

Д. Тереник, В. С. Харченко

Національний аерокосмічний університет «Харківський авіаційний інститут», Харків, Україна

АВТОМАТИЗАЦІЯ СТВОРЕННЯ КЛАСИФІКАТОРІВ: АЛГОРИТМИ І ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ФОРМУВАННЯ МНОЖИНИ СТРАТЕГІЙ РОЗГОРТАННЯ ТА ЗАБЕЗПЕЧЕННЯ НАДІЙНОСТІ РОЮ БПЛА

Анотація. Актуальність. Забезпечення надійності та автономності безпілотних літальних апаратів (БПЛА), а також їх роїв (РБА) потребує ефективних методів розроблення і вибору стратегій їх розгортання, а також забезпечення функціональної надійності в складних умовах застосування. Кількість таких стратегій зростає, що вимагає відповідного інструментарію їх розроблення, класифікації та вибору. **Об'єкт дослідження:** процеси класифікації та автоматизації побудови класифікаторів стратегій розгортання і забезпечення надійності (СРЗН) мобільних систем, зокрема РБА. **Мета статті:** розроблення методу автоматизації процесу створення та модифікації (розширення і зміни структури) класифікаторів СРЗН мобільних систем у різних предметних областях, демонстрація його застосування, на прикладі РБА, задля підвищення ефективності прийняття рішень за наявності великої кількості чинників предметної області, які треба брати до уваги. **Результати дослідження.** Проведено аналіз існуючих методів, запропоновано концепцію та алгоритми автоматизації процесів класифікації та моделювання СРЗН, розроблено і валідовано програмне забезпечення, надано приклади використання, які демонструють його здатність підтримувати рішення щодо формування множини і видалення суперечливих або практично неприйнятних стратегій. **Висновки.** Основним науковим внеском даного дослідження є запропонований і реалізований метод автоматизації, що базується на фасетно-ієрархічному описі ознак і декартових моделях для формування можливих конфігурацій стратегій, що може підвищити надійність та ефективність застосування РБА у складних та мінливих умовах завдяки формуванню повних і внутрішньо несуперечливих множин для вибору СРЗН.

Ключові слова: класифікатор стратегій, розгортання рою БПЛА, забезпечення надійності, фасетна класифікація, автоматизація прийняття рішень.

Вступ

Автономні мобільні системи, зокрема, рої безпілотних літальних апаратів (РБА), стрімко набувають поширення в таких сферах, як пошуково-рятувальні операції, моніторинг критичної інфраструктури, військові застосування, комунікація у складних умовах під час техногенних катастроф. Здатність швидко адаптуватися до динамічних умов, зокрема, до впливу фізичних та кіберзагроз, робить РБА важливим інструментом вирішення різноманітних задач у надзвичайних ситуаціях.

Забезпечення надійності та автономності РБА потребує ефективних алгоритмів і методів розроблення і вибору стратегій їх розгортання, а також забезпечення функціональної надійності при використанні [1]. Кількість і різноманітність таких стратегій зростає для задач моніторингу, подолання, забезпечення комунікацій та зменшення рівня впливу кіберфізичних впливів у небезпечних просторах (великі індустриальні споруди з руйнуваннями та підвищеним рівнем шкідливих речовин, території суходолу та акваторії, забруднені вибухонебезпечними предметами (ВНП), зони лісових пожеж та ін. [2, 3]). Вибір стратегій має передувати обстеженню предметної області та формуванню їх множини. Множина стратегій розгортання і забезпечення надійності (СРЗН) має бути повною, несуперечливою, детальною та систематизованою. Ця задача є важливою для забезпечення ефективності та надійності, важко формалізованою і потребує відповідних процедур та засобів програмної підтримки.

Мета статті – розроблення концепції, алгоритмів і програмного забезпечення для автоматизації процесів створення та модифікації класифікаторів

СРЗН для різних систем (на прикладі РБА), використовуючи фасетно-ієрархічний підхід для структурування ознак і врахування обмежень.

Сучасний стан досліджень. Проаналізуємо різні методи побудови класифікаторів стратегій.

Ієрархічна класифікація: базується на традиційному підході, за яким множина об'єктів послідовно декомпонується на підпорядковані підмножини на основі певної ознаки, утворюючи деревоподібну структуру. Перевагами цього підходу є відносна простота побудови. Однак недоліком є слабка гнучкість структури, обумовлена фіксованою основою поділу, що не допускає включення нових об'єктів без переробки всієї класифікаційної схеми [4].

Фасетна класифікація: дозволяє організувати складні предметні області за незалежними атрибутами (фасетами). Цей метод забезпечує гнучкість та масштабованість, особливо у випадках, коли додавання нових атрибутів не порушує загальної структури. У дослідженні [5] запропоновано фасетну класифікацію особливих випадків під час польоту БПЛА або РБА. Автори виділяють такі категорії, як потрапляння повітряного судна у небезпечні метеорологічні умови, втрата орієнтування, втрата просторового орієнтування та незаконне втручання. Цей підхід дозволяє систематизувати та ефективно аналізувати різноманітні ситуації, що можуть виникнути під час експлуатації БПЛА, забезпечуючи більш структурований підхід до управління ризиками та підвищення безпеки польотів.

Декартові моделі: використовуються для генерації множини варіантів шляхом комбінаторного аналізу. З математичного погляду, декартовий добуток дозволяє отримати повну множину варіантів рішень, у кожному з яких кожна ознака приймає

одне зі своїх допустимих значень. Водночас декартові моделі потребують обов'язкової фільтрації неефективних чи конфліктних комбінацій, оскільки не всі теоретично згенеровані варіанти виявляться допустимими або корисними.

