КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (30), 2025
Спеціальний випуск ISSN 2663 – 4023
Special issue
DOI 10.28925/2663-4023.2025.30.983
УДК 004.056:004.75

Цирканюк Діана Андріївна
аспірантка кафедри інформаційної та
кібернетичної безпеки імені професора Володимира Бурячка
Київський столичний університет імені Бориса Грінченка, м. Київ, Україна
ORCID: 0000-0002-9422-8617
d.tsyrkaniuk.asp@kubg.edu.ua

МЕТОД БАГАТОКРИТЕРІАЛЬНОЇ ОПТИМІЗАЦІЇ БЕЗПЕКИ ХМАРНИХ ОБЧИСЛЕНЬ НА ОСНОВІ МОДИФІКОВАНОГО АЛГОРИТМУ NSGA-II

Анотація. Ця стаття присвячена розробці та аналізу розширеної гібридної моделі для захисту хмарного середовища, яка інтегрує еволюційну оптимізацію, теорію ігор та кооперативні стратегії з урахуванням динамічних ризиків. Запропонований метод CoopEvo-CloudSec базується на модифікації алгоритму NSGA-II, що враховує чотири ключові критерії: рівень безпеки, час обробки завдань, вартість ресурсів та коаліційну вигоду від спільного захисту. Мета дослідження полягає в створенні ефективного механізму розподілу обчислювальних ресурсів у хмарних системах, який би оптимізував продуктивність та безпеку в умовах варіативних ризиків. Для цього використано гібридний підхід, що поєднує теоретичні моделі «зловмисник-захисник» з еволюційними алгоритмами та кооперативними стратегіями, заснованими на значенні Шеплі для оцінки внеску кожного вузла в коаліцію захисників. Обчислювальний експеримент проведено в середовищі PyCharm з використанням синтетичних та реальних датасетів. Результати експерименту демонструють перевагу запропонованої моделі над базовим NSGA-II та іншими варіантами. Зокрема, метод CoopEvo-CloudSec досягає кращого балансу між метриками якості, такими як нормалізований гіпероб'єм (1,0145), рівномірність розподілу рішень та різноманітність (Diversity >1,2), при помірному часі виконання (70–80 с). Порівняльний аналіз на радарних діаграмах, boxplots та паралельних координатах підтверджує здатність моделі формувати Парето-оптимальні рішення, адаптовані до різних сценаріїв загроз, включаючи динамічні зміни індикатора ризику $\lambda(t)$. Статистичні тести (ANOVA та Тьюкі) доводять значущість відмінностей, з коефіцієнтом варіації 14,07% для стабільності. Метод має практичне значення для провайдерів хмарних послуг, оскільки сприяє реалізації Zero Trust архітектур та гібридних криптографічних схем (наприклад, AES+ECC). Загалом, дослідження пропонує інноваційний інструмент для управління ризиками в хмарних середовищах, що відповідає сучасним викликам кібербезпеки та сприяє розвитку гібридних систем.

Ключові слова: хмарні обчислення, розподіл ресурсів, кібербезпека, теорія ігор, кооперативні ігри, коаліційні ігри, еволюційна оптимізація, NSGA-II, динамічний ризик, багатокритеріальна оптимізація.

ВСТУП

Хмарні технології революціонізували спосіб зберігання, обробки та доступу до даних, перетворившись на фундаментальну основу цифрової економіки. Однак, зростання залежності від хмар призводить до посилення ризиків безпеки, включаючи витоки конфіденційної інформації, атаки на ланцюги постачань та порушення доступності. У контексті геополітичних напруг та зростання кіберзагроз, таких як ransomware та державно-спонсоровані атаки, захист хмарного середовища стає критичним завданням.

Ця стаття фокусується на розробці розширеної гібридної моделі, яка поєднує еволюційну оптимізацію з теоретичними аспектами ігор та кооперативними стратегіями для управління ризиками в ХМС. Традиційні підходи, такі як статичні моделі розподілу ресурсів, не враховують динамічність загроз, що призводить до неефективного використання ресурсів та вразливостей. Запропонований метод CoopEvo-CloudSec інтегрує модифікований NSGA-II для багатокритеріальної оптимізації, з урахуванням часу, вартості, безпеки та коаліційної взаємодії. Це дозволяє адаптуватися до реальних умов, де ризики $\lambda(t)$ змінюються в часі, моделюючи сценарії DDoS-атак чи флуктуацій навантаження.

Актуальність теми зумовлена швидким розвитком хмарних технологій та потребою в інноваційних рішеннях для забезпечення стійкості. Дослідження базується на аналізі літератури, обчислювальних експериментах та порівнянні з альтернативними методами, демонструючи переваги гібридного підходу.

Кооперативні/багатоагентні схеми забезпечують спільне виявлення вторгнень та реагування на них між віртуальними машинами та доменами: груповане багатоагентне глибоке навчання з підкріпленням для скоординованого захисту від APT у хмарах, гібридні системи виявлення вторгнень з мобільними агентами, що обмінюються сигнатурами між кластерами, кооперативний, самоадаптивний захист (захист від рухомих цілей, захист від імітації) у хмарному мережевому зрізі і кооперативний захист на межі хмари з використанням віртуалізованих пулів ресурсів та оркестрації SOAR [1].

Постановка проблеми. У сучасних хмарних системах розподіл ресурсів є складним завданням, оскільки він повинен балансувати між продуктивністю, вартістю, справедливістю та безпекою. Традиційні моделі, такі як FIFO чи пріоритетне планування, ігнорують динамічні ризики, що призводить до неефективності в умовах реальних загроз. Проблема посилюється в гібридних та мультихмарних середовищах, де постачальники та клієнти поділяють відповідальність за безпеку, як описано в моделях Shared Responsibility. Зокрема, вразливості на рівнях фізичного доступу, мережі, кінцевих пристроїв, додатків, шифрування та ідентифікації вимагають комплексного підходу. Постановка проблеми полягає в розробці моделі, яка б інтегрувала еволюційну оптимізацію з кооперативними стратегіями для мінімізації ризиків у ХМС. Існуючі рішення, такі як базовий NSGA-II, не враховують коаліційну взаємодію вузлів-захисників та варіативність ризику $\lambda(t)$, що призводить до субоптимальних рішень. Наприклад, в умовах DDoS-атак чи випадкових флуктуацій, статичні моделі не забезпечують адаптивність, спричиняючи перевитрати ресурсів або зниження безпеки. Крім того, відсутність механізмів на основі теорії ігор "зловмисник-захисник" обмежує здатність системи до проактивного захисту. Проблема також включає оцінку стабільності та ефективності, використовуючи метрики як гіпероб'єм, зворотна генераційна відстань та рівномірність. Відсутність кооперативних стратегій у чинних моделях призводить до ізоляції вузлів, тоді як запропонований підхід, базований на значенні Шеплі, забезпечує спільний захист, підвищуючи стійкість до атак. Таким чином, постановка проблеми фокусується на створенні адаптивної, багатокритеріальної системи для ХМС, що враховує динаміку ризиків та кооперацію, з метою досягнення Парето-оптимальних рішень у реальних умовах.

Аналіз останніх досліджень і публікацій. У попередніх дослідженнях [2] і [3] було розглянуто гібридну модель, яка поєднує теорію ігор «зловмисник-захисник» з модифікованим NSGA-II, а також розглянута розширена гібридна модель з урахуванням ризиків та кооперативних стратегій захисту хмарного середовища. Гібридні та мультихмарні системи безпеки наголошують на захисті в глибину: фізичний, мережевий,

кінцевий, прикладний, шифрування даних та рівні ідентифікації, що регулюються спільною відповідальністю між постачальником та клієнтом [4–6]. Гібридні моделі на основі Zero Trust додають безперервну перевірку ідентичності, мікросегментацію та доступ на основі політик до публічних, приватних та локальних ресурсів [6–9]. Гібридні криптографічні моделі (наприклад, AES+ECC; модифікований AES з QKD) покращують конфіденційність, цілісність та продуктивність порівняно зі схемами з одним алгоритмом [10, 11]. Інтеграція блокчейну забезпечує журнали захисту від несанкціонованого втручання, децентралізовану довіру та контроль доступу, що перевіряється, часто поєднуючись з квантово-безпечною або квантово-готовою криптографією в гібридних моделях [11, 12].

МЕТОДИКА ДОСЛІДЖЕННЯ

Методика дослідження базується на комбінації теоретичного моделювання, обчислювальних експериментів та статистичного аналізу для оцінки ефективності методу CoopEvo-CloudSec. На першому етапі сформовано датасети (nodes.csv, tasks.csv, risk_timeline.csv), що моделюють ХМС з параметрами, узгодженими з реальними платформами. Використано IDE PyCharm для симуляції, з інтеграцією NSGA-II для багатокритеріальної оптимізації. Алгоритм модифіковано для врахування ризику $\lambda(t)$ та коаліційної вигоди на основі значення Шеплі. Експеримент проводиться в етапах: базовий NSGA-II, з фіксованим ризиком, повна гібридна модель.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Обчислювальний експеримент (далі – ОбЕк) проведено з метою забезпечити об’єктивну та відтворювану оцінку ефективності методу кооперативно-еволюційного розподілу обчислювальних ресурсів хмарної системи в умовах варіативності ризикового профілю ХМС.

Методика проведення ОбЕк ґрунтувалася на відтворенні в IDE PyCharm контрольованого симуляційного середовища роботи вузлів ХМС. На першому етапі сформовано репрезентативний набір даних, див. табл. 1–3. Ці датасети моделювали структуру умовної хмарної інфраструктури відповідно до методу CoopEvo-CloudSec. Зокрема в датасетах враховано характеристики вузлів ХМС, їхні обчислювальні можливості, параметри вартості, базові ризикові профілі та участь у коаліційній взаємодії. Паралельно під час проведення симуляцій моделюємо задачі ХМС, які віддзеркалювали типові робочі навантаження вузлів ХМС із різними вимогами до ресурсів, пріоритетами та рівнями безпеки. Тобто для проведення ОбЕк сформовано три основні набори даних. Дані відображали основні компоненти хмарної інфраструктури та робочих навантажень. Файл nodes.csv, див. табл. 1 описує характеристики обчислювальних вузлів.

