КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (28), 2025
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2025.28.836
УДК 004.056

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

МОДЕЛЬ РОЗПОДІЛУ ОБЧИСЛЮВАЛЬНИХ ЗАДАЧ У ХМАРНІЙ ІНФРАСТРУКТУРІ З УРАХУВАННЯМ ПРОДУКТИВНОСТІ, ВАРТОСТІ ТА БЕЗПЕКИ

Анотація. Хмарні системи (ХС), які наразі стали часткою бізнес-процесів багатьох компаній та установ, стикаються зі складними викликами, включаючи цілеспрямовані атаки, експлуатацію вразливостей програмного забезпечення, та ризики витоку конфіденційних даних, що суттєво підвищує вимоги до безпеки та ефективності розподілу ресурсів. В умовах інтенсифікації кіберзагроз та зростання вимог до ефективності ХС релевантною є розробка нових моделей розподілу обчислювальних задач, які одночасно враховують критерії безпеки, продуктивності та вартості хмарної інфраструктури. Наявні теоретичні підходи часто аналізують ці фактори ізольовано. При цьому не враховується взаємодія між атакуючим і захисником ХС. А це обмежує практичну придатність таких моделей. У цій статті запропоновано гібридну модель, яка поєднує теорію антагоністичних ігор для оцінки ризиків хмарним системам із багатокритеріальною оптимізацією на основі модифікованого алгоритму NSGA-II. Стратегії атакуючого формалізуються через параметр агресивності λ, що впливає на ймовірність атаки вузлів, а захисника — через адаптивні механізми розподілу задач. Оптимізаційна задача мінімізує три цільові функції — сумарний ризик розміщення задач (безпека ХС), сумарний час обробки задач (продуктивність ХС), та сумарна вартість використання ресурсів (економічність ХС). Симуляція в моделі середовищі Python підтвердила ефективність підходу. Значення метрик якості дорівнює IGD = 0.2263, Spacing = 0.0106, Hypervolume ≈ 1.3310. Такі результати свідчать про збіжність, рівномірність розподілу та різноманітність Парето-оптимальних рішень для захищеної ХС. Запропонована модель забезпечує гнучкий компроміс між конфліктними критеріями та адаптується до різних сценаріїв поведінки зловмисника.

Ключові слова: хмарні обчислення; розподіл задач; багатокритеріальна оптимізація; теорія ігор; кібербезпека; ризик, алгоритм NSGA-II; Парето-оптимальність.

ВСТУП

Хмарні інфраструктури стали вагомим компонентом інформаційних систем. Отже вони потребують ефективного управління обчислювальними ресурсами [1], [2]. Однак характер кіберзагроз, суперечливі вимоги до продуктивності та вартості хмарних систем (ХС), а також обмеженість адаптації існуючих оптимізаційних методів до реальних умов експлуатації ХС ускладнюють пошук збалансованих рішень. Тому робота спрямована на подолання фрагментарності в дослідженнях шляхом інтеграції механізмів оцінки ризиків, заснованих на взаємодії суб’єктів захисту та атаки, із методами багатокритеріальної оптимізації. Наукова новизна полягає у поєднанні ігрової моделі з еволюційними алгоритмами для врахування фактору безпеки у процесі розподілу ресурсів ХС.

Постановка проблеми. Проблематика дослідження визначається двома ключовими аспектами. По-перше, більшість проаналізованих публікацій аналізує безпеку, продуктивність і вартість хмарних систем ізольовано, не враховуючи їхню взаємозалежність та вплив дій зловмисника (хакера). Наприклад, алгоритми на кшталт NSGA-II демонструють ефективність у оптимізації ресурсів, але не інтегрують оцінку ризиків для ХС, пов'язаних із цілеспрямованими атаками. По-друге, існуючі моделі часто ігнорують адаптивність до змінних умов експлуатації, таких як коливання навантаження чи еволюція загроз (наприклад, EDoS, експлуатація вразливостей), що обмежує їхню практичну цінність. Ці прогалини обумовлюють необхідність розробки гібридних підходів, здатних забезпечити комплексний аналіз безпеки ХС, продуктивності та економічних показників.

Аналіз останніх досліджень і публікацій. Дослідження в галузі безпеки хмарних систем концентруються на аналізі фундаментальних загроз, зокрема DDoS-атак [2] та багаторівневої класифікації ризиків [3]–[6]. Оптимізація розподілу ресурсів досліджується в [7]–[9], де алгоритми на кшталт NSGA-II [7]–[11] демонструють значне покращення продуктивності порівняно з традиційними методами. Суттєвий внесок у оцінку якості оптимізаційних рішень вносять метрики Hypervolume [9] та вдосконалений IGD [10].

Проте існують значні прогалини у цих дослідженнях. Більшість робіт розглядає безпеку [3], [4] та продуктивність [7], [8] ізольовано, не враховуючи їх комбінований вплив цих факторів. Відзначимо, що існуючі оптимізаційні алгоритми [7]–[9], [11] недостатньо адаптовані до реальних умов хмарних середовищ із змінним навантаженням та релевантними загрозами. Це підтверджується обмеженими результатами апробації в роботах [8]–[15]. Таким чином, актуальним завданням є розробка інтегрованих моделей, що поєднують адаптивну оцінку ризиків та багатокритеріальну оптимізацію ресурсів з урахуванням змін хмарного середовища.

Мета статті. Мета — розробка інтегрованої оптимізаційної моделі розподілу обчислювальних задач у хмарній системі (ХС), яка одночасно мінімізує сумарний ризик кіберзагроз, час виконання задач та вартість використання ресурсів.

