Тернопільський національний економічний університет

На правах рукопису

Биковий Павло Євгенович

УДК 004.415 + 654.924

МЕТОДИ І ЗАСОБИ ОПТИМІЗАЦІЇ ФУНКЦІОНАЛЬНО-ВАРТІСНИХ ХАРАКТЕРИСТИК КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ СИГНАЛІЗАЦІЇ НА ОСНОВІ ГЕНЕТИЧНОГО АЛГОРИТМУ

05.13.05 – комп'ютерні системи та компоненти

Дисертація на здобуття наукового ступеня
кандидата технічних наук

Науковий керівник
Кочан Володимир Володимирович,
кандидат технічних наук,
доцент

Тернопіль – 2010

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ГА – генетичний алгоритм

СТС - система тривожної сигналізації

КСТС – комп'ютеризована система тривожної сигналізації

СФЗ – система фізичного захисту

СКД – система контролю доступу

КПП – контрольно-пропускний пункт

КС - комп'ютерна система

ПКП – приймально-контрольний прилад

МКПКП - мікроконтролер приймально-контрольного пристрою

гомогенне покриття – для охорони ділянок периметру використовуються сповіщувачі однакової моделі;

гетерогенне покриття – для охорони ділянок периметру можуть використовуватися сповіщувачі різних типів;

тип сповіщувача – конкретна реалізація сповіщувача того чи іншого принципу дії;

умови ділянки – специфіка ділянки з врахуванням загроз та завад

Z – кількість ділянок, з яких складається периметр;

n_i – кількість типів сповіщувачів, які згідно умов i-тої ділянки придатні для її охорони;

M_i – кількість Парето-оптимальних варіантів покриття i-тої ділянки;

n – кількість альтернативних варіантів сповіщувачів, що дорівнює $\sum n_i$;

S_i – протяжність і-тої ділянки периметру;

L_j – область дії компонента j-того типу;

p_j – імовірність працездатності сповіщувача j-того типу;

r_j – імовірність невиявлення загрози працездатним сповіщувачем j-того типу;

q_j – імовірність хибного спрацювання сповіщувача j-того типу.

ВСТУП

Актуальність теми. Згідно ДСТУ IEC 60839-2001, система тривожної сигналізації (СТС) є розподіленою системою, яка містить сповіщувачі (detectors – англ., извещатели – рос.), що за допомогою провідного або безпровідного зв'язку взаємодіють з приймально-контрольним приладом, який, в свою чергу, обробляє сигнали від сповіщувачів та передає результат до засобів візуалізації та прийняття рішень. СТС периметру території відіграють роль першої основної захисної лінії, вони є важливим елементом різноманітних охоронних систем. Такі СТС в більшості випадків є багатокутником, сторони якого є зонами СТС, кожна з яких має свою специфіку охорони відносно порушень, які потрібно виявляти на ній. Найбільш ефективними з точки зору сукупності функціонально-вартісних характеристик є комп'ютеризовані системи тривожної сигналізації (КСТС), які мають всі ознаки спеціалізованих комп'ютерних систем.

Значний вклад в розробку СТС та оцінювання їхньої ефективності внесли науковці М.Л. Гарсіа, К.Дж. Тар, Р.Г. Магауенов, В.В. Домарев, Б.С. Введенський, С.С. Звежинський, И.В. Иванов, О.А. Панін та ін.

На сьогодні фірми, які встановлюють СТС, зазвичай використовують готові шаблонні рішення, котрі можуть не врахувати всіх особливостей кожної із ділянок периметру відповідної території, зокрема, при використанні багатошарового захисту, де виникає багато питань взаємодії різних видів і засобів захисту. Як правило, оцінка СТС при розробці враховує лише два фактори – покриття зони областями дії компонентів і ціну системи, проте доцільність використання саме цих компонентів не завжди відповідає їхнім характеристикам та індивідуальним особливостям ділянок підохоронного об'єкту. Крім того, часто не враховують велику різноманітність компонентів різних фірм та масштабів об'єкту, що суттєво впливає на ресурсоємність СТС (зокрема, її ціну). Тому актуальною є задача пошуку компромісів між ресурсоємністю СТС та її ефективністю (ймовірністю виявлення загроз і рівнем хибних тривог).

Аналіз наведених недоліків виявив шляхи вирішення поставленої задачі, які розділено на два напрямки – оптимальне використання компонентів СТС та їх вдосконалення. Перший напрям полягає у створенні методів, що забезпечують розробку всієї СТС із необхідною ефективністю при мінімумі ресурсоємності. Другий напрям може включати розробку нового типу компонентів, що мають кращі параметри, ніж їх попередники (це є складною задачею), або модифікацію існуючих компонентів, що вимагає менше зусиль. Реалізація вказаних напрямків вимагає переходу від компонування СТС до їх розробки, що може бути рентабельним тільки при її автоматизації. Тому покращення функціонально-вартісних характеристик СТС (і КСТС, як їх підмножини), зокрема, їх ефективності та ресурсоємності, а також їх компонентів, шляхом автоматизованого синтезу та відбору кращих рішень є актуальною задачею.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконана в рамках плану наукових досліджень, які проводилися кафедрою інформаційно-обчислювальних систем і управління та науково-дослідним інститутом Інтелектуальних комп'ютерних систем Тернопільського національного економічного університету, зокрема:

- міжнародного американсько-українського науково-дослідного проекту "Проектування дистрибутивної сенсорної мережі для безпеки острова Ayers Island з використанням технології функціонально-вартісного аналізу", грант CRDF FSTM UM2-5012-TE-03 (2003-2005 рр.), котрий виконувався згідно програми "Перші кроки до ринку" Фонду Цивільних Досліджень і Розвитку США спільно з корпорацією Trefoil, штат Мейн, США;
- міжнародного українсько-турецького науково-дослідного проекту "Розробка методів проектування та оптимізації систем виявлення порушників безпеки", згідно договору № М/47-2008 від 27.03.08, що виконувався за підтримки МОН України спільно з Інститутом технологій м. Гебзе, Республіка Туреччина;

- теми НДІ ІКС 0106U010731: “Розробка теоретичних основ підтримки прийняття рішень для синтезу розподілених систем безпеки” (2006-2009 рр.).

