing. 2019. Vol. 84. 105743. DOI: 10.1016/j.asoc.2019.105743.
4. Landau L.D., Lifshitz E.M. Mechanics: Volume 1 (Course of Theoretical Physics S). 3rd Edition. Butterworth-Heinemann. 1976. 200 p.
5. Table of metaheuristics. URL: https://en.wikipedia.org/wiki/Table_of_metaheuristics (дата звернення 12.11.2023).
6. Metaheuristic Optimization. URL: http://www.scholarpedia.org/article/Metaheuristic_Optimization (дата звернення 12.11.2023).
7. Yang X.-Sh. Nature-Inspired Metaheuristic Algorithms. School of Science and Technology. Middlesex University London. London. Elsevier. 258 p.
8. Handbook of Metaheuristics. Second Edition. Edited by Gendreau M., Potvin J.-Yv. Springer New York Dordrecht Heidelberg London. 648 p. DOI: 10.1007/978-1-4419-1665-5.
9. Kennedy J., Eberhart R. Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks. 1995. Vol. IV. P. 1942–1948. DOI: 10.1109/ICNN.1995.488968.
10. Bratton D., Kennedy J. Defining a standard for particle swarm optimization. Proc. IEEE Swarm Intell. Symp. 2007. P. 120–127. DOI: 10.1109/SIS.2007.368035.

11. Ratnaweera A., Halgamuge S.K., Watson H.C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE transactions on evolutionary computation*. 2004. Vol. 8, Iss. 3. P. 240–255. DOI: 10.1109/tevc.2004.826071.
12. Zhan Z.H., Zhang J., Li Y., Chung HSH. Adaptive Particle Swarm Optimization. *IEEE transactions on systems man and cybernetics part B (Cybernetics)*. 2009. Vol. 39. Iss. 6. P. 1362-1381. DOI: 10.1109/TSMCB.2009.2015956.
13. Shi Y., Eberhart R.C. A modified particle swarm optimizer. *Proceedings of the IEEE Conference on Evolutionary Computation*. 1998. P. 69–73. DOI: 10.1109/ICEC.1998.699146.
14. Kennedy J., Mendes R. Population structure and particle swarm performance. *Proc. of the 2002 Congress on Evolutionary Computation*. 2002. P. 1671–1675. DOI: 10.1109/CEC.2002.1004493.
15. Tang J., Liu G., Pan Q. A Review on Representative Swarm Intelligence Algorithms for Solving Optimization Problems: Applications and Trends. *IEEE/CAA Journal of Automatica Sinica*. 2021. Vol. 8(10). P. 1627–1643. DOI: 10.1109/jas.2021.1004129.
16. Slowi A., Kwasnicka H. (2018). Nature Inspired Methods and Their Industry Applications—Swarm Intelligence Algorithms. *IEEE Transactions on Industrial Informatics*. 2018. Vol. 14(3). P. 1004–1015. DOI: 10.1109/tii.2017.2786782.
17. Yang X.-Sh. *Nature-Inspired Optimization Algorithms*. 2nd Edition. Academic Press. 2020. 687 p. ISBN 9780128219898.
18. Ezugwu A. E., Shukla A. K., Nath R., Akinyelu A. A., Agushaka J. O., Chiroma H., Muhuri P. K. Metaheuristics: a comprehensive overview and classification along with bibliometric analysis. *Artificial Intelligence Review*. 2021. Vol. 54(6). P. 4237–4316. DOI: 10.1007/s10462-020-09952-0.
19. Fan X., Sayers W., Zhang S., Han Z., Re, L., Chizari H. Review and Classification of Bio-inspired Algorithms and Their Applications. *Journal of*

- Bionic Engineering. 2020. Vol. 17(3). P. 611–631. DOI: 10.1007/s42235-020-0049-9.
20. Piotrowski A. P., Napiorkowski J. J., Piotrowska A. E. (2020). Population size in Particle Swarm Optimization. *Swarm and Evolutionary Computation*. 2020. Vol. 58. 100718. DOI: 10.1016/j.swevo.2020.100718.
 21. Eberhart R.C., Kennedy J. A new optimizer using particle swarm theory. *Proceedings of the 6th International Symposium on Micromachine Human Science*. Nagoya, Japan. IEEE. Piscataway, NJ, USA. 1995. P. 39–43.
 22. Nakisa B., Nazri M.Z.A., Rastgoo M.N., Abdullah S. A survey: Particle swarm optimization based algorithms to solve premature convergence problem. *Journal of Computer Science*. 2014. Vol. 10(10). P. 1758–1765. DOI: 10.3844/jcssp.2014.1758.1765.
 23. Romasevych Y., Loveikin V., Loveikin A. Investigation of an Influence of a Ratio „ Swarm Size - Iterations Number” on Particle Swarm Optimization Performance. 2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek). Kharkiv. Ukraine. 2022. P. 1–4, DOI: 10.1109/KhPIWeek57572.2022.9916498.
 24. Weibo W., Chuan L., Yongkang Z. Experiment and analysis of particle swarm algorithm parameters. *Journal of Xihua University*. 2008. Vol. 27(1), P. 76–80.
 25. Carlisle A., Dozier G. An Off-The-Shelf PSO. *Proceedings of the Workshop on Particle Swarm Optimization*. 2001. P. 1-6.
 26. Clerc M. *Particle Swarm Optimization*. ISTE Ltd, 2006. 243 p.
 27. Pant M., Thangaraj R., Abraham A. Particle Swarm Optimization: Performance Tuning and Empirical Analysis. In: Abraham, A., Hassanien, AE., Siarry, P., Engelbrecht, A. (eds) *Foundations of Computational Intelligence Volume 3. Studies in Computational Intelligence*, vol 203. Springer. Berlin. Heidelberg. 2009. DOI: 10.1007/978-3-642-01085-9_5