Кластерне групування: використовується для групування об'єктів у кластери на основі схожості їхніх характеристик. Зокрема, алгоритм k-means є популярним підходом, який дозволяє ефективно класифікувати дані. У дослідженні [6] запропоновано вдосконалення цього алгоритму шляхом визначення початкових центроїдів та автоматичного розрахунку кількості кластерів, що підвищує точність класифікації. k-means передбачає, що кластери мають сферичну форму та однаковий розмір. Це обмеження може бути неприйнятним для даних зі складною структурою кластерів. Також алгоритм чутливий до різних масштабів ознак, тому необхідно проводити попереднє нормалізування даних, щоб уникнути домінування ознак з більшими значеннями.

Онтологічний підхід: забезпечує семантичну основу для організації знань, дозволяє моделювати взаємозв'язки між об'єктами, атрибутами та їх залежностями, що виходить за рамки класичної класифікації. Використання мови OWL (Web Ontology Language) дозволяє автоматично формувати нові знання.

Зазначимо, що вибір типу класифікатора обумовлюється, в першу чергу, особливостями предметної області (типом систем, умовами застосування, задачами використання тощо) і тому, зазвичай, можуть реалізовуватися комбіновані схеми. Зважаючи на необхідність гнучкої та масштабованої системи класифікації, яка дозволяє враховувати різноманітні комбінації атрибутів, а також з огляду на особливості РБА у цьому дослідженні обрано поєднання фасетної класифікації та декартових моделей. Такий підхід забезпечує можливість ефективної систематизації та аналізу стратегій завдяки врахуванню всіх можливих комбінацій незалежних атрибутів без порушення загальної структури класифікації.

Постановка проблеми. При роботі систем з поетапним виконанням завдань, місія є успішною за умов успішності на будь-якому з етапів. Отже, важливо забезпечити надійність системи на всіх етапах використання РБА [4]. Основна проблема полягає в тому, що планування розгортання та забезпечення надійності систем відбувається за умов невизначеності характеристик середовища, доступних ресурсів та обмежень, які можуть швидко змінюватися. З одного боку, необхідно формалізувати множину різних ознак (наприклад, параметри комунікацій, можливість ремонту тощо). З іншого, – чітко визначити обмеження і залежності, пов'язані з реальною експлуатацією (наприклад, варіанти резервування з дробовою чи цілою кратністю, вимоги з безперервності зв'язку тощо). Тому постає завдання розробити уніфіковану систему для підтримки опису ознак і обмежень, що охоплює різні сценарії застосування.

Автоматизація створення класифікаторів стратегій розгортання і забезпечення надійності є багатоаспектною задачею, що охоплює:

- *Формалізацію вимог* до класифікатора, яка передбачає структурування ознак та визначення їх залежностей та обмежень.

- *Забезпечення гнучкості оновлення* класифікаційних структур без порушення загальної схеми.

- *Розроблення алгоритмів* для автоматизації процесу створення множини стратегій, базуючись на заданому списку ознак та налаштувань залежностей.

- *Генерацію валідних стратегій*, що відповідають заданим обмеженням, на основі структурованих даних про ознаки та їхні обмеження.

Зважаючи на ці вимоги, необхідно розробити автоматизовану систему, яка дозволяє на основі структурованих даних про ознаки та обмеження генерувати множину валідних стратегій, усуваючи неефективні або суперечливі комбінації.

Крім того, перспективними є наступні задачі, які в даному дослідженні тільки окреслені:

- *Інтеграція програмного забезпечення* для моделювання процесу розгортання в реальному часі.

- *Побудова системи критеріїв*, що дозволяють оцінювати, ранжувати та обирати найкращі варіанти стратегій для конкретних умов.

Основні завдання розроблення алгоритмів і програмного забезпечення

Розроблення автоматизованої системи класифікації та генерації конфігурацій передбачає створення уніфікованих структур для опису ознак, обмежень і логічних зв'язків між ними.

Ця система має бути універсальною, придатною для використання в різних предметних областях, включаючи, але не обмежуючись, стратегічним плануванням, управлінням ресурсами та оптимізацією процесів.

Основними завданнями на даному етапі є наступні:

1. *Формалізація структур даних:* використання фасетно-ієрархічного підходу для опису ознак і обмежень включає:

- розробка уніфікованої схеми опису ознак;
- створення уніфікованого формату для обмежень;
- формалізація вихідних результатів.

2. *Автоматизація генерації конфігурацій*, що включає:

- розробка алгоритму створення множини можливих конфігурацій шляхом застосування декартового добутку ознак;
- фільтрація невідповідних (неприйнятних за певних причин) конфігурацій за визначеними обмеженнями, що забезпечує отримання тільки валідних варіантів.

3. *Реалізація програмного забезпечення*, що включає:

- розробка JSON-структури для зберігання ознак та обмежень у стандартизованому форматі;
- імплементація алгоритмів обробки конфігурацій для побудови множини допустимих конфігурацій;
- зберігання результатів.

Методологія побудови класифікаторів і розроблення структури даних

Розробимо методологію побудови класифікаторів і уніфіковану структуру даних для ознак (features) та обмежень (constraints), а також загальні правила формування валідних конфігурацій для подальшої генерації стратегій.