Для досягнення цього вирішуються такі завдання:

формалізувати взаємодію між атакуючим і захисником за допомогою антагоністичної ігрової моделі з параметром агресивності (λ), що кількісно визначає ймовірність атак на вузли через модифіковану softmax-функцію;

інтегрувати оцінку ризиків, отриману з ігрової компоненти, у багатокритеріальну оптимізаційну задачу з критеріями безпеки ХС;

реалізувати гібридний алгоритм на основі модифікованого NSGA-II для пошуку Парето-оптимальних рішень, здатних адаптуватися до змінних умов (навантаження, типи загроз);

експериментально довести ефективність моделі за метриками якості (Hypervolume, IGD, Spacing) у середовищі Python.

МЕТОДИ ТА МОДЕЛІ

Нехай маємо множину обчислювальних вузлів хмарної інфраструктури:

$$N = \{1,2, \dots, n\}. \quad (1)$$

$$T = \{T_1, T_2, \dots, T_m\}. \quad (2)$$

Етап 1. Ігрова модель моделювання атакуючого та захисника.

Розглядаємо двох гравців:
захисник (індекс — D (Defender));
атакуючий (індекс — A (Attacker)).
Визначимо стратегії гравців.
Стратегія захисника. Розподіл задач $x_{ij} \in \{0,1\}$, де

$$x_{ij} = \begin{cases} 1, \text{якщо задача } T_i \text{ розміщена на вузлі } i, \\ 0, \text{інакше.} \end{cases} \quad (3)$$

Стратегія атакуючого. Вибір вузлів для атаки задач $a_i \in \{0,1\}$, де

$$a_i = \begin{cases} 1, \text{якщо атакуючий атакуя вузол } i, \\ 0, \text{інакше.} \end{cases} \quad (4)$$

Тоді для забезпечення безпеки ХС актуальним є моделювання стратегічної взаємодії між учасниками, які мають суперечливі цілі. Для формалізації цієї взаємодії використовується підхід теорії антагоністичних ігор. Це дозволяє врахувати можливу поведінку зловмисника в процесі планування захисних заходів та розподілу обчислювальних ресурсів. Зазначимо, що ігрова модель базується на припущенні, що обидва гравці діють раціонально в умовах конфлікту. Ціль атакуючого полягає у максимізації шкоди, завданої ХС, через компрометацію обчислювальних вузлів або порушення функціонування сервісів. Водночас, захисник прагне мінімізувати ймовірні втрати, реалізуючи стратегії захисту та розподілу задач з урахуванням ризиків. У запропонованій моделі розглядається скінченна множина стратегій кожного гравця.

Таблиця 1

Порівняльний аналіз стратегій атакуючого та захисника у хмарних системах (складено автором на підставі аналізу літератури [1]–[4])

Категорія	Стратегія атакуючого	Стратегія захисника	Приклад реальної атаки/захисту
Навантаження системи	Атака DDoS на вузли з піковим навантаженням.	Балансування навантаження, використання резервних вузлів.	Атака на AWS (2020), яка призвела до збоїв через перевантаження.
Вразливості ПЗ	Експлуатація сервісів, які не оновили механізми безпеки.	Автоматизоване оновлення та моніторинг вразливостей.	Використання CVE-2021-44228 (Log4j) для несанкціонованого доступу.
Конфіденційність даних	Цілеспрямований витік даних із вузлів.	Шифрування даних, сегментація мережі.	Витік даних із хмарного сховища Microsoft Azure (2019).
Адаптивний захист ХС	Адаптація до заходів безпеки.	Маскування ресурсів, зміна IP-адрес вузлів.	Уникнення виявлення під час атак на хмарні Kubernetes-кластери.

Стратегії атакуючого $S_A = \{a_1, a, \dots, a_m\}$ відповідають різним типам атак, спрямованих на певні вузли або сервіси ХС. Наприклад, це можуть бути [2], [3]:

a_1 — атака на вузли з високим навантаженням (DoS);
a_2 — цілеспрямована атака на вузли з високою цінністю даних (APT);
a_3 — експлуатація вразливостей в ПЗ з низьким рівнем оновлення;
тощо.

Стратегії захисника $S_D = \{d_1, d, \dots, d_n\}$ моделюють різні конфігурації розподілу задач та механізми захисту:

d_1 — реплікація задач для зменшення ймовірності повної втрати;

d_2 — мігрування задач у відповідь на загрозу;

d_3 — приоритезація задач залежно від критичності та вартості;

тощо.

Стратегії атакуючого базуються на виборі цілей, які забезпечують максимальний ефект при мінімальних витратах. Наприклад, атаки типу DDoS спрямовані на вузли з високим навантаженням. Це призводить до їхньої недоступності та порушення роботи критичних сервісів ХС. Іншим прикладом є атаки на вразливе програмне забезпечення (Exploit-Based Attacks), де зловмисник експлуатує недоліки в системах віртуалізації або керування хмарою для отримання несанкціонованого доступу. Також поширені атаки на конфіденційність даних, які передбачають цілеспрямований витік інформації з критичних вузлів, що обробляють чутливі дані, див. табл. 1.

Захисник використовує адаптивні механізми для протидії атакам, зокрема розподіл задач для уникнення перевантаження окремих вузлів ХС. Іншим ефективним підходом є реплікація критичних сервісів. Це дозволяє зменшити ймовірність повного виведення системи з ладу. Також застосовуються механізми пріоритезації, які забезпечують обслуговування найважливіших задач на найбільш захищених вузлах.

Функція втрат $L(d_i, a_j)$ є відображенням у простір дійсних чисел ($\mathbb{R}$), яке визначає потенційні втрати захисника в результаті застосування певної пари стратегій [4], [5]: $L = S_D \times S_A \longrightarrow \mathbb{R}^+$, $L(d_i, a_j) =$ очікувані втрати при захисті d_i та атаці a_j.

