DOI: 10.18372/2410-7840.20.13070
УДК 004.056.53(045)

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ФОРМУВАННЯ ЕТАЛОНІВ ПАРАМЕТРІВ ДЛЯ СИСТЕМ ВИЯВЛЕННЯ КІБЕРАТАК

Анна Корченко, Олег Заріцький, Тарас Паращук, Володимир Бичков

Переважна більшість систем виявлення вторгнень стають невід'ємною частиною захисту будь якої мережевої безпеки, вони використовуються для моніторингу підозрілої активності в системі та виявлення атакуючих дій неавторизованої сторони. Активізація кібератак ініціює створення спеціальних технічних рішень, здатних залишатись ефективними при появі нових або модифікованих видів кіберзагроз з невстановленими або нечітко визначеними властивостями. Більшість таких систем направлена на виявлення підозрілої активності чи втручання в мережу для прийняття адекватних заходів щодо запобігання кібератакам. Актуальними системами виявлення вторгнень є ті, які орієнтовані на ідентифікацію аномальних станів але вони мають низку недоліків. Більш ефективні в цьому відношенні є експертні підходи, засновані на використанні знань і досвіду фахівців відповідної предметної галузі. Побудова технічних рішень і створення спеціальних засобів (наприклад, програмного забезпечення для систем виявлення атак, що дозволяють детектувати раніше невідомі кібератаки шляхом контролю поточного стану нечітко визначених параметрів в слабоформалізованому середовищі оточення), заснованих на експертних підходах, є перспективним напрямком досліджень. На основі відомої системи виявлення кібератак, яка базується на методології виявлення аномалій (породжених кібератаками) та множини відповідних методів і моделей запропоноване програмне забезпечення, яке, за рахунок базового алгоритму та низки розроблених процедур (конструювання координатної сітки; ініціалізації величин на основі набору баз даних та модулів; графічного формування параметрів; пошуку спільних точок відповідно базових правил та графічної інтерпретації результату) дозволяє автоматизувати процес формування еталонів параметрів для сучасних систем виявлення атак та відображати результати детектування аномального стану у заданий проміжок часу.

Ключові слова: *атаки, кібератаки, аномалії, системи виявлення вторгнень, системи виявлення атак, системи виявлення кібератак, виявлення аномалій в інформаційних системах.*

На сьогодні більшість систем виявлення вторгнень (СВВ) стають невід'ємною частиною захисту будь-якої мережевої безпеки, вони використовуються для моніторингу підозрілої активності в системі та виявлення атакуючих дій неавторизованої сторони (НАС).

Останні дослідження, які проведені фахівцями в відповідних галузях за 2018 рік показали, що корпоративні мережі переважної більшості компаній не здатні забезпечити належний захист від існуючих кіберзагроз. Відзначені дві масштабні кібератаки, які потрясли світову спільноту – віруси WannaCry та NotPetya інфікували сотні тисяч комп'ютерів в різних країнах. Також, користувачі зіткнулися і з низкою інших, менш значних атак, що пов'язані з вірусами-вимагачами, DDoS-атаками, викраданням персональних даних [1].

Активізація таких кібератак ініціює створення спеціальних технічних рішень, засобів та систем протидії, здатних залишатись ефективними при появі нових або модифікованих видів кіберзагроз з невстановленими або нечітко визначеними властивостями. Загалом такі СВВ направлені на виявлення підозрілої активності (наприклад, шляхом формування відповідного звіту для адміністратора інформаційної системи) чи втручання в мережу для прийняття адекватних заходів щодо запобігання кібератакам. Ці системи, як правило, достатньо дорогі, мають закритий код та вимагають періодичної підтримки висококваліфікованих фахівців для налаштування до умов конкретних підприємств.

Достатньо актуальними і необхідними СВВ є ті, які орієнтовані на виявлення аномальних станів. Основними їх недоліками є, наприклад, надлишок помилкових спрацювань, складність процесу налаштування, тривалий процес навчання та створення відповідного профілю нормального стану системи. Більш ефективними в цьому є експертні підходи, засновані на використанні знань і досвіду фахівців відповідної предметної галузі [2].

Виходячи з цього, побудова технічних рішень і створення спеціальних засобів (наприклад, програмного забезпечення для систем виявлення атак (СВА), що дозволяють детектувати раніше невідомі кібератаки шляхом контролю поточного стану нечітко визначених параметрів в слабоформалізованому середовищі оточення), заснованих на експертних підходах, є актуальною науковою задачею.

На основі системи [3], яка базується на методології виявлення аномалій, породжених кібератa-

ками [4] (в основу якої закладено логіко-лінгвістичний підхід [1] і кортежна модель [5-6]) та розроблених методах (формування лінгвістичних еталонів [7-10]; фазифікації параметрів на лінгвістичних еталонах [11]; α-рівневої номіналізації нечітких чисел [12]; визначення ідентифікуючих термів [13]; формування базових детекційних правил [14]), побудуємо і проведемо експериментальне дослідження алгоритмічного та програмного забезпечення формування еталонів параметрів для систем виявлення аномалій. Такий засіб дозволить автоматизувати процес детектування в слабоформалізованому нечітко визначеному середовищі аномальний стан у заданий проміжок часу.

Відповідно до запропонованої методології [4] програмний засіб формування еталонів параметрів для систем виявлення аномалій функціонує на основі базового алгоритму System_level_Click (рис. 1), що поєднує низку наступних зумовлених процесів (процедур):

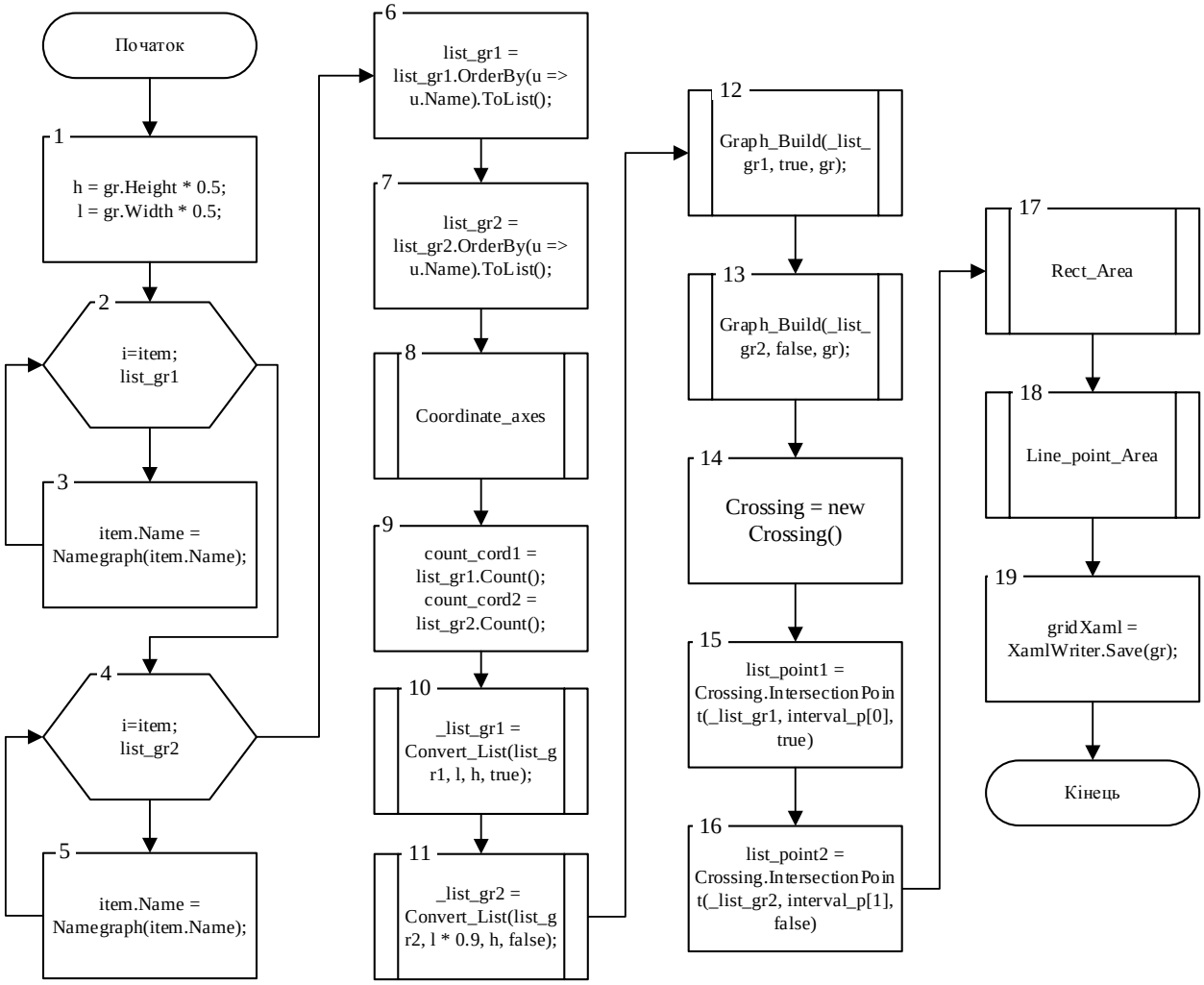

Рис. 1. Базовий алгоритм System_level_Click

– Coordinate_axes (конструювання координатної сітки для μ і x);

– Convert_List (ініціалізація величин на основі баз даних кібератак (БДК) і еталонів (БДЕ) [3] та формування поточних значень (МФПЗ). Відповідно до структури системи виявлення кібератак (СВК) в [3] (рис. 1) визначаються координати еталонних $\underset{\sim}{T}^{e}_{ijs}$ та поточних $\underset{\sim}{P}^{\tau_f}_{ij}$ НЧ в m_i-мірному параметричному підсередовищі);

– Graph_Build (графічне формування параметрів, наприклад, $P_{31} = P_{SPKOП} = КОП$ і $P_{32} = P_{SPKПОА} = КПОА$ та їх відображення на об'єкті Canvas відповідно до етапу 3 [4];

– Crossing (реалізуються процедури IntersectionPoint і GetPoint та здійснюється відображення поточного стану системи відповідно до базових правил $\mathbf{DR_i}$ в детекційному середовищі (див. етап 5-7 в [4]));

– Rect_Area (будуються двомірні опорні області (дивись етап 5 в [4]) відповідно до заданих правил, які надходять з баз даних правил (БДП) [3], що формуються на основі $\mathbf{DR_i}$ $(i=\overline{1,n})$ та використовуються для виявлення i-ї кібератаки на основі параметричних підсередовищ різної розмірності;

– Line_point_Area (відповідно до етапу 5 в [4] будуються і відображаються спільні точки ліній проектування еталонних $\underset{\sim}{T}^{e}_{ijs}$ і поточних $\underset{\sim}{P}^{\tau_f}_{ij}$ НЧ, наприклад, для параметрів $P_{31}=P_{SPKOП}=КОП$ і $P_{32}=P_{SPKПОА}=КПОА$ в 2-мірному параметричному підсередовищі).