Мета і завдання дослідження. Метою дисертаційної роботи є створення методів і засобів оптимізації функціонально-вартісних характеристик комп’ютеризованих систем тривожної сигналізації на основі автоматичного агрегування їх структур та відбору кращих рішень, що базуються на використанні генетичного алгоритму (ГА).

Для досягнення мети необхідно вирішити наступні завдання:

- проаналізувати та оцінити функціонально-вартісні характеристики типових СТС з метою виявлення їх основних недоліків і шляхів покращення;
- розробити метод формування множини структур СТС, що є допустимими розв’язками задачі оптимізації функціонально-вартісних характеристик СТС;
- розробити методи оптимального відбору кращих варіантів на множині можливих структур СТС;
- розробити методологію застосування пропонованих методів при розробці СТС;
- розробити автоматизовану систему оптимального відбору кращих варіантів серед множини можливих структур СТС для представлення користувачеві;
- розробити СТС та провести їх функціонально-вартісний аналіз;
- вдосконалити функціонально-вартісні характеристики компонентів КСТС для забезпечення кращих функціонально-вартісних характеристик мережі;
- розробити методи захисту зв’язку в запропонованих КСТС;
- дослідити зразки впроваджених КСТС на базі відібраних кращих варіантів та проаналізувати їх функціонально-вартісні характеристики.

***Об’єкт дослідження*:** процес розробки комп’ютеризованих систем тривожної сигналізації.

Предмет дослідження: методи і засоби оптимізації функціонально-вартісних характеристик комп’ютеризованих систем тривожної сигналізації та вдосконалення їх компонентів.

***Методи дослідження*:** функціонально-вартісний аналіз, нечіткі множини, методи багатокритеріальної оптимізації, зокрема, генетичні алгоритми, морфологічний аналіз, методи аналізу та синтезу електричних кіл, структурний синтез.

Наукова новизна одержаних результатів.

1. *Вперше розроблено* метод відображення хромосом генетичного алгоритму в область аргументів комбінаторної задачі багатокритеріальної оптимізації функціонально-вартісних характеристик комп'ютеризованих систем тривожної сигналізації, що, на відміну від класичного генетичного алгоритму та схем перебору, зосереджує пошук на множині лише допустимих розв'язків задачі, дозволяючи зменшити часову складність алгоритму оптимізації до 69 % і збільшити кількість отриманих за одиницю часу Парето-оптимальних СТС в середньому на 15 %.

2. *Вперше* запропоновано організацію послідовного інтерфейсу, що, на відміну від існуючих, базується на виявленій максимальній кількості елементів інтерфейсу, які можна використати для захисту повідомлень, їх комбінованому застосуванні, а також прихованій детермінованій та псевдовипадковій їх заміні, що забезпечує надійний захист мережі від імітації сповіщувачів та індивідуальний захист систем, перешкоджає зловмиснику вивчати повідомлення сповіщувачів при штучно створених атаках і дає можливість створення пасток для зловмисника.

3. *Удосконалено* метод оцінювання функціональних характеристик комп'ютеризованих систем тривожної сигналізації, який, на відміну від існуючих, враховує невизначеності інтенсивностей завад і вразливість до них компонентів систем шляхом використання нечітких множин та їх дефазифікації при оцінюванні ризику проникнення порушника, що дозволило формалізувати ризики невиявлення загроз сповіщувачами в конкретних умовах їх роботи.

4. *Отримав подальший розвиток* метод розробки комп'ютеризованих систем тривожної сигналізації, який, на відміну від існуючих, базується на запропонованих методах їх структурної оптимізації та оцінки функціональних

характеристик при автоматизованому агрегуванні і відборі кращих рішень, що дозволило покращити функціонально-вартісні характеристики таких систем.

Практичне значення одержаних результатів. На основі запропонованих методів реалізовано програмні модулі, розроблені в середовищах MATLAB і Microsoft Visual Studio 2008 на мовах програмування Visual Basic та C#, об'єднані в систему відбору кращих варіантів серед множини структур СТС. При її використанні виявлено протиріччя серед існуючих СТС і запропоновано його усунення шляхом розробки інтерфейсного контролера для сповіщувачів і приймально-контрольного приладу. Результати роботи використані при:

- створенні КСТС острова Ayers Island (грант № CRDF FSTM UM2-5012-TE-03), що дозволило за прийнятний час виявити кращі рішення, що не були виявлені обмеженим за часом послідовним перебором, та формалізувати оцінки ризику невиявлення загроз сповіщувачами (акт про впровадження від 10.11.2009 р.);
- розробці системи оптимізації структур СТС (договір № М/47-2008), що забезпечило зменшення часової складності алгоритму оптимізації до 69 % (акт про впровадження від 22.06.2010 р.);
- створенні СТС НДІ ІКС (тема НДІ ІКС 0106U010731), що дозволило зменшити довжину ліній зв'язку в 2,5 разів без втрат інформативності (акт про впровадження від 17.12.2009 р.);
- викладанні дисциплін "Системи передачі даних", "Мікроконтролери", "Мікроконтролери і спецпроцесори" (акт про впровадження від 31.05.2010 р.).

Особистий внесок здобувача. Основні результати, що виносяться на захист, отримані здобувачем особисто [8, 10, 12, 13, 14, 18, 90]. У роботах, опублікованих у співавторстві, здобувачу належать: у [9] – методика уніфікації техніко-економічних показників компонентів СТС; у [73, 74] – алгоритм визначення ключових функціональних показників компонентів систем, структура бази даних, аналіз множини формалізованих показників, модифікація методу морфологічних матриць; у [88] – спеціалізований мережевий контролер систем тривожної сигналізації; у [87, 126] – методи розробки та оптимізації

СТС; у [92, 93] формалізовано обмеження та критерії комбінаторної задачі покриття сповіщувачами підохоронного об'єкта та ГА для її розв'язання; у [15, 17, 89] – програмно-апаратний метод захисту повідомлень в мережі сповіщувачів КСТС; у [16, 91] – метод оцінювання вразливості СТС периметру території з врахуванням неповної інформації про характеристики сповіщувачів та охоронних зон.