Згідно [4], [5] ця функція може включати компоненти, пов’язані з: прямими фінансовими втратами (порушення SLA); втратами конфіденційності чи цілісності даних; витратами на відновлення після атаки.

Тоді функція виграшу атакуючого, подамо так [4], [5]:

$$U_A = \sum_{i=1}^{n} a_i \cdot D_i \tag{5}$$

де D_i — очікувані збитки внаслідок успішної атаки на вузол i.

Функція виграшу атакуючого подано так [4], [5]:

$$U_D = -\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} \cdot a_i \cdot D_i \tag{6}$$

Після формалізації стратегічної взаємодії між атакуючим та захисником у межах ігрової моделі наступним логічним етапом є побудова кількісної оцінки ризику для кожного вузла хмарної інфраструктури. Отримані в результаті гри стратегії гравців, а також їхні очікувані виграші, будуть слугувати підґрунтям для розрахунку потенційних втрат. Тобто ці потенційні втрати пов’язані із можливою компрометацією окремих обчислювальних ресурсів ХС. Подібна оцінка ризику буде відігравати ключову роль у подальшій оптимізаційній моделі. Це пов’язано із тим, що можна на наступному кроці інтегрувати вплив стратегій дій зловмисника у процес прийняття рішень щодо призначення задач. У цьому розрізі завдані дослідження ризик розглядається як функція ймовірності атаки на вузол ХС та очікуваного збитку, який завданий у разі її успішного здійснення, і слугує одним з основних критеріїв багатокритеріальної оптимізації.

На підставі цих суджень можемо записати ймовірність атаки на вузол i так:

$$P_i^{(attack)} = \frac{e^{\lambda \cdot D_i}}{\sum_{k=1}^{n} e^{\lambda \cdot D_k}}, \tag{7}$$

де λ — параметр, який характеризує агресивність атакуючого;

D_i — оцінка втрат у разі успішної атаки на вузол i.

Даний вираз є модифікованою softmax-функцією, яка трансформує вектор потенційних збитків D_i у ймовірнісний розподіл. Застосування експоненційної функції $e^{\lambda \cdot D_i}$ забезпечує невід'ємність значень та підсилює вплив вузлів із високими D_i. А нормування на суму $\sum_{k=1}^{n} e^{\lambda \cdot D_k}$ гарантує, що сума ймовірностей для всіх вузлів дорівнює одиниці, що відповідає аксіомам теорії ймовірностей.

Тобто параметр λ, який ми умовно називаємо параметром агресивності атакуючого, відіграє вагому роль у моделюванні поведінки зловмисника в межах ігрової частини нашої гібридної моделі. Інтуїтивно цей параметр визначає наскільки активно або ризиковано атакуючий діє у виборі цілей (вузлів хмарної інфраструктури). Зокрема — чи схильний він концентрувати зусилля на найбільш вразливих вузлах ХС. Чи, навпаки, розподіляє свої атаки більш рівномірно.

Зазначимо, що не всі атакуючі поводяться однаково. Деякі діють обережно, намагаючись уникнути виявлення. Інші агресивно, концентруючи атаки на найцінніших або найменш захищених цілях. Параметр λ регулює розподіл ймовірностей вибору цілей атакуючим. При малому значенні λ зловмисник атакує цілі майже випадково (низька агресивність, розосереджені атаки). Та при високому λ зловмисник майже завжди атакує вузли з найбільшим потенційним виграшом (висока агресивність — таргетовані атаки). Наведемо приклад. Припустимо, що у нас є множина вузлів $N = \{1,2,\dots,n\}$, а L_j — потенційний виграш (або збиток для захисника) у разі успішної атаки на вузол ХС. Значить ми можемо задати ймовірність того, що атакуючий обирає для атаки вузол j, за допомогою модифікованого softmax-розподілу (розподіл ймовірносі при K різних можливих варіантах), див. вираз (7) [6]. Якщо $\lambda \longrightarrow 0$, тоді

$$P_i^{(attack)} = \frac{1}{n}. \tag{8}$$

Тобто атакуючий розподіляє свої зусилля рівномірно по всіх вузлах. Це неагресивна стратегія, з погляду вибору цілей.

Якщо $\lambda \longrightarrow \infty$ тоді

$$P_i^{attack} \longrightarrow \begin{cases} 1, \text{якщо } L_j = max_k \cdot L_k, \\ 0, \text{інакше.} \end{cases}, \tag{9}$$

Це означає, що атакуючий зосереджується лише на одному найбільш вразливому/вигідному вузлі. Тобто це максимально агресивна стратегія. При помірних значеннях λ, наприклад λ=1 або λ=3, атакуючий вибирає вузли з вищим значенням L_j частіше, але не виключає інші вузли повністю. Це збалансована стратегія, яка відповідає реалістичному атакуючому, що враховує і вигоду, і ризики. Наприклад, нехай у нас є 3 вузли з такими потенційними втратами у разі атаки: $L_1 = 10; L_2 = 5; L_3 = 1$. Для $\lambda = 0{,}5$ маємо

$$P_1 = \frac{e^{0,5\cdot 10}}{e^{0,5\cdot 10}+e^{0,5\cdot 5}+e^{0,5\cdot 1}} \approx 0{,}9,\ P_2 \approx 0{,}074,\ P_{32} \approx 0{,}01.$$

У нашій гібридній моделі параметр λ дозволяє моделювати різні сценарії поведінки зловмисника. Від хаотичних атак до цілеспрямованих високоризикових операцій. Тоді для одного рівня λ оптимальні стратегії розподілу задач будуть одні, для іншого кардинально інші. У подальшій оптимізації (через NSGA-II) ці оцінки атакуючого напряму впливають на функцію ризику, яка, в свою чергу, є складовою одного з критеріїв багатокритеріальної задачі.