Розглянемо принцип роботи головного алгоритму System_level_Click (рис. 1), який інтегрує зазначені процедури для побудови повного переліку графічних компонентів необхідних для ефективного виявлення аномального стану в інформаційних системах.

На початку обчислювального процесу (рис. 1, вершина 1) відбувається ініціалізація необхідних характеристик екрану. Далі (рис. 1, вершина 2-3 та 4-5) відповідно отримуємо в циклах початкові дані з БДЕ [3], наприклад, для параметрів КОП та КПОА. Наступним (рис. 1, вершина 6-7) формуються множини порядку параметрів відповідно до (14) та (16) в [5].

Далі (рис. 1, вершина 8), виконується зумовлений процес (клас **Coordinate_axes**), відповідно до якого реалізується процедура Main_coordinate_axes (рис. 2), що здійснює послідовну обробку трьох підпрограм градуювання координатних осей та сітки:

– Grid_coordinates (відповідає за побудову координатної сітки з використанням отриманих меж області вертикальних та горизонтальних ліній);

– Graduation_axes (відповідає за маркування осей μ і x, інтервали градуювання в обмеженій області);

– Drawing_axes (відповідає за побудову координатних осей для відображення параметрів, наприклад, $P_{31}=P_{SPKOП}=КОП$ і $P_{32}=P_{SPKПОА}=КПОА$ в 2-мірному параметричному підсередовищі [7]).

Далі (рис. 1, вершина 9) відповідно до структури СВК в [3] (рис. 1) визначається кількість даних (count_cord1 і count_cord2) у БДЕ для кожного з параметрів, наприклад, $P_{31}=P_{SPKOП}=КОП$ і $P_{32}=P_{SPKПОА}=КПОА$.

На наступному етапі (рис. 1, вершини 10 та 11) викликається процедура **Convert_List** (рис. 3), яка дозволяє отримувати дані з таблиць за визначеними параметрами (відповідно до етапу 3 в [4]), наприклад, КОП та КПОА, які конвертуються в форму необхідну для побудови графічних зображень заданих параметрів, наприклад, $P_{31}=P_{SPKOП}=КОП$ і $P_{32}=P_{SPKПОА}=КПОА$ в 2-мірному параметричному підсередовищі.

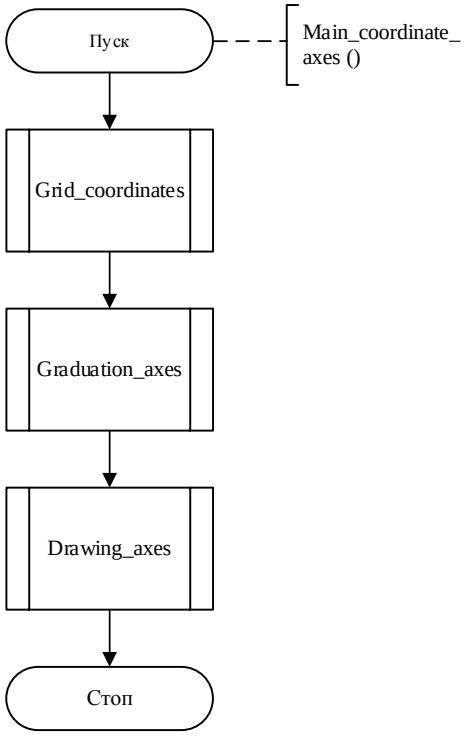

Рис. 2. Алгоритм Main_coordinate_axes

Процес конвертації даних складається з двох етапів. Перший, відповідно до етапу 4, в [4] визначає інтервал, якому належить вершина графічного зображення поточного НЧ $\underset{\sim}{P}^{\tau_f}_{ij}$ в m_i-мірному поточному підсередовищі. Це необхідно для пошуку спільних точок графічних зображень еталонних $\underset{\sim}{T}^{e}_{ijs}$ та поточного $\underset{\sim}{P}^{\tau_f}_{ij}$ НЧ, оскільки зазначені точки повинні лежати в одних межах з вершиною $\underset{\sim}{P}^{\tau_f}_{ij}$ будь-якого з параметрів. Другий – конвертує еталонні $\underset{\sim}{T}^{e}_{ijs}$ та поточні $\underset{\sim}{P}^{\tau_f}_{ij}$ НЧ (які знаходяться в БДЕ та МФПЗ [3]) у значення, що відповідають системі координат Canvas. Зазначена процедура повертає список конвертованих значень НЧ та меж, де можуть знаходитись спільні точки графічного зображення поточного стану (див. етап 4 в [4]), наприклад, для параметрів $P_{31}=P_{SPKOП}=КОП$ і $P_{32}=P_{SPKПОА}=КПОА$. Отриманні дані необхідні для побудови ліній проектування та спільних точок графічних зображень по заданим параметрам КПОА і КОП.

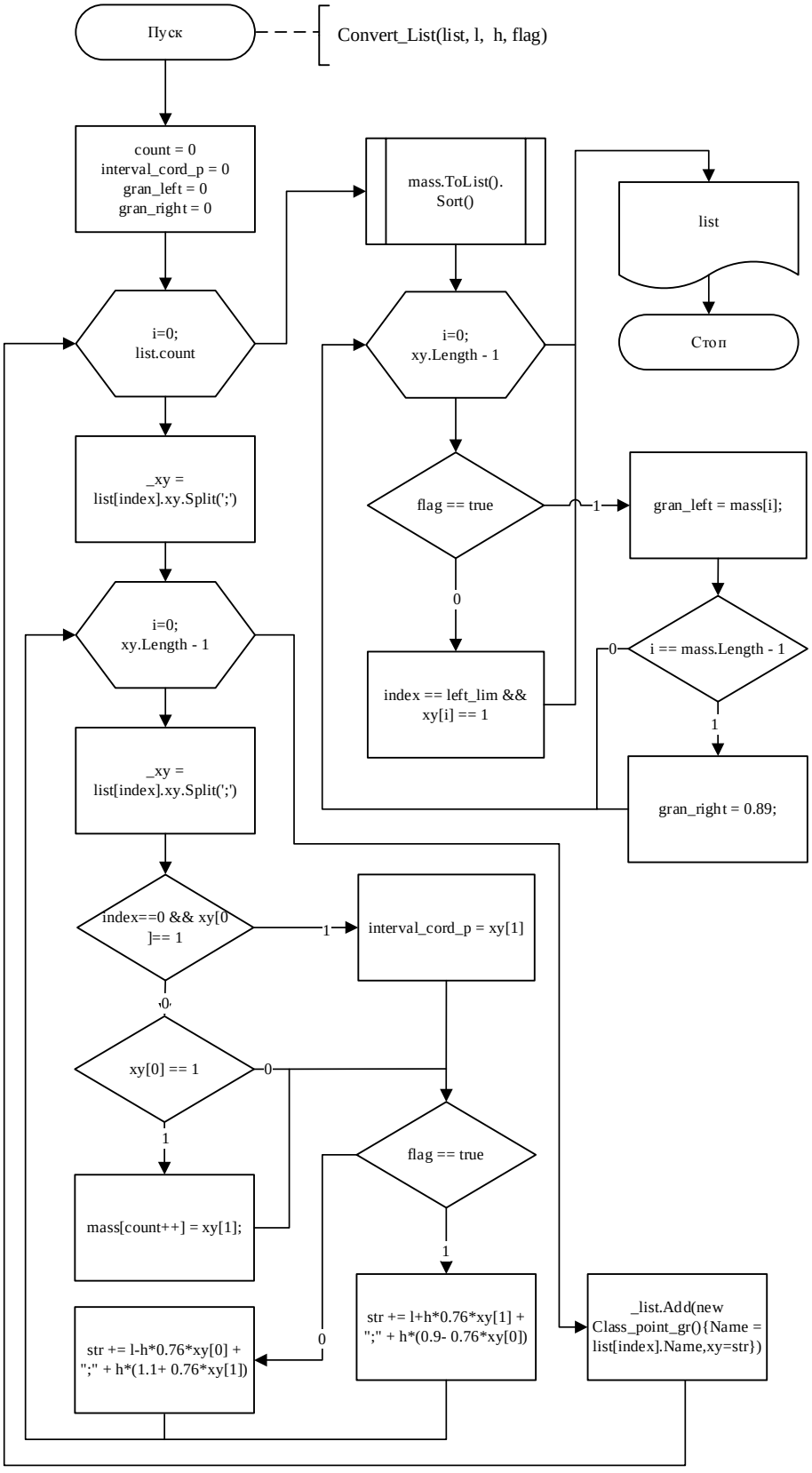

Рис. 3. Алгоритм реалізації Convert_List

Далі (рис. 1, вершини 12-13) викликається процедура **Graph_Build** (рис. 4), що на підставі БДЕ та МФПЗ в [3] дозволяє будувати графічні зображення еталонних $\underset{\sim}{T}^{e}_{ijs}$ та поточних $\underset{\sim}{P}^{\tau_f}_{ij}$ НЧ. Після проведення процесу конвертації даних з БДЕ вони передаються в підпрограму Graph_Build, яка при виклику отримує список конвертованих даних у вигляді індексу для отримання кольору та зміну графічного об'єкта Canvas для побудови базових графічних значень.