Апробація результатів дисертації. Основні положення дисертації представлені і обговорювалися на 11-ти міжнародних конференціях і симпозіумах: міжнародній конференції IEEE "Інтелектуальні системи" (Варна, Болгарія, 2004 р.); 9-тій та 13-тій науковій конференції Тернопільського державного технічного університету імені Івана Пулюя (Тернопіль, 2005 р. та 2009 р.); 3-тьому, 4-тому та 5-тому міжнародному науково-технічному симпозіумі "Інтелектуальні засоби збору даних і сучасні обчислювальні системи: розробка та застосування" (Софія, Болгарія, 2005 р.; Дортмунд, Німеччина, 2007 р.; Ренде, Італія, 2009 р.); другій міжнародній конференції IEEE "Технології для охорони батьківщини та безпека" (Стамбул, Туреччина, 2006 р.); міжнародній науково-технічній конференції "Комп'ютерні системи та мережні технології" (Київ, 2008); міжнародній конференції IEEE "Обчислювальний інтелект для систем вимірювання та застосувань" (Стамбул, Туреччина, 2008 р.); IX міжнародній конференції "Контроль і управління в складних системах" (Вінниця, 2008 р.), 10-тій та 11-й міжнародних науково-практичних конференціях "Сучасні інформаційні і електронні технології" (Одеса, 2009 р., 2010 р.).

Публікації. За результатами проведених наукових досліджень опубліковано 20 друкованих робіт, серед них 6 статей у фахових виданнях, з них 4 одноосібні, 14 доповідей і тез в збірниках вітчизняних та міжнародних наукових конференцій.

Структура та об'єм роботи. Дисертаційна робота складається зі вступу, чотирьох розділів, висновків, додатків і списку використаних джерел. Загальний обсяг роботи становить 187 сторінок, із них 143 сторінки основного

тексту, що включає 52 рисунки і 9 таблиць. Список використаних джерел 127 найменувань, додатки – на 29 сторінках.

РОЗДІЛ 1
АНАЛІЗ СУЧАСНОГО СТАНУ СИСТЕМ ТРИВОЖНОЇ СИГНАЛІЗАЦІЇ

1.1. Узагальнена структура систем тривожної сигналізації

Згідно [51], системи охорони об’єкта є складними комплексами, що включають фізичний захист (служба охорони, наряд міліції), інженерний захист (огорожі, решітки, сталеві двері, замки, сейфи), технічний захист (засоби тривожної сигналізації, системи телевізійного (відео) спостереження, системи контролю доступу, системи виявлення зброї, системи пожежної сигналізації та ін.), системи спеціального захисту (захист комунікаційних мереж, захист приміщень від прослуховування). Відповідно до [32], поняття підохоронний об’єкт може включати територію, розміщені на ній споруди (та приміщення в них), транспортні засоби тощо. Прикладом може служити підохоронний об’єкт, представлений на рис. 1.1, де представлена огороджена територія, на якій розміщені будівлі 5.

Як показує досвід, узагальнений в [20, 36, 50, 51, 117], найбільш ефективними є системи охорони об’єкта, побудовані за принципом створення послідовних рубежів охорони, котрі мають виявляти порушника і перешкоджати йому дістатись до цілі. Першим та одним з найважливіших рубежів охорони є периметр території об'єкта [19, 35, 50].

На рис. 1.1 територія підохоронного об'єкта розділена на декілька можливих рубежів, відповідно до особливостей їх функціонування. Рубіж 1, буферний, призначений для попередньої оцінки загрози, де зазвичай використовують активні інфрачервоні, мікрохвильові та інфрачервоні сповіщувачі [5, 33, 117]. Рубіж 2, огорожа території, призначений для відлякування випадкових порушників (які не є зловмисниками, тобто не мають злочинних намірів [51]) та затримки порушників, що цілеспрямовано намагаються проникнути на підохоронний об’єкт (збільшення часу, необхідного для проникнення). Для виявлення останніх використовують

сповіщувачі вібрації, натягу проводів, оптоволоконні, індуктивні сенсорні кабелі механічної дії на огорожу, ємнісні та сповіщувачі електричного поля [117]. Рубіж 3, внутрішній, призначений для підвищення надійності виявлення порушників, а також можливої їх класифікації з допомогою сповіщувачів іншого принципу дії. Тут використовують приховані в землі сенсори тиску (тензочутливі коаксіальні та магнітні кабелі, оптоволоконні лінії та рідинні трубчаті тензометри), акустичні сенсори (геофони) та кабельні сенсори електромагнітного поля [117]. Рубіж 4, захищена територія, може не мати засобів суцільного захисту, але мати локальні захищені ділянки. Тут часто використовують пасивні та активні інфрачервоні сповіщувачі. Рубіж 5, будівлі на захищеній території, найчастіше мають свої засоби захисту, відповідні до їх призначення та особливостей. Тут використовують пасивні інфрачервоні, магнітоконтактні, розбиття скла та ін. сповіщувачі [5]. Рубіж 6, місце доступу на захищену територію, є обов'язковим елементом системи охорони об'єкта, але його функції значно відрізняються від функцій інших рубежів, тому тут використовують системи контролю доступу, які в роботі не розглядатимуться. Будь-який рубіж може бути оснащений системою відеоспостереження, яка є складним універсальним засобом, однак він вимагає постійної взаємодії з людиною-оператором [5]. Зазвичай системи відеоспостереження виконують комплекс функцій, який включає не тільки виявлення порушника, тому специфіка їх використання вимагає окремого дослідження.