1. Методологія побудови класифікаторів

Процес побудови класифікаторів стратегій у межах цього дослідження спирається на такі положення:

1. Аналіз предметної області. Виділяється й формалізується множина можливих класифікаційних ознак (Set of Classification Features - SCF):

$$SCF = \{CF_i | i = 1, 2, \dots, n\}.$$

2. Встановлення відношень між ознаками. Запроваджуються фасетні, ієрархічні, матричні або змішані відношення, що утворюють множину взаємозв'язків ознак (Set of Feature Relationships - SFR). Отже, визначається як саме ознаки поєднуються або успадковуються.

3. Упорядкування ознак. З урахуванням відношень із SFR визначається кортеж класифікаційних ознак (Classification Features Tuple - CFT) з огляду на їх пріоритетність і логіку слідування.

4. Побудова підмножин сутностей. Для кожної ознаки CF_i формується підмножина сутностей (Subset of the Set of Entities - SSE_i) SSE_i. Сукупність усіх таких підмножин можна розглядати як множину сутностей класифікації (Set of Classification Entities - SCE).

5. Декартовий добуток. Повну множину класифікованих сутностей (Full set of Classified Entities FCE) розглядають як декартовий добуток усіх SSE_i:

$$FCE = SSE_1 \times SSE_2 \times \dots \times SSE_n.$$

6. Задання обмежень. Формально обмеження можна подати як сукупність множини (Conditions - C),

де кожне обмеження $ck \in C$ описує певне правило:

- Domain-обмеження:

$$ck: CF_i \in D_i,$$

де D_i - дозволений піднабір (домен) можливих значень ознаки CF_i.

- Conditional-обмеження (умовний тип):

$$ck: \text{якщо}(CF_i = v) \Rightarrow CF_j \in A_{ij},$$

де v - конкретне значення ознаки CF_i; A_{ij} - дозволений набір для CF_j за умови CF_i= v .

7. Застосування обмежень (фільтрація). Для кожного елемента $x \in FCE$ (тобто для кожної комбінації ознак) перевіряємо виконання всіх правил із C . Якщо x порушує бодай одне з обмежень, такий елемент вилучається. Формально це можна відобразити так:

$$FCE^* = x \in FCE \forall ck \in C: ck(x) = True,$$

тут $ck(x)=True$ означає, що комбінація x не порушує обмеження ck .

8. Фінальне відсіювання. У результаті залишається кінцева множина FCE^* , елементи якої відповідають класифікованим сутностям, що не суперечать жодному з визначених обмежень і мають практичну реалізованість.

У подальших розділах наведені приклади формату опису ознак CF_i та обмежень ck . В якості формату зберігання обрано JSON, оскільки він забезпечує зручність обробки, легкість інтеграції з іншими компонентами та можливість швидкого розширення структури.

2. Уніфікація опису ознак

Для кожної ознаки (feature) було визначено: назву, короткий чи розширений опис, тип даних (enum чи boolean), перелік можливих значень (domain), а також опційно групу. У табл. 1 наведено назву, опис та доступні значення полів сутності ознаки.

Таблиця 1 - Опис полів для features

Назва поля	Особливості	Допустимі значення	Опис
name	Обов'язкове Унікальне	Строка	Унікальний ідентифікатор ознаки
description	Не обов'язкове	Строка	Додатковий опис сутності ознаки (пояснює призначення або особливості цієї ознаки).
type	Обов'язкове	Одне зі значень: • enum; • bool	Вказує, як трактувати можливі значення: enum – перелік можливих опцій (наприклад, "ПН", "ПП"); bool – логічне значення, де допустимими значеннями будуть [true, false].
domain	Обов'язкове	Для типу enum це масив унікальних строкових полів ["Значення1", "Значення2", ...]. Для типу bool - це масив [true, false].	Перелік доступних значень для ознаки.
group	Не обов'язкове	Строка	Дає змогу об'єднувати ознаки за темами (наприклад, "Комунікаційні", "Надійності", "Архітектурні" тощо).

На рис. 1 наведено приклади опису ознак у форматі JSON.

Порядок, у якому зазначено ознаки у JSON (масив "features"), є критично важливим. Ознаки, розташовані вище у списку, вважаються більш важливими або базовими, і саме їх значення використовуються для формування конфігурації наступних за списком ознак. Таким чином, ключові параметри впливають на подальший вибір додаткових ознак.

```
{
  "name": "Можливість переривання зв'язку",
  "description": "Можливість переривання зв'язку",
  "type": "enum",
  "domain": [
    "ПН",
    "ПП"
  ],
  "group": "Комунікаційні"
}
```

Рис. 1. Приклади опису ознак

Розробник повинен уважно обрати порядок розміщення ознак у JSON, оскільки цей порядок буде застосовуватися при обчисленні комбінацій.

Вихідний рядок конфігурації також формується, використовуючи визначений порядок (список назв ознак). Це дає однозначне представлення кожного варіанту.

3. Уніфікація опису обмежень

Було визначено, що обмеження (constraints) може бути двох основних типів:

- *domain*: безпосередньо встановлює звужений набір дозволених значень для певної ознаки;

- *conditional*: описує залежності між декількома ознаками за принципом "якщо – тоді".

У табл. 2 наведено повний список всіх полів з описом та доступними значеннями.

На рис. 2-3 наведено приклади опису domain (для уточнення домену конкретної ознаки) та conditional (за однією або кількома умовами застосовуються певні дії щодо іншої ознаки) обмежень у форматі JSON.