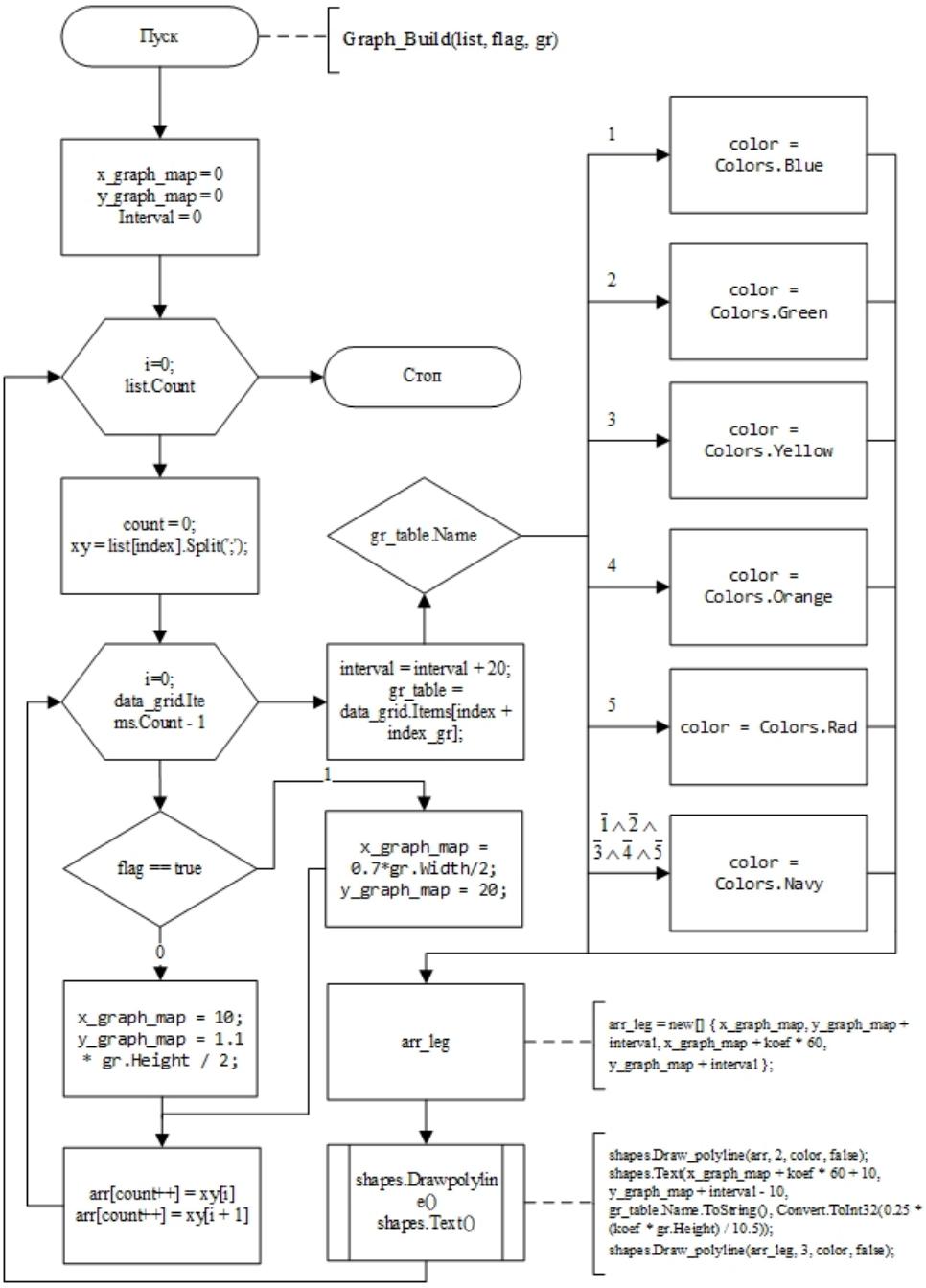

Рис. 4. Алгоритм реалізації Graph_Build

Використовуючи Main_figures створюється об'єкт класу shapes в тілі Graph_Build і далі за допомогою об'єкта shapes викликається Draw_polyline та дані зі списку записуються в масив. Також визначаються варіації кольорів та типів ліній.

Виконання в циклі зазначеної послідовності дій пов'язане з побудовою графічних зображень та їх легенд (наприклад, Р, ОМ, М, С, Б та ОБ для параметрів $P_{31} = P_{SPKOP} = KOP$ і $P_{32} = P_{SPKPOA} = KPOA$ (рис. 5-6) в 2-мірному параметричному підсередовищі [7]).

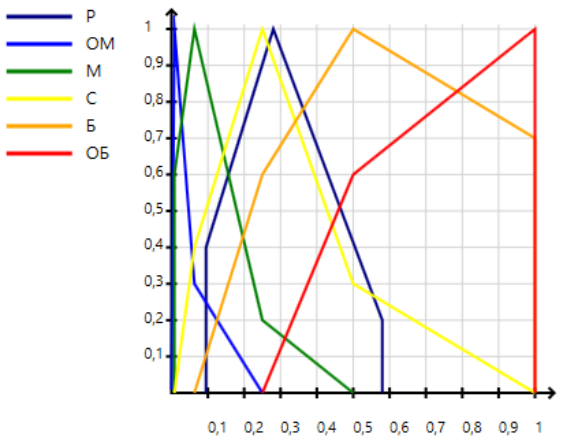

Рис. 5. Результат роботи процедури Graph_Build для параметра КОП

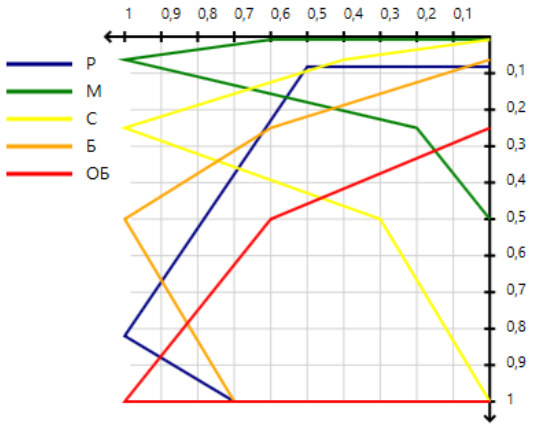

Рис. 6. Результат роботи процедури Graph_Build для параметра КПОА

На наступному етапі (рис. 1, вершина 14) створюється об'єкт класу **Crossing**, і викликається процедура IntersectionPoint та формується список координат спільних точок, необхідних для відображення поточного стану системи. Отримавши список та ідентифікатори еталонів НЧ за допомогою Convert_List (рис. 1, вершини 10-11), визначаються параметри ідентифікації опорних областей за допомогою Draw_main_rect. Слід зазначити, що клас Crossing складається з двох процедур: IntersectionPoint (рис. 7) та GetPoint (рис. 8).

Перша процедура IntersectionPoint дозволяє отримати спільні точки графічних зображень $\underset{\sim}{T}^{e}_{ijs}$ та $\underset{\sim}{P}^{\tau_f}_{ij}$ (окремо для кожного з параметрів), а також для $\underset{\sim}{T}^{e}_{ijs}$ з суміжними еталонними НЧ (див. етап 5-6 в [4]).

Друга процедура GetPoint (див. етап 5 та 6 в [4]) відповідає за отримання координат вищезазначених точок, тобто, наприклад, пара значень $(\mu_1; x_1)$ та $(\mu_2; x_2)$, які характеризують складову першого графічного зображення і пара значень $(\mu_3; x_3)$ та $(\mu_4; x_4)$ – другого. Далі обчислюються всі можливі значення для обраної складової першого графічного зображення відносно всіх можливих складових другого. Обчислення проводиться за допомогою Intersection_point, яка визначає спільні точки складових для заданих координат і повертає до GetPoint.