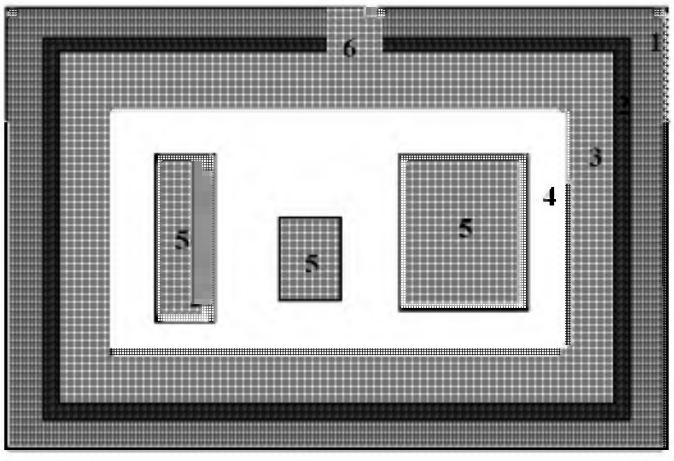


Рис. 1.1. Підохоронний об'єкт та виділені рубежі охорони

Традиційно до системи охорони периметра відносять рубежі 1…3. Для цієї системи характерним є виконання двох взаємопов'язаних задач – виявлення порушника та збільшення часу проникнення порушника на підохоронний об'єкт (захищену територію). Таке збільшення часу необхідне для того, щоб полегшити службі охорони об'єкта нейтралізацію порушника [20]. В даній дисертаційній роботі розглядаються питання, пов'язані з системами, що реалізують першу функцію – виявлення порушника. Такі системи, згідно [32, 33] називаються системами тривожної сигналізації.

Виходячи із сказаного, охоронні системи і СТС як їх різновид є складним розподіленим комплексом засобів, які повинні тісно взаємодіяти в процесі охорони об'єкта. Сучасні СТС переважно будуються як дистрибутивні сенсорні мережі, послідовність взаємодії компонентів яких (апаратних та програмних) можна представити у вигляді графу взаємодії компонентів (рис. 1.2) [28]. В такій СТС сповіщувачі 1…6 можуть взаємодіяти з приймально-контрольним приладом (ПКП) 11 наступним чином:

- безпосередньо з допомогою провідного зв'язку (сповіщувач 3);
- з допомогою провідного зв'язку через розширювачі (периферійні блоки) 7 (сповіщувачі 1-2);
- безпосередньо з допомогою безпровідного зв'язку (сповіщувач 4);
- з допомогою безпровідного зв'язку через безпровідний розширювач (периферійний блок) 8 (сповіщувачі 5-6).

При поступленні від сповіщувачів сигналу про виникнення порушення ПКП 11 включає звукову сигналізацію 9 та індикує зону порушення з допомогою алфавітно-цифрового табло або мнемосхеми 10. Пульт керування 12 служить для встановлення і зняття з охорони окремих сповіщувачів. Сигнал тривоги з ПКП 11 поступає також на вищий ієрархічний рівень, в центр прийняття рішень 13 охоронної системи. В центр 13 сходиться інформація також від інших підсистем, наприклад, відео-спостереження 14, пожежної сигналізації – 15 та контролю доступу - 16.

Центр прийняття рішень 13 представляє собою людино-машинний комплекс, завданнями якого є: оцінити загрозу (по можливості класифікувати її), прийняти рішення про використання методів протидії, віддати наказ підсистемі реагування 17.

На рис. 1.2 показано також традиційний поділ функцій між СТС і охоронною системою [33]: як видно, СТС є однією з найважливіших складових охоронної системи, яка виконує найскладнішу функцію – здобуття інформації. В цю функцію входять задачі виявлення факту порушення безпеки підохоронного об'єкта, виявлення місця порушення, класифікація характеру порушення та формування повідомлення про порушення (індикація, звуковий сигнал). Таким чином, СТС є складною, функціонально закінченою підсистемою, яка входить в систему вищого ієрархічного рівня – охоронну систему. Тому СТС можна розглядати як самостійну систему і вдосконалювати її незалежно від інших підсистем.

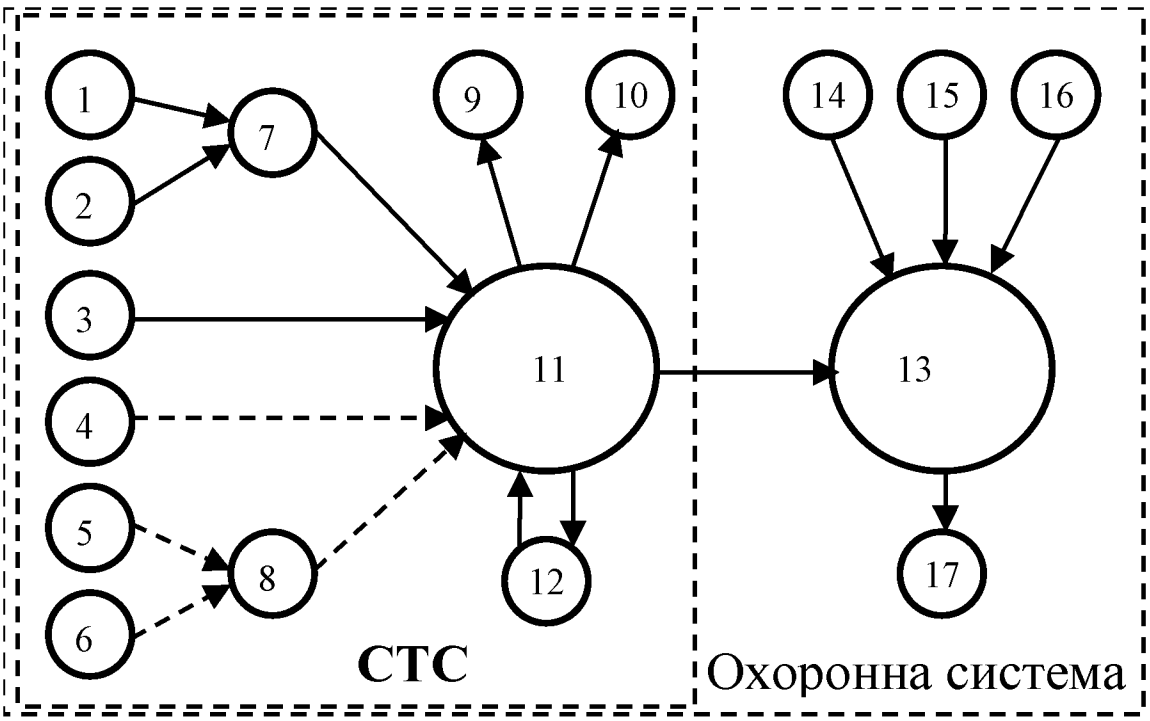


Рис. 1.2. Граф взаємодії компонентів охоронної системи

Відповідно до загальних принципів побудови інформаційних систем та руху в них даних [55], структура СТС визначається розподілом функцій