28. Li Q., Liu S.-Y., Yang X.-S. Influence of initialization on the performance of metaheuristic optimizers. *Applied Soft Computing*. 2020. Vol. 91. 106193. DOI: 10.1016/j.asoc.2020.106193.
29. Karaboga D. An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report. Technical Report-Tr06. Erciyes University. Engineering Faculty. Computer. 2005. 10 p.
30. Yang X.-S., Deb S. Cuckoo search via Lévy flights. 2009 World Congress on Nature & Biologically Inspired Computing. NaBIC. IEEE. 2009, P. 210–214. DOI: 10.1109/nabic.2009.5393690.
31. Storn R., Price K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*. 1997. Vol. 11 (4). P. 341–359. DOI: 10.1023/a:1008202821328.
32. Pal S.K., Wang P.P. *Genetic Algorithms for Pattern Recognition*. CRC Press. 2017. 336 p.
33. Isiet M., Gadala M. (2020). Sensitivity analysis of control parameters in particle swarm optimization. *Journal of Computational Science*. 2020. Vol. 41. 101086. DOI: 10.1016/j.jocs.2020.101086.
34. Poli R., Broomhead D. Exact analysis of the sampling distribution for the canonical particle swarm optimiser and its convergence during stagnation, in *Genetic and Evolutionary Computation Conference*. New York. NY. USA: ACM Press. 2007. P. 134–141.
35. Poli R. Mean and variance of the sampling distribution of particle swarm optimizers during stagnation. *IEEE Transactions on Evolutionary Computation*. 2009. Vol. 13. No. 4. P. 712–721.
36. Cleghorn C. W., Engelbrecht A. P. Particle swarm convergence: An empirical investigation. 2014 IEEE Congress on Evolutionary Computation (CEC). 2014. P.2524–2530 DOI: 10.1109/cec.2014.6900439.
37. Van den Bergh F., Engelbrecht A. A study of particle swarm optimization particle trajectories. *Information Sciences*. 2006. Vol. 176. No. 8. P. 937–971.

38. Van den Bergh F. An analysis of particle swarm optimizers. PhD thesis. Department of Computer Science. University of Pretoria. Pretoria. South Africa. 2002. 284 p.
39. Trelea I. The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Information Processing Letters*. 2003. Vol. 85. No. 6. P. 317–325.
40. Kadiramanathan V., Selvarajah K., Fleming P. Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Transactions on Evolutionary Computation*. 2006. Vol. 10. No. 3. P. 245–255.
41. Gazi V. Stochastic stability analysis of the particle dynamics in the PSO algorithm,” in *Proceedings of the IEEE International Symposium on Intelligent Control*. Dubrovnik. Croatia: IEEE Press. 2012. P. 708–713.
42. Clerc M., Kennedy J. The particle swarm: explosion, stability and convergence in a multi-dimensional space. *IEEE transactions on Evolutionary Computation*. 2002. Vol. 6. P. 58–73.
43. Ozcan E., Mohan C. K. Particle swarm optimization: surfing the waves. *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*. (Cat. No. 99TH8406). 1999. Vol. 3. P. 1939–1944 DOI: 10.1109/cec.1999.785510.
44. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей. Монографія під заг. ред. С. О. Субботіна. Запоріжжя. ЗНТУ. 2009. 375 с.
45. Jain M., Saijpal V., Singh N., Singh S.B. An Overview of Variants and Advancements of PSO Algorithm. *Applied Sciences*. 2022. Vol. 12(17). 8392. DOI: 10.3390/app12178392.
46. Kennedy J., Mendes R. Neighborhood topologies in fullyinformed and best-of-neighborhood particle swarms. 2003 IEEE international workshop on soft computing in industrial applications (SMCia/03). Binghamton. New York. USA. Oct 12–14, 2003. P. 45–50.

47. Sengupta S., Basak S., Peters R. Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives. *Machine Learning and Knowledge Extraction*. 2018. Vol. 1(1). P. 157–191. DOI: 10.3390/make1010010.
48. Kennedy J. Small Worlds and Mega-Minds: Effects of Neighbourhood Topology on Particle Swarm Performance. *IEEE Congress on Evolutionary Computation*, Washington. DC. USA. 6–9 July 1999. Vol. 3. P. 1931–1938.
49. Mendes R., Kennedy, J., Neves J. Watch thy Neighbour or How the Swarm can Learn from its Environment. *IEEE Swarm Intelligence Symposium*, Indianapolis. IN. USA. 26 April 2003. P. 88–94.
50. Liu Q., Wei W., Yuan H., Zhan Z.H., Li Y. Topology selection for particle swarm optimization. *Inf. Sci.* 2016. Vol. 363. P. 154–173.
51. Blackwell T., Kennedy J. Impact of communication topology in particle swarm optimization. *IEEE Transactions on Evolutionary Computation*. 2019. Vol. 23, No. 4. P. 689–702. DOI: 10.1109/tevc.2018.2880894.
52. Гальченко В.Я., Якимов А.Н. Популяционные метаэвристические алгоритмы оптимизации роём частиц. Учебное пособие. Черкассы: ФЛП Третьяков А. Н. 2015. 160 с.
53. Clerc M. Back to random topology. URL: http://clerc.maurice.free.fr/psorandom_topology.pdf (дата звернення 12.11.2023).
54. Dabiri A., Moghaddam B. P., Machado J. A. T. Optimal variable-order fractional PID controllers for dynamical systems. *Journal of Computational and Applied Mathematics*. 2018. Vol. 339. P. 40–48. DOI: 10.1016/j.cam.2018.02.029.
55. Khalifa B., Abdelfateh C. Optimal tuning of fractional order $P \lambda D \mu A$ controller using Particle Swarm Optimization algorithm. *IFAC-PapersOnLine*. 2017. Vol. 50(1). P. 8084–8089. DOI: 10.1016/j.ifacol.2017.08.1241.

56. Irshad M., Ali A. Robust PI-PD controller design for integrating and unstable processes. *IFAC-PapersOnLine*. 2020. Vol. 53(1). P. 135–140. DOI: 10.1016/j.ifacol.2020.06.023.
57. Belkadi A., Oulhadj H., Touati Y., Khan S. A., Daachi B. On the robust PID adaptive controller for exoskeletons: A particle swarm optimization based approach. *Applied Soft Computing*. 2017. Vol. 60. P. 87–100. DOI: 10.1016/j.asoc.2017.06.012.
58. Hashem Zadeh S. M., Khorashadizadeh S., Fateh M. M., Hadadzarif M. Optimal sliding mode control of a robot manipulator under uncertainty using PSO. *Nonlinear Dynamics*. 2016. Vol. 84(4). P. 2227–2239. DOI: 10.1007/s11071-016-2641-4.
59. Yadav N., Yadav A., Kumar M., Kim J. H. An efficient algorithm based on artificial neural networks and particle swarm optimization for solution of nonlinear Troesch's problem. *Neural Computing and Applications*. 2015. Vol. 28(1). P. 171–178. DOI: 10.1007/s00521-015-2046-1.
60. Romasevych Yu., Loveikin V., Ohiienko M., Shymko L., Łukawiecki K. Innovation management in agriculture. *Agrotronics and design of optimal controllers based on new modifications of particle swarm optimization*. Publishing House: Wyższa Szkoła Zarządzania i Administracji w Opolu. 2021. 122 p. (ISBN 978-83-66567-39-9).
61. Zheng Y., Liao Y. Parameter identification of nonlinear dynamic systems using an improved particle swarm optimization. *Optik - International Journal for Light and Electron Optics*. 2016. Vol. 127(19). P. 7865–7874. DOI: 10.1016/j.ijleo.2016.05.145.
62. Carrillo-Santos C. A., Seck-Tuoh-Mora J. C., Hernández-Romero N., Ramos-Velasco L. E. Wavenet identification of dynamical systems by a modified PSO algorithm. *Engineering Applications of Artificial Intelligence*. 2018. Vol. 73. P. 1–9. DOI: 10.1016/j.engappai.2018.04.025.
63. Mousavian S. H., Koofgar H. R. Identification-Based Robust Motion Control of an AUV: Optimized by Particle Swarm Optimization Algorithm. *Journal*