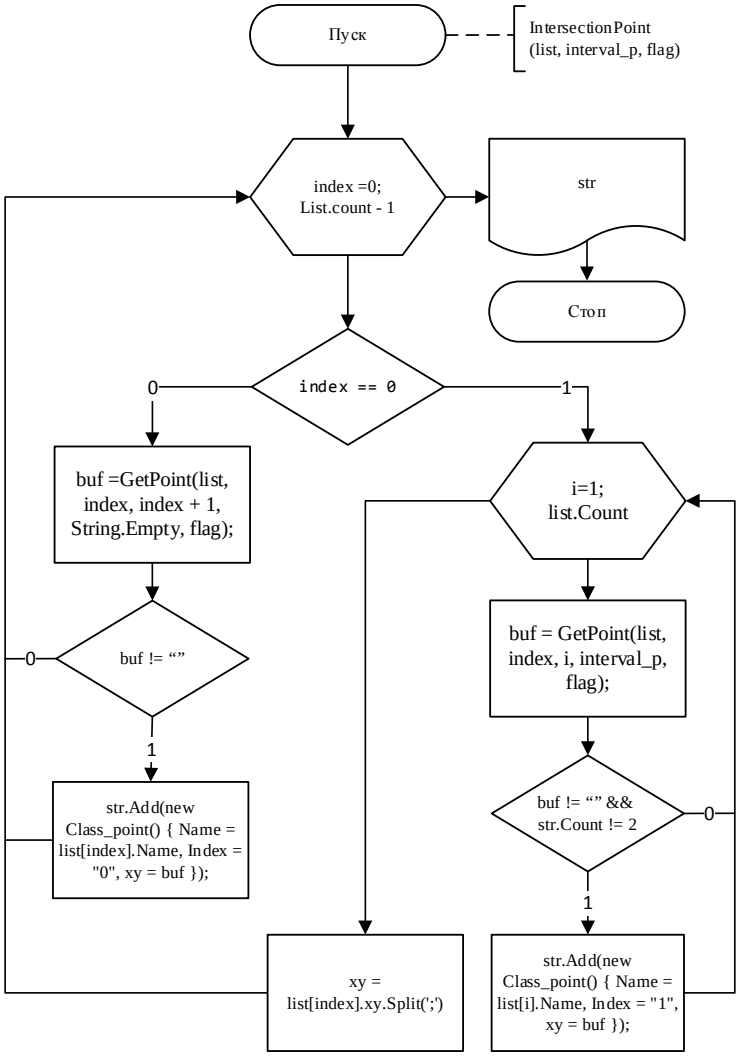

Рис. 7. Алгоритм реалізації процедури IntersectionPoint

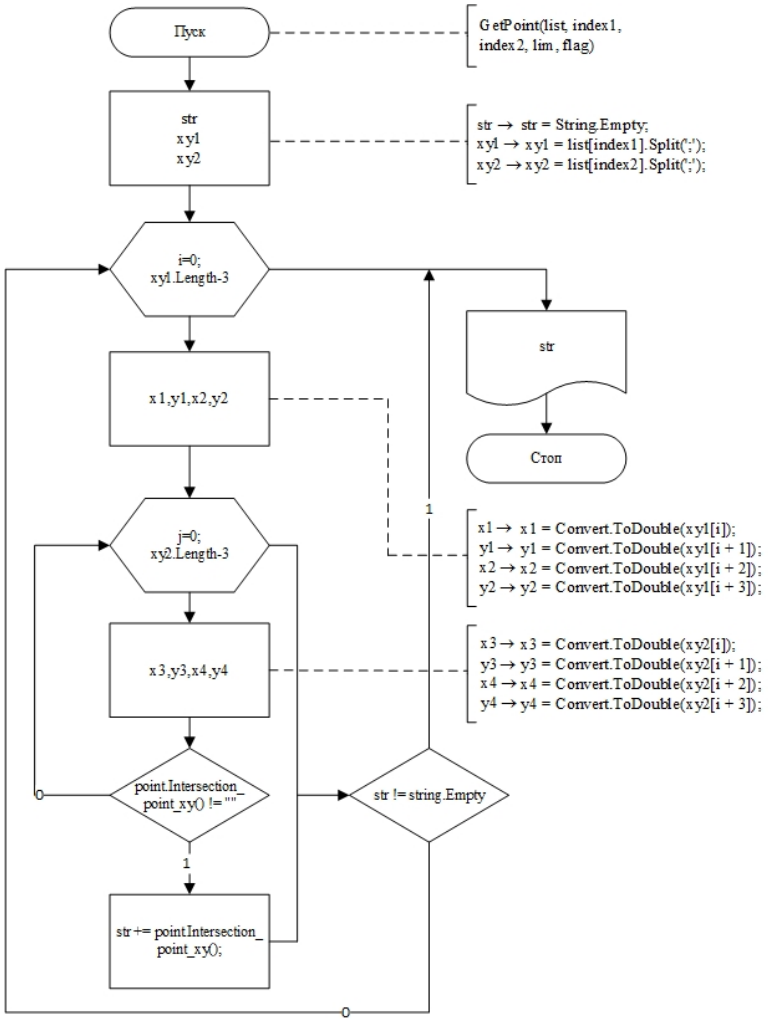

Рис. 8. Алгоритм реалізації процедури GetPoint

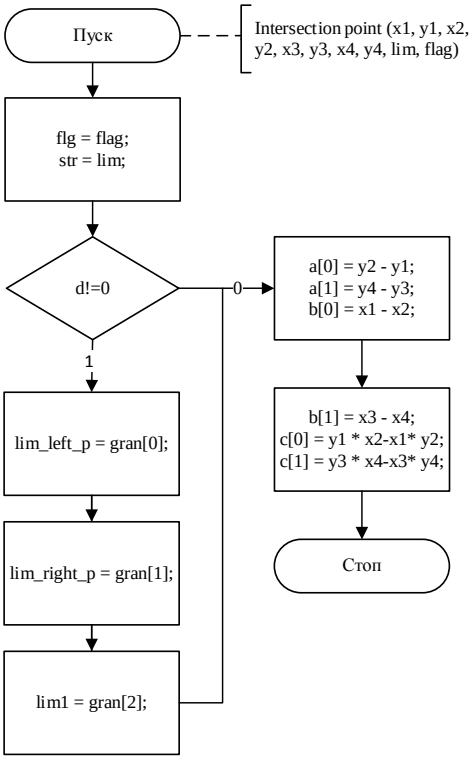

Рис. 9. Алгоритм реалізації процедури Intersection_point

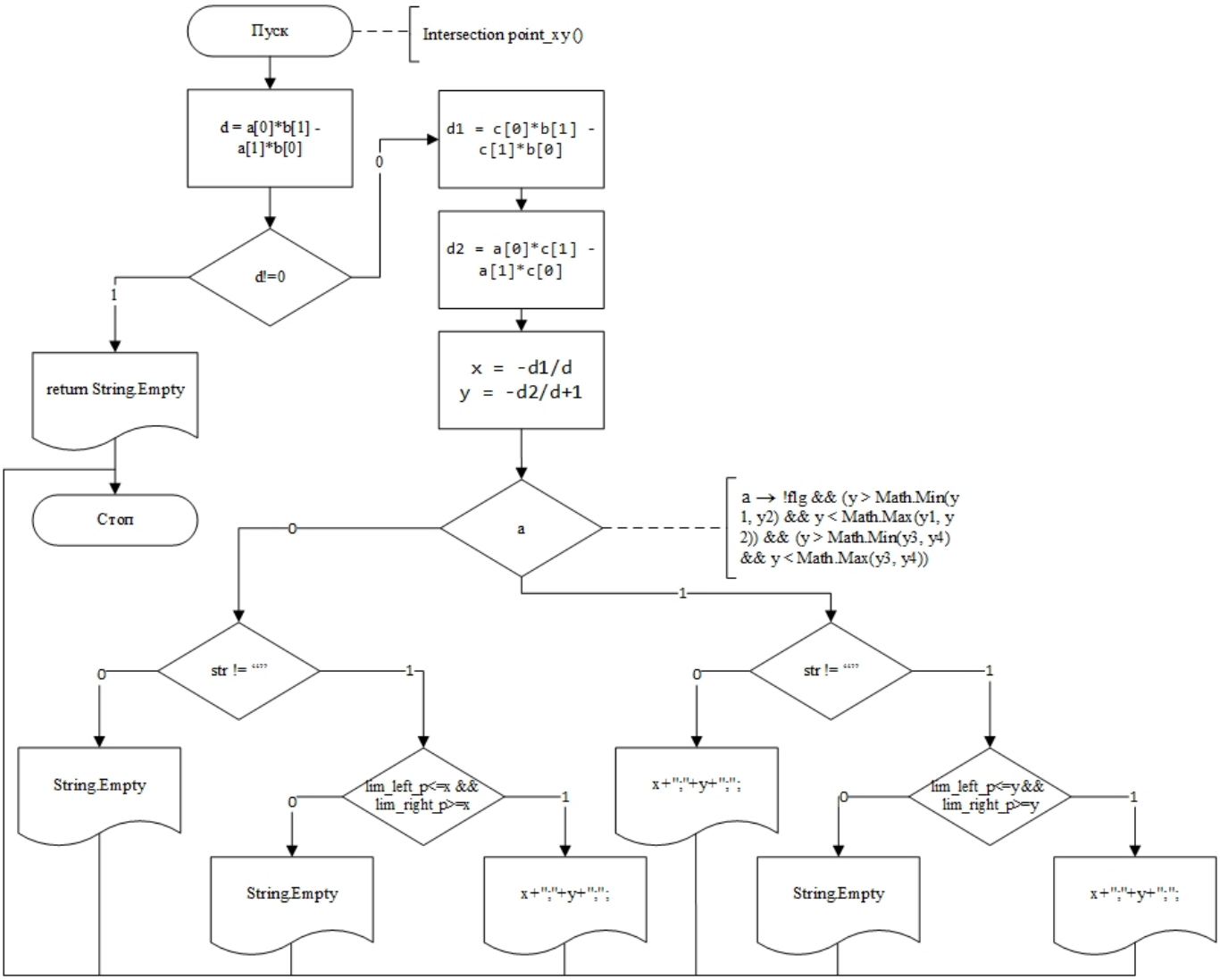

Рис. 10. Алгоритм реалізації процедури Intersection_point_xy

Далі (рис. 1, вершини 15-16) реалізується клас **Intersection_point** (відповідно до функціоналу МАРН і МІТ – див. структуру СВК в [3]), що складається з двох процедур Intersection_point (рис. 9) та Intersection_point_xy (рис. 10). У свою чергу, перша є конструктором, що отримує дані з GetPoint (рис. 8) та визначає коефіцієнти при μ і x, які передаються в другу – Intersection_point_xy, де розраховуються спільні точки і повертаються в GetPoint.

На наступному етапі (рис. 1, вершина 17) викликається процедура **Rect_Area** (відповідно до функціоналу БДП і МРА – див. структуру СВК в [3]) (рис. 11), що відповідає за побудову базових двомірних областей та областей поточного стану (див. етап 7 в [4]) і послідовно активує Draw_main_rect (рис. 12) та Draw_Rect. Процедура Draw_main_rect відповідає за побудову двомірних опорних областей з урахуванням правил $\mathbf{DR_i}$, на основі яких буде визначатися рівень аномального стану системи.

Залежно від отриманих даних про візуалізацію, наприклад, параметрів $P_{31} = P_{SPKOП} = KOП$ і $P_{32} = P_{SPKПОА} = KПОА$ та на основі спільних точок графічних зображень еталонних $\mathcal{I}_{ijs}^{e}$ НЧ та проекцій лінійних компонент, побудованих за допомогою класу Draw_main_object, отримаємо необхідні опорні області. Їх генерація проходить за вище визначеними правилами, тому на графічному зображенні генеруються кольорові області, які відображають рівень аномального стану системи в детекційному середовищі відповідно до $\mathbf{DR_i}$ (див. етап 7 в [4]).