обробки інформації від сповіщувачів між ПКП та іншими блоками, а також способом зв'язку їх із сповіщувачами [51]. На вибір типу структури впливають: якісний і кількісний склад сповіщувачів та інших блоків; степінь централізації управління приймально-контрольним приладом; структурні особливості об'єкту, який охороняється; фактори вартості та надійності.

В процесі розвитку охоронні системи, і СТС зокрема, постійно вдосконалювалися та ускладнювалися. Це пов'язано з конкурентною боротьбою, суперниками були все більш кваліфіковані, треновані та оснащені щоразу досконалішими технічними засобами порушники. Перші, механічні СТС вже на початку 20-го століття були витіснені електромеханічними, а потім електронними системи. При цьому проходила спеціалізація СТС, а кращі рішення ставали типовими [51].

Особливістю периметра території об'єкта є відносно велика довжина та замкнута форма. Тому його недоцільно охоплювати зоною дії одного сповіщувача, хіба такий сповіщувач є настільки складним, що сам визначає місце порушення, наприклад, радіолокаційний сповіщувач [66]. Тому зазвичай необхідною є взаємодія ряду сповіщувачів для забезпечення суцільності захисту (в нашому випадку першої функції захисту – виявлення порушника). Адже рівень захищеності об'єкта завжди оцінюється за ланкою, яка забезпечує мінімальний рівень захисту ("найслабша ланка") [51]. Таким чином, необхідним є системний підхід до побудови охоронних систем, зокрема до СТС, як їх обов'язкового компонента. В такій системі сповіщувачі можуть бути як однотипними (коли умови їх роботи та можливі загрози однакові), так і різнотипними (відповідно до дійсних умов роботи та можливих загроз). Для організації тісної взаємодії компонентів системи обов'язковою умовою є уніфікація каналів зв'язку та інтерфейсів.

Така уніфікація веде до типізації структури СТС. Типова узагальнена структура переважної більшості сучасних СТС представлена на рис. 1.3. Ця структура дозволяє використовувати різнотипні сповіщувачі, які мають не тільки різне виконання, а і різні принципи дії та використовують різні фізичні

явища для виявлення порушника (єдиною вимогою є уніфікація їх вихідного сигналу). Наслідком такої типізації структури СТС є перенесення функції прийняття рішення про порушення від ПКП до сповіщувача. Інакше ПКП повинен би вміти приймати рішення про порушення для великої номенклатури сповіщувачів різного принципу дії.

Рис. 1.3. Узагальнена структура більшості сучасних СТС

Відповідно до такого розподілу функцій між сповіщувачем і ПКП, основна обробка сигналу щодо виявлення порушника зазвичай проводиться в сповіщувачі. Такий сповіщувач представляє собою досить складний пристрій, в структурі якого можуть бути виділені чотири функціональні вузли: (i) власне сенсор – чутливий елемент, який сприймає інформацію про загрозу і перетворює її в електричний сигнал (звичайно за назвою цього чутливого елемента називають сповіщувач в цілому і, при наявності в складі СТС тільки одного типу сповіщувачів, СТС в цілому); (ii) схему обробки вихідного електричного сигналу чутливого елемента, яка звичайно виконує функції підсилення, фільтрування, селекції, виявлення (детектування) наявності загрози і прийняття рішення про видачу тривожного сигналу (зазвичай при перевищенні сигналом сенсора деякого порогового значення); (iii) схему забезпечення нормального функціонування сповіщувача, захисту від несанкціонованого доступу до нього або спроб порушення його функціонування (зазвичай в останніх випадках генерується і передається на ПКП сигнал тривоги); (iv) один або декілька пристроїв, які формують вихідну схему сповіщувача.

Необхідність прийняття рішення привела до ускладнення самого сповіщувача і спрощення ПКП. Основними функціями останнього стали: (і) виявлення навмисних і випадкових коротких замикань та обривів лінії зв'язку; (іі) представлення інформації про стан окремих зон захисту (стан сповіщувачів) – є або немає порушення безпеки (спрацював або не спрацював відповідний сповіщувач); (ііі) встановлення режиму роботи окремих сповіщувачів – постановка під охорону їх зон і зняття з охорони.

На сьогодні як сповіщувачі, так і ПКП, можуть бути виконані на базі наступних класів електронних схем:

1. На базі аналогових компонентів електронних схем. При цьому типова схема виявлення навмисних і випадкових коротких замикань та обривів складається з тригера Шмідта на базі операційного підсилювача, що вимірює напругу в діагоналі мостової схеми, в одне з плеч якої з допомогою двопровідної лінії зв'язку ввімкнено вихід сповіщувача. Уніфікована вихідна схема сповіщувача представляє собою нормально замкнуті або розімкнуті контакти, які змінюють свій стан при виявленні загрози. При відсутності порушень сповіщувач має типовий вихідний опір 2 кОм (допуск ±10%). Спрацювання сповіщувача, коротке замикання, обрив або підключення імітаційних пристроїв змінює вихідний опір сповіщувача, міст розбалансовується, тригер Шмідта вмикає сигнал тривоги та індикує місце порушення на мнемосхемі. Такі СТС на сьогодні масово випускаються багатьма фірмами. Їх перевагами є низька ціна та легкість пошуку несправностей (для цього достатньо тестера), а недоліками (як це буде показано в §3.1) – відносно великі затрати на лінії зв'язку та велика трудомісткість прокладання цих ліній. Крім того, досить добре розроблені методи імітації окремих сповіщувачів, що сприяє зловмисникам і знижує надійність таких СТС. Для підтримання рівня надійності захисту для таких СТС необхідно виконувати заходи щодо додаткового захисту ліній зв'язку від доступу до них порушників, що веде до додаткових затрат.