Таблиця 2 - Опис полів для обмежень

Назва поля	Особливості	Допустимі значення	Опис
id	Обов'язкове, унікальне	Строка	Унікальний ідентифікатор обмеження
description	Не обов'язкове	Строка	Додатковий опис сутності обмеження (пояснює логіку застосування)
rule_type	Обов'язкове	Одне зі значень: • conditional; • domain	Визначає, як обробляти обмеження: domain – обмеження на домен однієї ознаки; conditional – умовне обмеження (з умовами і діями).
feature	Обов'язкове для rule_type = domain	Строка	Посилання на назву ознаки, до якої застосовується обмеження.
allowed_values	Обов'язкове для rule_type = domain	Список значень	Перелік дозволених значень для ознаки.
conditions	Обов'язкове для rule_type = conditional	Список об'єктів з наступною структурою: feature : строка (назва ознаки); value : строка (значення, яке активує правило).	Перелік умов, які повинні бути виконані для активації обмеження.
actions	Обов'язкове для rule_type = conditional	Список об'єктів. Кожен об'єкт може містити такі поля: feature : строка (назва ознаки, на яку впливає дія); mode : строка (режим дії, наприклад, "null" або "block"); allowed_values : список значень (використовується для режиму "block").	Дії, що виконуються при виконанні умов. Режим null означає, що значення ознаки має бути скинуто (заборонено); режим block – дозволяються тільки значення, перелічені у allowed_values .

```
{
  "id": "c0",
  "description": "Для заданої задачі не можливі значення, відмінні від ПН для Резервування системи",
  "rule_type": "domain",
  "feature": "Резервування системи",
  "allowed_values": [
    "ПН"
  ]
},
```

Рис. 2. Приклади опису обмеження типу domain

```
{
  "id": "c1",
  "description": "Якщо 'Резервування системи' = 'БР', тоді 'Метод резервування' недоступний але не блокуючий",
  "rule_type": "conditional",
  "conditions": [
    {
      "feature": "Резервування системи",
      "value": "БР"
    }
  ],
  "actions": [
    {
      "feature": "Метод резервування",
      "mode": "null"
    }
  ]
},
```

Рис. 3. Приклади опису обмеження типу conditional

Обмеження визначають логіку відсіювання недопустимих конфігурацій, звужуючи множину комбінацій значень ознак відповідно до бізнес-правил або технічних вимог, і застосовуються до кожної згенерованої комбінації. Алгоритм перевіряє кожну комбінацію послідовно. Порядок перевірки constraints може вплинути на кінцевий результат. Вона відбувається за порядком, в якому обмеження задані у вхідних даних. Розробник має забезпечити, щоб логіка бізнес-правил була узгодженою і послідовністю обробки відповідала пріоритетам завдання. У наведеній структурі використовуються два типи обмежень:

- **Conditional:** Обмеження цього типу описують залежність за принципом "якщо – тоді". Якщо умови (conditions) виконуються, виконуються відповідні дії (actions), наприклад, обмеження конфігурації (mode "block") або скидання значення окремої ознаки (mode "null").

- **Domain:** Цей тип обмежень безпосередньо задає дозволені значення (домен) для конкретної

ознаки. Якщо значення ознаки не входить до дозволеного списку, конфігурація відкидається.

Тестування та результати

Для перевірки дієвості та оцінки ефективності розробленого алгоритму та структур даних був реалізований прототип на мові Python, який генерує множину валідних стратегій розгортання РБА з урахуванням заданих ознак і обмежень. Реалізація алгоритму включає наступні етапи:

1. Завантаження конфігураційних даних із JSON-файлу, де зберігаються опис ознак (features) та обмежень (constraints). Для тестового сценарію було виокремлено 6 ознак та 6 обмежень (рис. 4) на базі класифікації, наданої в [7].

2. Генерація повного набору можливих комбінацій шляхом обчислення декартового добутку всіх можливих значень ознак. Для наведеного прикладу з 6 ознаками (з доменами розміром 2, 3, 2, 2, 3 та 2) було отримано 144 потенційні конфігурації (рис. 5, а).