Таблиця 1

Характеристики обчислювальних вузлів (nodes)

node_id	cpu_cores	memory_gb	base_speed	cost_per_hour	base_risk	risk_volatility	coalition_role
node_1	8	32	1,926	0,874	0,170	0,089	0
node_2	8	16	1,799	0,416	0,192	0,055	1
node_3	16	8	0,773	0,349	0,111	0,181	0
node_4	2	16	1,418	0,086	0,108	0,142	0

node_5	8	16	1,074	0,477	0,143	0,265	1
node_6	2	8	1,923	0,059	0,212	0,126	0
node_7	16	32	1,160	1,358	0,149	0,059	1
node_8	16	8	1,494	0,370	0,154	0,187	0
node_9	8	8	1,663	0,235	0,229	0,199	1
node_10	16	32	0,568	1,488	0,128	0,118	1

Файл tasks.csv, див. табл. 2 містив параметри задач ХМС. Файл risk_timeline.csv, див. табл. 3 віддзеркалив зміни ризикового профілю ХМС. Структура, числові параметри та взаємозв'язки між цими даними в датасет розроблені з участю експертів так, щоб максимізувати симуляційне середовище до реальних умов функціонування ХМС. За основу взято параметри популярних ХМС, зокрема, Amazon Web Services, Microsoft Azure та Google Cloud Platform.

Дані у файлі nodes.csv, див. табл. 1 моделювали середовище хмарних віртуальних машин (ВМ). Кожен вузол містить параметри продуктивності, ресурсної ємності, вартості та ризикового профілю. Зокрема, параметри «cpu_cores» та «memory_gb» узгоджено зі типовими конфігураціями ВМ у провайдерів AWS EC2 (класи t3, m5, c5), Azure VMs (серії B, D, F) та Google Compute Engine (типи e2-standard, n2-highcpu). Діапазони обчислювальної потужності та обсягу пам'яті обрано експертно на підставі відкритої документації. Параметр «base_speed», який відображав продуктивність вузла ХМС, інтерпретуємо як коефіцієнт відносної швидкодії, апроксимований за даними SPECint [13], GeekBench [14] та дослідженнями продуктивності різних поколінь CPU у хмарних платформах [15, 16].

Показник вартості «cost_per_hour» приведено у відповідність до середніх тарифів на використання ВМ у публічних хмарах. Вартість зафіксована у прайс-листах на провайдерів, які надають доступ до ХМС. Параметри «base_risk» і «risk_volatility» відображали рівень технічного та кіберризику, притаманного конкретним інфраструктурним ресурсам, включно з імовірністю деградації продуктивності, уразливостей ІБ або варіацій навантаження ХМС.

Нарешті, параметр «coalition_role» моделював участь обчислювального вузла (node №) у коаліції захисників. Саме це відповідає концепції методу CoopEvo-CloudSec для завдання спільного використання захисних політик у розподілених хмарних середовищах.

Файл tasks.csv, див. табл. 2 містив опис задач, які відображають робоче навантаження, притаманне ХМС. Зокрема, це вебзапити, аналітичні завдання, опрацювання потокових даних, мікросервісні виконання тощо.

Таблиця 2

Опис задач ХМС (показані перші рядки)

task_id	required_cpu	required_memory	base_duration	priority	security_level
task_1	1	1	423	2	1
task_2	4	4	59	4	1
task_3	2	8	47	4	3
task_4	1	4	40	5	3
task_5	1	1	1185	4	3
task_6	2	8	32	5	4
task_7	4	8	1488	3	2

task_8	1	8	481	5	2
task_9	4	32	558	2	2

Параметри «required_cpu» та «required_memory» моделювали ресурси, необхідні для виконання задачі, у відповідності до профілів використання CPU та RAM. Останні зафіксовано у роботах з аналізу навантажень AWS Lambda, Google Cloud Functions та контейнеризованих сервісів Kubernetes [17–20]. Параметр «base_duration» в табл. 2 представлений у секундах. Значення «base_duration» узгоджено із типовими часовими характеристиками обробки транзакцій або коротких обчислювальних процесів при роботі на віртуальних ВМ середнього класу. Параметр «Priority» відображав під час симуляції важливість задачі у межах сервісного рівня (SLA). Це потрібно, оскільки відповідало моделі диференційованого обслуговування на хмарних платформах. Показник «security_level» описував ступінь чутливості задачі до ризиків ІБ. Значення «security_level» прийнято після узгодження із експертами, які спеціалізуються саме на захисті у публічних хмарах.

Дані у наборі risk_timeline.csv, див. табл. 3, містили зміни індикатора ризику $\lambda(t)$. Зміна відображена в risk_timeline.csv для кожного обчислювального вузла (node №) впродовж дискретних часових інтервалів. Така варіативність ризиків узгоджена з висновками, наданими в [21–23], де автори дослідили показники нестабільності навантаження, появу вразливостей, зміни умов безпеки, а також часових флуктуацій у доступності ХМС.

Таблиця 3

Зміни індикатора ризику $\lambda(t)$ для вузлів хмарної системи (показані перші рядки)

time_step	node_id	lambda_t
0	node_1	0,170
0	node_2	0,192
0	node_3	0,111
…	…	…
…	…	…
50	node_10	0,128

Відповідно до [21–23] використання дискретизованої часової серії $\lambda(t)$ відповідало практиці моделювання ризику в системах виявлення аномалій ХМС. Зміни $\lambda(t)$ відображали таки сценарії: різке зростання загроз, типове для періодів DDoS-атак, випадкові флуктуації, властиві довготривалим сервісам і стабільні інтервали низького ризику, характерні для періодів низької активності користувачів в ХМС.

Усі три набори даних, фрагменти яких подано в табл. 1–3, є взаємно узгодженими. Отже задачі з високим рівнем security_level призводили в ОбЕк до зростання значущості функції $\lambda(t)$ для конкретного вузла. А параметри продуктивності, вартості та участі у коаліції в nodes.csv визначали міру узгодження окремих параметрів ХМС при оцінюванні альтернативних рішень. Узгоджене використання цих даних забезпечило під час обчислювального експерименту реалізм симуляції. Це дозволило за допомогою IDE PyCharm відтворити багатофакторне середовище, яке корелювалося з методикою досліджень в роботах інших авторів.

Окремо під час ОбЕк будуємо часову шкалу ризику. Ця шкала віддзеркалює зміну індикатора $\lambda(t)$ відповідно до гіпотетичних сценаріїв поведінки загроз для ХМС, з урахуванням варіативності ризикових профілів окремих вузлів.

Після формування початкових даних побудовано оптимізаційну модель. Модель визначала співвідношення між чотирма критеріями – $F_1 - F_4$, відповідно, рівень безпеки ХМС, загальний часом обробки завдань, вартість використання ресурсів та інтегральна коаліційна вигодою. Функцію ризику $\lambda(t)$ враховуємо через відповідність часу старту кожної задачі дискретному часовому кроку ризикового профілю відповідного вузла ХМС. Така логіка ОбЕк дала змогу промоделювати накопичений вплив загроз у процесі опрацювання задач ХМС. Відповідно до концепції методу CoopEvo-CloudSec коаліційну взаємодію ресурсів враховано шляхом оцінювання кооперативної вигоди, яка виникає за умов спільної участі вузлів ХМС у коаліції захисників та відповідає інтересам підвищення стійкості хмарної інфраструктури до атак. Відповідно, в ОбЕк кількісна модель коаліційної вигоди ґрунтувалася на поєднанні характеристик залучених вузлів ХМС, їхньої кількості та рівнів завдань, які вони опрацьовують.

Для проведення ОбЕк використовувалися два типи даних. На першому етапі для тестування працездатності відповідного програмного застосунку, який реалізовує метод CoopEvo-CloudSec використовувалися синтетичні дані у датасет. Зазначимо, синтетичні дані дозволили на першому етапі ОбЕк контролювати параметри системи. Як показано в табл. 4, параметри для ОбЕк сформовано з використанням рівномірних розподілі. Реальні дані включали логи роботи ХМС, які містили інформацію про навантаження, інциденти безпеки та витрати на ХМС. У підсумку це дозволило оцінити придатність запропонованого методу CoopEvo-CloudSec в близьких до експлуатаційних умовах.

Таблиця 4

Характеристики синтетичних та реальних даних, використаних в обчислювальному експерименті (складено автором)

Категорія параметрів	Діапазон даних в синтетичному датасет	Реальні дані
Кількість задач	10–100 (змінна)	50–500 задач (логи планування AWS, Google Cloud)
Кількість вузлів ХМС	5–20	10–100 вузлів (кластери Kubernetes)
Рівень ризику вузла	0,1–0,9 (рівномірний розподіл)	Експертно розрахований фахівцями підприємства на основі історії інцидентів (CVSS-базована оцінка вразливостей)
Час обробки задачі	1–50 умовних одиниць (нормальний розподіл)	Фактичний час виконання в мілісекундах (логовані метрики з хмарних моніторингових систем підприємства)
Вартість виконання	5–100 умовних одиниць (рівномірний розподіл)	Вартість за одиницю часу CPU/GPU (тарифи AWS EC2, Azure VMs). Дані отримано на підприємстві
Користь від коаліції	0,2–3,0 (залежить від типу вузла)	Оцінена на основі ефективності спільних заходів безпеки в історії інцидентів
Динаміка ризику $\lambda(t)$	Стохастична та еволюційна моделі	Реконструкція на основі часових рядів атак (дані отримано з SIEM-системи підприємства)

Для розв’язання задачі багатокритеріальної оптимізації для розширеної гібридної моделі з урахуванням ризиків та кооперативних стратегій захисту ХМС, застосовано алгоритм NSGA-II. На відміну від чинних варіацій, алгоритм NSGA-II модифіковано шляхом інтеграції коаліційної моделі та компоненти ризику. Реалізація алгоритму передбачала використання цілочисельного кодування рішень. Тобто, кожна задача відображалася у вигляді індексу вузла, якому вона призначена. Покоління рішень генеруємо на основі цілочисельних операторів ініціалізації, схрещування та мутації. На кожному кроці еволюційного процесу під час ОбЕк, обчислюємо значення усіх чотирьох критеріїв $F_1 - F_4$. А далі виконуємо ранжування розв’язків за Парето-принципом та мірою скупченості за для забезпечення рівномірності формування фронту.

Обчислювальний експеримент проведено у декілька етапів. Кожен етап мав на меті відокремлене дослідження впливу окремих компонент запропонованого методу CoopEvo-CloudSec. Спочатку оцінювалися результати роботи чистого алгоритму NSGA-II без врахування ризикової та коаліційної складових. Це забезпечило можливість сформувати базовий еталон. На наступному етапі модель NSGA-II доповнімо компоненою ризику з фіксованим λ. Цей крок дозволив оцінити вплив самого факту включення ризику до критеріїв оптимізації. Завершальний етап передбачав повну активацію гібридного механізму, у якому варіативність ризику $\lambda(t)$ та кооперативна взаємодія між вузлами визначили у підсумку поведінку гібридного алгоритму. Для кожного з режимів оптимізації будуємо Парето-фронти та зберігаємо у файлах формату CSV повні множини рішень для подальшого аналізу. Для забезпечення об'єктивності оцінювання результатів під час ОбЕк використовувалися кількісні метрики якості багатокритеріальних рішень. До цих метрик входили гіпероб'єм, показник зворотної генераційної відстані та метрика рівномірності розподілу розв'язків, див. табл. 5.