Можна стверджувати, що функція (7) відображає принцип раціонального вибору атакуючого в умовах обмежених ресурсів. Зловмисник оптимізує свої дії, максимізуючи очікуваний виграш $U_A = \sum_{i=1}^{n} a_i \cdot D_i$, де $a_i \in \{0,1\}$ - індикатор атаки. Вираз (7) фактично

є рішенням цієї задачі раціонального вибору атакуючого при обмежені $\sum_{i=1}^{n} a_i = 1$ (атака на один вузол ХС за раз).

Ризик виконання задачі на вузлі i

$$R_i = P_i^{(attack)} \cdot D_i, \tag{10}$$

R_i — очікуваний ризик використання вузла i.

На підставі побудованої ігрової моделі (1) — (10), що відображає стратегічну взаємодію між атакуючим та захисником, а також сформованої на її основі оцінки ризику кожного вузла хмарної інфраструктури (10), на наступному кроці доцільним є перехід до формалізації задачі багатокритеріальної оптимізації. Отримані значення ризику (10) відображають потенційні загрози, що виникають у результаті дій зловмисника. Тобто вони виступають одним із ключових факторів, які слід враховувати під час ухвалення рішень щодо розподілу обчислювальних задач у хмарному середовищі.

Проте, окрім безпекових складових, ефективне функціонування хмарної інфраструктури передбачає також досягнення високої продуктивності, мінімізацію часу обробки та оптимізацію вартості використання ресурсів. Формалізація задачі зводиться до багатокритеріального підходу, в якому необхідно одночасно враховувати суперечливі цілі, зокрема забезпечення безпеки, продуктивності та економічної доцільності захисту вузлів ХС. Тоді наступним кроком є побудова частини математичної моделі, яка дозволить знайти компромісні рішення з урахуванням усіх зазначених критеріїв.

Шукаємо такий розподіл задач $x = \{x_{ij}\}$, який мінімізує векторну функцію (11):

$$\begin{cases} F_1(x) = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \cdot R_i; \\ F_2(x) = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \cdot T_{ij}; \\ F_3(x) = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \cdot C_{ij}; \end{cases} \tag{11}$$

де $F_1(x)$ — сумарний ризик розміщення задач (безпека ХС);

$F_2(x)$ — сумарний час обробки задач (продуктивність ХС);

$F_3(x)$ — сумарна вартість використання ресурсів (економічність ХС);

T_{ij} — час виконання задачі T_i на вузлі i;

C_{ij} — вартість виконання задачі на вузлі i.

Зазначимо, що з огляду на сформульовану багатокритеріальну постановку задачі, яка враховує ключові параметри — безпеку, продуктивність та вартість розподілу обчислювальних задач у хмарній інфраструктурі, необхідним є визначення обмежень. Ці обмеження формалізують технічні, функціональні та логічні характеристики досліджуваного ХС. У межах оптимізаційної частини моделі обмеження відіграють значну роль. Це пов’язано із тим, що саме вони забезпечують коректність та практичну доцільність знайдених рішень, унеможливлюючи порушення допустимих меж використання ресурсів, перевищення навантаження на вузли ХС або призначення задач у непридатні до обробки умови. Крім того, обмеження дозволяють ураховувати специфіку поведінки атакуючого та обмеження захисника у виборі захисних стратегій. Формалізація цих обмежень є необхідним етапом для побудови повної моделі, яка дозволить адекватно відображати

реальні умови експлуатації ХС та забезпечить можливість застосування алгоритмів оптимізації до практичних сценаріїв розподілу задач.

В рамках опису обмежень кожна задача повинна бути призначена одному вузлу:

$$\sum_{i=1}^{n} x_{ij} = 1,\ \forall j \in \{1, \dots, m\}, \tag{12}$$

де n — кількість обчислювальних вузлів у хмарі;

m — кількість задач для розміщення.

Загальне навантаження на вузол ХС не повинно перевищувати його можливостей:

$$\sum_{j=1}^{m} x_{ij} \cdot L_j \le C_i^{(cap)},\ \forall i \in \{1, \dots, n\}, \tag{13}$$

де L_j — обчислювальне навантаження задачі T_i на вузлі;

$C_i^{(cap)}$ — обчислювальна потужність вузла i.

Після формалізації обмежень (12) та (13), що визначають допустимі межі та умови роботи хмарної інфраструктури для завдання забезпечення безпечного, продуктивного та економічно ефективного розподілу обчислювальних задач ХС, постає необхідність у виборі результативного підходу до розв'язання поставленої багатокритеріальної задачі. Враховуючи складність досліджуваного середовища, наявність суперечливих цілей, а також природу загроз, що моделюються в рамках ігрової взаємодії між атакуючим та захисником, використання традиційних детермінованих методів оптимізації виявляється недостатнім або малоефективним. Це зумовлює необхідність залучення релевантних евристичних підходів, здатних знаходити компромісні розв'язки у складних, багатовимірних просторах рішень з множинними цілями та обмеженнями.