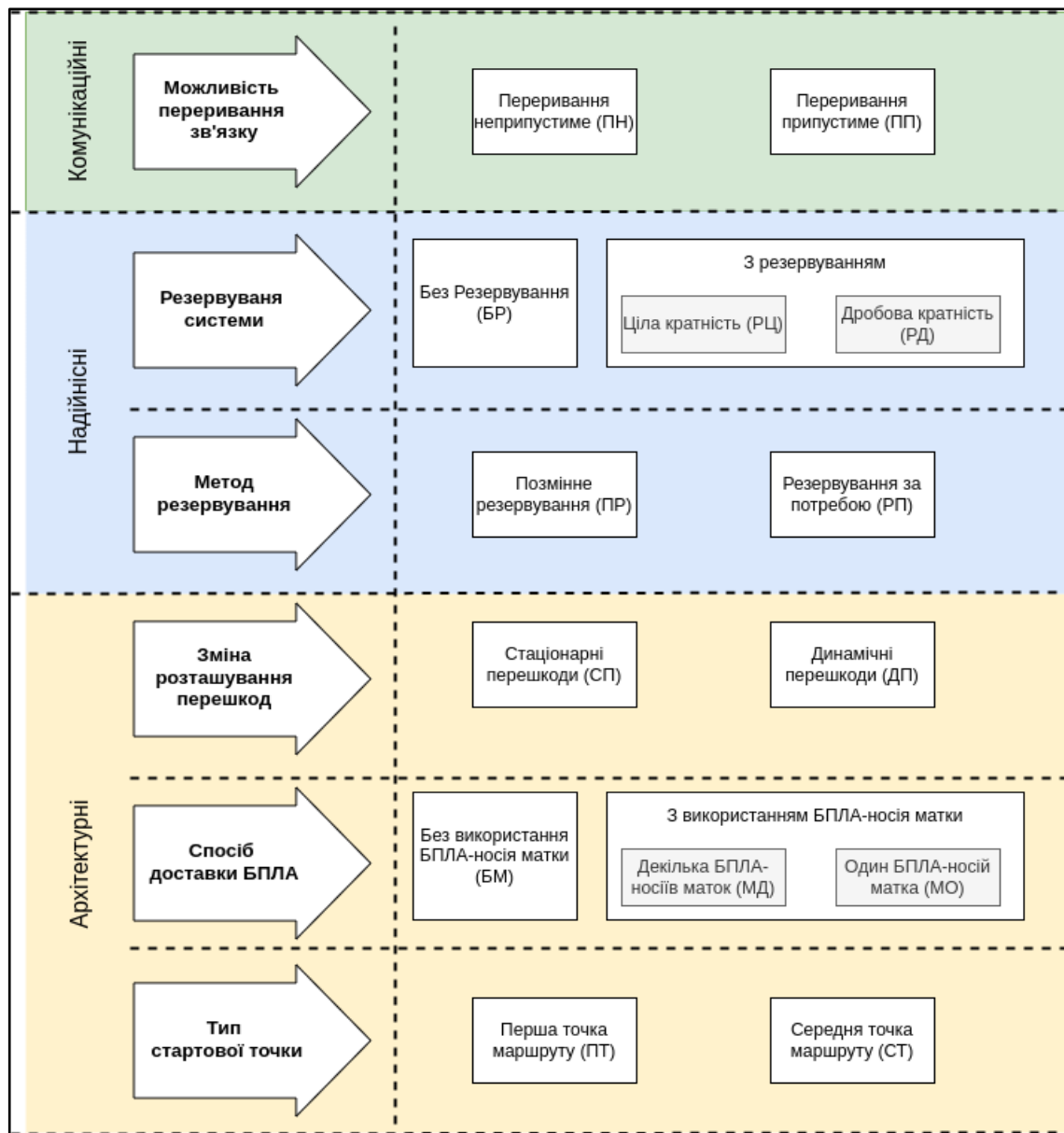


Рис. 4. Класифікатор стратегій розгортання мережі БПЛА

<pre>/home/dimat/Documents/projects/classifier/.venv All possible configurations 144: 1. ПН/БР/ПР/СП/БМ/ПТ 2. ПН/БР/ПР/СП/БМ/СТ 3. ПН/БР/ПР/СП/МО/ПТ 4. ПН/БР/ПР/СП/МО/СТ 5. ПН/БР/ПР/СП/МД/ПТ 6. ПН/БР/ПР/СП/МД/СТ 7. ПН/БР/ПР/ДП/БМ/ПТ 8. ПН/БР/ПР/ДП/БМ/СТ 9. ПН/БР/ПР/ДП/МО/ПТ 10. ПН/БР/ПР/ДП/МО/СТ</pre>	<pre>Valid configurations (48): 1. ПН/РЦ/ПР/СП/БМ/ПТ 2. ПН/РЦ/ПР/СП/БМ/СТ 3. ПН/РЦ/ПР/СП/МО/ПТ 4. ПН/РЦ/ПР/СП/МО/СТ 5. ПН/РЦ/ПР/ДП/МО/ПТ 6. ПН/РЦ/ПР/ДП/МО/СТ 7. ПН/РЦ/РП/СП/БМ/ПТ 8. ПН/РЦ/РП/СП/БМ/СТ 9. ПН/РЦ/РП/СП/МО/ПТ 10. ПН/РЦ/РП/СП/МО/СТ</pre>	<pre>Blocked configurations (96): 1. Blocked by (c2): ПН/БР/None/СП/БМ/ПТ 2. Blocked by (c2): ПН/БР/None/СП/БМ/СТ 3. Blocked by (c2): ПН/БР/None/СП/МО/ПТ 4. Blocked by (c2): ПН/БР/None/СП/МО/СТ 5. Blocked by (c2): ПН/БР/None/СП/МД/ПТ 6. Blocked by (c2): ПН/БР/None/СП/МД/СТ 7. Blocked by (c2): ПН/БР/None/ДП/БМ/ПТ 8. Blocked by (c2): ПН/БР/None/ДП/БМ/СТ 9. Blocked by (c2): ПН/БР/None/ДП/МО/ПТ 10. Blocked by (c2): ПН/БР/None/ДП/МО/СТ</pre>
--	--	--

Рис. 5. Список: а – всіх згенерованих конфігурацій; б – валідних конфігурацій; в – заблокованих конфігурацій з наведенням ідентифікаторів обмежень

У наведеному прикладі стратегії кодуються за такою схемою:

Можливість переривання зв'язку
/ {Резервування} / {Метод резервування}
/ {Зміна розташування перешкод}
/ {Спосіб доставки БПЛА} / {Тип стартової точки}

Код конфігурації «ПП/РД/РП/СП/БМ/СТ»: переривання зв'язку припустимі (ПП), дробова кратність резервування (РД), резервування за потреби (РП), стаціонарні перешкоди (СП), без використання БПЛА-носія матки (БМ), середня точка старту маршруту (СТ).

Код конфігурації «ПН/РЦ/РП/ДП/МД/СТ»: переривання зв'язку неприпустимі (ПН), резервування цілої кратності (РЦ), резервування за потреби (РП), динамічні перешкоди (ДП), використання декількох БПЛА-носіїв матки (МД), середня точка старту маршруту (СТ).