2. На базі цифрових компонентів електронних схем. При цьому сповіщувач і/або ПКП виконуються з широким використанням цифрових мікросхем малої та середньої степені інтеграції. Однак зазвичай інтерфейс сповіщувач – ПКП залишається аналогічним до описаного в п. 1 інтерфейсу аналогових схем. Перевагою такого рішення є можливість використання комбінацій – аналогові сповіщувачі та цифрові ПКП або навпаки. Цифрові сповіщувачі та ПКП, які на сьогодні випускаються серійно, зазвичай мають ширші функціональні можливості порівняно з аналоговими, однак такі СТС дорожчі.
3. На базі мікроконтролерів. При цьому існують два напрямки розвитку таких СТС – на базі описаного в п. 1 аналогового інтерфейсу та на базі комп'ютерних мереж. Перший варіант на сьогодні є найбільш розповсюдженим – зазвичай ПКП будується на широко вживаному мікроконтролері, який може взаємодіяти з сповіщувачами, побудованими на базі всіх перелічених класів електронних схем. Перевагою такого рішення є універсалізм і широкі функціональні можливості – простота постановки окремих зон на охорону та зняття з охорони, легкість організації індивідуального рівня захисту за рахунок індивідуальних паролів, простота заміни паролів та можливість використання алгоритмічних паролів, що підвищує рівень захисту, можливий зв'язок з комп'ютерними охоронними системами вищого рівня, автоматичний виклик міліції та ін. Такі СТС легко розширювати та модифікувати, їх ціна відносно невисока, особливо при використанні широко вживаних сповіщувачів. Однак такі СТС повністю повторюють недоліки СТС на базі аналогових компонентів електронних схем, вказані в п. 1 – невисока стійкість таких СТС до імітації сповіщувачів і відносно великі затрати на лінії зв'язку та велика трудомісткість прокладання цих ліній.
4. На базі мікроконтролерів та комп'ютерних мереж. Такі СТС зазвичай називають комп'ютерними. Вони появилися на ринку відносно недавно і починають його завойовувати. Їх перевагами є відносно невисокі затрати на

лінії зв'язку та широкі функціональні можливості, які відповідають переліченим в п. 3. Крім того, імітувати окремі сповіщувачі (або їх групи) таких систем значно складніше, методи зламу таких мереж недостатньо розроблені. Однак такі СТС використовують спеціалізовані сповіщувачі, ціна яких (порівняно з аналогічними за призначенням і принципом дії аналоговими сповіщувачами) в декілька разів вища. Тому, незважаючи на істотні переваги, такі системи в Україні не знайшли широкого застосування – на думку споживачів, зменшені затрати на лінії зв'язку не компенсують високу ціну СТС.

5. На базі безпровідних технологій. Такі СТС появилися на ринку останнім часом і характеризуються відсутністю затрат на лінії зв'язку. Така суттєва перевага повинна була би сприяти швидкому поширенню цих СТС за рахунок витіснення інших класів. Однак такої тенденції в Україні не спостерігається. Імовірно це пов'язано із значно вищою ціною таких СТС та виявленими недоліками таких систем – малим радіусом дії надійного безпровідного зв'язку в умовах залізобетонних будівель, значним затуханням радіохвиль в несучих стінах і перекриттях цегляних будівель, екрануючим впливом залізних дверей та обладнання. Крім того, деякі з наявних на ринку систем відносно легко вивести з робочого режиму (створити хибну тривогу) дистанційно з допомогою розміщеного на невеликій віддалі передавача радіосигналів в робочому діапазоні сповіщувачів, який введе в насичення його радіоприймальний тракт. Часте повторення хибних тривог веде до відключення СТС як несправної, хоча в дійсності маємо справу з цілеспрямованою атакою зловмисника.

Відповідно до розглянутої бази побудови ПКП і СТС в цілому використовуються різні топології мереж. Зіркова (радіальна, променева) топологія (рис. 1.4) характеризується простотою виконання та технічного обслуговування (підключення, налаштування, ремонт), надійністю та живучістю всієї системи, оскільки вихід з ладу однієї лінії з'єднання із сповіщувачем С не впливає на роботу інших сповіщувачів та приймально-

контрольного приладу. Однак з точки зору затрат на лінії зв'язку такий тип є найменш економічним. Для зменшення затрат на лінії зв'язку при використанні аналогових ліній (пп. 1, 2, 3 приведеної вище класифікації) часто до однієї лінії підключають декілька виходів сповіщувачів. При цьому нормально замкнуті контакти з'єднують послідовно, а нормально розімкнуті – паралельно (тоді зміна стану будь-якого контакту буде розпізнана приймально-контрольним приладом як сигнал тривоги). Однак в цьому випадку економія досягається за рахунок втрати інформативності СТС – приймально-контрольний прилад не зможе розпізнати, який сповіщувач прореагував на порушення і подав сигнал тривоги.

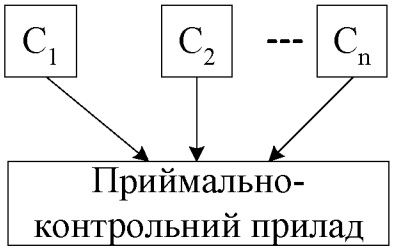


Рис. 1.4. Зіркова топологія СТС

Збільшити кількість каналів та ущільнити інформацію, що передається, дозволяють периферійні блоки ПБ (так звані концентратори), котрі беруть на себе функцію попередньої обробки сигналів від сповіщувачів (рис. 1.5). Таке рішення вимагає наявності відповідних входів приймально-контрольного приладу. При його використанні інформативність СТС не зменшується через те, що зв'язок між ПБ і приймально-контрольним приладом передбачає передачу адреси сповіщувача, який послав тривожне повідомлення. Однак в такому випадку, через відносно високу ціну ПБ, споживач не отримує практично ніякої економії. Особливістю такої топології є теоретично нижча надійність, порівняно із чисто зірковою топологією, через збільшення кількості компонентів СТС і через те, що вихід з ладу ПБ унеможливлює роботу під'єднаних до нього сповіщувачів (проте не впливає на загальну роботу інших блоків і сповіщувачів) [51]. Однак надійність функціонування СТС, на відміну від більшості інших класів систем, визначається, по-перше, імовірністю

невиявлення порушника та, по-друге, імовірністю “виявлення” неіснуючого порушника (хибна тривога). Вказані імовірності на декілька порядків більші, ніж імовірності виходу з ладу електронних та механічних компонентів СТС. Тому оцінка надійності СТС через напрацювання на відмову її компонентів не має практичного змісту.