Як показано [1]–[10] релевантною є задача побудови гібридного алгоритму, який поєднує можливості теоретико-ігрової оцінки ризику для ХС із потужним механізмом багатокритеріальної еволюційної оптимізації. Подібний підхід дозволить інтегрувати стратегічну інформацію про потенційні дії зловмисника безпосередньо в процес оптимального розподілу задач. А це, у свою чергу, підвищує адаптивність і стійкість ХС до змін у ландшафті кібернетичних загроз для ХС. У якості оптимізаційного ядра пропонується застосування модифікованого алгоритму NSGA-II [7], [8]. Як зазначено у першому розділі роботи, а також у [7], [8] NSGA-II, є ефективним при розв'язанні багатокритеріальних задач, а також забезпечує генерацію Парето-оптимального фронту рішень.

Уведення ігрової компоненти у вигляді параметрів ризику, що змінюються залежно від стратегій гравців, на нашу думку, створює умови для детального врахування безпекових факторів у процесі оптимізації.

А відтак, наступним кроком є побудова гібридного алгоритму, див. рис. 1, структура і механізми якого забезпечать поєднання результатів стратегічного аналізу з багатокритеріальним еволюційним пошуком ефективних конфігурацій розподілу ресурсів.

У рамках побудованої моделі гібридний алгоритм оптимізації, див. рис. 1, виконує роль обчислювального ядра, яке інтегрує результати стратегічного аналізу з боку потенційного атакуючого та захисника з можливістю пошуку оптимального балансу між суперечливими критеріями — безпекою, продуктивністю та вартістю. В основі алгоритму лежить ідея циклічної взаємодії між оцінкою ризику (як вихідної функції ігрової моделі) та багатокритеріальним еволюційним пошуком, який реалізується за допомогою модифікованого алгоритму NSGA-II.

Робота гібридного алгоритму окреслює декілька взаємопов'язаних фаз. На початковому етапі здійснюється генерація початкової популяції можливих розподілів

обчислювальних задач між вузлами хмарної інфраструктури. Для кожного потенційного рішення, що описує конфігурацію розподілу, проводиться розрахунок рівня ризику з урахуванням стратегічної поведінки атакуючого, яка моделюється через параметр агресивності та ймовірнісну функцію вибору цілей. Таким чином, кожен індивід у популяції оцінюється не лише за критеріями продуктивності й вартості, але також із позиції стійкості до потенційних атак.

На наступній фазі виконується багатокритеріальний відбір за принципом Парето-домінування, з урахуванням оновлених функцій придатності, які включають у себе як традиційні техніко-економічні метрики, так і показники ризику.

Відбір, схрещування та мутація здійснюються згідно до механізмів NSGA-II. Це дозволяє результативне досліджувати простір рішень та уникати локальних екстремумів. На кожній ітерації алгоритму відбувається повторна оцінка рівня ризику згідно до зміненої конфігурації розподілу задач, що забезпечує гнучку інтеграцію безпекових факторів у процес оптимізації.

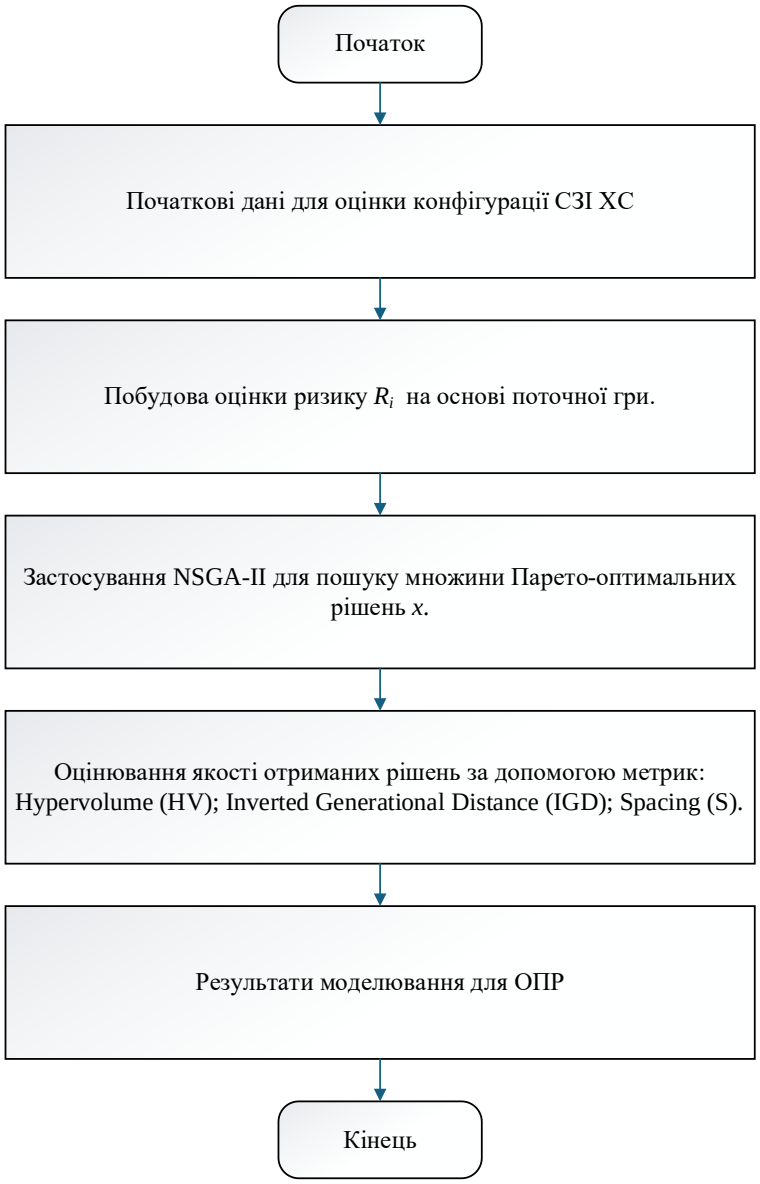

Рис. 1. Концептуальна схема гібридного алгоритму

Гібридна природа алгоритму полягає саме в постійному обміні інформацією між ігровим компонентом, який моделює зміну поведінки атакуючого, та еволюційною частиною, що відповідає за поступове вдосконалення стратегій розподілу. Запропонований підхід, на нашу думку, дозволяє не лише отримати набір Парето-оптимальних рішень, але й забезпечити, щоб ці рішення були релевантними до актуальних загроз ХС. Це, відображає компроміс між продуктивністю, безпекою та вартісними показниками ХС.