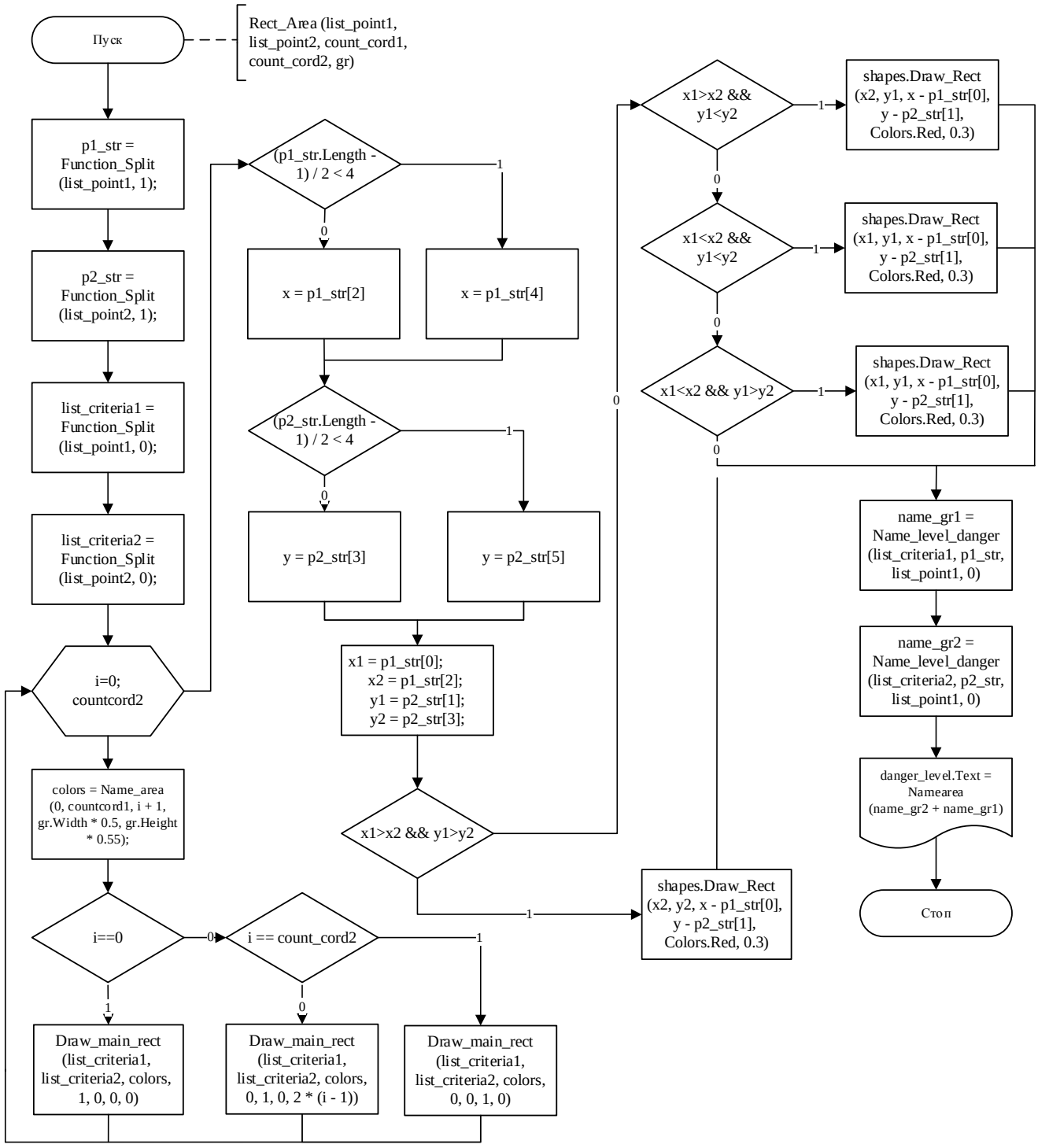

Рис. 11. Алгоритм реалізації процедури Rect_Area

Далі (рис. 1, вершина 18) викликається підпрограма **Line_point_Area** (відповідно до функціоналу МВ – див. структуру СВК в [3]) (рис. 13) і на графічному об'єкті Canvas за допомогою класу Draw_main_object, викликаючи його процедури Draw_main_point (рис. 14) та Draw_main_line (рис. 15), будуються проекції лінійних компонент та спільні точки як на початкових графічних зображеннях, так і на кінцевому зображенні поточного стану. Приклад реалізації роботи цих двох процедур наведений на рис. 16.

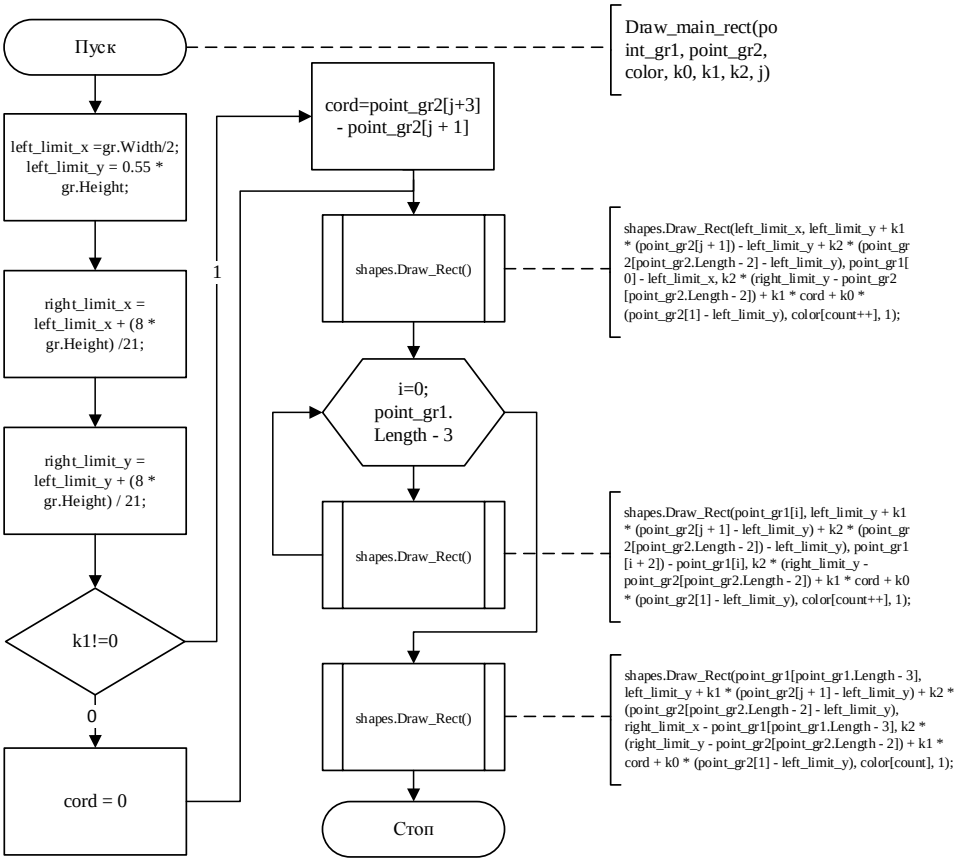

Рис. 12. Алгоритм реалізації процедури Draw_main_rect

Також при побудові графічних елементів в алгоритмі System_level_Click (рис. 1) використовуються додаткові складові, наприклад, клас **Main_figures** включає в себе процедури Draw_polyline (рис. 17), Draw_point і Draw_Rect (рис. 18), які відповідно здійснюють побудову лінійних компонент з різними вхідними характеристиками точок на графічному об'єкті Canvas і прямокутних компонент.

Після отримання всіх даних System_level_Click здійснює побудову області поточного стану (відповідно до функціоналу МВ – див. структуру СВК в [3]), що дозволяє візуально оцінити аномальний стан в системі для прийняття необхідного рішення.

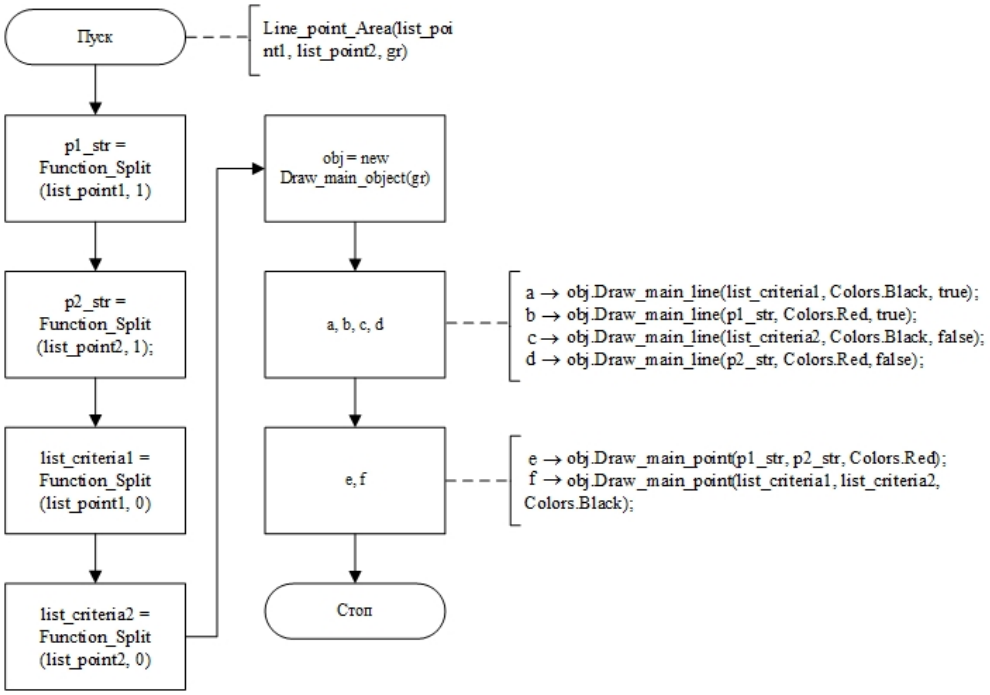

Рис. 13. Алгоритм реалізації процедури Line_point_Area

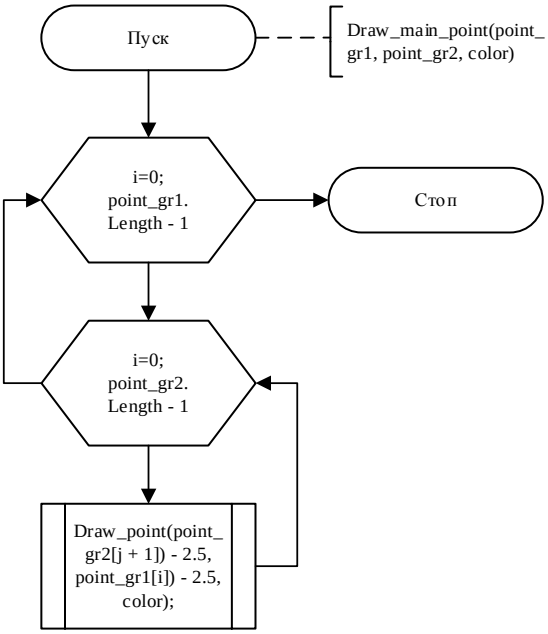

Рис. 14. Алгоритм реалізації процедури Draw_main_point

Фактично процедура генерує поточний блок, наприклад, у вигляді червоної прямокутної області, утвореної за допомогою $\underset{\sim}{P}_{31}^{\tau_f}$ і $\underset{\sim}{P}_{32}^{\tau_f}$, що інтерпретує аномалію в 2-мірному параметричному КОП-КПОА-підсередовищі, породжену відповідним атакуючим SP-середовищем в момент часу τ_f [14]. Приклад роботи ПЗ формування еталонів параметрів з різними вхідними даними наведений на рис. 19 і рис. 20).