3. Послідовне застосування алгоритму фільтрації конфігурацій. Для наведеного прикладу було виокремлено 48 валідних (рис. 5, б) та 96 заблокованих (рис. 5, в) конфігурацій, причому для заблокованих фіксується ідентифікатор обмеження, що спричинило відхилення.

Під час аналізу отриманих результатів було виявлено, що розроблене програмне забезпечення продемонструвало кращу ефективність порівняно з ручним перебором, результати якого надано у статті [7]. Зокрема, у зазначеній статті стратегія «ПП/РД/РП/ДП/БМ/ПТ» була помилково визначена як валідна для запропонованого класифікатора стратегій розгортання мережі БПЛА (рис. 6). Водночас, розроблені алгоритми і ПЗ правильно визначили стратегію «ПП/РД/РП/ДП/БМ/ПТ» як недопустиму (рис. 7), оскільки ця стратегія не задовольняє обмеження «с5».

Множини стратегій розгортання комунікаційної мережі БПЛА			
Множина стратегій розгортання по ознакам надійності	Множина стратегій розгортання по архітектурним ознакам	Остаточна множина стратегій розгортання	
		Переривання неприпустиме (ПН)	Переривання припустиме (ПП)
1. БР 2. РЦ-ПР 3. РЦ-РП 4. РД-РП	1. СП-БМ-ПТ 2. СП-БМ-СТ 3. СП-МО-ПТ 4. СП-МД-СТ 5. ДП-БМ-ПТ 6. ДП-БМ-СТ 7. ДП-МО-ПТ	1. ПН/РЦ-ПР/СП-БМ-ПТ 2. ПН/РЦ-ПР/СП-БМ-СТ 3. ПН/РЦ-ПР/СП-МО-ПТ 4. ПН/РЦ-ПР/СП-МД-СТ 5. ПН/РЦ-ПР/ДП-БМ-ПТ 6. ПН/РЦ-ПР/ДП-БМ-СТ 7. ПН/РЦ-ПР/ДП-МО-ПТ 8. ПН/РЦ-ПР/ДП-МД-СТ 9. ПН/РЦ-РП/СП-БМ-ПТ 10. ПН/РЦ-РП/СП-БМ-СТ 11. ПН/РЦ-РП/СП-МО-ПТ 12. ПН/РЦ-РП/СП-МД-СТ 13. ПН/РЦ-РП/ДП-БМ-ПТ 14. ПН/РЦ-РП/ДП-БМ-СТ 15. ПН/РЦ-РП/ДП-МО-ПТ 16. ПН/РЦ-РП/ДП-МД-СТ 17. ПН/РД-РП/СП-БМ-ПТ 18. ПН/РД-РП/СП-БМ-СТ 19. ПН/РД-РП/СП-МО-ПТ 20. ПН/РД-РП/СП-МД-СТ 21. ПН/РД-РП/ДП-БМ-ПТ 22. ПН/РД-РП/ДП-БМ-СТ 23. ПН/РД-РП/ДП-МО-ПТ 24. ПН/РД-РП/ДП-МД-СТ	1. ПП/БР/СП-БМ-ПТ 2. ПП/БР/СП-БМ-СТ 3. ПП/БР/СП-МО-ПТ 4. ПП/БР/СП-МД-СТ 5. ПП/БР/ДП-БМ-ПТ 6. ПП/БР/ДП-БМ-СТ 7. ПП/БР/ДП-МО-ПТ 8. ПП/БР/ДП-МД-СТ 9. ПП/РЦ-ПР/СП-БМ-ПТ 10. ПП/РЦ-ПР/СП-БМ-СТ 11. ПП/РЦ-ПР/СП-МО-ПТ 12. ПП/РЦ-ПР/СП-МД-СТ 13. ПП/РЦ-ПР/ДП-БМ-ПТ 14. ПП/РЦ-ПР/ДП-БМ-СТ 15. ПП/РЦ-ПР/ДП-МО-ПТ 16. ПП/РЦ-ПР/ДП-МД-СТ 17. ПП/РЦ-РП/СП-БМ-ПТ 18. ПП/РЦ-РП/СП-БМ-СТ 19. ПП/РЦ-РП/СП-МО-ПТ 20. ПП/РЦ-РП/СП-МД-СТ 21. ПП/РЦ-РП/ДП-БМ-ПТ 22. ПП/РЦ-РП/ДП-БМ-СТ 23. ПП/РЦ-РП/ДП-МО-ПТ 24. ПП/РЦ-РП/ДП-МД-СТ 25. ПП/РД-РП/СП-БМ-ПТ 26. ПП/РД-РП/СП-БМ-СТ 27. ПП/РД-РП/СП-МО-ПТ 28. ПП/РД-РП/СП-МД-СТ 29. ПП/РД-РП/ДП-БМ-ПТ 30. ПП/РД-РП/ДП-БМ-СТ 31. ПП/РД-РП/ДП-МО-ПТ 32. ПП/РД-РП/ДП-МД-СТ