Результатом роботи гібридного алгоритму є фінальна популяція рішень, що формує фронт Парето, на підставі якого зацікавлені сторони (адміністратори, системні інтегратори, аналітики з кібербезпеки ХС) можуть здійснювати вибір оптимальної конфігурації системи згідно до поточних пріоритетів або обмежень. Подібне рішення надає не лише гнучкість і масштабованість, але й враховує поведінковий характер кібернетичних загроз ХС, що суттєво підвищує рівень реалізму та практичної значущості моделі.

Тоді можна нашу гібридну модель формалізувати так.

Приймемо такі позначення для параметрів моделі:

$N -$ кількість вузлів хмарної інфраструктури;

$M -$ кількість обчислювальних задач;

$x_{ij} \in \{0,1\} -$ бінарна змінна, що вказує, чи призначена задача j вузлу i;

$w_j -$ обсяг ресурсів, необхідний для виконання задачі j;

$c_i -$ продуктивність обчислювального вузла i;

$cost_i -$ вартість використання вузла i за одиницю ресурсу;

$C_i -$ загальна обчислювальна ємність вузла i;

$u_i -$ доцільність атаки на вузол i з точки зору атакуючого (може враховувати значущість даних, критичність функцій тощо);

$\lambda -$ параметр агресивності атакуючого, що моделює рівень його націленості на високоризикові або легкодоступні вузли;

$p_i -$ ймовірність атаки на вузол i;

$L_i -$ потенційні збитки у разі успішної атаки на вузол i;

$R_i -$ очікуваний ризик для вузла i;

$F_1(x) -$ критерій мінімізації ризику (максимізація безпеки);

$F_2(x) -$ критерій мінімізації загального часу обробки;

$F_3(x) -$ критерій мінімізації вартості використання хмарних ресурсів.

Етап 1. Ігрова модель — ймовірність атаки на вузол ХС.

$$p_i = \frac{e^{\lambda \cdot u_i}}{\sum_{j=1}^{N} e^{\lambda \cdot u_i}}, i = 1,2, \dots, N. \quad (14)$$

Етап 2. Оцінка очікуваного ризику атаки на вузол i.

$$R_i = p_i \cdot L_i. \quad (15)$$

Етап 3. Функція безпеки ХС для розподілу задач.

$$F_1(x) = \sum_{i=1}^{N} R_i \cdot \left(\sum_{j=1}^{M} x_{ij} \cdot w_j \right) \quad (16)$$

Етап 4. Функція продуктивності (мінімізація часу обробки).

$$F_2(x) = \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij} \cdot \frac{w_j}{c_i}. \quad (17)$$

Етап 5. Функція вартості (вартість використання обчислювальних вузлів).

$$F_3(x) = \sum_{i=1}^{N}\sum_{j=1}^{M} x_{ij} \cdot w_j \cdot cost_i. \quad (18)$$

Етап 6. Обмеження ємності вузлів.

$$\sum_{j=1}^{M} x_{ij} \cdot w_j \le C_i, \forall i \in \{1,2,\dots,N\}. \quad (19)$$

Етап 7. Повна розподіленість задач у ХС.

$$\sum_{i=1}^{N} x_{ij} = 1, \forall j \in \{1,2,\dots,M\}. \quad (20)$$

Етап 8. Двоїстість змінних розподілу.

$$x_{ij} \in \{0,1\}, \forall i,j. \quad (21)$$

Етап 9. Задача багатокритеріальної оптимізації.

$$min\big(F_1(x), F_2(x), F_3(x)\big) \quad (22)$$

Етап 10. Оцінка якості рішення.

10.1. Оцінка Hypervolume (HV) [9]:

$$HV = \text{об'єм простору, домінованого фронтом Парето.} \quad (23)$$

10.2. Inverted Generational Distance (IGD) [10]:

$$IGD = \frac{1}{|P^*|}\sum_{v\in P} \min_{u\in P}\|u - v\|, \quad (24)$$

де $P -$ отриманий Парето-фронт;

$P^* -$ справжній (референсний) Парето-фронт;

$d(v,u) -$ відстань між точками v і u.

10.3. Spacing (S) [11]:

$$S = \sqrt{\frac{1}{|P|-1}\sum_{i=1}^{|P|}\big(d_i - \bar{d}\big)^2} \quad (25)$$

де $d_i -$ відстань між рішенням i та його найближчим сусідом;

$\bar{d} -$ середнє значення d_i.

Відповідно моделі була проведена симуляція роботи моделі, в середовищі програмування PyCharm, див. рис. 2.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Отримані значення метрик якості, демонструють дієвість запропонованого гібридного підходу до оптимізації розподілу обчислювальних ресурсів у хмарному середовищі із урахуванням ризиків для безпеки. Значення метрики Inverted Generational Distance (IGD) на рівні 0.2263 свідчить про відносно невелику середню відстань між знайденим Парето-оптимальним фронтом та еталонним набором рішень. Це вказує на задовільну збіжність алгоритму. Метрика Spacing, яка становить 0.0106, підтверджує рівномірний розподіл знайдених рішень у просторі критеріїв. Таке низьке значення свідчить про відсутність значних скупчень рішень у локальних областях.