- of Intelligent & Robotic Systems. 2016. Vol. 85(2). P. 331–352. DOI: 10.1007/s10846-016-0401-9.
64. Quaranta G., Lacarbonara W., Masri S. F. A review on computational intelligence for identification of nonlinear dynamical systems. *Nonlinear Dynamics*. 2020. Vol. 9. P. 1709–1761. DOI: 10.1007/s11071-019-05430-7.
65. Suresh K., Ghosh S., Kundu D., Sen A. Inertia-adaptive particle swarm optimizer for improved global search. *Intelligent Systems Design and Applications*. 2008. Vol. 2. P. 253–258. DOI: 10.1109/ISDA.2008.199.
66. Джитгер.
<https://uk.wikipedia.org/wiki/%D0%94%D0%B6%D0%B8%D1%82%D0%B5%D1%80> (дата звернення 12.11.2023).
67. Romasevych Yu., Loveikin V., Loveikin Y. Development of a PSO Modification with Varying Cognitive Term. 2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek). Kharkiv. Ukraine. 2022. P. 1–5. DOI: 10.1109/KhPIWeek57572.2022.9916413.
68. Romasevych Yu., Loveikin V., Makarets V. Optimal Constrained Tuning of PI-Controllers via a New PSO-Based Technique, *International Journal of Swarm Intelligence Research*. 2020. Vol. 11. Issue 4. P. 87–105. DOI: 10.4018/IJSIR.2020100104.
69. Romasevych Yu., Loveikin V. A Novel Multi-Epoch Particle Swarm Optimization Technique. *Cybernetics and Information Technologies*. 2018. Vol. 18(3). P. 62–74. DOI: 10.15199/48.2019.07.08.
70. Romasevych Y., Loveikin V. Development of a PSO Modification with Soft Particle Reinitialization. 2023 IEEE 4rd KhPI Week on Advanced Technology (KhPIWeek). Kharkiv. Ukraine. 2023 - In press.
71. Romasevych Y., Loveikin V., Loveikin Y. Development of new rotating ring topology of PSO-algorithm. 2021 IEEE 2nd KhPI Week on Advanced Technology. KhPI Week 2021 - Conference Proceedings. 2021. P. 79–82. DOI: 10.1109/KhPIWeek53812.2021.9569973.

72. Romasevych Y., Loveikin V., Bakay B., Rudko I. Optimal tuning of belt conveyor soft-starter via PSO-Rot-Ring method. U.P.B. Sci. Bull. Series D. 2023. Vol. 85. Iss. 2. P. 73-84.
73. Deb's Function No.02. URL: <https://al-roomi.org/benchmarks/unconstrained/n-dimensions/232-deb-s-function-no-02> (дата звернення 12.11.2023).
74. Optimization & Eye Pleasure: 78 Benchmark Test Functions for Single Objective Optimization. URL: <https://towardsdatascience.com/optimization-eye-pleasure-78-benchmark-test-functions-for-single-objective-optimization-92e7ed1d1f12> (дата звернення 12.11.2023).
75. Virtual Library of Simulation Experiments: Test Functions and Datasets. URL: <https://www.sfu.ca/~ssurjano/optimization.html>. (дата звернення 12.11.2023).
76. HappyCat – A Simple Function Class Where Well-Known Direct Search Algorithms Do Fail. URL: https://homepages.fhv.at/hgb/New-Papers/PPSN12_BF12.pdf. (дата звернення 12.11.2023).
77. Jamil M., Yang X.-Sh. A literature survey of benchmark functions for global optimization problems. Int. Journal of Mathematical Modelling and Numerical Optimisation. 2013. Vol. 4. No. 2. P. 150–194. DOI: 10.1504/IJMMNO.2013.055204.
78. Montaz Ali M., Khompatraporn Ch., Zabinsky Z. B. A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems. Journal of Global Optimization. 2005. Vol. 31. P. 635–672. DOI 10.1007/s10898-004-9972-2.
79. Vanaret Ch., Gotteland J.-B., Durand N., Alliot J.-M. Certified Global Minima for a Benchmark of Difficult Optimization Problems. 2014. fihal-00996713. URL: <https://arxiv.org/pdf/2003.09867.pdf> (дата звернення 12.11.2023).

80. Wolpert D.H., Macready W.G. No Free Lunch Theorems for Optimization. In: IEEE Transactions on Evolutionary Computation. 1997. Vol. 1, Iss. 1, pp. 67–82. DOI: 10.1109/4235.585893.
81. Romasevych Y., Loveikin V., Bakay B. A Real-World Benchmark Problem for Global Optimization. Cybernetics and Information Technologies. 2023. Vol. 23(3). P. 23–39. DOI: 10.2478/cait-2023-0022.
82. Storn R., Price K.V. Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces, Institute of Company Secretaries of India. Chennai. Tamil Nadu. Tech. Report TR-95-012, 1995.
83. Caraffini F., Kononova A.V., Corne D. Infeasibility and structural bias in differential evolution – Information Sciences. 2019. Vol. 496. P. 161–179. DOI: 10.1016/j.ins.2019.05.019.
84. Brest J., Greiner S., Boskovic B., Mernik M., Zumer V. Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. IEEE Transactions on Evolutionary Computation. 2006. Vol. 10(6). P. 646–657. DOI: 10.1109/tevc.2006.872133.
85. Mittal N., Singh U., Sohi B.S. Modified Grey Wolf Optimizer for Global Engineering Optimization. Applied Computational Intelligence and Soft Computing. 2016. P. 1–16. DOI: 10.1155/2016/7950348.
86. Saremi S., Mirjalili S.Z., Mirjalili S.M. Evolutionary population dynamics and grey wolf optimizer – Neural Computing and Applications. 2014. Vol. 26(5). P. 1257–1263. DOI: 10.1007/s00521-014-1806-7.
87. Wang H., Wang W., Sun H., Rahnamayan S. Firefly algorithm with random attraction – International Journal of Bio-Inspired Computation. 2016. Vol. 8(1). P. 33. DOI: 10.1504/ijbic.2016.074630
88. Guo Z., Yang H., Wang S., Zhou C., Liu X. Adaptive harmony search with best-based search strategy – Soft Computing. 2016. Vol. 22(4). P. 1335–1349. DOI: 10.1007/s00500-016-2424-3
89. Kumar V., Chhabra J. K., Kumar D. Parameter adaptive harmony search algorithm for unimodal and multimodal optimization problems. Journal of