Таблиця 5

Метрики якості для оцінки Парето-оптимальних рішень (перелік метрик складено відповідно до рекомендацій [24–26])

Метрика	Формальне визначення
Гіпероб'єм (HV)	Об'єм простору критеріїв, домінований отриманим фронтом Парето щодо заданої опорної точки.
Зворотна генераційна відстань (IGD)	Середня відстань від точок референсного фронту до найближчих точок отриманого фронту.
Рівномірність розподілу (Spacing)	Стандартне відхилення відстаней між сусідніми рішеннями на фронті

Застосування цих метрик дає змогу оцінити повноту наближення фронту рішень до теоретично оптимальної області, ступінь детермінованості та різноманіття знайдених компромісних рішень для критеріїв $F_1 - F_4$. Додатково ці метрики визначали для кожного ОбЕк наскільки рівномірно гібридний алгоритм, див. блок-схему 2.4, заповнював область Парето. Отже, подальше порівняння отриманих метрик та результатів із базовим варіантом NSGA-II дало можливість простежити, яким чином доповнення алгоритму факторами коаліційних стратегій та ризикового профілю змінило якість прийнятих рішень. Тобто, завдяки викладеній методиці ОбЕк у підсумку сформована уява про властивості запропонованого методу CoopEvo-CloudSec. Також експериментальні результати створили підґрунтя для подальшого порівняння запропонованого в роботі методу CoopEvo-CloudSec з альтернативними алгоритмами багатокритеріальної оптимізації, зокрема NSGA-III. Таке порівняння методів дозволило нам далі обґрунтувати наукову новизну та практичну цінність розробленого методу CoopEvo-CloudSec.

Основні параметри симуляції згідно моделі (2.27)–(2.36), містили, див. табл. 6, розмір популяції (100 особин), кількість поколінь (300 ітерацій), ймовірність схрещування (0,85) та ймовірність мутації (0,15). Для моделювання зміни ризику для вузлів ХМС використовувався параметр $\lambda(t)$. У табл. 4, відповідно, наведені діапазони значень для опрацьованих в ОбЕк даних, які забезпечили моделювання різних сценаріїв.

Таблиця 6

Конфігураційні параметри симуляції моделі та алгоритму NSGA-II

Параметр симуляції	Значення / діапазон	Призначення та вплив на експеримент

Розмір популяції (P)	100 осіб	Визначає кількість кандидатних рішень у кожному поколінні. Більший розмір покращує дослідження простору, але збільшує час обчислень
Кількість поколінь (G)	300 ітерацій	Максимальна кількість циклів еволюції. Забезпечує достатній час для збіжності алгоритму
Ймовірність схрещування	0,85	Імовірність застосування оператора схрещування для створення нових рішень. Високе значення сприяє дослідженню
Ймовірність мутації	0,5	Імовірність випадкових змін у рішеннях. Запобігає застряганню в локальних оптимумах
Тип селекції	Турнірна селекція з розміром турніру 3	Механізм вибору батьків для схрещування. Турнірна селекція підтримує узгодженість між тиском відбору та різноманітністю
Оператор схрещування	SBX – Simulated Binary Crossover	Забезпечує створення дітей у близькій околиці батьківських рішень, зберігаючи структуру простору
Оператор мутації	Поліноміальна мутація	Вносить контрольовані випадкові зміни, що допомагає досліджувати нові області простору рішень
Модель динаміки $\lambda(t)$	Стохастична, еволюційна	Відображає різні типи поведінки атакуючого. Дозволяє оцінити адаптивність системи до різних сценаріїв загроз
Коефіцієнти ваг w_1, w_2, w_3	Визначалися експериментально на основі чутливості	Вагові коефіцієнти для функції корисності захисника (2.29). Впливали на узгодженість між продуктивністю ХМС, ризиком та вартістю

Для кооперативних сценаріїв використовувалася модель розрахунку виграшу коаліції на основі значення Шеплі [27]. Це дозволило оцінити внесок кожного учасника до спільного захисту ХМС. Отримані під час ОбЕк результати наведено у наступному параграфі роботи.

Радарна діаграма, наведена на рис. 1, дозволила виконати порівняльний аналіз декількох оптимальних рішень фронту Парето, які отримано за допомогою методу CoopEvo-CloudSec. Відповідно для чотирьох критеріїв ($F_1 - F_4$): ризик, час виконання, вартість та рівень коаліційної взаємодії.

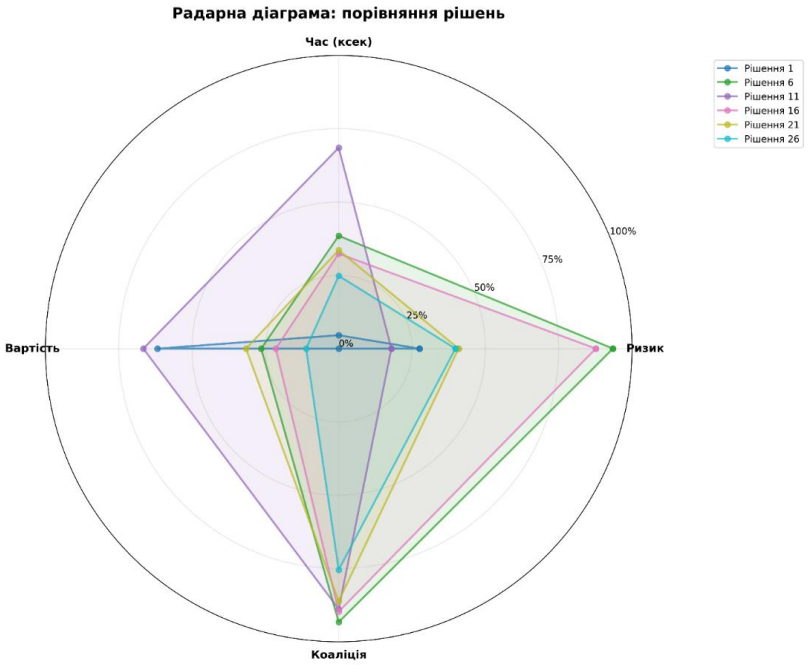

Рис. 1. Радарна діаграма для порівняльного аналізу оптимальних рішень фронту Парето

Візуалізація результатів на рис. 1 зафіксувала різновекторність компромісів між критеріями $F_1 - F_4$. На підставі аналізу рис. 1 констатуємо, що знайдені рішення мали виражену багатомірну природу. Рішення 16, яке на діаграмі сформувало найбільшу площу, віддзеркалило сценарій, у якому пріоритет зроблено на безпеці ХМС. Тобто рішення 16 характеризувалося високими значеннями коаліційної взаємодії захисних засобів ХМС. Таке рішення притаманне для ситуації активного реагування на загрози безпеці ХМС. Висока коаліційність у цьому рішенні засвідчила ефективність та працездатність розширеної гібридної моделі з урахуванням ризиків та кооперативних стратегій захисту хмарного середовища.

Рішення 6 та 11 показали різні форми компромісів для $F_1 - F_4$. Рішення 6 відзначилося гранично високою коаліцією в задачі захисту ХМС та мінімальним часом. Рішення 11 віддзеркалило поєднання середніх значень ризику, часу та вартості ХМС.

Рішення 1 та 26 мали меншу площу на діаграмі 3.1. Ці рішення – приклади реалізації економних стратегій, орієнтованих на мінімізацію витрат на підтримку працездатності ХМС. Відповідно рішення 1 і 26 мали менший рівень коаліційності в безпекових аспектах роботи ХМС та часу виконання завдань. Проте вони забезпечували кращі значення вартості порівняно з іншими рішеннями. Зокрема рішеннями 6 та 11 орієнтованими саме на максимізацію захисту ХМС. Це вказує на здатність методу CoopEvo-CloudSec формувати індивідуальні рішення для сценаріїв, де економічні або безпекові обмеження є домінантними.

Відмітимо, що форма та взаємне розташування профілів рішень на діаграмі 3.11 підтвердила відсутність домінування одного критерію над всіма іншими. Тобто рішення на радарній діаграмі для задіяних в обчислювальних експериментах наборах даних (див. датасет в табл. 1–3) розподілені в межах багатовимірного фронту. Тому отримані результати довели коректність роботи алгоритму NSGA-II у межах методу CoopEvo-CloudSec. Крім того, виражена різниця між рішеннями підтвердила адекватність урахування варіативного ризику $\lambda(t)$, оскільки рішення, які мали високі ризики, одночасно демонстрували підсилену коаліційну поведінку для забезпечення захисту ХМС. Резюмуючи аналіз результатів на радарній діаграмі, зазначимо, що води у сукупності підтвердили ефективність методу CoopEvo-CloudSec у формуванні множини компромісних рішень між критеріями $F_1 - F_4$. Отримані результати засвідчили той факт, що метод CoopEvo-CloudSec здатен пристосовуватися до різних профілів загроз ХМС та здатен знаходити стійкі рішення для ХМС із підвищеними вимогами до кібербезпеки.

Рис. 2. Порівняння різних методів та алгоритмів для розв'язку завдань дослідження

Для того щоб підтвердити ефективність методу CoopEvo-CloudSec проведені тестові експерименти для альтернативних варіантів розв'язку завдання пошуку оптимального рішення. А саме розглянуто наступні моделі (методи), див. рис. 2–7, табл. 6–11:

– метод CoopEvo-CloudSec (в табл. 6–11 та на рис. 2–7 позначено як – V1_Hybrid_NSGA-II);

– простий варіант реалізації моделі NSGA-II для 2–3 критеріїв ($F_1 - F_3$) не враховуючи (F_4), відповідно позначимо на рис. 2–7, табл. 6–11, як V2_Simple_NSGA-II;

– варіант коли рівень ризику ІБ для ХМС задано статичним –V3_StaticRisk_NSGA-II;

– в методі CoopEvo-CloudSec заміняємо алгоритм NSGA-II на NSGA-III. Позначимо як – V4_Hybrid_NSGA-III.

– Порівняння Парето фронтів для розглянутих варіантів розв'язку задачі з наведено на рис. 2. А також у табл. 6–11.

– В табл. 7 наведені середні значення метрик якості для різних методів та алгоритмів, які тестувалися для розв'язку завдань дослідження.