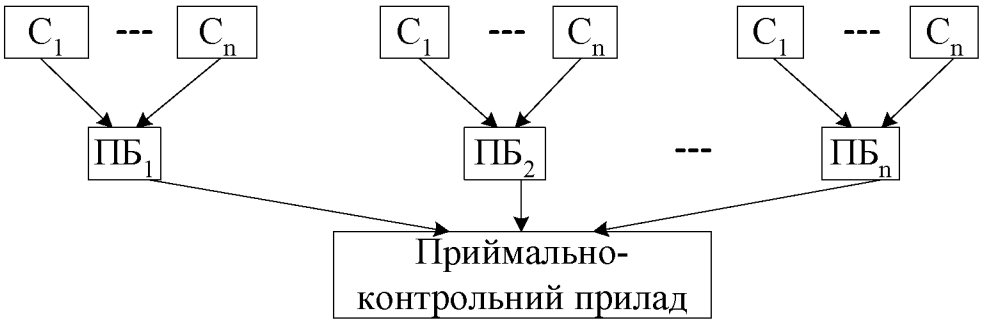


Рис. 1.5. Зіркова топологія з використанням концентраторів

В СТС на базі мікроконтролерів та комп’ютерних мереж найчастіше використовують топології спільна шина (інші назви – шлейфова, магістральна) в двох варіантах – без концентраторів та з концентраторами (рис. 1.6). Така топологія дозволяє суттєво зекономити на лініях зв’язку між сповіщувачами та приймально-контрольним приладом. Щоб не зменшувати інформативності СТС використовують спеціалізовані адресні блоки, які з’єднуються з кожним сповіщувачем. Проте вони дорогі (вартість може перевищувати вартість самих сповіщувачів) і їх використання рентабельне на великих об’єктах, де великі затрати на вартість довгих ліній кабелю для зіркової топології [51]. Спеціалізовані сповіщувачі для топології спільна шина також дорожчі від своїх аналогів з аналоговими лініями зв’язку в 2,5…3 рази. Теоретичне зменшення надійності, викликане ускладненням сповіщувачів і збільшенням кількості компонентів в мережі, як це було показано вище, при розгляді топології, що використовує концентратори, є неістотним. Вихід з ладу основної лінії зв’язку, який унеможливить роботу всіх наступних сповіщувачів та ПБ, слід інтерпретувати не як зниження надійності СТС, а як дію порушника (випадкового або зловмисника). В такому випадку надійна СТС (її приймально-контрольний прилад) швидко (в реальному часі) виявить пошкодження лінії

зв'язку і видасть відповідне повідомлення. Тому пошкодження лінії зв'язку та його наступне швидке виявлення слід інтерпретувати як нормальне функціонування СТС в реальних умовах дій порушників.

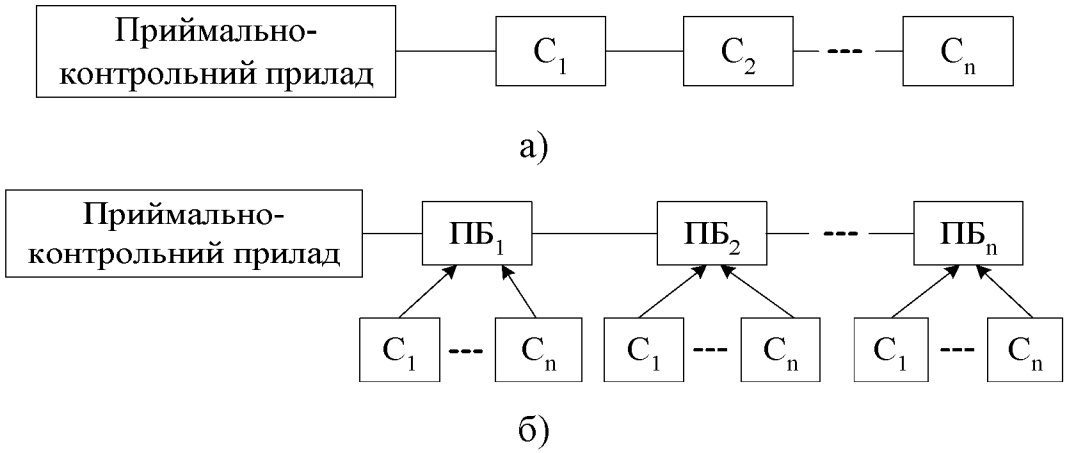


Рис. 1.6. СТС з використанням топології спільна шина:
а – без концентраторів, б – з концентраторами.

На базі мікроконтролерів побудовані також найсучасніші універсальні ПКП, які забезпечують роботу сповіщувачів з використанням всіх описаних вище топологій – зірка, зірка з концентраторами, спільна шина (як з концентраторами, так і без них). При цьому, для складних об'єктів великої площі можлива побудова ієрархічних мереж (рис. 1.7). Такі ПКП можуть взаємодіяти з іншими ПКП по телефонній лінії, радіоканалах, спеціальних лініях зв'язку тощо [29, 30]. Однак, через високу ціну, вони виправдані тільки для об'єктів, що використовують багато сповіщувачів різного принципу дії, які побудовані на базі різних класів електронних схем.