- Computational Science. 2014. Vol. 5(2). P. 144–155. DOI: 10.1016/j.jocs.2013.12.001
90. Naik M., Nath M. R., Wunnava A., Sahany S., Panda R. A new adaptive Cuckoo search algorithm. 2015 IEEE 2nd International Conference on Recent Trends in Information Systems. 2015. P. 1–5. DOI: 10.1109/retis.2015.7232842
91. Shi Cheng, Yuhui Shi. Diversity control in particle swarm optimization. In: 2011 IEEE Symposium on Swarm Intelligence. 11-15 April 2011. INSPEC Accession Number: 12134332. Conference Location: Paris, France. Publisher: IEEE. DOI: 10.1109/SIS.2011.5952581
92. S. Mirjalili, S. M. Mirjalili, A. Lewis. Grey Wolf Optimizer. *Advances in Engineering Software*. 2014. Vol. 69, p. 46-61. doi: 10.1016/j.advengsoft.2013.12.007
93. R. Mendes, J. Kennedy, J. Neves Watch thy Neighbour or How the Swarm can Learn from its Environment. Presented at the IEEE Swarm Intelligence Symposium, Indianapolis. IN. USA. 2003. p. 88-94
94. P. N. Suganthan. Particle swarm optimiser with neighbourhood operator. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406). doi: 10.1109/cec.1999.785514
95. J. J. Liang, A. K. Qin, P. N. Suganthan, S. Baskar. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *Evolutionary Computation*, IEEE Transactions, 2006. vol. 10(3), pp. 281-295
96. Åström K. J., Hägglund T. Benchmark Systems for PID Control. International Federation of Automatic Control. 2000. P. 165–166.
97. Synthesis of advanced automatic control systems. Monograph / Yuriy Romasevych, Viatcheslav Loveikin, Alla Dudnyk, Vitaliy Lysenko, Natalia Zaets. Kõima. – 2020. – 140 p. ISBN 978-9916-4-0282-5. URL: http://www.ester.ee/record=b5392805~S1*est

98. Amir Hossein Gandomi, Xin-She Yang, Amir Hossein Alavi. Mixed variable structural optimization using Firefly Algorithm. *Computers and Structures* 89 (2011) 2325–2336. [doi: 10.1016/j.compstruc.2011.08.002](https://doi.org/10.1016/j.compstruc.2011.08.002)
99. Hadi Bayzidi, Siamak Talatahari, Meysam Saraee, Charles-Philippe Lamarche. Social Network Search for Solving Engineering Optimization Problems. Hindawi. *Computational Intelligence and Neuroscience*. Volume 2021, Article ID 8548639, 32 pages. [doi: 10.1155/2021/8548639](https://doi.org/10.1155/2021/8548639)
100. C.A.C. Coello. Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, Vol. 41, No. 2, pp. 113–127, 2000.
101. Kaveh A., Dadras A. A novel meta-heuristic optimization algorithm: Thermal exchange optimization. *Advances in Engineering Software* 110 (2017) 69–84. [doi: 10.1016/j.advengsoft.2017.03.014](https://doi.org/10.1016/j.advengsoft.2017.03.014)
102. H. Nowcki, Optimization in pre-contract ship design. In *Computer Applications. In the Automation of Shipyard Operation and Ship Design* in: Y. Fujita, K. Lind and T. J. Williams, Eds. North- Holland, Elsevier, New York, vol. 2, 1974, pp. 327-338
103. Gandomi, A. H., & Roke, D. A. Engineering optimization using interior search algorithm. 2014 IEEE Symposium on Swarm Intelligence. 2014. [doi: 10.1109/sis.2014.7011771](https://doi.org/10.1109/sis.2014.7011771)
104. E. Sandgren. Nonlinear integer and discrete programming in mechanical design optimization. *Journal of Mechanical Design*, vol. 112, no. 2, pp. 223–229, 1990.
105. H. Chickermane and H. C. Gea. Structural optimization using a new local approximation method. *International Journal for Numerical Methods in Engineering*, vol. 39, no. 5, pp. 829–846, 1996
106. A. Ravindran, K. M. Ragsdell, G. V. Reklaitis, *Engineering Optimization*, John Wiley & Sons, Inc., Hoboken, NJ, USA, 2006
107. B. Hossain, T. Hasegawa. Nonlinear mixed-discrete structural optimization. *Journal of Structural Engineering*, vol. 115, pp. 626–646, 1989

108. Youn B.D., Choi K.K. A new response surface methodology for reliability-based design optimization. *Computers and Structures* 2004; vol. 82, no. 2-3: 241–256.
109. Bassi, S.J., Gbenga, E.D., Abidemi, A., Opeoluwa Oyewola, D., Khammas B.M. Metaheuristic algorithms for PID controller parameters tuning: review, approaches and open problems – *Heliyon*, Vol. 8, Issue 5, 2022, e09399. doi: 10.1016/j.heliyon.2022.e09399
110. Toshev, A. Particle Swarm Optimization and Tabu Search Hybrid Algorithm for Flexible Job Shop Scheduling Problem – Analysis of Test Results – *Cybernetics and information technologies*, Vol. 19, No 4, 2019, 26-44. doi: 10.2478/cait-2019-0034
111. Stoilova, K., Stoilov, T., Dimitrov, S. Bi-Level Optimization Model for Urban Traffic Control – *Cybernetics and information technologies*, Vol. 21, No 3, 2021, 108-126. doi: 10.2478/cait-2021-0033
112. Yadav, D. Blood coagulation algorithm: a novel bio-inspired meta-heuristic algorithm for global optimization – *Mathematics*, Vol 9, 2021, 3011. doi: 10.3390/math9233011
113. Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V., Mishchenko, E.F. *Mathematical Theory of Optimal Processes* [in Russian], Nauka, Moscow, 1969.
114. Oleinikov, V.A., Zotov, N.S., Pryshvin, A.M. *Basics of optimal and extremal control* [in Russian], Vyschaya shkola, 1969.
115. Romacevych, Y., Loveikin, V., Stekhno, O. Closed-loop optimal control of a system Trolley – Payload – *UPB Scientific Bulletin, Series D: Mechanical Engineering*, Vol 81(2), 312, 2019, 3-12
116. Boom, T.V., Klaassens, J., Meiland, R. Real-Time Time-Optimal Control for a Nonlinear Container Crane Using a Neural Network. In *Informatics in Control, Automation and Robotics II*. Springer, Dordrecht, 2007, 79-84. doi: 10.1007/978-1-4020-5626-0_10