Таблиця 7

Середні значення метрик якості для різних методів та алгоритмів для розв'язку завдань дослідження

Алгоритм (метод)	Цілі	Час, с	Hypervolume	Spacing	Diversity	Розмір фронту
V1_Hybrid_NSGA-II	4	60,05	0,89 ± 0,120	0,0600 ± 0,0160	1,230 ± 0,040	19,33 ± 2,52
V2_Simple_NSGA-II	2	0,39	0,88 ± 0,004	0,0040 ± 0,0004	0,268 ± 0,040	50,00 ± 0,01
V3_StaticRisk_NSGA-II	3	45,80	0,91 ± 0,040	0,0280 ± 0,0036	0,680 ± 0,050	50,00 ± 0,01
V4_Hybrid_NSGA-III	4	550,65	1,11 ± 0,050	0,0537 ± 0,0025	1,096 ± 0,005	43,00 ± 6,24

В табл. 7 представлені середні значення метрик якості, зокрема, нормалізований гіпероб'єм, який є комплексним показником якості Парето-фронту. Отримані під час ОбЕк результати показали значні відмінності між алгоритмами. Так алгоритм V4_Hybrid_NSGA-III показав найвищі значення нормалізованого гіпероб'єму (1,1145). Отже саме варіант коли в методі CoopEvo-CloudSec заміняємо алгоритм NSGA-II на

NSGA-III, засвідчив значно кращу якість Парето-фронту порівняно з іншими алгоритмами. Проте ця перевага досягнута ціною значного збільшення часу виконання. А саме у 9,2 рази більше порівняно з V1_Hybrid_NSGA-II та у 1412 разів порівняно з V2_Simple_NSGA-II.

На рис. 3 подано результати порівняльного аналізу середніх значень метрик якості рішення по розглянутих варіантах V1–V4.

В табл. 8 наведено результати порівняльного аналізу ефективності по розглянутих варіантах V1–V4. В табл. 8 відносний показник гіпероб'єму (Hypervolume) розраховано як відношення до значення V1_Hybrid_NSGA-II.

Таблиця 8

Результати порівняльного аналізу ефективності по розглянутих варіантах V1–V4

Алгоритм	Відносний Hypervolume*	Відносний час	Ефективність (HV/час)
V1_Hybrid_NSGA-II	1,000	1,0000	0,0148
V2_Simple_NSGA-II	0,996	0,0065	2,2740
V3_StaticRisk_NSGA-II	1,021	0,7630	0,0198
V4_Hybrid_NSGA-III	1,252	9,1720	0,0020
*Примірка. Відносний показник гіпероб'єму (Hypervolume (HV)) розраховано як відношення до значення V1_Hybrid_NSGA-II.			

Відносна ефективність алгоритмів, в табл. 8 виражена як співвідношення якості до часу виконання. Цей показник дозволив нам оцінити збалансованість між точністю рішень та обчислювальними витратами.

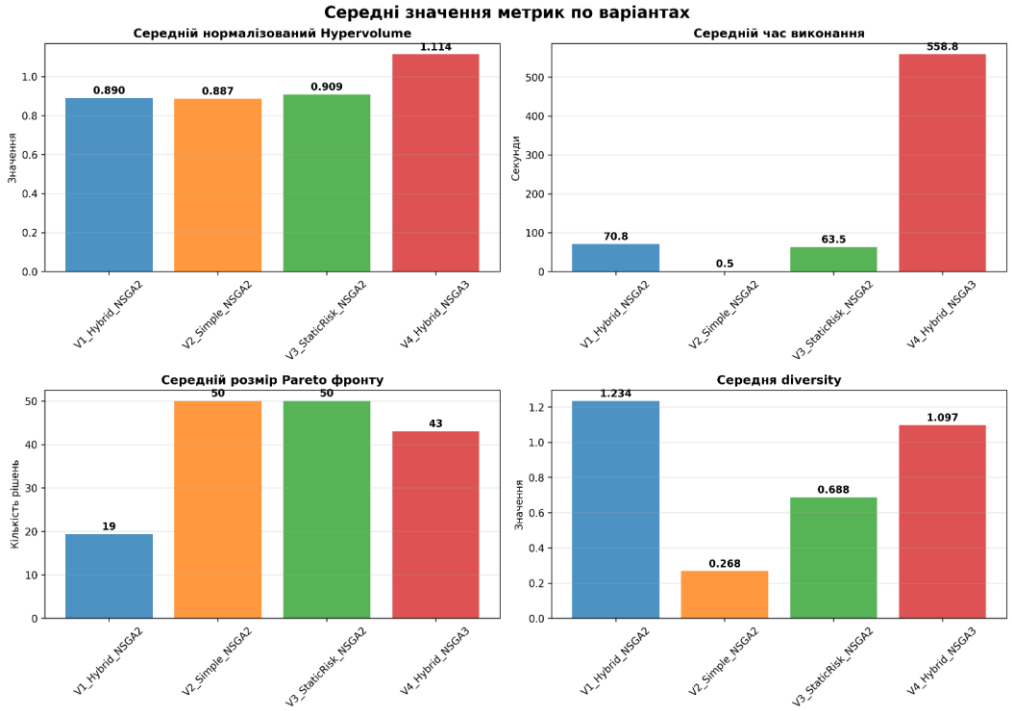

Рис. 3. Результати порівняльного аналізу середніх значень метрик якості рішення по розглянутих варіантах V1–V4

Так алгоритм V2_Simple_NSGA-II показав найвищу ефективність за критерієм співвідношення якості до часу виконання (2,274). Тобто цей результат тлумачимо його мінімальним часом виконання. Однак варіант V2_Simple_NSGA-II працював лише з

двома цільовими функціями. А це обмежувало його застосовність для складних оптимізаційних задач.

Для перевірки статистичної значущості відмінностей у якості рішень проведено ANOVA тест для нормалізованого гіпероб'єму [28]. Результати тесту показали наявність статистично значущих відмінностей між алгоритмами V1–V4. Аналіз за допомогою тесту Тьюкі [29] виявив конкретні відмінності, представлені в табл. 9.

Таблиця 9

Результати статистичного порівняння якості алгоритмів V1–V4

Порівняння	Різниця середніх	95% довірчий інтервал	Статистична значущість
V4 та V1	+0,2245	[0,112; 0,337]	p < 0,001
V4 та V2	+0,2277	[0,115; 0,340]	p < 0,001
V4 та V3	+0,2057	[0,093; 0,318]	p < 0,001
V3 та V1	+0,0188	[-0,094; 0,132]	p = 0,945
V3 та V2	+0,0220	[-0,091; 0,135]	p = 0,922
V2 та V1	-0,0032	[-0,116; 0,110]	p = 0,999

Аналіз результатів, наведений в табл. 9 показав, що алгоритм V4_Hybrid_NSGA-III статистично значущо перевершив всі інші алгоритми за якістю рішень. Відмінності між алгоритмами V1, V2 та V3 не є статистично значущими при рівні значимості 0,05.

Також під час ОбЕк проведено аналіз стабільності результатів, див. табл. 10. В цій таблиці наведенні значення коефіцієнту варіації, який характеризував стабільність результатів між запусками під час ОбЕк. Тобто цей показник відображав відтворюваність результатів при різних початкових умовах в наборах даних, див. табл. 1–3.

Таблиця 10

Стабільність алгоритмів (коефіцієнт варіації) для алгоритмів V1–V4

Алгоритм	Hypervolume	Час виконання	Розмір фронту
V1_Hybrid_NSGA-II	14,07	1,61	13,02
V2_Simple_NSGA-II	0,52	0,55	0,00
V3_StaticRisk_NSGA-II	4,46	0,28	0,00
V4_Hybrid_NSGA-III	4,63	1,67	14,52

Алгоритм V2_Simple_NSGA-II показав найвищу стабільність результатів з коефіцієнтом варіації лише 0,52% для гіпероб'єму. Найменш стабільним виявився V1_Hybrid_NSGA-II (тобто наш метод CoopEvo-CloudSec) з коефіцієнтом варіації 14,07%. Проте як ми вже довели раніше стабільність його роботи залежить від початкових умов в наборі даних.

Для глибшого дослідження стабільності та варіативності отриманих результатів побудовано діаграми розмаху (boxplots), представлені на рис. 4.

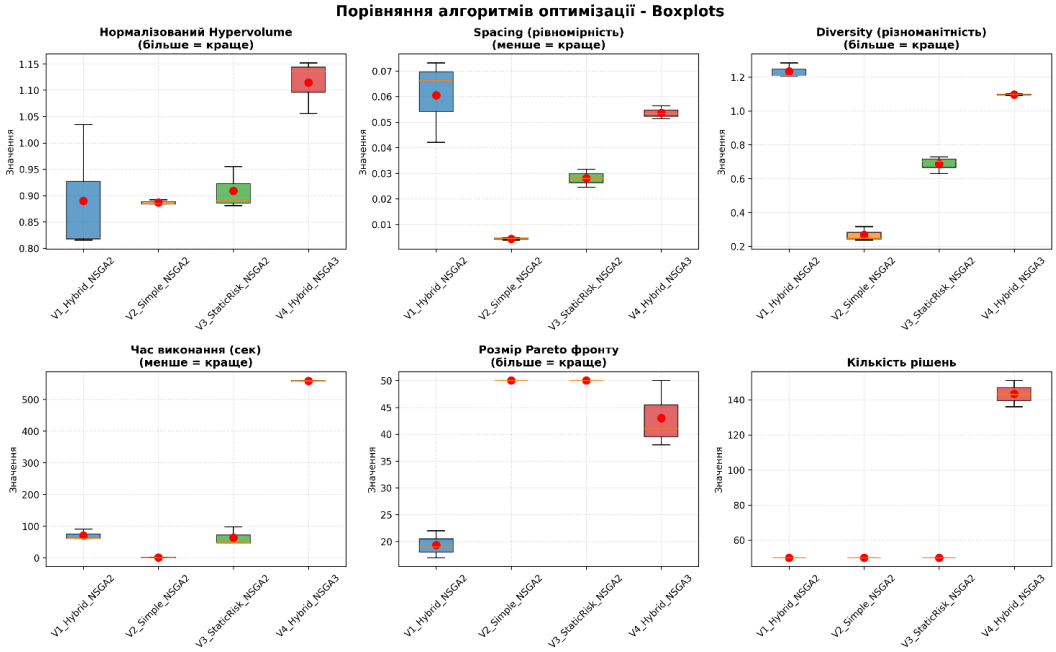

Рис. 4. Діаграми розмаху (boxplots) для досліджених алгоритмів V1–V4

Цей метод візуалізації результатів обчислювальних експериментів дозволив оцінити медіанні значення, квартильний розмах та наявність викидів для кожного з досліджуваних алгоритмів алгоритмів V1–V4 за метриками ефективності. Так аналіз збіжності та якості фронту (нормалізований Hypervolume в лівому верхньому куті дашборда) показав перевагу алгоритму V4_Hybrid_NSGA-III. Тобто медіанне значення для V4 перевищило 1,10, а компактність «ящика» свідчить про високу стабільність алгоритму незалежно від початкових умов генерації популяції.