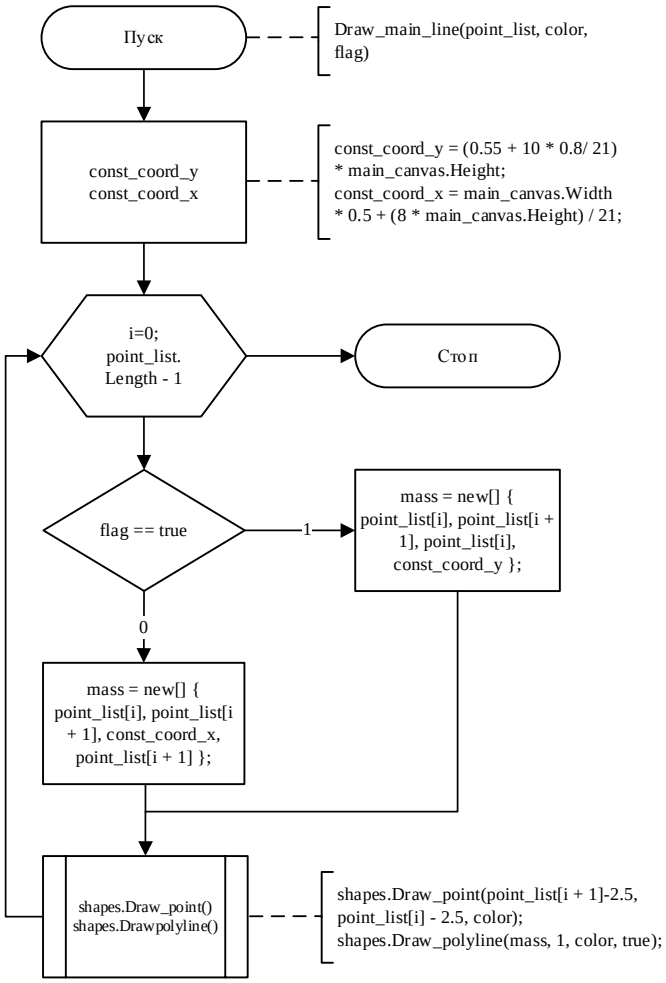

Рис. 15. Алгоритм реалізації процедури Draw_main_line

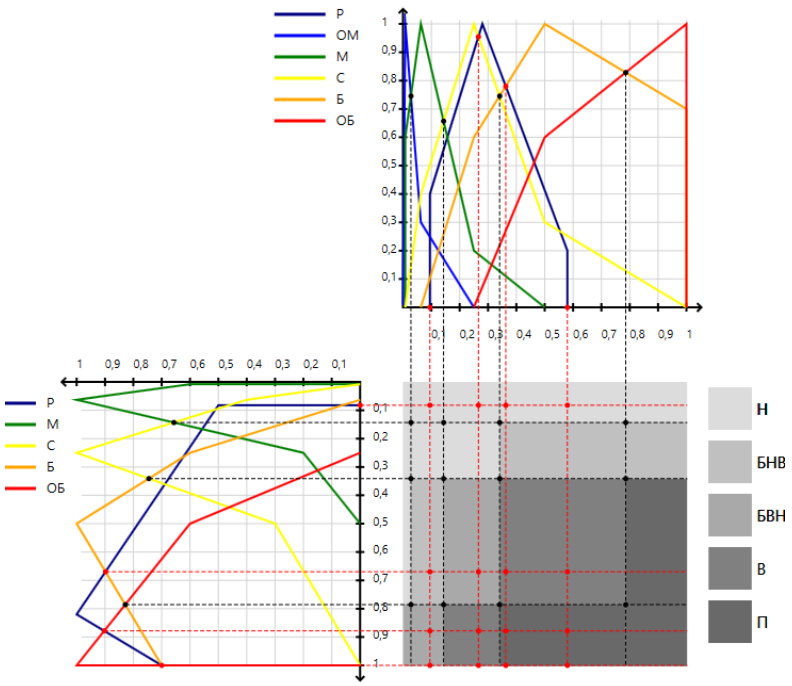

Рис.16. Приклад побудови опорних областей відповідно до Draw_main_rect

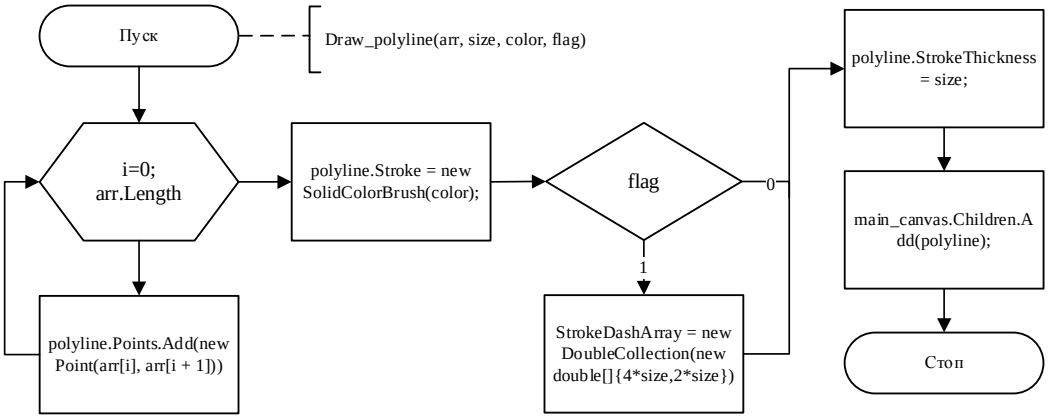

Рис. 17. Алгоритм реалізації процедури Draw_polyline

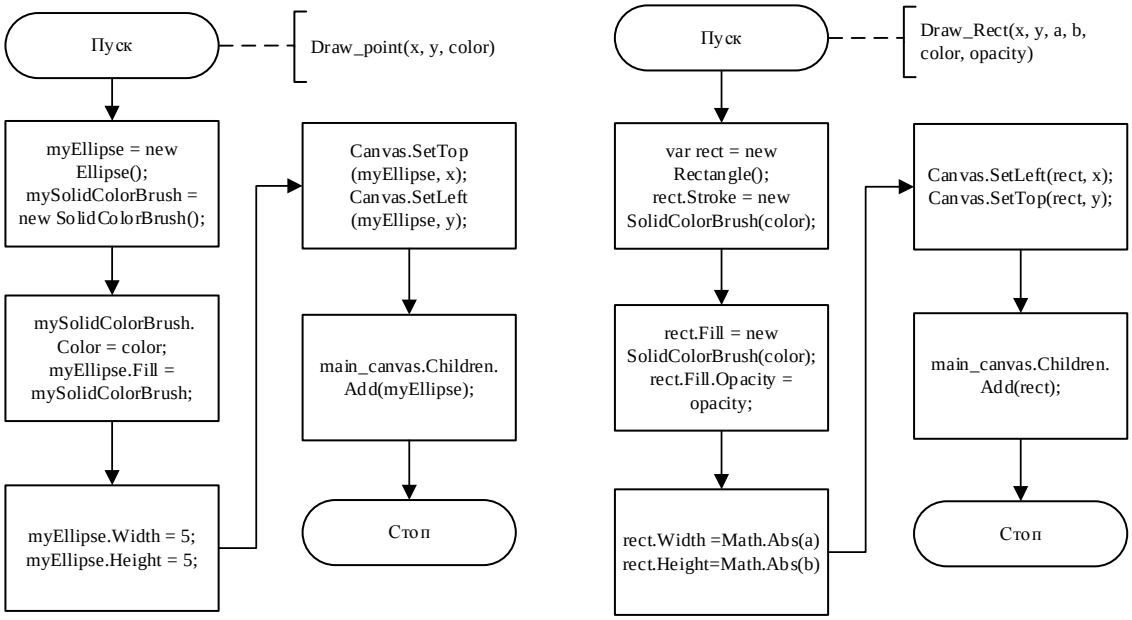

Рис. 18. Алгоритми реалізації процедур Draw_point і Draw_Rect

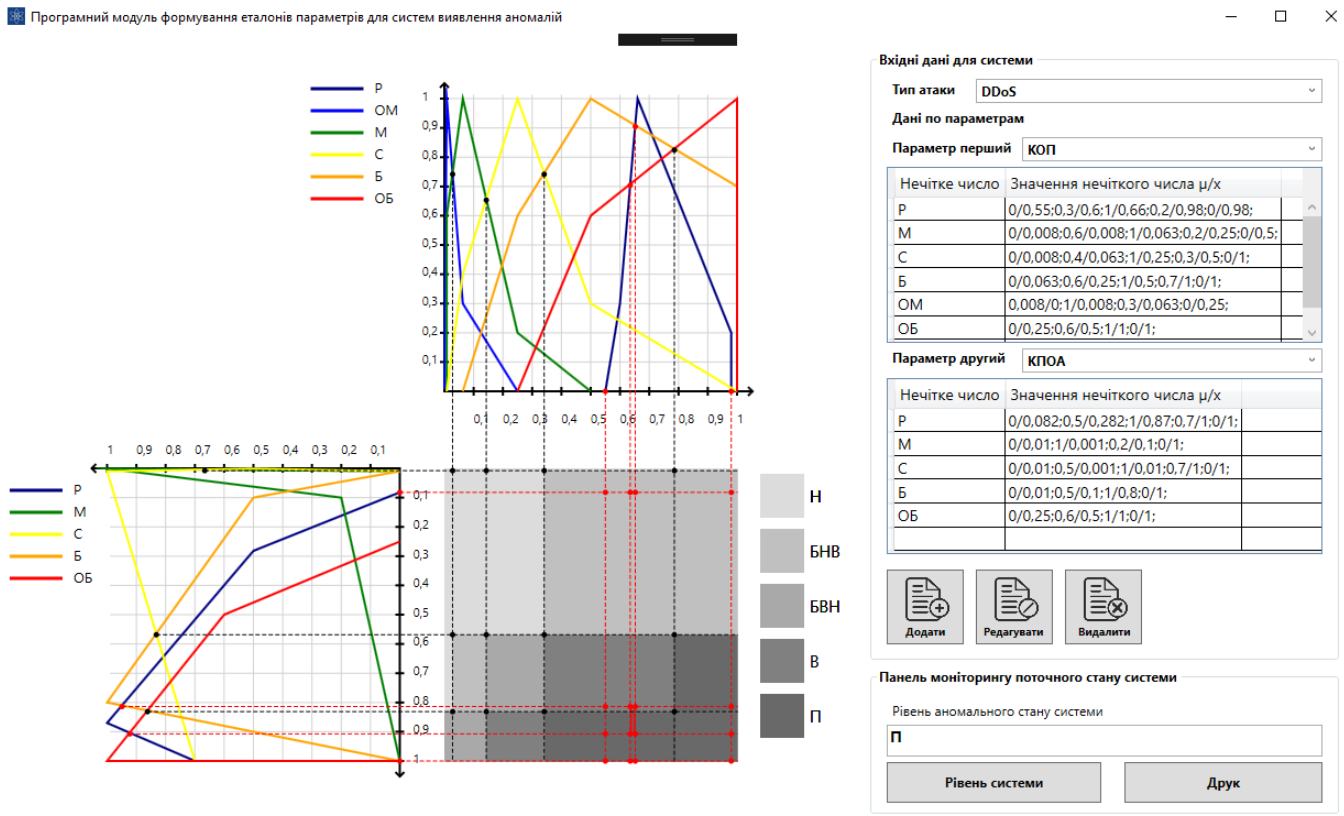

Рис.19. Приклад роботи ПЗ формування еталонів параметрів (визначення поточного стану системи)

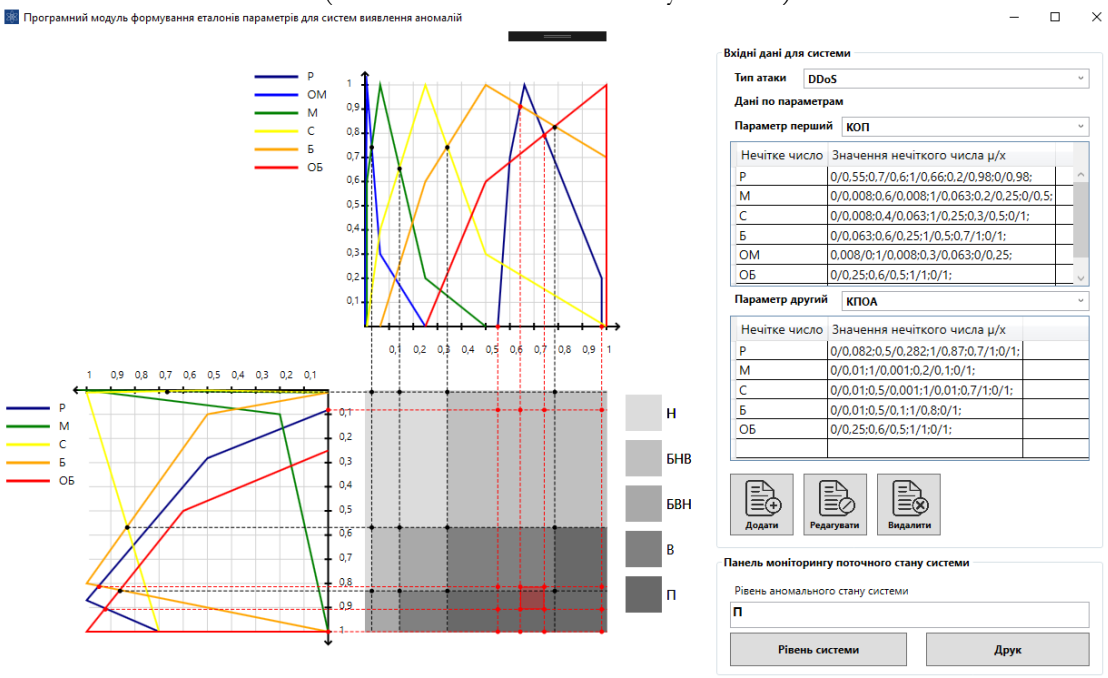

Рис. 20. Приклад роботи ПЗ формування еталонів параметрів (визначення поточного стану системи)

На заключному етапі (рис. 1, вершина 19) використовуються класи **PrintPreviewWindow** та **Print**, що відповідають за створення файлу звіту та його попереднього перегляду. Тобто, користувач за необхідністю у момент часу τ_f може ініціалізувати режим друку, що приведе до створення у буферній пам'яті файлу попереднього перегляду (рис. 21), який можна роздрукувати (рис. 22) або зберегти у форматі pdf (рис. 23).

Друк ініціюється кнопкою «Рівень системи», у результаті чого графічний об'єкт Canvas конвертується в Xaml файл, а решта тексту, заголовок звіту і правило, що спрацювало (відповідно до функціоналу МВ – див. структуру СВК в [3]), генеруються за допомогою стандартного класу FixedDocument,

який дозволяє зручно розмістити текст в звіті. Файл звіту передається у буферну пам'ять, після чого його можна переглядати, змінювати налаштування друку тощо.

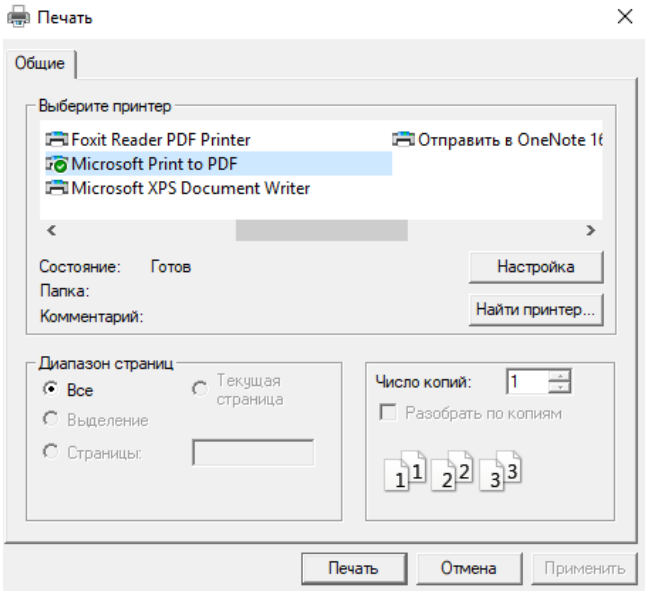

Рис. 22. Вибір варіанту друку документа

У звіті формується відображення рівня аномального стану системи (у тому числі в момент часу τ_f).