117. Ermidoro, M., Formentin, S., Cologni, A., Previdi, F., Savaresi, S.M. On time-optimal anti-sway controller design for bridge cranes. In 2014 American Control Conference, 2014, 2809-2814. doi: 10.1109/acc.2014.6858939
118. Loveikin, V.S., Romasevich, Yu. A., Khoroshun, S.A., Shevchuck, A.G. Time-optimal control of a simple pendulum with a movable pivot. Part 1 – International Applied Mechanics, Vol. 54, No. 3, 2018, 358–365. doi: 10.1007/s10778-018-0887-x
119. Loveikin, V.S., Romasevich, Yu.A., Khoroshun, S.A., Shevchuck, A.G. Time-optimal control of a simple pendulum with a movable pivot. Part 2 – International Applied Mechanics, Vol. 56, No. 2, 2020, 208–215. doi: 10.1007/s10778-020-01007-9
120. Da Cruz, J.J., Leonardi, F. Minimum-time anti-swing motion planning of cranes using linear programming. – Optimal Control Applications and Methods, 34(2), 2012, 191–201. doi: 10.1002/oca.2016
121. Chen, H., Fang, Y., Sun, N. A swing constrained time-optimal trajectory planning strategy for double pendulum crane systems – Nonlinear Dynamics, Vol. 89(2), 2017, 1513–1524. doi: 10.1007/s11071
122. CVX: Matlab Software for Disciplined Convex Programming. <http://cvxr.com/cvx/>
123. Yiming, W., Ning, S., He C., Jianyi, Z., Yongchun, F. Nonlinear time-optimal trajectory planning for varying rope-length overhead cranes – Assembly Automation, Vol. 35, no 5, 2018, 587-594. doi: 10.1108/AA-12-2017-183
124. Rao, A.V.; Benson, D.A., Darby, C.L., Patterson, M.A., Francolin, C., Sanders, I., Huntington, G. T. GPOPS: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using the Gauss Pseudospectral Method – ACM Transactions on Mathematical Software. 37 (2): 22:1, 2010, 22:39. doi: 10.1145/1731022.1731032. S2CID 15375549
125. Wu, Q., Wang, X., Hua, L., Xia, M. Dynamic analysis and time optimal anti-swing control of double pendulum bridge crane with distributed mass beams

- Mechanical Systems and Signal Processing, Vol. 144, 2020, 106968. doi: 10.1016/j.ymssp.2020.106968
126. Smekhov, A.A., Erofeev, N.I. Optimal Control of Carrying-and-Lifting Machines [in Russian], Mashinostroenie, Moscow, 1975.
127. Yoshida, Y. Feedback Control and Time-Optimal Control about Overhead Crane by Visual Servo and These Combination. In book Control – Intelligent Mechatronics, 2011. doi: 10.5772/15198
128. Gerasimiak, R.P., Lestchev, V.A. Analysis and synthesis of crane electromechanical systems [in Russian], SMIL, Odessa, 2008
129. Perelmuter, M.M., Poliakov, L.N. Load oscillation elimination, which is suspended to a crane trolley, via its drive control [in Russian] – *Isvestia vuzov. Electromechanica*, № 7, 1971, 769-774.
130. Brand, Z., Cole, M.O.T. Mini-max optimization of actuator/sensor placement for flexural vibration control of a rotating thin-walled cylinder over a range of speeds – *Journal of Sound and Vibration*, 506, 2021, 116105. doi: 10.1016/j.jsv.2021.116105
131. Li S., Zou Y., Lai X., Liu Z., Wang X. Performance-maximum optimization of the intelligent lifting activities for a polar ship crane through trajectory planning. *Proceedings of the Institution of Mechanical Engineers. Part C. Journal of Mechanical Engineering Science*. 2023 Vol. 4. Pp 765-781. DOI: [10.1177/0954406222111117](https://doi.org/10.1177/0954406222111117)
132. Montonen, J.-H., Nevaranta N., Niemelä, M., Lind, T. Comparison of Extrainsensitive Input Shaping and Swing-Angle- Estimation-Based Slew Control Approaches for a Tower Crane. *Appl. Sci*. 2022. Vol 12. P. 5945. DOI: [10.3390/app12125945/](https://doi.org/10.3390/app12125945/).
133. Englert T. Lastpendeldämpfung für Turmdrehkrane. 30. Internationale Kranfachtagung 2022. *Kran 4.0. Erfolge der Digitalisierung* Herausgegeben als Begleitband zur gleichnamigen Fachtagung am 11. und 12. Juli 2022 in Magdeburg. Pp. 41-49.

134. Kostikov A.A., Perig A.V., Larichkin O.V., Stadnik A.N., Gribkov E.P. Research Into Payload Swaying Reduction Through Cable Length Manipulation During Boom Crane Motion. FME Transactions, 2019, 47, P. 464-476. DOI: [10.5937/fmet1903464K/](https://doi.org/10.5937/fmet1903464K/).
135. Qian Y., Fang Y., Lu B. Adaptive repetitive learning control for an offshore boom crane. Automatica. 2017. Vol. 82. №. 8. Pp. 21–28. DOI: [10.1016/j.automatica.2017.04.003](https://doi.org/10.1016/j.automatica.2017.04.003)
136. He J., Chen Y., Wu K., Zhao Y., Wang Z., Chen Z. Energy flow analysis of crane hoisting system and experimental of potential energy. Journal of Jilin University (Engineering and Technology Edition). 2018. Vol. 48. Issue 4. Pp. 1106-1113. DOI: [10.13229/j.cnki.jdxbgxb20170539](https://doi.org/10.13229/j.cnki.jdxbgxb20170539)
137. Sun N., Wu Y., Chen H., Fang Y. An energy-optimal solution for transportation control of cranes with double pendulum dynamics: Design and experiments. Mechanical Systems and Signal Processing. Vol. 102. 2018. Pp. 87–101. DOI: [10.1016/j.ymsp.2017.09.027](https://doi.org/10.1016/j.ymsp.2017.09.027)
138. Ho T., Suzuki K., Tsume M., Tasaki R., Miyoshi T., Terashima K. A switched optimal control approach to reduce transferring time, energy consumption, and residual vibration of payload's skew rotation in crane systems. Control Engineering Practice. 2019. Vol. 84. Pp. 247–260. DOI: [10.1016/j.conengprac.2018.11.018](https://doi.org/10.1016/j.conengprac.2018.11.018)
139. Rams H., Schöberl M., Schlacher K. Optimal Motion Planning and Energy-Based Control of a Single Mast Stacker Crane. IEEE Transactions On Control Systems Technology. 2017. Vol. 26. Issue 4. 1449–1457. DOI: [10.1109/TCST.2017.2710953](https://doi.org/10.1109/TCST.2017.2710953)
140. Ловейкін В.С., Ромасевич Ю.О., Стехно О.В. Оптимізація режиму пуску механізму зміни вильоту вантажу баштового крана. Підйомно-транспортна техніка. Одеса, 2020. Вип. 1 (62), С. 4-18. DOI: [10.15276/pidtt.1.62.2020.01](https://doi.org/10.15276/pidtt.1.62.2020.01)
141. Ловейкін В.С., Ромасевич Ю.О., Стехно О.В. Аналіз динаміки зміни вильоту вантажу баштового крана. Науковий вісник Національного