Рис. 6. Приклад ручного перебору стратегій з роботи [7], що містить помилково визначену як валідну стратегію «ПП/РД/РП/ДП/БМ/ПТ»

```

87. Blocked by (c5): ПП/РД/ПР/ДП/БМ/ПТ
88. Blocked by (c5): ПП/РД/ПР/ДП/БМ/СТ
89. Blocked by (c3): ПП/РД/ПР/ДП/МО/СТ
90. Blocked by (c4): ПП/РД/ПР/ДП/МД/ПТ
91. Blocked by (c3): ПП/РД/РП/СП/МО/СТ
92. Blocked by (c4): ПП/РД/РП/СП/МД/ПТ
93. Blocked by (c5): ПП/РД/РП/ДП/БМ/ПТ
94. Blocked by (c5): ПП/РД/РП/ДП/БМ/СТ
95. Blocked by (c3): ПП/РД/РП/ДП/МО/СТ
96. Blocked by (c4): ПП/РД/РП/ДП/МД/ПТ

Constraints description:
ID: c1. Description: Якщо 'Резервування системи' = '(БР) Без Резервування', тоді 'Метод резервування' недоступний а
ID: c2. Description: Якщо 'Резервування системи' = '(РД) Дробова кратність', тоді 'Метод резервування' = '(ПР) Позм
ID: c3. Description: Якщо 'Спосіб доставки БПЛА' = '(МО) Один БПЛА-носії матка', тоді 'Тип стартової точки' = '(СТ)
ID: c4. Description: Якщо 'Спосіб доставки БПЛА' = '(МД) екілька БПЛА-носіїв маток', тоді 'Тип стартової точки' = '
ID: c5. Description: Якщо 'Зміна розташування перешкод' = '(ДП) Динамічні перешкоди', тоді 'Спосіб доставки БПЛА' =
ID: c6. Description: При '(ПН) Неприпустимому перериванню' Системи '(БР) без резервування' неефективні, оскільки та

```

Рис. 7. Результати роботи програми: стратегія «ПП/РД/РП/ДП/БМ/ПТ»
коректно заблокована відповідно до обмеження «с5»

А саме: якщо параметр «Зміна розташування перешкод» дорівнює «(ДП) Динамічні перешкоди», то «Спосіб доставки БПЛА», визначений як «Без використання БПЛА-носія матки (БМ)», стає неефективним.

Причиною цього є значне зростання обсягу додаткових розрахунків, які має виконувати кожен окремий БПЛА за таких умов.

За результатами тестування було виявлено, що:

- генерація 144 можливих стратегій дозволяє охопити весь простір варіантів розгортання, що забезпечує повноту аналізу;

- автоматична перевірка з використанням заданих обмежень значно зменшує кількість помилок у виборі стратегій, оскільки некоректні комбінації відсіюються на етапі формування валідних конфігурацій;

- детальна діагностика через фіксацію заблокованих конфігурацій із зазначенням відповідних ID обмежень допомагає швидко виявляти і коригувати потенційні неточності у визначенні правил.

Нижче наведено ключові можливості реалізованого прототипу:

- вивід усіх згенерованих комбінацій для подальшого аналізу;
- перевірка кожної конфігурації за заданими умовами (conditional) та безпосередньо за доменами (domain);
- підрахунок валідних конфігурацій, що сприяє оцінці ефективності застосування алгоритму.

Висновки та перспективи

Основним науковим внеском даного дослідження є запропонований і реалізований метод автоматизації генерації СРЗН, що базується на фасетно-ієрархічному описі ознак і декартових моделях

для формування можливих конфігурацій. Ключовими результатами роботи є концептуальна схема і комплекс алгоритмів процесу автоматизації, що складається з етапів:

- формування уніфікованих структур ознак і обмежень у JSON-форматі;
- генерування повного переліку комбінацій;
- послідовної фільтрації із застосуванням domain та conditional обмежень.

Наведений Python-код демонструє практичну реалізацію алгоритму, що включає завантаження даних, генерацію всіх можливих конфігурацій, застосування обмежень і виведення результатів. Тестові результати підтвердили, що такий підхід дає змогу швидко відсікати недійсні або суперечливі комбінації, фіксувати причини блокування (ідентифікатор обмеження) й отримувати кінцеву множину валідних стратегій. У подальших дослідженнях планується:

- інтегрувати розроблений інструментарій із системами динамічного моделювання роботи РБА;
- розробити систему багатокритеріального оцінювання та ранжування згенерованих стратегій;
- деталізувати ознаки небезпечних просторів та врахувати можливості використання гетерогенних систем (комплексів безпілотних літальних, безпекопажних і наземних апаратів-роботів);
- розширити типи представлення обмежень, зокрема, з використанням нечіткої логіки або онтологічного оброблення.

Отже, автоматизований підхід до проектування та аналізу стратегій може підвищити надійність та ефективність застосування РБА у складних та мінливих умовах завдяки формуванню повних і внутрішньо несуперечливих множин для вибору СРЗН з використанням процедур [1].

СПИСОК ЛІТЕРАТУРИ

1. Terenyk D. Choosing strategies for deployment and ensuring the reliability of a UAV swarm to support communications in destruction conditions / Dmytro Terenyk, Vyacheslav Kharchenko // Innovative technologies and scientific solutions for industries. – 2024. – № 3 (29). – С. 91–103. – URL: <https://doi.org/10.30837/2522-9818.2024.3.091>.