Також у розробленому ПЗ використовується модуль **ChildWindow**, який відповідає за створення та редагування $\mathcal{I}^e_{ijs}$ і $\mathcal{P}^{\tau_f}_{ij}$ [7]. Він представлений окремим вікном програми із базовим інтерфейсом для виконання сформованих вище задач. Дані в БДЕ за допомогою функціоналу цього модуля можна модифікувати та переглядати. Аналогічна процедура реалізується при активізації кнопок Додати, Редагувати та Видалити.

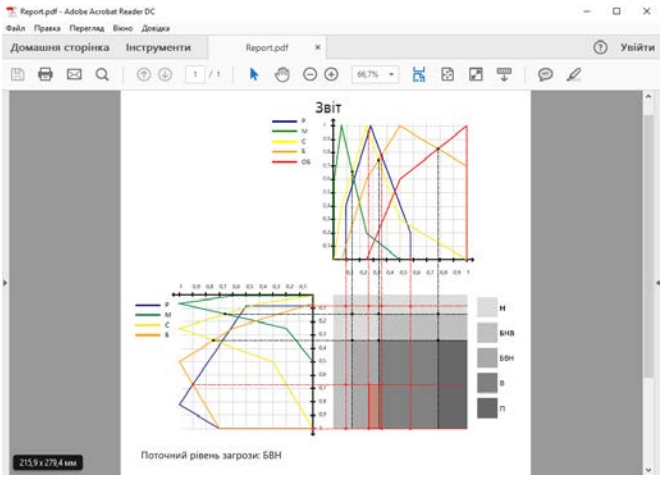

Рис. 23. Приклад друку звіту у форматі pdf

Додавання запису (рис. 24). За допомогою функціоналу вікна «Додати графік» є можливість доповнити такі дані як: назва графічного зображення еталонного та поточного НЧ (обирається за допомогою ComboBox і списку назв), кількість та ініціалізація значень їх координат (за допомогою «+» реалізується додавання/видалення нової пари координат).

Редагування існуючого запису (рис. 25). Процес редагування подібний процесу додавання, оскільки базис роботи цих процедур схожий. Тому, після використання в головному вікні кнопки «Редагувати» з'являється відповідне вікно, де за допомогою функціоналу «Редагування даних графіка» є можливість його модифікації.

Видалення даних (рис. 26). При обранні необхідного рядка запису для видалення використовується функціональна клавіша «Видалити», в результаті чого відбувається видалення даних з БДЕ і її автоматичне оновлення.

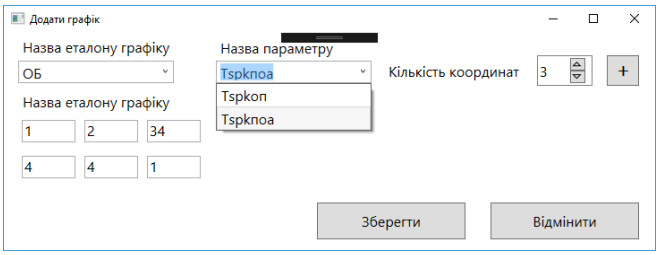

Рис. 24. Вікно додавання запису еталонних значень

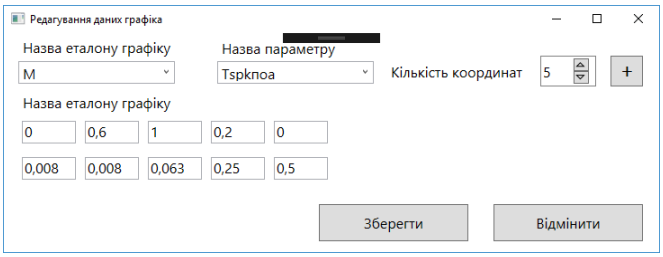

Рис. 25. Вікно редагування запису

Експериментальне дослідження та практичне використання запропонованого ПЗ підтвердило дієвість теоретичних положень, які стали основою розробленого алгоритмічного забезпечення.