- університету біоресурсів і природокористування України. Серія: техніка та енергетика АПК. Київ, 2018. Вип. 282. С. 74-87.
142. Ловейкін В.С., Ромасевич Ю.О., Стехно О.В. Оптимізація режимів зміни вильоту вантажу баштового крана. Монографія. Київ. ЦП «КОМПРИНТ». 2017. 172 с.
143. O. Grigorov, E. Druzhynin, G. Anishchenko, M. Strizhak, V. Strizhak. Analysis of various approaches to modeling of dynamics of lifting-transport vehicles. *International Journal of Engineering and Technology (UAE)*. (2018). Vol. 7(4). Pp. 64-70. DOI: 10.14419/ijet.v7i4.3.19553.
144. Ловейкін В.С., Ромасевич Ю.О., Стехно О.В., Муштин Д.І. Наукове обґрунтування і розробка методів динамічного моделювання та режимно-параметричної оптимізації сучасних вантажопідйомних машин. Монографія. ЦП „КОМПРИНТ”. Київ. 2023. 458 с.
145. Ловейкін В.С., Ромасевич Ю.О. Динамічна оптимізація механізму підйому вантажу. Монографія. Київ. ЦП «КОМПРИНТ». 2015. 197 с.
146. Kim A., Kim J. Dynamic analysis and control of an overhead crane with multiple hoists using a sliding mode control approach // *J. of Mechanical Sci. and Technology*. – 2021. – 35, N 8. – P. 4055 – 4065.
147. Lee J.W., Kim D.H. Dynamic analysis of a tower crane using multibody system simulation // *J. of Mechanical Sci. and Technology*. – 2016. – 30, N 8. – P. 3475 – 3481.
148. Lee S.W., Lee J.W. Dynamic analysis of a luffing jib tower crane using Kane's method // *Proc. of the Institution of Mechanical Engineers, Part K: J. of Multi-body Dynamics*. – 2017. – 231, N 3. – P. 435 – 448
149. Wu M., Li L., Li Y., Dynamic analysis of a container crane considering the coupling effect between spreader and cargo // *J. of Vibroengineering*. – 2019. – 21, N 2. – P. 360 – 373.
150. Xue G., Wu Y., Cai L., Dynamic analysis and control of a cable crane with dual winches // *J. of Sound and Vibration*. – 2013. – 332, N 12. – P. 2937 – 2956

151. Sakawa Y., Nakazumi A. Modeling and Control of a Rotary Crane // *J. of Dynamic Systems. Measurement and Control.* – 1985. – 107, N 3. – P. 200.
152. Герасимьяк Р.П., Лещев В.А. Анализ и синтез крановых электромеханических систем. – Одесса: СМІЛ, 2008. – 192 с.
153. Герасимьяк Р.П., Найденко О.В. Особливості керування електроприводом механізму вильоту стріли під час обертання крана з підвішеним вантажем // *Електромашинобудування та електрообладнання.* – 2007. – 68. – С. 11 – 15.
154. Найденко Е.В. Управление электроприводом механизмов горизонтального перемещения с подвешенным грузом // *Електромашинобудування та електрообладнання.* – 2007. – 69. – С. 17 – 22.
155. Lee H.P. Dynamic responses of a beam with a moving mass // *J. of Sound and Vibration.* – 1996. – 191, N 2. – P. 289 – 294.
156. Oguamanam D.C.D., Hansen J.S., Heppler G.R. Dynamics of a three-dimensional overhead crane system// *J. of Sound and Vibration.* – 2001. – 242, N 3. – P. 411 – 426.
157. Qian Y., Fang Y. Switching Logic-Based Nonlinear Feedback Control of Offshore Ship-Mounted Tower Cranes: A Disturbance Observer-Based Approach // *IEEE Transactions on Automation Science and Engineering.* – 2018. – P. 1 – 12.
158. Sun G., Kleeberger M. Dynamic responses of hydraulic mobile crane with consideration of the drive system // *Mechanism and Machine Theory.* – 2003. – 38, N 12. – P. 1489 – 1508.
159. Loveikin V., Romasevych Y., Loveikin A., Shymko L., Liashko A. Minimization of The Drive Torque of The Trolley Movement Mechanism During Tower Crane Steady Slewing // *J. of Theoretical and Appl. Mech. (Bulgaria).* – 2023. – 53, N 1. – P. 19 – 33.

160. Loveikin V., Romasevych Y., Kadykalo I., Liashko A. Optimization of the swinging mode of the boom crane upon a complex integral criterion // *J. of Theoretical and Appl. Mech. (Bulgaria)*. – 2019. – 49, N 3. – P. 285 – 296.
161. Loveikin V., Romasevych Y., Loveikin A., Lyashko A., Korobko M. Minimization of high-frequency oscillations of trolley movement mechanism during steady tower crane slewing // *UPB Scientific Bulletin, Series D: Mechanical Engineering*. – 2022. – 84, N 1. – P. 31 – 44.
162. Loveikin V.S., Romasevich Yu.A., Loveikin A.V., Khoroshun A.S. Optimizing the Start of the Trolley Mechanism during Steady Slewing of Tower Crane // *Int. Appl. Mech.* – 2022. – 58, N 5. – P. 594 – 604.
163. Loveikin V.S., Romasevych Y.O., Loveikin A.V., Korobko M.M. Optimization of the trolley mechanism acceleration during tower crane steady slewing // *Archive of Mechanical Eng.* – 2022. – 69, N 3. – P. 411 – 429.
164. Chen J., Li M., Li Z. Optimal control strategy for minimizing sway motion of a crane's payload // *IEEE Transactions on Control Systems Technology*. – 2016. – 24, N 2. – P. 570 – 577.
165. Li L., Wang X., Zhang J. Optimal trajectory planning and control of a container crane considering dynamic effects // *J. of Mechanical Sci. and Technology*. – 2014 – 28, N 1. – 237 – 248.
166. Li Z., Wang J., Zhang D. Optimal trajectory planning for container cranes with anti-sway control // *IEEE Transactions on Automation Sci. and Engng.* – 2018. – 15, N. 1. – P. 44 – 56.
167. Sun H., Chen X., Ma J. Optimal motion planning of overhead cranes with variable-length cable // *IEEE/ASME Transactions on Mechatronics*. – 2018. – 23, N. 2. – P. 795 – 804.
168. Zhang S., Zhou J., Zhao Y. Optimal control of crane systems within put saturation and state constraints // *Mechanical Systems and Signal Processing*. – 2019. – 117. – P. 141 – 158.