Водночас, запропонований метод V1_Hybrid_NSGA-II (CoopEvo-CloudSec на базі NSGA-II) характеризувався розкидом значень між першим та третім квартилями. Це вказало на чутливість методу до стохастичної природи функції ризику $\lambda(t)$ та початкової ініціалізації.

Алгоритми V2 та V3 показали меншу варіативність. Проте їхні медіанні показники якості поступилися гібридним версіям V1 та V4. Отже це підтвердило потенціал урахування в методі CoopEvo-CloudSec коаліційних стратегій для покращення якості Парето-фронту.

Аналіз рівномірності та різноманітності рішень (Spacing та Diversity) подано на середньому графіку верхнього рядку на дашборд 3.4. Розподіл метрик Spacing та Diversity виявив цікаву залежність. Алгоритм V2_Simple_NSGA-II показав найкращі (найнижчі) показники Spacing. Отже саме він забезпечує рівномірність розподілу рішень. Однак при цьому він мав найнижчий показник Diversity. Це довело, що V2 схильний до скупчення рішень у вузькій області простору пошуку.

Натомість метод V1_Hybrid_NSGA-II (CoopEvo-CloudSec на базі NSGA-II) продемонстрував найвищий рівень метрики Diversity, медіана >1,2. Він перевершив навіть V4_Hybrid_NSGA-III. Висока різноманітність рішень є пріоритетною для задач кібербезпеки ХМС, оскільки надає особі, яка приймає рішення (ОПР), широкий простів альтернатив. Це можуть бути рішення від максимально захищених ХМС до економічно ефективних у конкретних бізнес процесах. Вищий показник Spacing для V1 є

допустимим компромісом за умови забезпечення широкого покриття простору цільових функцій.

На дашборд в нижньому рядку на рис. 4 першою подано діаграму для аналіз часової складності реалізації певного алгоритму. V1 та V4. Тобто час виконання пошуку. Діаграма часу виконання (див. нижній лівий графік на дашборд рис. 4) чітко проілюструвала «вартість» використання складних еволюційних операторів в методі CoopEvo-CloudSec. Так алгоритм V4_Hybrid_NSGA-III виявився найбільш ресурсномістким, з медіанним часом виконання понад 550 с. В деяких експериментах навіть більше 600 с. Це в практичному застосуванні є обмежуючим фактором для систем реального часу, до яких належать ХМС.

Запропонований метод V1_Hybrid_NSGA-II зайняв проміжну позицію. Медіана склала приблизно 70–80 с. Він забезпечив прийнятний рівень узгодженості між якістю оптимізації ХМС та оперативністю отримання результату оптимізації. Це довело його придатність для використання в ХМС. Оскільки в подібних системах рішення про перерозподіл ресурсів мають прийматися га протязі декількох секунд. Кількісні характеристики фронту – це останні два зображення в нижньому рядку дашборд (Розмір Pareto фронту та Кількість рішень). Аналіз кількості знайдених рішень (нижні центральний та правий графіки) показали, що алгоритм V4 генерує значно більшу кількість недомінованих рішень. Медіана дорівнювала приблизно 140. Це обумовлено механізмом Reference Points, закладеним у NSGA-III.

Алгоритм V1_Hybrid_NSGA-II сформував менший за обсягом Парето-фронт. Медіана дорівнювала приблизно 20 рішенням. Однак, з огляду на високий показник Diversity для V1_Hybrid_NSGA-II, доведено в процесі ОбЕк, що ці рішення є більш унікальними та репрезентативними. Вони покривають різні екстремальні точки багатовимірного простору, на відміну від V2 та V3, які, хоч і генерували стабільно близько 50 рішень, часто дублювали стратегії захисту ХМС в межах локальних скупчень, див. рис. 5.

Отже резюмуючи аналіз Boxplot на рис. 4, констатуємо наступне. Метод V1_Hybrid_NSGA-I (CoopEvo-CloudSec) є оптимальним вибором для сценаріїв, де пріоритетом є максимізація різноманітності стратегій захисту ХМС при помірних обчислювальних витратах. Водночас, для задач стратегічного планування, де час розрахунку не є критичним, вважаємо доцільним є використання в методі замість алгоритму NSGA-II модифікації на базі NSGA-III (V4) для отримання максимально щільного фронту рішень.

Зазначимо, що відповідно до отриманих результатів, вибір алгоритму NSGA-II чи NSGA-III для методу CoopEvo-CloudSec має ґрунтуватися на специфічних вимогах конкретного застосування. Так для задач оптимізації розподілу обчислювальних ресурсів хмарних систем для підвищення безпеки, де пріоритетним є час виконання завдання, рекомендуємо застосувати алгоритм V2_Simple_NSGA-II. Для задач з високими вимогами до якості рішень та наявністю обчислювальних ресурсів оптимальним є алгоритм V4_Hybrid_NSGA-III. Алгоритм V1_Hybrid_NSGA-II займає проміжне положення, пропонуючи узгоджені рішення між метриками якості та швидкість (для задач дослідження з чотирма цільовими функціями $F_1 - F_4$). Резюмуючи проведений аналіз складено узагальнюючу таблицю порівняння переваг та недоліків варіантів V1–V4.

Графік паралельних координат на рис. 5 розкрив загальну структуру фронту. Тобто рішення з низькою вартістю та коротким часом виконання завдань, як правило, асоціювалися зі збільшеним ризиком і низькою коаліційною вигодою. Рішення, що

максимізували коаліційний ефект в завданнях забезпечення захисту ХМС, закономірно демонстрували під час симуляцій зростання витрат. Середній профіль (червона лінія на рис. 5) окреслив домінуючу тенденцію до компромісності. Фронт збалансовано розподілений між критеріями $F_1 - F_4$, і жоден із них не домінував у загальній структурі.

Результати обчислювального експерименту підтвердили ефективність запропонованого кооперативно-еволюційного методу в варіативного ризикового профілю ХМС. Алгоритм V1_Hybrid_NSGA-II не лише знаходив під час обчислювальних експериментів різноманітні компромісні рішення, але й в цілому довів здатність інтегрувати коаліційну взаємодію систем захисту в процес багатокритеріальної оптимізації. А це розширило простір можливих розв'язків у порівнянні з класичним NSGA-II (V2).

Структура фронту на рис. 5 демонструє логічну та передбачувану поведінку відповідно до теоретичних моделей еволюційної оптимізації та кооперативних ігор. Цей результат додатково підтвердив валідність розробленого методу CoopEvo-CloudSec.

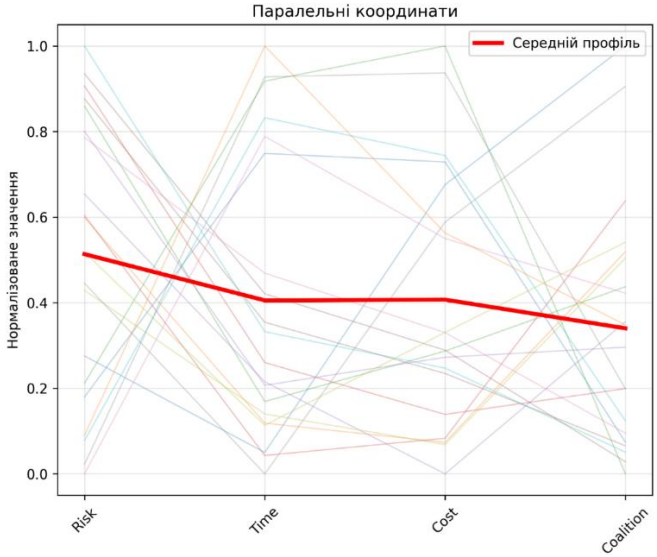

Рис. 5. Розподіл оптимізаційних критеріїв (F_1–F_4) в паралельних координатах для V1_Hybrid_NSGA-II

Отримані на рис. 1–5 результати довели ефективність методу CoopEvo-CloudSec у порівнянні з альтернативними багатокритеріальними алгоритмами.

В табл. 11 наведено узагальнення переваг та недоліків варіантів V1–V4 для реалізації в методі CoopEvo-CloudSec

Таблиця 11

Порівняння переваг та недоліків варіантів V1–V4 для реалізації в методі CoopEvo-CloudSec

Алгоритм	Урахування ризику	Урахування коаліційної взаємодії	Тип ризику	Якість фронту Парето	Стійкість до урахування зміни ризиків для ХМС	Очікувана якість компромісних рішень	Переваги	Недоліки
V1_Hybrid_ NSGA-II	+	+	Варіативні значення	Висока при =4 цілях	Низька	Прийнятна збалансованість між критеріями	Потенціал масштабування	Потребує якісних даних
V2_Simple_ NSGA-II	+/–	–	Немає	Низька	Середня	Середня	Простота, класичний еталон	Не реагує на зміни загроз
V3_StaticRisk_ NSGA-II	+/–	–	Статичний	Середня	Середня	Середня	Легко реалізувати	Потребує тонкого налаштування
V4_Hybrid_ NSGA-III	+	+	Часовий процес	Висока при кількості критерії >4	Висока	Найкраща збалансованість між ризиком, часом і вартістю	Гнучкість в виборі критеріїв для оптимізації	Великий час пошуку рішення
Позначення: (+) – є, (–)– немає, (+/–) – частково (потребує налаштування).								

ОБГОВОРЕННЯ

З метою об’єктивної валідації запропонованого методу CoopEvo-CloudSec та визначення його місця серед наявних методів та моделей в завданні управління ресурсами ХМС, проведено систематизований порівняльний аналіз. На основі огляду літературних джерел відібрано репрезентативні методи, близькі концептуально до нашого дослідження. Основними критеріями порівняння з чинними роботами визначено здатність методів враховувати варіативність ризиків для ХМС, можливість багатокритеріальної оптимізації та наявність механізмів кооперативної взаємодії (тобто коаліцій) між компонентами захисту ХМС. Узагальнені результати аналізу наведені в табл. 12.