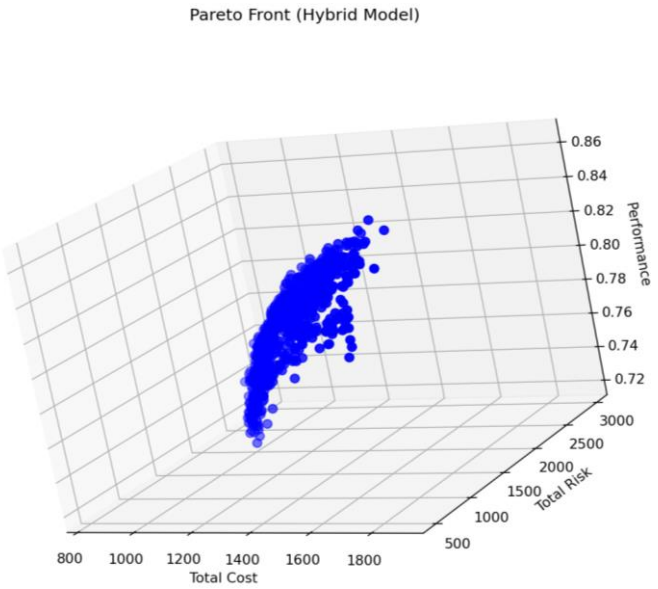

Рис. 2. Результати симуляції для запропонованої гібридної моделі та візуалізація Парето-оптимальних рішень

Апроксимоване значення гіпероб'єму (Hypervolume) 1.3310 відображає обсяг простору критеріїв, який домінують знайдені рішення. Ця величина, хоча й є приблизною, дозволяє зробити висновок про достатню різноманітність отриманого набору Парето-оптимальних рішень.

Загалом, отримані результати підтверджують працездатність запропонованого підходу.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Результати дослідження підтверджують ефективність запропонованої гібридної моделі для оптимізації розподілу обчислювальних задач у хмарних середовищах (ХС). Поєднання теорії ігор (з параметром агресивності λ) та модифікованого алгоритму NSGA-II дозволило сформувати множину Парето-оптимальних рішень, що забезпечують баланс між суперечливими критеріями оптимізації ХС. Симуляція показала збіжність підходу IGD = 0.2263, рівномірний розподіл рішень у просторі цільових функцій Spacing = 0.0106 та їхню різноманітність Hypervolume ≈ 1.3310. Перевагою моделі є адаптивність до різних сценаріїв поведінки атакуючого через регулювання λ. Майбутні дослідження мають спрямувати на інтеграцію статистичних даних про атаки на ХС, а також розширення ігрової компоненти для колективних стратегій зловмисників та оптимізацію обчислювальної складності алгоритму для великомасштабних хмарних середовищ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Mykhailiv, V. I. (2015). Experimental research of information technology for data protection in cloud computing systems. *Current Issues of Automation and Information Technologies, (19),* 52–66.
2. Kobevko, A. T., & Tymchenko, O. V. (2019). Features of DDoS attacks on cloud services. *Mechanical Engineering, 11(3),* 1–15.

3. Hussain, S. A., Fatima, M., Saeed, A., Raza, I., & Shahzad, R. K. (2017). Multilevel classification of security concerns in cloud computing. *Applied Computing and Informatics, 13(1),* 57–65.
4. Hosseini, S., & Vakili, R. (2019). Game theory approach for detecting vulnerable data centers in cloud computing network. *International Journal of Communication Systems, 32(8), e3938*.
5. Kakkad, V., Shah, H., Patel, R., & Doshi, N. (2019). A comparative study of applications of game theory in cyber security and cloud computing. *Procedia Computer Science, 155,* 680–685.
6. Banerjee, K., Gupta, R. R., Vyas, K., & Mishra, B. (2020). *Exploring alternatives to softmax function.* arXiv preprint arXiv:2011.11538. https://arxiv.org/abs/2011.11538
7. Sun, Y., Lin, F., & Xu, H. (2018). Multi-objective optimization of resource scheduling in fog computing using an improved NSGA-II. *Wireless Personal Communications, 102,* 1369–1385.
8. Tsai, J. T., Fang, J. C., & Chou, J. H. (2013). Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. *Computers & Operations Research, 40(12),* 3045–3055.
9. Chołodowicz, E., & Orłowski, P. (2017). Comparison of SPEA2 and NSGA-II applied to automatic inventory control system using hypervolume indicator. *Studies in Informatics and Control, 26(1),* 67–74.
10. Tian, Y., Zhang, X., Cheng, R., & Jin, Y. (2016, July). A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric. *In 2016 IEEE Congress on Evolutionary Computation (CEC),* 5222–5229. IEEE. https://doi.org/10.1109/CEC.2016.7748362
11. Ramesh, S., Kannan, S., & Baskar, S. (2012). Application of modified NSGA-II algorithm to multi-objective reactive power planning. *Applied Soft Computing, 12(2),* 741–753.
12. Glorfeld, L. W., & Palko, J. (1988). A comparison of novice algorithm composition performance using flowcharting or pseudocode as the design tool. *Journal of Research on Computing in Education, 21(1),* 82–96.
13. Vayadande, K., Bailke, P. A., Dombale, A. B., Dange, V. R., & Kulkarni, A. M. (2024). Converting pseudo code to code: A review. *In How Machine Learning is Innovating Today's World: A Concise Technical Guide,* 57–68.
14. Choi, J., Choi, C., Lynn, H. M., & Kim, P. (2015, November). Ontology based APT attack behavior analysis in cloud computing. *In 2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA),* 375–379. IEEE. https://doi.org/10.1109/BWCCA.2015.69
15. Alshaikh, A., Alanesi, M., Yang, D., & Alshaikh, A. (2023, June). Advanced techniques for cyber threat intelligence-based APT detection and mitigation in cloud environments. *In International Conference on Cyber Security, Artificial Intelligence, and Digital Economy (CSAIDE 2023), 12718,* 147–157). SPIE. https://doi.org/10.1117/12.2676532