169. A.N. Sharkawy, P. Koustoumpardis. Dynamics and Computed-Torque Control of a 2-DOF manipulator: Mathematical Analysis, *International Journal of Advanced Science and Technology*, 2019, 28 (12), pp. 201-212
170. F. Massaoudi, D. Elleuch, T. Damak. Robust Control for a Two DOF Robot Manipulator, *Journal of Electrical and Computer Engineering*, 2019, pp. 1-11. doi: 10.1155/2019/3919864
171. A. Doroudchi, R. Khodambashi, M. Sharifzadeh, D. Li, S. Berman, D.M. Aukes. Tracking Control of a Miniature 2-DOF Manipulator With Hydrogel Actuators, *IEEE Robotics and Automation Letters*, 2021, 6(3), pp. 4774-4781. doi: 10.1109/lra.2021.3067622
172. M. Esfandiari, S. Chan, G. Sutherland, D. Westwick. Nonlinear Model Predictive Control of Robot Manipulators Using Quasi-LPV Representation, 2019 7th International Conference on Control, Mechatronics and Automation (ICCMA). 2019, pp. 116-120. doi: 10.1109/iccma46720.2019.8988747
173. J.A.G. Luz Junior, A.M. Tusset, M.A. Ribeiro, J.M. Balthazar. Optimal Control for Path Planning on a 2 DOF Robotic Arm with Prismatic and Revolute Elastic Joints, *Vibration Engineering and Technology of Machinery. Mechanisms and Machine Science*, 2021, Vol 95, pp. 209-218. doi: 10.1007/978-3-030-60694-7_13
174. A.T. Azar, H.H. Ammar, M.Y. Beb, S.R. Garces, A. Boubakari. Optimal Design of PID Controller for 2-DOF Drawing Robot Using Bat-Inspired Algorithm, *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2019. AISI 2019. Advances in Intelligent Systems and Computing*, 2020, Vol 1058, pp. 175-186. doi: 10.1007/978-3-030-31129-2_17
175. A.T. Sadiq, F.A. Raheem, N. Alhuda, F. Abbas. Robot Arm Path Planning Using Modified Particle Swarm Optimization based on D* algorithm, *Al-Khwarizmi Engineering Journal*, 2017, Vol. 13, No. 3, pp. 27-37. doi: 10.22153/kej.2017.02.001

176. Yu. Romasevych, V. Loveikin. 2-DOF Robot Optimal Control via Artificial Neural Network Reinforcement Learning, 2023 IEEE 4th KhPI Week on Advanced Technology (KhPIWeek). Kharkiv. Ukraine. 2023. pp. 1-5, doi: 10.1109/KhPIWeek61412.2023.10312842
177. V.S. Loveikin, Y.O. Romasevich, A.V. Loveikin, A.S. Khoroshun, M.M. Korobko. Minimizing the Driving Torque of Tower Crane Slewing Mechanism During Steady Trolleying, *International Applied Mechanics*, 2023, Vol. 59(6), pp. 695-707. doi: 10.1007/s10778-024-01252-2
178. Talatahari S., Azizi M., Gandomi, A.H. Material Generation Algorithm: A Novel Metaheuristic Algorithm for Optimization of Engineering Problems. *Processes* 2021, 9, 859. doi: [10.3390/pr9050859](https://doi.org/10.3390/pr9050859)
179. Mahdavi M, Fesanghary M, Damangir E. An improved harmony search algorithm for solving optimization problems. *Appl Math Comput* 2007;188:1567–79
180. Kaveh A, Bakhshpoori T. Water evaporation optimization: a novel physically inspired optimization algorithm. *Comput Struct* 2016; 167: 69–85
181. Eskandar H. Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct* 2012;110–111:151–66.
182. Gandomi A. Bat algorithm for constrained optimization tasks. *Neural Comput Appl* 2013;22(6):1239–55
183. Wang L, Li L-P. An effective differential evolution with level comparison for constrained engineering design. *Struct Multidiscipl Optim* 2010;41:947–63.
184. Gong Wenyin, Cai Zhihua, Liang Dingwen. Engineering optimization by means of an improved constrained differential evolution. *Comput Methods Appl Mech. Eng* 2014;268:884–904 1 January
185. Zhang M, Luo W, Wang X. Differential evolution with dynamic stochastic selection for constrained optimization. *Inf Sci* 2008;178:3043–74.

186. Wang Y, Cai Z, Zhou Y, Fan Z. Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint handling technique. *Struct Multidisc Optim* 2009;37:395–413
187. J. Tsai. Global optimization of nonlinear fractional programming problems in engineering design, *Eng. Optimiz.* vol. 37, no. 4, 2005, pp. 399-409.
188. Talatahari S., Azizi M., Gandomi, A.H. Material Generation Algorithm: A Novel Metaheuristic Algorithm for Optimization of Engineering Problems. *Processes* 2021, 9, 859. [doi: 10.3390/pr9050859](https://doi.org/10.3390/pr9050859)
189. Amir Hossein Gandomi, Xin-She Yang, Amir Hossein Alavi. Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with Computers* (2013) 29:17–35. doi: 10.1007/s00366-011-0241-y
190. A. Kaveh and A. Dadras Eslamlou. Water strider algorithm: a new metaheuristic and applications. *Structures*, vol. 25, pp. 520–541, 2020
191. S. Mirjalili. Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.
192. S. Mirjalili. The ant lion optimizer. *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.
193. B. Akay and D. Karaboga. Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of Intelligent Manufacturing*, vol. 23, no. 4, pp. 1001–1014, 2012.
194. X. Han, L. Yue, Y. Dong, Q. Xu, G. Xie, and X. Xu. Efficient hybrid algorithm based on moth search and fireworks algorithm for solving numerical and constrained engineering optimization problems. *Journal of Supercomputing*, vol. 76, pp. 9404–9429, 2020.
195. A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi. Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing*, vol. 13, no. 5, pp. 2592–2612, 2013.

196. S. Mirjalili, S. M. Mirjalili, and A. Hatamlou. Multi-Verse Optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications*, vol. 27, no. 2, pp. 495–513, 2016.
197. S. N. Chegini, A. Bagheri, and F. Najafi. PSOSCALF: a new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems. *Applied Soft Computing*, vol. 73, pp. 697–726, 2018.
198. M.-Y. Cheng and D. Prayogo. Symbiotic Organisms Search: a new metaheuristic optimization algorithm. *Computers & Structures*, vol. 139, pp. 98–112, 2014.
199. S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, S. M. Mirjalili. Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.