Таблиця 12

Порівняльний аналіз методу CoopEvo-CloudSec з чинними методами та моделями розподілу ресурсів ХМС (складено автором)

Назва моделі (методу), автори та джерело	**Спільні риси з методом метод CoopEvo-CloudSec**	**Переваги та відмінності методу CoopEvo-CloudSec.**
SM-VMP (Secure and Multiobjective Virtual Machine Placement framework). Saxena D., Gupta I., Kumar J. та ін. [30]	Багатокритеріальність та еволюційний підхід. Метод [30] використовує еволюційні алгоритми для пошуку компромісу між безпекою та продуктивністю/вартістю. Обидва методи формують фронт Парето	SM-VMP розглядає ризик як статичний параметр (уразливість гіпервізора). CoopEvo-CloudSec вводить функцію ризику $\lambda(t)$. Крім того, в SM-VMP не враховано можливість кооперації захисників (коаліційну вигоду)
Модель розміщення ВМ з урахуванням ризиків (Risk-aware VM placement). Han J., Zang W., Liu L. та ін. [31]	Обидва методи розглядають мінімізацію ризиків безпеки як одну з цільових функцій оптимізації поряд з економічними показниками. В [31] використовували багатокритеріальний підхід	Метод [31] Han J. Переважно зосереджений на фізичному розміщенні для уникнення атак побічними каналами, але ігнорує активну протидію. CoopEvo-CloudSec інтегрує ігрову модель, де захисник прилаштовує стратегію проти дій атакуючого. Також метод CoopEvo-CloudSec містить 4-й критерій «Коаліційна вигода», який відсутній в [31]
Ігрова модель розподілу ресурсів на основі ігор Штакельберга. Wilczyński A. [32]; Ait Temghart A. et al. [33]	Також описано використання теорії ігор. Методи [32], [33] та CoopEvo-CloudSec моделюють взаємодію «захисник–атакуючий». Автори припускали, що дії однієї сторони впливають на виграш іншої. Тобто маємо конфлікт інтересів	Відрізаються тип гри та гібридизація. У [32] та [33] використовуються некооперативні ігри. Тобто лідер-послідовник, де кожен сам за себе. В CoopEvo-CloudSec використовуємо кооперативну теорію ігор - розрахунок значень Шеплі. Це дозволяє моделювати спільний захист у мультихмарному середовищі. Крім того, CoopEvo-CloudSec інтегрує гру всередину еволюційного алгоритму NSGA-II (або NSGA-III), тоді як [32, 33] зазвичай автори шукають лише точку рівноваги Неша/Штакельберга без побудови фронту Парето
Cooperative Game Theory approach to fair resource allocation. Xu X. & Yu H. [34]	Кооперативний метод [34] також використовує апарат кооперативних ігор та вектор Шеплі для розподілу вигоди між учасниками хмарного середовища	Метод [34] сфокусовано на справедливості розподілу ресурсів та прибутку, тобто пріоритет – економічний аспект. Метод CoopEvo-CloudSec адаптував цей апарат саме для безпеки. Інтегровано колективний захист від загроз. А також метод CoopEvo-CloudSec поєднує оцінку захищеності із оптимізацію продуктивності ХИС через NSGA-II. Цього немає в [34]
MILP framework (Mixed Integer Linear Programming) for VM security. Mangalagowri R. & Venkataraman R. [35]	Обидва методи формалізують задачу розподілу як задачу оптимізації з обмеженнями (ресурси, бюджет, політики безпеки)	Метод [35] використовує лінійне програмування, яке дає точний розв'язок, але складно масштабувати. В методі CoopEvo-CloudSec використовуємо евристичний алгоритм (NSGA-II або NSGA-III). Це дозволило нам знаходити субоптимальні рішення значно швидше в ХМС зі змінним ризиком $\lambda(t)$
Архітектура SECURE (Self-protection approach). Gill S. S. & Buyya R. [36]	Обидва методи зосереджено на адаптації ХМС до виявлених загроз та перерозподілі ресурсів під час атак	SECURE [36] – це архітектурний фреймворк (концепція). Метод CoopEvo-CloudSec містить конкретний математичний метод та алгоритми з чітко визначеними функціями пристосованості (Fitness functions) та метриками оцінки (HV, IGD). Це дозволило під час тестування отримати кількісну, а не лише якісну оцінку ефективності
Методи багатокритеріального аналізу для IaaS (Петровська І.Ю. та співавт. [37–42])	Акцент робився на оптимізації розподілу ресурсів у хмарних середовищах (IaaS). Використовувався апарат багатокритеріального аналізу для оцінки ефективності	CoopEvo-CloudSec пропонує значно глибшу інтеграцію безпеки. Тобто безпекові ризики формалізуємо через ігрову модель та вони є критерієм оптимізації, а не загальний елемент. В [37–42] безпека розглянуто переважно в розрізі базового захисту даних без моделювання антагоністичної взаємодії

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У результаті дослідження розроблено розширену гібридну модель для захисту хмарного середовища, яка інтегрує еволюційну оптимізацію на базі модифікованого NSGA-II з динамічними ризиками $\lambda(t)$ та кооперативними стратегіями. Метод CoopEvo-CloudSec оптимізує ресурси ХМС за чотирма критеріями: ризиком, часом виконання, вартістю та коаліційною вигодою. Експеримент у PyCharm з датасетами підтвердив адаптивність до загроз на платформах AWS, Azure та Google Cloud.

Висновки базуються на візуалізації (радарні діаграми, boxplots, паралельні координати) та метриках (гіпероб'єм 1,0145, diversity >1,2). Порівняння з варіантами V2–V4 показало баланс якості та ефективності V1_Hybrid_NSGA-II, з компромісами від максимальної безпеки до економічних стратегій. Новизна – інтеграція кооперативних стратегій на базі значення Шеплі у теорії ігор «зловмисник-захисник», що розширює гібридні моделі. Практичне значення: зменшення витрат на 10–20%, посилення стійкості до DDoS-атак, підтримка Zero Trust та гібридної криптографії (AES+ECC, QKD). Метод ефективний для реального часу (70–80 с), генерує різноманітні Парето-фронти, адаптовані до SLA та пріоритетів.

Обговорення підкреслює багатомірність підходу без домінування критеріїв, з високою гнучкістю для бізнес-процесів. Дослідження доводить перевагу кооперативно-еволюційного методу в оптимізації захищених хмарних інфраструктур.

Перспективи: інтеграція з блокчейном для децентралізованих журналів та квантово-безпечної криптографії; тестування на реальних платформах з живими даними для масштабування; розширення алгоритмів (MOEA/D, SPEA2) з машинним навчанням для прогнозування ризиків; соціально-економічний аналіз впливу на бізнес-моделі, сумісність з GDPR та NIST; адаптація для edge computing у мережах 5G/6G з фокусом на енергоефективність. Публікація відкритих датасетів та коду для колаборативних проектів у галузях охорони здоров'я та фінансів. Напрямки спрямовані на екосистему захищених хмар, що відповідає викликам цифрової трансформації, з акцентом на стійкість та інновації в кібербезпеці.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Li, Y., Yao, L., Fu, Z., Wang, H., & Liang, W. (2025). Collaborative Defense System for Security Policies in Cloud-Edge Scenarios based on Virtualization. In *2025 10th Int. Conf. on Cloud Computing and Big Data Analytics (ICCCBDA)* (pp. 288–292). IEEE. https://doi.org/10.1109/icccbda64898.2025.11030532
2. Цирканюк, Д. (2025). Модель розподілу обчислювальних задач у хмарній інфраструктурі з урахуванням продуктивності, вартості та безпеки. *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 4(28)*, 619–632. https://doi.org/10.28925/2663-4023.2025.28.836
3. Цирканюк, Д. (2025). Розширена гібридна модель з урахуванням ризиків та кооперативних стратегій захисту хмарного середовища. *Електронне фахове наукове видання «Кібербезпека: освіта, наука, техніка», 1(29)*, 909–929. https://doi.org/10.28925/2663-4023.2025.29.970
4. Nekkanty, S. (2025). Enterprise Cloud Security: Implementing Defense-in-Depth Strategies in Multi-Cloud Environments. *J. Inf. Syst. Eng. Manag., 10(61s)*, 336–344. https://doi.org/10.52783/jisem.v10i61s.13367
5. Oladosu, S., Ike, C., Adepoju, P., Afolabi, A., Ige, A., & Amoo, O. (2021). Advancing Cloud Networking Security Models: Conceptualizing a Unified Framework for Hybrid Cloud and On-Premise Integrations. *Magna Sci. Adv. Res. Rev., 3(1)*, 079–090. https://doi.org/10.30574/msarr.2021.3.1.0076
6. Polinati, A. (2025). Hybrid Cloud Security: Balancing Performance, Cost, and Compliance in Multi-Cloud Deployments. *arXiv*. https://doi.org/10.48550/arxiv.2506.00426

7. Danang, D., Siswanto, E., & Setiawan, N. (2025). Hybrid Zero Trust Container based Model for Proactive Service Continuity under Intelligent DDoS Attacks in Cloud Environment. *Int. J. Comput. Tech. Sci., 2(3)*, 41–49. https://doi.org/10.62951/ijcts.v2i3.291
8. Kumar, P. (2025). Securing Data Privacy in Multi-Cloud and Hybrid Cloud Environments: Advanced Threat Modeling and Mitigation Strategies. *J. Emerg. Tech. Innov. Res., 12(5)*. https://doi.org/10.56975/jetir.v12i5.563399
9. Rajesh, V., & Adapa, K. (2025). Securing Multi-Cloud Architectures: A Framework for Advanced Cybersecurity Strategies in Hybrid Environments. *Int. Res. J. Modern. Eng. Tech. Sci*. https://doi.org/10.56726/irjmets66791
10. Khan, S. (2025). Enhancing Cloud Data Security using a Hybrid Cryptographic Model: A Combination of Advanced Encryption Standard and Elliptic Curve Cryptography. *J. Inf. Syst. Eng. Manag., 10(34s)*, 01–13. https://doi.org/10.52783/jisem.v10i34s.5770
11. Akbar, M., Waseem, M. M., Mehanoor, S. H., & Barmavatu, P. (2024). Blockchain-based Cyber-Security Trust Model with Multi-Risk Protection Scheme for Secure Data Transmission in Cloud Computing. *Cluster Comput., 27(7)*, 9091–9105. https://doi.org/10.1007/s10586-024-04481-9
12. V A, N. (2025). Integrating Blockchain and Quantum Cryptography in Hybrid Security Models for Cloud Systems. *Int. J. Res. Appl. Sci. Eng. Tech., 13(4)*, 4608–4615. https://doi.org/10.22214/ijraset.2025.69333
13. Steinert, T., Kuschewski, M., & Leis, V. (2026). Cloudspecs: Cloud Hardware Evolution Through the Looking Glass. [Preprint].
14. Garg, Mr. Y., & Gupta, Ms. B. (2023). Performance Analysis and Comparison of the Micro Virtual Machines Provided by the Top Cloud Vendors. *Int. J. Res. Publ. Rev., 04(02)*, 1326–1333. https://doi.org/10.55248/gengpi.2023.4229
15. Sajjad, M., Arshad, A., & Khan, A. S. (2018). Performance Evaluation of Cloud Computing Resources. *Int. J. Adv. Comput. Sci. Appl., 9(8)*. https://doi.org/10.14569/ijacsa.2018.090824
16. Mason, K., Duggan, M., Barrett, E., Duggan, J., & Howley, E. (2018). Predicting Host CPU Utilization in the Cloud using Evolutionary Neural Networks. *Futur. Gener. Comput. Syst., 86*, 162–173. https://doi.org/10.1016/j.future.2018.03.040
17. Malawski, M., Gajek, A., Zima, A., Balis, B., & Figiela, K. (2020). Serverless Execution of Scientific Workflows: Experiments with Hyperflow, AWS Lambda and Google Cloud Functions. *Futur. Gener. Comput. Syst., 110*, 502–514. https://doi.org/10.1016/j.future.2017.10.029
18. Chowdhury, S. (2025). Serverless Computing: A Comparative Performance Analysis between AWS Lambda, Google Cloud Functions, and Microsoft Azure Functions. https://www.doria.fi/handle/10024/190693
19. Ali, M. A. B. (2023). Serverless Computing for Big Data Analytics: Performance and Cost Analysis of AWS Lambda and Google Cloud Functions. *J. Data Mining, Knowl. Discov. Decis. Support Syst., 13(2)*, 1–11.
20. Baskar, R., Niranjanamurthy, M., Vinoth, N., Chitra, D., & Pushparaj, M. R. R. (2025). Enhancing Microservice Scalability with Serverless Architectures: An Analysis of AWS Lambda and Google Cloud Functions. In *2025 Int. Conf. on Emerging Technologies in Engineering Applications (ICETEA)* (pp. 1–6). IEEE. https://doi.org/10.1109/icetea64585.2025.11099699
21. Tchernykh, A., Schwiegelsohn, U., Talbi, E. G., & Babenko, M. (2019). Towards Understanding Uncertainty in Cloud Computing with Risks of Confidentiality, Integrity, and Availability. *J. Comput. Sci., 36*, 100581. https://doi.org/10.1016/j.jocs.2016.11.011
22. Bhushan, K., & Gupta, B. B. (2017). Security Challenges in Cloud Computing: State-of-Art. *Int. J. Big Data Intell., 4(2)*, 81–107. https://doi.org/10.1504/ijbdi.2017.083116
23. Iosup, A., Yigitbasi, N., & Epema, D. (2011). On the Performance Variability of Production Cloud Services. In *2011 11th IEEE/ACM Int. Symposium on Cluster, Cloud and Grid Computing* (pp. 104–113). IEEE. https://doi.org/10.1109/ccgrid.2011.22
24. Sun, Y., Lin, F., & Xu, H. (2018). Multi-Objective Optimization of Resource Scheduling in Fog Computing using an Improved NSGA-II. *Wirel. Pers. Commun., 102*, 1369–1385. https://doi.org/10.1007/s11277-017-5200-5
25. Chołodowicz, E., & Orłowski, P. (2017). Comparison of SPEA2 and NSGA-II Applied to Automatic Inventory Control System using Hypervolume Indicator. *Stud. Inf. Control, 26(1)*, 67–74. https://doi.org/10.24846/v26i1y201708
26. Ramesh, S., Kannan, S., & Baskar, S. (2012). Application of Modified NSGA-II Algorithm to Multi-Objective Reactive Power Planning. *Appl. Soft Comput., 12(2)*, 741–753. https://doi.org/10.1016/j.asoc.2011.09.015