2. Robotic-biological systems for detection and identification of explosive ordnance: concept, general structure, and models / Gennadiy Fedorenko [та ін.] // Radioelectronic and Computer Systems. – 2023. – № 2. – С. 143–159. – URL: <https://doi.org/10.32620/reks.2023.2.12>.
3. Methods and software tools for reliable operation of flying LiFi networks in destruction conditions / Herman Fesenko [та ін.] // Sensors. – 2024. – Т. 24, № 17. – С. 5707. – URL: <https://doi.org/10.3390/s24175707>.
4. Balalaieva O. Y. Фасетні класифікації електронних засобів навчального призначення / Olena Yu Balalaieva // Information technologies and learning tools. – 2013. – Т. 38, № 6. – С. 41–52. – URL: <https://doi.org/10.33407/itlt.v38i6.926>.
5. Перегуда О. М. Фасетна класифікація особливих випадків у польоті безпілотної літальної апарату 1 класу / О. М. Перегуда, А. В. Родіонов, А. І. Бобунов // Наука і техніка Повітряних Сил Збройних Сил України. – 2022. – № 1(46). – С. 85–91. – URL: <https://doi.org/10.30748/nitps.2022.46.12>.
6. Жигайло О. М. Кластерний аналіз даних в автоматизованих системах простежуваності / О. М. Жигайло, В. В. Борис // Automation of technological and business processes. – 2018. – Т. 10, № 1. – URL: <https://doi.org/10.15673/atbp.v10i1.879>.
7. Тереник Д. Класифікація стратегій забезпечення надійності літаючої мережі для забезпечення комунікацій в умовах руйнувань / Дмитро Тереник, Вячеслав Харченко // Measuring and computing devices in technological processes. – 2024. – № 2. – С. 357–370. – URL: <https://doi.org/10.31891/2219-9365-2024-78-41>.
8. Phase combination for reliability analysis of dynamic k-out-of-n Phase-AND mission systems / Chaonan Wang [та ін.] // Reliability Engineering & System Safety – 2025. – С. 110817. – URL: <https://doi.org/10.1016/j.res.2025.110817>.
9. Wei Y. Fleet service reliability analysis of self-service systems subject to failure-induced demand switching and a two-dimensional inspection and maintenance policy / Yian Wei, Yao Cheng, Haitao Liao // IEEE Transactions on Automation Science and Engineering. – 2024. – С. 1–16. – URL: <https://doi.org/10.1109/tase.2024.3516049>.
10. Ruiz C. Generalized functional mixed models for accelerated degradation-based reliability analysis / Cesar Ruiz, Haitao Liao, Edward A. Pohl // IEEE Transactions on Reliability. – 2024. – С. 1–12. – URL: <https://doi.org/10.1109/tr.2024.3505077>.
11. Dui H. Attack strategies and reliability analysis of Wireless Mesh Networks considering cascading failures / Hongyan Dui, Jiabao Zhai, Xiuwen Fu // Reliability Engineering & System Safety. – 2025. – С. 110832. – URL: <https://doi.org/10.1016/j.res.2025.110832>.
12. Kucherenko O. I. Classification of methods and algorithms for constructing UAV routes / O. I. Kucherenko, T. A. Vakaliuk // Scientific notes of Taurida national V.I. Vernadsky university. Series: Technical sciences. – 2024. – Т. 1, № 3. – С. 111–116. – URL: <https://doi.org/10.32782/2663-5941/2024.3.1/18>.

Received (Надійшла) 10.04.2025

Accepted for publication (Прийнята до друку) 11.06.2025

ВІДОМОСТІ ПРО АВТОРІВ / ABOUT THE AUTHORS

Тереник Дмитро – аспірант кафедри комп'ютерних систем, мереж і кібербезпеки, Національний аерокосмічний університет "Харківський авіаційний інститут", Харків, Україна;

Terenyk Dmytro – PhD Student at the Department of Computer Systems, Networks, and Cybersecurity, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine;

e-mail: d.terenik@student.csn.khai.edu; ORCID Author ID: <https://orcid.org/0009-0003-4173-8757>;

Scopus Author ID: <https://www.scopus.com/authid/detail.uri?authorId=57222131855>.

Харченко Вячеслав Сергійович – доктор технічних наук, професор, завідувач кафедри комп'ютерних систем, мереж і кібербезпеки, Національний аерокосмічний університет "Харківський авіаційний інститут", Харків, Україна;

Kharchenko Vyacheslav – Doctor of Sciences (Engineering), Professor, Head at the Department of Computer Systems, Networks, and Cybersecurity, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine;

e-mail: v.kharchenko@csn.khai.edu; ORCID ID: <https://orcid.org/0000-0001-5352-077X>;

Scopus Author ID: <https://www.scopus.com/authid/detail.uri?authorId=22034616000>.

Automated development of classifiers: algorithms and software for generating strategies of deployment and reliability assurance for UAV swarm

Dmytro Terenyk, Vyacheslav Kharchenko

Abstract. Motivation. Ensuring reliability and autonomy of unmanned aerial vehicles (UAVs), as well as their swarms (UAV swarms), requires effective methods for developing and selecting deployment strategies, along with ensuring functional reliability under complex operational conditions. As the number of such strategies grows, appropriate tools for their development, classification, and selection become necessary. **Object of research:** processes of classification and automation in constructing classifiers for deployment and reliability assurance strategies (DRAS) for mobile systems, particularly UAV swarms. **Purpose of the article.** Development of a method for automating the process of creating and modifying (expanding and restructuring) classifiers of DRAS for mobile systems in various subject areas, demonstrating its application through the example of UAV swarms to enhance decision-making efficiency in conditions with a large number of domain-specific factors. **Research results.** An analysis of existing methods has been conducted; a concept and algorithms for automating classification and modeling processes of DRAS have been proposed; software has been developed and validated; and practical examples of its application have been presented, demonstrating its capability to support decision-making regarding the formation of sets and elimination of conflicting or practically unacceptable strategies. **Conclusions.** The primary scientific contribution of this study is the proposed and implemented automation method based on facet-hierarchical attribute descriptions and Cartesian models for generating possible strategy configurations. This approach can enhance the reliability and effectiveness of UAV swarm deployment under complex and dynamic conditions by generating complete and internally consistent sets for selecting DRAS.

Keywords: strategy classifier, UAV swarm deployment, reliability assurance, facet classification, decision-making automation.