ЗМІСТ

Вступ.....	3
1 Короткий аналіз досліджень за темою PSO.....	6
1.1 Кількісний аналіз.....	6
1.2 Контент-аналіз окремих наукових праць.....	14
1.2.1 Налаштування базових параметрів PSO	14
1.2.2 Застосування PSO у задачах керування та ідентифікації динамічних систем.....	20
Висновки до першого розділу.....	25
2 Розробка та математичний опис модифікацій PSO.....	27
2.1 Постановка задачі оптимізації.....	27
2.2 Відомі оптимізаційні алгоритми, на основі яких виконується розробка нових модифікацій PSO.....	27
2.2.1 Алгоритм PSO.....	27
2.2.2 Алгоритм IA-PSO.....	29
2.2.3 Алгоритм VCT-PSO.....	30
2.2.4 Алгоритм Ring-PSO.....	31
2.2.5 Алгоритм ME-D-PSO.....	32
2.3 Розробка нових модифікацій PSO.....	33
2.3.1 Алгоритми PV-IA-PSO.....	33
2.3.2 Алгоритм SR-D-PSO.....	36
2.3.3 Алгоритм M-PSO-Hybrid (гібридизація модифікацій PSO).....	37
2.3.4 Алгоритм LG-Ring-PSO.....	41
2.4 Інші модифікації PSO, що використані в дослідженні.....	43
Висновки до другого розділу.....	46
3 Опис тестових функцій.....	48
Висновки до третього розділу.....	65

4	Обґрунтування чисельних значень параметрів розроблених модифікацій PSO.....	66
4.1	Методика визначення оптимальних значень параметрів розроблених модифікацій PSO.....	66
4.2	Визначення оптимального значення параметра σ_{\max} алгоритма PV-IA-PSO-1.....	67
4.3	Визначення оптимального значення параметра B алгоритма PV-IA-PSO-2.....	74
4.4	Визначення оптимального значення параметра ψ алгоритма M-PSO-Hybrid.....	78
4.5	Визначення оптимального значення параметра a алгоритма LG-Ring-PSO.....	83
	Висновки до четвертого розділу.....	87
5	Розробка оцінок ефективності роботи модифікацій PSO.....	88
5.1	Класи критеріїв оцінки роботи алгоритмів.....	88
5.2	Клас статичних оцінок.....	90
5.2.1	Відносний успіх алгоритму.....	91
5.2.2	Відносна кількість успішних запусків алгоритму.....	95
5.2.3	Клас статично-динамічних оцінок (показник залученості частинок у пошук розв'язку задачі).....	97
5.3	Клас динамічних оцінок.....	102
5.3.1	Зміна базових показників рою за ітераціями.....	102
5.3.2	Рух частинок по контурній поверхні функції.....	105
	Висновки до п'ятого розділу.....	111
6	Проведення комплексу чисельних експериментів та статистична оцінка отриманих даних	112
6.1	Умови проведення чисельних експериментів щодо оптимізації тестових функцій.....	112
6.2	Оцінка результатів чисельних експериментів.....	112

	374
6.3 Аналіз різноманітності рою.....	114
6.4 Оцінювання за додатковими показниками.....	121
6.4.1 Вибір і розробка додаткових показників.....	121
6.4.2 Оцінка роботи алгоритмів за числовими значеннями додаткових показників.....	124
6.5 Дослідження впливу ступеня зв'язності рою алгоритмів l-best PSO на їхню ефективність.....	126
6.5.1 Топології зв'язків PSO l-best та g-best.....	127
6.5.2 Постановка чисельного експерименту.....	129
6.5.3 Визначення впливу J на ефективність RSTC _{J/K} -PSO...	130
6.5.4 Визначення впливу структури зв'язків частинок на ефективність RSTC _{J/K} -PSO.....	133
Висновки до шостого розділу.....	137
7 Застосування PSO-алгоритмів для задач оптимального керування динамічних систем	138
7.1 Постановка лінійних задач (налаштування ПІ-регуляторів)...	138
7.2 Постановка нелінійних задач.....	142
7.3 Розробка оператора керування (структура ШНМ).....	150
7.4 Результати розв'язання задач оптимального керування.....	151
7.4.1 Лінійні задачі.....	151
7.4.2 Нелінійні задачі.....	154
Висновки до сьомого розділу.....	157
8 Застосування PSO-алгоритмів для розв'язання задач інженерії.....	158
8.1 Постановка задач оптимізації інженерних конструкцій.....	158
8.1.1 Зварна балка.....	158
8.1.2 Ємність під тиском.....	159
8.1.3 Пружина.....	160
8.1.4 Дерев'яна ферма.....	161
8.1.5 Редуктор.....	162

	375
8.1.6 Зубчаста передача.....	164
8.1.7 Консольна балка.....	165
8.1.8 Двотаврова балка.....	166
8.1.9 Гофрована перегородка.....	167
8.1.10 Залізобетонна балка.....	168
8.1.11 Трубчаста колона.....	169
8.1.12 Металоконструкція автомобіля при дії бокового удару.....	170
8.2 Аналіз результатів застосування PSO-алгоритмів.....	172
8.3 Задача глобальної оптимізації (визначення оптимального за швидкодією керування).....	174
8.3.1 Постановка та зведення задачі глобальної оптимізації	177
8.3.2 Короткий аналіз цільової функції.....	181
8.4 Аналіз результатів розв'язання задачі глобальної оптимізації.....	185
Висновки до восьмого розділу.....	190
9 Застосування PSO-алгоритмів для розв'язання задач керування рухом механізмів вантажопідйомних машин та робототехніки.....	191
9.1 Параметрична оптимізація частотно-керованого пуску динамічної системи «візок-вантаж» механізму зміни вильоту баштового крана.....	191
9.2 Оптимізація за енергетичним критерієм сумісного пуску механізмів зміни вильоту та повороту баштового крану.....	204
9.2.1 Постановка задачі оптимального керування.....	207
9.2.2 Розв'язання задачі.....	211
9.2.3 Аналіз результатів розв'язання задачі.....	222
9.3 Планування оптимальної траєкторії руху дволанкового маніпулятора.....	227
9.3.1 Постановка задачі.....	229

	376
9.3.2 Розв'язання задачі.....	232
9.3.3 Короткий аналіз оптимального керування.....	235
Висновки до дев'ятого розділу.....	240
Висновки.....	242
Додатки.....	245
Перелік джерел посилання.....	351

Ромасевич Ю.О.
Ловейкін В.С.

МОДИФІКАЦІЇ МЕТОДУ ОПТИМІЗАЦІЇ PSO В ІНЖЕНЕРІЇ

НАУКОВЕ ВИДАННЯ