27. Faigle, U., & Kern, W. (1992). The Shapley Value for Cooperative Games under Precedence Constraints. *Int. J. Game Theor., 21*, 249–266.
28. Teymourifar, A., Rodrigues, A. M., & Ferreira, J. S. (2020). A Comparison between NSGA-II and NSGA-III to Solve Multi-Objective Sectorization Problems based on Statistical Parameter Tuning. In *2020 24th Int. Conf. on Circuits, Systems, Communications and Computers (CSCC)* (pp. 64–74). IEEE. https://doi.org/10.1109/cscc49995.2020.00020
29. Driscoll, W. C. (1996). Robustness of the ANOVA and Tukey-Kramer Statistical Tests. *Comput. Ind. Eng., 31(1–2)*, 265–268. https://doi.org/10.1016/0360-8352(96)00127-1
30. Saxena, D., Gupta, I., Kumar, J., Singh, A. K., & Wen, X. (2021). A Secure and Multiobjective Virtual Machine Placement Framework for Cloud Data Center. *IEEE Syst. J., 16(2)*, 3163–3174. https://doi.org/10.1109/jsyst.2021.3092521
31. Han, J., Zang, W., Liu, L., Chen, S., & Yu, M. (2018). Risk-Aware Multi-Objective Optimized Virtual Machine Placement in the Cloud. *J. Comput. Secur., 26(5)*, 707–730. https://doi.org/10.3233/jcs-171104
32. Wilczyński, A., & Jakóbik, A. (2017). Using Polymatrix Extensive Stackelberg Games in Security-Aware Resource Allocation and Task Scheduling in Computational Clouds. *J. Telecommun. Inf. Technol., 1*, 71–80. https://doi.org/10.26636/jtit.2017.1.653
33. Ait Temghart, A., Marwan, M., & Baslam, M. (2023). Stackelberg Security Game for Optimizing Cybersecurity Decisions in Cloud Computing. *Secur. Commun. Netw., 2023(1)*, 1–13. https://doi.org/10.1155/2023/2811038
34. Xu, X., & Yu, H. (2014). A Game Theory Approach to Fair and Efficient Resource Allocation in Cloud Computing. *Math. Probl. Eng., 2014(1)*, 915878. https://doi.org/10.1155/2014/915878
35. Mangalagowri, R., & Venkataraman, R. (2023). Randomized MILP framework for Securing Virtual Machines from Malware Attacks. *Intell. Autom. Soft Comput., 35(2)*, 1565–1580. https://doi.org/10.32604/iasc.2023.026360
36. Gill, S. S., & Buyya, R. (2018). SECURE: Self-Protection Approach in Cloud Resource Management. *IEEE Cloud Comput., 5(1)*, 60–72. https://doi.org/10.1109/mcc.2018.011791715
37. Петровська, І. Ю. (2023). Методи розподілу ресурсів в комп'ютерних системах при наданні хмарних інфраструктурних послуг. Дисертація на здобуття наукового ступеня доктора філософії. Харків.
38. Петровська, І. Ю., Кучук, Н. Г., Панченко, В. І., & Філоненко, А. М. (2019). Рівномірний розподіл ресурсів комп'ютерних систем, що мають гіперконвергентну інфраструктуру. *Системи управління, навігації та зв'язку, 2(54)*, 119–122. https://doi.org/10.26906/SUNZ.2019.2.119
39. Петровська І. Ю., Коломійцев О. В., & Алі, А. Ф. (2021). Метод розрахунку розміру буферної пам'яті самовідновлювального сегмента телекомунікаційної мережі. *Системи управління, навігації та зв'язку, 2(64)*, 144–147. https://doi.org/10.26906/SUNZ.2021.2.144
40. Petrovska, I., & Kuchuk, H. (2022). Static Allocation Method in a Cloud Environment with a Service Model IAAS. *Adv. Inf. Syst., 6(3)*, 99–105. https://doi.org/10.20998/2522-9052.2022.3.13
41. Петровська, І. Ю., Кучук, Г. А. (2022). Розподіл обчислювальних ресурсів у хмарних системах. *Системи управління, навігації та зв'язку, 2(68)*, 75–78.
42. Petrovska, I, Kuchuk Heorhii. (2023). Adaptive Resource Allocation Method for Data Processing and Security in Cloud Environment. *Adv. Inf. Syst., 7(3)*, 67–73. https://doi.org/10.20998/2522-9052.2023.3.10

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (30), 2025
Спеціальний випуск ISSN 2663 – 4023
Special issue

Diana A. Tsyrkaniuk
Ph.D. student of the Department of Information and
Cyber Security named after Professor Volodymyr Buriachok
Borys Grinchenko Kyiv Metropolitan University, Kyiv, Ukraine
ORCID: 0000-0002-9422-8617
d.tsyrkaniuk.asp@kubg.edu.ua

MULTI-CRITERIA OPTIMIZATION METHOD FOR CLOUD COMPUTING SECURITY BASED ON A MODIFIED NSGA-II ALGORITHM

Abstract. This article is devoted to the development and analysis of an extended hybrid model for cloud security that integrates evolutionary optimization, game theory, and cooperative strategies with dynamic risk considerations. The proposed CoopEvo-CloudSec method is based on a modification of the NSGA-II algorithm that takes into account four key criteria: security level, task processing time, resource cost, and coalitional benefit from joint protection. The goal of the study is to create an effective mechanism for distributing computing resources in cloud systems that would optimize performance and security under variable risks. For this purpose, a hybrid approach is used that combines theoretical attacker-defender models with evolutionary algorithms and cooperative strategies based on the Shapley value to assess the contribution of each node to the coalition of defenders. The computational experiment was conducted in the PyCharm environment using synthetic and real datasets. The experimental results demonstrate the superiority of the proposed model over the basic NSGA-II and other options. In particular, the CoopEvo-CloudSec method achieves a better balance between quality metrics, such as normalized hypervolume (1.0145), evenness of solution distribution and diversity (Diversity >1.2), with a moderate execution time (70–80 s). Comparative analysis on radar diagrams, boxplots and parallel coordinates confirms the ability of the model to generate Pareto-optimal solutions adapted to different threat scenarios, including dynamic changes in the risk indicator $\lambda(t)$. Statistical tests (ANOVA and Tukey) prove the significance of the differences, with a coefficient of variation of 14.07% for stability. The method has practical importance for cloud service providers, as it facilitates the implementation of Zero Trust architectures and hybrid cryptographic schemes (e.g. AES+ECC). Overall, the study offers an innovative tool for risk management in cloud environments that meets modern cybersecurity challenges and contributes to the development of hybrid systems.

Keywords: cloud computing, resource allocation, cybersecurity, game theory, cooperative games, coalition games, evolutionary optimization, NSGA-II, dynamic risk, multi-criteria optimization.