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (28), 2025
ISSN 2663 - 4023

Diana 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 ID: 0000-0002-9422-8617
d.tsyrkaniuk.asp@kubg.edu.ua

A MODEL FOR THE DISTRIBUTION OF COMPUTATIONAL TASKS IN CLOUD INFRASTRUCTURE INCORPORATING PERFORMANCE, COST, AND SECURITY CONSIDERATIONS

Abstract. Cloud systems (CSs) — now integral to the business processes of many organisations— face sophisticated challenges such as targeted attacks, exploitation of software vulnerabilities and the leakage of confidential data. These threats greatly increase the demands on both security and efficient resource allocation. Against a backdrop of escalating cyber-risks and rising performance requirements, it is essential to design new task-allocation models that simultaneously account for security, performance and cost within cloud infrastructures. Existing theoretical approaches often study these factors in isolation and ignore the strategic interaction between attacker and defender, limiting their practical usefulness.This paper presents a hybrid model that couples antagonistic game theory for cloud-risk assessment with multi-objective optimisation based on a modified NSGA-II algorithm. Attacker behaviour is represented by an aggressiveness parameter (λ) that influences the probability of node compromise, whereas defender behaviour relies on adaptive task-allocation mechanisms. The optimisation problem minimises three objectives: total task-placement risk (cloud security), total task-processing time (cloud performance) and total cost of resource usage (cloud cost-efficiency). Simulations carried out in a Python environment confirm the effectiveness of the method, yielding IGD = 0.2263, Spacing = 0.0106 and Hypervolume $\approx$ 1.3310. These metrics indicate good convergence, a uniform spread and high diversity of the Pareto-optimal front for a protected cloud system. The proposed model therefore offers a flexible trade-off among conflicting criteria and can adapt to diverse adversary-behaviour scenarios.

Keywords: cloud computing; task allocation; multi-objective optimization; game theory; cybersecurity; risk; NSGA-II algorithm; Pareto optimality.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Mykhailiv, V. I. (2015). Experimental research of information technology for data protection in cloud computing systems. *Current Issues of Automation and Information Technologies, (19),* 52–66.
2. Kobevko, A. T., & Tymchenko, O. V. (2019). Features of DDoS attacks on cloud services. *Mechanical Engineering, 11(3),* 1–15.
3. Hussain, S. A., Fatima, M., Saeed, A., Raza, I., & Shahzad, R. K. (2017). Multilevel classification of security concerns in cloud computing. *Applied Computing and Informatics, 13(1),* 57–65.
4. Hosseini, S., & Vakili, R. (2019). Game theory approach for detecting vulnerable data centers in cloud computing network. *International Journal of Communication Systems, 32(8), e3938.*
5. Kakkad, V., Shah, H., Patel, R., & Doshi, N. (2019). A comparative study of applications of game theory in cyber security and cloud computing. *Procedia Computer Science, 155,* 680–685.
6. Banerjee, K., Gupta, R. R., Vyas, K., & Mishra, B. (2020). *Exploring alternatives to softmax function.* arXiv preprint arXiv:2011.11538. https://arxiv.org/abs/2011.11538
7. Sun, Y., Lin, F., & Xu, H. (2018). Multi-objective optimization of resource scheduling in fog computing using an improved NSGA-II. *Wireless Personal Communications, 102,* 1369–1385.
8. Tsai, J. T., Fang, J. C., & Chou, J. H. (2013). Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. *Computers & Operations Research, 40(12),* 3045–3055.
9. Chołodowicz, E., & Orłowski, P. (2017). Comparison of SPEA2 and NSGA-II applied to automatic inventory control system using hypervolume indicator. *Studies in Informatics and Control, 26(1),* 67–74.

10. Tian, Y., Zhang, X., Cheng, R., & Jin, Y. (2016, July). A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric. *In 2016 IEEE Congress on Evolutionary Computation (CEC)*, 5222–5229. IEEE. https://doi.org/10.1109/CEC.2016.7748362
11. Ramesh, S., Kannan, S., & Baskar, S. (2012). Application of modified NSGA-II algorithm to multi-objective reactive power planning. *Applied Soft Computing, 12(2),* 741–753.
12. Glorfeld, L. W., & Palko, J. (1988). A comparison of novice algorithm composition performance using flowcharting or pseudocode as the design tool. *Journal of Research on Computing in Education, 21(1),* 82–96.
13. Vayadande, K., Bailke, P. A., Dombale, A. B., Dange, V. R., & Kulkarni, A. M. (2024). Converting pseudo code to code: A review. *In How Machine Learning is Innovating Today's World: A Concise Technical Guide*, 57–68.
14. Choi, J., Choi, C., Lynn, H. M., & Kim, P. (2015, November). Ontology based APT attack behavior analysis in cloud computing. *In 2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA),* 375–379. IEEE. https://doi.org/10.1109/BWCCA.2015.69
15. Alshaikh, A., Alanesi, M., Yang, D., & Alshaikh, A. (2023, June). Advanced techniques for cyber threat intelligence-based APT detection and mitigation in cloud environments. *In International Conference on Cyber Security, Artificial Intelligence, and Digital Economy (CSAIDE 2023), 12718,* 147–157). SPIE. https://doi.org/10.1117/12.2676532