Таким чином, у роботі запропоновано ПЗ, яке, за рахунок базового алгоритму та низки розроблених процедур (конструювання координатної сітки; ініціалізації величин на основі набору баз даних та модулів; графічного формування параметрів; пошуку спільних точок відповідно базових правил та графічної інтерпретації результату), дозволяє автоматизувати процес формування еталонів параметрів для сучасних СВА та відображати результати детектування аномального стану у заданий проміжок часу.

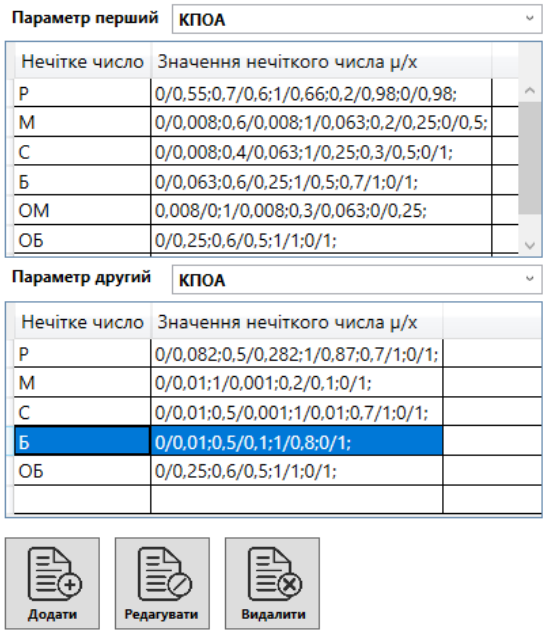

Рис. 26. Виділення рядка необхідного запису та його видалення

ЛІТЕРАТУРА

[1]. Газета.ru. Вымогатели терроризируют интернет [Електронний ресурс]. Режим доступу: https://www.gazeta.ru/tech/2017/08/23/10839932/cyberthreats_2017.shtml?updated (дата звернення 20.08.2018).

[2]. А.Г. Корченко, *Построение систем защиты информации на нечетких множествах [Текст]: Теория и практические решения.* К. : МК-Пресс, 2006, 320 с.

[3]. И. Терейковский, А. Корченко, "Система выявления кибератак", *Безпека інформації*, Т. 23, № 3., С. 176-180, 2017.

[4]. А. Корченко, В. Щербина, Н. Вишневская, "Методология построения систем выявления аномалий порожденных кибератаками", *Захист інформації,* Т. 18, №1, С. 30-38, 2016.

[5]. А. Корченко, "Кортежная модель формирования набора базовых компонент для выявления кибератак", *Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні,* вып. 2 (28), С. 29-36, 2014.

[6]. A. Korchenko, K. Warwas, A. Klos-Witkowska, "The Tupel Model of Basic Components' Set Formation for Cyberattacks", *Proceedings of the 2015 IEEE 8th International Conference on «Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications» (IDAACS'2015),* Warsaw, Poland, September 24-26, 2015, vol. 1, pp. 478-483.

[7]. А. Корченко, "Метод формирования лингвистических эталонов для систем выявления вторжений", *Захист інформації*, Т. 16, №1, С. 5-12, 2014.

[8]. D. Wijayasekara, O. Linda., M. Manic, C.G Rieger, "Mining Building Energy Management System Data Using Fuzzy Anomaly Detection and Linguistic Descriptions", *IEEE Trans. Industrial Informatics*, vol. 10, no. 3, pp 1829-1840, 2014.

[9]. И. Терейковский, А. Корченко, П. Викулов, А. Шаховал, "Модели эталонов лингвистических переменных для обнаружения сниффинг-атак", *Захист інформації*, Т. 19, №3, С. 228-242, 2017.

[10]. І. Терейковський, А. Корченко, П. Вікулов, І. Ірейфідж, "Моделі еталонів лінгвістичних змінних для систем виявлення email-спуфінг-атак", *Безпека інформації*, Т. 24, № 2, С. 99-109, 2018.

[11]. А. Корченко, "Метод фаззификации параметров на лингвистических эталонах для систем выявления кибератак", *Безпека інформації,* № 1 (20), С. 21-28, 2014.

[12]. А. Корченко, "Метод α-уровневой номинализации нечетких чисел для систем обнаружения вторжений", *Захист інформації*, Т. 16, №4, С. 292-304, 2014.

[13]. А. Корченко, "Метод определения идентифицирующих термов для систем обнаружения вторжений", *Безпека інформації,* Т. 20, № 3, С. 217-223, 2014.

[14]. Н. Карпинский, А. Корченко, С. Ахметова, "Метод формирования базовых детекционных правил для систем обнаружения вторжений", *Захист інформації,* Т. 17, №4, С. 312-324, 2015.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ФОРМИРОВАНИЯ ЭТАЛОНОВ ПАРАМЕТРОВ ДЛЯ СИСТЕМ ОБНАРУЖЕНИЯ КИБЕРАТАК

Большинство систем обнаружения вторжений становятся неотъемлемой частью защиты какой-либо сетевой безопасности, они используются для мониторинга подозрительной активности в системе и выявления атакующих действий неавторизованной стороны. Активизация кибератак инициирует создание специальных технических решений способных оставаться эффективными при появлении новых или модифицированных видов киберугроз с неустановленными или нечетко определенными свойствами. Большинство таких систем направлено на выявление подозрительной активности или вмешательства в сеть для принятия адекватных мер по предотвращению кибератак. Актуальными системами обнаружения вторжений являются те, которые ориентированы на идентификацию аномальных состояний, но они имеют ряд недостатков. Более эффективными в этом являются экспертные подходы, основанные на использовании знаний и опыта специалистов соответствующей предметной области. Построение технических решений и создания специальных средств (например, программного обеспечения для систем обнаружения атак, позволяющего детектировать ранее неизвестные кибератаки путем контроля текущего состояния нечетко определенных параметров в слабоформализованной среде окружения), основанных на экспертных подходах, является перспективным направлением исследований. На основе известной системы обнаружения кибератак, которая базируется на методологии выявления аномалий (порожденных кибератаками) и множества соответствующих ме-

тодов и моделей, предложено программное обеспечение, которое, за счет базового алгоритма и ряда разработанных процедур (конструирование координатной сетки; инициализации величин на основе набора баз данных и модулей; графического формирования параметров; поиска общих точек соответственно базовых правил и графической интерпретации результата), позволяет автоматизировать процесс формирования эталонов параметров для современных систем обнаружения атак и отображать результаты детектирования аномального состояния в заданный промежуток времени.

Ключевые слова: атаки, кибератаки, аномалии, системы обнаружения вторжений, системы обнаружения атак, системы выявления кибератак, выявление аномалий в информационных системах.

THE SOFTWARE FOR THE FORMATION OF PARAMETERS ETALONS FOR CYBER-ATTACKS DETECTION SYSTEMS

The overwhelming majority of intrusion detection systems become an integral part of the protection of any network security, they are used to monitor suspicious activity in the system and detect an attack by an unauthorized party. Activating of cyberattacks initiates the creation of special technical solutions that can remain effective when new or modified types of cyber threats appear with unidentified or unclearly defined properties. The majority of such systems are aimed to detect suspicious activity or interfering in the network to take adequate measures to prevent cyber attacks. Current systems for detecting intrusions are those that are aimed to identify abnormal states, but they have a number of disadvantages. More effective in this regard are expert approaches based on the use of knowledge and experience of specialists in the relevant subject field. The construction of technical solutions and the creation of special tools (for example, software for detection systems that allow detection of previously unknown cyber attacks by controlling the current state of unclear parameters in a poorly formalized environment, based on expert approaches, is a promising area of research. Based on the well-known cyberattack detection system which is based on the methodology for detecting anomalies generated by cyber attacks and the set of appropriate methods and models, software provided by the basic algorithm and a number of developed procedures (grid construction, initialization of values based on a set of databases and modules; graphic forming of parameters, search of common points in accordance with the basic rules and graphic interpretation of the result) allows to automate the process of forming parameters etalons for modern attack detection systems and display the results of detecting abnormal states at a given time interval.

Keywords: attacks, cyber attacks, anomalies, intrusion detection systems, attack detection systems, cyberattack detection systems, detection of anomalies in information systems.

Корченко Анна Олександрівна, кандидат технічних наук, доцент, доцент кафедри безпеки інформаційних технологій Національного авіаційного університету.
E-mail: annakor@ukr.net.
Корченко Анна Александровна, кандидат технических наук, доцент, доцент кафедры безопасности информационных технологий Национального авиационного университета.
Korchenko Anna, PhD in Eng., Associate Professor of Academic Department of IT-Security, National Aviation University (Kyiv, Ukraine).

Заріцький Олег Володимирович, доктор технічних наук, доцент кафедри безпеки інформаційних технологій Національного авіаційного університету.
E-mail: olegzaritskyi@gmail.com.
Зарицкий Олег Владимирович, доктор технических наук, доцент кафедры безопасности информационных технологий Национального авиационного университета.
Zaritskyi Oleg, Doctor of Engineering Sciences, Associate Professor of Academic Department of IT-Security, National Aviation University (Kyiv, Ukraine).

Паращук Тарас Іванович, студент кафедри безпеки інформаційних технологій Національного авіаційного університету.
Email: taras1039@ukr.net.
Паращук Тарас Иванович, студент кафедры безопасности информационных технологий Национального авиационного университета.
Paraschuk Taras, student of Academic Department of IT-Security, National Aviation University (Kyiv, Ukraine).

Бичков Володимир Вячеславович, старший викладач кафедри безпеки інформаційних технологій Національного авіаційного університета.
Email: bychkov.volodymyr@gmail.com.
Бычков Владимир Вячеславович, старший преподаватель кафедры безопасности информационных технологий Национального авиационного университета.
Bychkov Volodymyr, National Aviation University, Academic Department of Information Technology Security, senior lecturer (Kyiv, Ukraine).