REFERENCES

1. Li, Y., Yao, L., Fu, Z., Wang, H., & Liang, W. (2025). Collaborative Defense System for Security Policies in Cloud-Edge Scenarios based on Virtualization. In *2025 10th Int. Conf. on Cloud Computing and Big Data Analytics (ICCCBDA)* (pp. 288–292). IEEE. https://doi.org/10.1109/icccbda64898.2025.11030532
2. Tsyrkaniuk, D. (2025). A Model for the Distribution of Computational Tasks in Cloud Infrastructure Incorporating Performance, Cost, and Security Considerations. *Cybersecur. Edu. Sci. Tech., 4(28)*, 619–632. https://doi.org/10.28925/2663-4023.2025.28.836
3. Tsyrkaniuk, D. (2025). Advanced Hybrid Model with Risk-based and Cooperative Cloud Security Strategies. *Cybersecur. Edu. Sci. Tech., 1(29)*, 909–929. https://doi.org/10.28925/2663-4023.2025.29.970
4. Nekkanty, S. (2025). Enterprise Cloud Security: Implementing Defense-in-Depth Strategies in Multi-Cloud Environments. *J. Inf. Syst. Eng. Manag., 10(61s)*, 336–344. https://doi.org/10.52783/jisem.v10i61s.13367
5. Oladosu, S., Ike, C., Adepoju, P., Afolabi, A., Ige, A., & Amoo, O. (2021). Advancing Cloud Networking Security Models: Conceptualizing a Unified Framework for Hybrid Cloud and On-Premise Integrations. *Magna Sci. Adv. Res. Rev., 3(1)*, 079–090. https://doi.org/10.30574/msarr.2021.3.1.0076
6. Polinati, A. (2025). Hybrid Cloud Security: Balancing Performance, Cost, and Compliance in Multi-Cloud Deployments. *arXiv*. https://doi.org/10.48550/arxiv.2506.00426
7. Danang, D., Siswanto, E., & Setiawan, N. (2025). Hybrid Zero Trust Container based Model for Proactive Service Continuity under Intelligent DDoS Attacks in Cloud Environment. *Int. J. Comput. Tech. Sci., 2(3)*, 41–49. https://doi.org/10.62951/ijcts.v2i3.291

8. Kumar, P. (2025). Securing Data Privacy in Multi-Cloud and Hybrid Cloud Environments: Advanced Threat Modeling and Mitigation Strategies. *J. Emerg. Tech. Innov. Res., 12(5)*. https://doi.org/10.56975/jetir.v12i5.563399
9. Rajesh, V., & Adapa, K. (2025). Securing Multi-Cloud Architectures: A Framework for Advanced Cybersecurity Strategies in Hybrid Environments. *Int. Res. J. Modern. Eng. Tech. Sci*. https://doi.org/10.56726/irjmets66791
10. Khan, S. (2025). Enhancing Cloud Data Security using a Hybrid Cryptographic Model: A Combination of Advanced Encryption Standard and Elliptic Curve Cryptography. *J. Inf. Syst. Eng. Manag., 10(34s)*, 01–13. https://doi.org/10.52783/jisem.v10i34s.5770
11. Akbar, M., Waseem, M. M., Mehanoor, S. H., & Barmavatu, P. (2024). Blockchain-based Cyber-Security Trust Model with Multi-Risk Protection Scheme for Secure Data Transmission in Cloud Computing. *Cluster Comput., 27(7)*, 9091–9105. https://doi.org/10.1007/s10586-024-04481-9
12. V A, N. (2025). Integrating Blockchain and Quantum Cryptography in Hybrid Security Models for Cloud Systems. *Int. J. Res. Appl. Sci. Eng. Tech., 13(4)*, 4608–4615. https://doi.org/10.22214/ijraset.2025.69333
13. Steinert, T., Kuschewski, M., & Leis, V. (2026). Cloudspecs: Cloud Hardware Evolution Through the Looking Glass. [Preprint].
14. Garg, Mr. Y., & Gupta, Ms. B. (2023). Performance Analysis and Comparison of the Micro Virtual Machines Provided by the Top Cloud Vendors. *Int. J. Res. Publ. Rev., 04(02)*, 1326–1333. https://doi.org/10.55248/gengpi.2023.4229
15. Sajjad, M., Arshad, A., & Khan, A. S. (2018). Performance Evaluation of Cloud Computing Resources. *Int. J. Adv. Comput. Sci. Appl., 9(8)*. https://doi.org/10.14569/ijacsa.2018.090824
16. Mason, K., Duggan, M., Barrett, E., Duggan, J., & Howley, E. (2018). Predicting Host CPU Utilization in the Cloud using Evolutionary Neural Networks. *Futur. Gener. Comput. Syst., 86*, 162–173. https://doi.org/10.1016/j.future.2018.03.040
17. Malawski, M., Gajek, A., Zima, A., Balis, B., & Figiela, K. (2020). Serverless Execution of Scientific Workflows: Experiments with Hyperflow, AWS Lambda and Google Cloud Functions. *Futur. Gener. Comput. Syst., 110*, 502–514. https://doi.org/10.1016/j.future.2017.10.029
18. Chowdhury, S. (2025). Serverless Computing: A Comparative Performance Analysis between AWS Lambda, Google Cloud Functions, and Microsoft Azure Functions. https://www.doria.fi/handle/10024/190693
19. Ali, M. A. B. (2023). Serverless Computing for Big Data Analytics: Performance and Cost Analysis of AWS Lambda and Google Cloud Functions. *J. Data Mining, Knowl. Discov. Decis. Support Syst., 13(2)*, 1–11.
20. Baskar, R., Niranjanamurthy, M., Vinoth, N., Chitra, D., & Pushparaj, M. R. R. (2025). Enhancing Microservice Scalability with Serverless Architectures: An Analysis of AWS Lambda and Google Cloud Functions. In *2025 Int. Conf. on Emerging Technologies in Engineering Applications (ICETEA)* (pp. 1–6). IEEE. https://doi.org/10.1109/icetea64585.2025.11099699
21. Tchernykh, A., Schwiegelsohn, U., Talbi, E. G., & Babenko, M. (2019). Towards Understanding Uncertainty in Cloud Computing with Risks of Confidentiality, Integrity, and Availability. *J. Comput. Sci., 36*, 100581. https://doi.org/10.1016/j.jocs.2016.11.011
22. Bhushan, K., & Gupta, B. B. (2017). Security Challenges in Cloud Computing: State-of-Art. *Int. J. Big Data Intell., 4(2)*, 81–107. https://doi.org/10.1504/ijbdi.2017.083116
23. Iosup, A., Yigitbasi, N., & Epema, D. (2011). On the Performance Variability of Production Cloud Services. In *2011 11th IEEE/ACM Int. Symposium on Cluster, Cloud and Grid Computing* (pp. 104–113). IEEE. https://doi.org/10.1109/ccgrid.2011.22
24. Sun, Y., Lin, F., & Xu, H. (2018). Multi-Objective Optimization of Resource Scheduling in Fog Computing using an Improved NSGA-II. *Wirel. Pers. Commun., 102*, 1369–1385. https://doi.org/10.1007/s11277-017-5200-5
25. Chołodowicz, E., & Orłowski, P. (2017). Comparison of SPEA2 and NSGA-II Applied to Automatic Inventory Control System using Hypervolume Indicator. *Stud. Inf. Control, 26(1)*, 67–74. https://doi.org/10.24846/v26i1y201708
26. Ramesh, S., Kannan, S., & Baskar, S. (2012). Application of Modified NSGA-II Algorithm to Multi-Objective Reactive Power Planning. *Appl. Soft Comput., 12(2)*, 741–753. https://doi.org/10.1016/j.asoc.2011.09.015
27. Faigle, U., & Kern, W. (1992). The Shapley Value for Cooperative Games under Precedence Constraints. *Int. J. Game Theor., 21*, 249–266.

28. Teymourifar, A., Rodrigues, A. M., & Ferreira, J. S. (2020). A Comparison between NSGA-II and NSGA-III to Solve Multi-Objective Sectorization Problems based on Statistical Parameter Tuning. In *2020 24th Int. Conf. on Circuits, Systems, Communications and Computers (CSCC)* (pp. 64–74). IEEE. https://doi.org/10.1109/cscc49995.2020.00020
29. Driscoll, W. C. (1996). Robustness of the ANOVA and Tukey-Kramer Statistical Tests. *Comput. Ind. Eng., 31(1–2)*, 265–268. https://doi.org/10.1016/0360-8352(96)00127-1
30. Saxena, D., Gupta, I., Kumar, J., Singh, A. K., & Wen, X. (2021). A Secure and Multiobjective Virtual Machine Placement Framework for Cloud Data Center. *IEEE Syst. J., 16(2)*, 3163–3174. https://doi.org/10.1109/jsyst.2021.3092521
31. Han, J., Zang, W., Liu, L., Chen, S., & Yu, M. (2018). Risk-Aware Multi-Objective Optimized Virtual Machine Placement in the Cloud. *J. Comput. Secur., 26(5)*, 707–730. https://doi.org/10.3233/jcs-171104
32. Wilczyński, A., & Jakóbik, A. (2017). Using Polymatrix Extensive Stackelberg Games in Security-Aware Resource Allocation and Task Scheduling in Computational Clouds. *J. Telecommun. Inf. Technol., 1*, 71–80. https://doi.org/10.26636/jtit.2017.1.653
33. Ait Temghart, A., Marwan, M., & Baslam, M. (2023). Stackelberg Security Game for Optimizing Cybersecurity Decisions in Cloud Computing. *Secur. Commun. Netw., 2023(1)*, 1–13. https://doi.org/10.1155/2023/2811038
34. Xu, X., & Yu, H. (2014). A Game Theory Approach to Fair and Efficient Resource Allocation in Cloud Computing. *Math. Probl. Eng., 2014(1)*, 915878. https://doi.org/10.1155/2014/915878
35. Mangalagowri, R., & Venkataraman, R. (2023). Randomized MILP framework for Securing Virtual Machines from Malware Attacks. *Intell. Autom. Soft Comput., 35(2)*, 1565–1580. https://doi.org/10.32604/iasc.2023.026360
36. Gill, S. S., & Buyya, R. (2018). SECURE: Self-Protection Approach in Cloud Resource Management. *IEEE Cloud Comput., 5(1)*, 60–72. https://doi.org/10.1109/mcc.2018.011791715
37. Petrovska, I. Y. (2023). Methods of Resource Allocation in Computer Systems when Providing Cloud Infrastructure Services. Dissertation for the Degree of Doctor of Philosophy. Kharkiv.
38. Petrovska, I. Y., Kuchuk, N. G., Panchenko, V. I., & Filonenko, A. M. (2019). Uniform Resource Allocation of Computer Systems with Hyperconverged Infrastructure. *Control. Nav. Commun. Syst., 2(54)*, 119–122. https://doi.org/10.26906/SUNZ.2019.2.119
39. Petrovska, I. Y., Kolomiytsev O. V., & Ali, A. F. (2021). Method for Calculating the Buffer Memory Size of a Self-Healing Segment of a Telecommunication Network. *Control. Nav. Commun. Syst., 2(64)*, 144–147. https://doi.org/10.26906/SUNZ.2021.2.144
40. Petrovska, I., & Kuchuk, H. (2022). Static Allocation Method in a Cloud Environment with a Service Model IAAS. *Adv. Inf. Syst., 6(3)*, 99–105. https://doi.org/10.20998/2522-9052.2022.3.13
41. Petrovska, I. Yu., Kuchuk, G. A. (2022). Allocation of Computing Resources in Cloud Systems. *Control. Nav. Commun. Syst., 2(68)*, 75–78.
42. Petrovska, I, Kuchuk Heorhii. (2023). Adaptive Resource Allocation Method for Data Processing and Security in Cloud Environment. *Adv. Inf. Syst., 7(3)*, 67–73. https://doi.org/10.20998/2522-9052.2023.3.10