Наприклад, інтегрована система охорони "Оріон", представлена в додатку А, може включати підмножину СТС, кожна з яких керується ПКП типу С2000-КДЛ. Такий ПКП взаємодіє з охоронною системою через інтерфейс RS-485, тобто ця СТС, по суті, є комп'ютеризованою СТС (КСТС). ПКП дозволяє підключення до 127 спровіщувачів через двохпровідну адресну лінію довжиною до 800 м. При цьому використовується інтерфейс ДПЛС_v2.xx, який забезпечує живлення сповіщувачів через мережу [47]. Хоча сповіщувачі системи "Оріон" є відносно дешевими, сам ПКП досить дорогий. Крім того,

інтерфейс ДПЛС_v2.xx, на якому базуються СТС, що входять в склад системи “Оріон”, не забезпечує захист від імітації одного або декількох сповіщувачів. Він не передбачає шифрування повідомлень, тому робота сповіщувачів може імітуватися досить простим мікроконтролерним пристроєм, який забезпечує формування імпульсів відповіді сповіщувача при відсутності спрацювань.

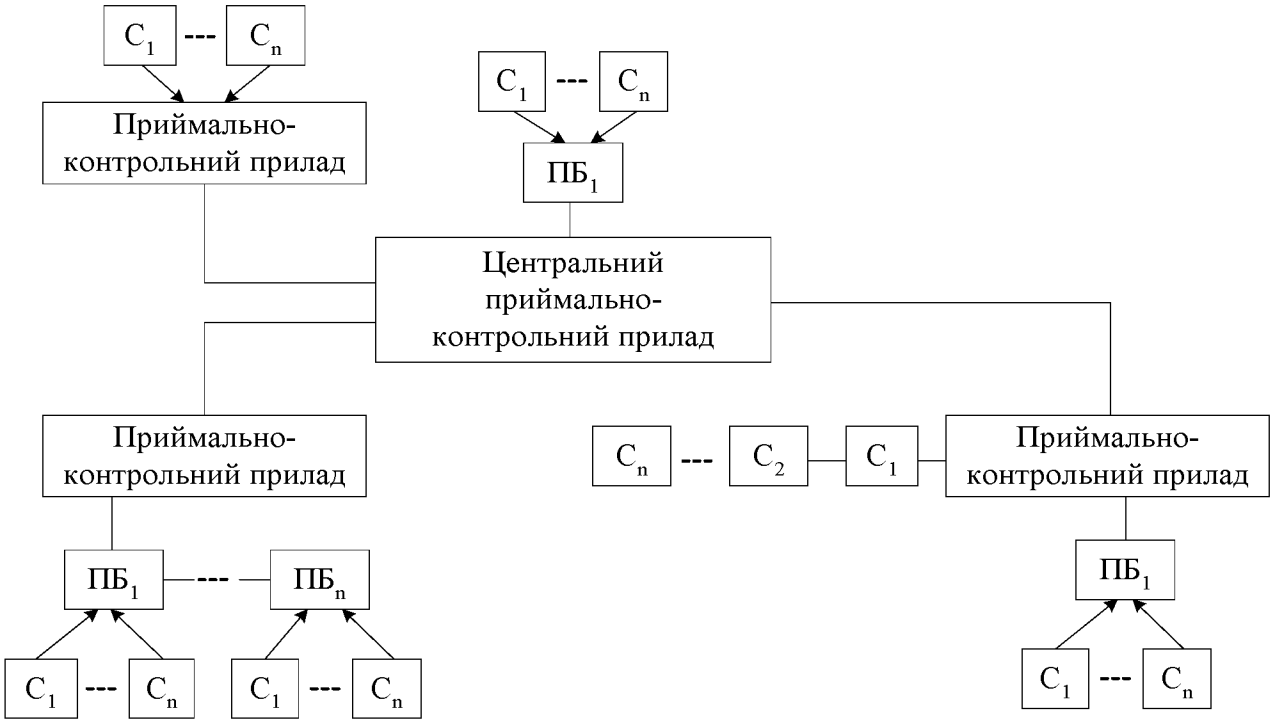


Рис. 1.7. Узагальнена структура дистрибутивної СТС

Як видно з викладеного, різні класи СТС мають різні переваги і недоліки, тому при проектуванні СТС слід було би провести аналіз відповідності властивостей конкретних реалізацій СТС тим задачам, які вона має виконувати (кількість і параметри охоронних зон, типи загроз і порушників для кожної охоронної зони, умови експлуатації, несприятливі зовнішні та внутрішні фактори), а також обмеження, які накладаються на проект (в першу чергу фінансові). Однак така розробка СТС, особливо не підтримана комп’ютерними засобами, вимагає надто багато часу, високої кваліфікації розробників і значних фінансових затрат. Тому більшість фірм, що встановлюють СТС, мають доволі вузьку спеціалізацію. Як показали опитування, вони використовують від двох до п’яти типових універсальних рішень в рамках продукції двох-трьох фірм, які

спеціалізуються в області СТС (з декількох сотень, представлених на ринку). Таким чином, такі фірми займаються не проектуванням СТС, а їх компонуванням, тобто адаптацією типових проектів фірм-виробників до елементарних вимог підохоронного об'єкта, таких як, наприклад, узгодження кількості сповіщувачів та їх радіусу дії з довжиною периметра підохоронної території. При цьому не іде жодної мови про оптимізацію СТС, доцільність використання якихось компонентів, що не входять у традиційний набір, який використовує фірма. Такий стан в області СТС створився останнім часом, коли загроза тероризму та криміналітету значно зросла, що привело до масового встановлення СТС.

Слід відзначити, що досить висока якість продукції фірм-виробників обладнання СТС та намагання запропонувати повний спектр обладнання для побудови типових СТС, веде до того, що в переважній більшості випадків скомпоновані СТС в першому наближенні виконують свої функції. Однак часто вони далеко не оптимальні з точки зору затрат на їх створення та експлуатацію, а також успішно протидіють тільки випадковим або слабо підготовленим та погано оснащеним порушникам.

Виходом з цієї ситуації може бути підтримка процесу розробки СТС (включаючи її оптимізацію за заданими параметрами) за допомогою комп'ютерної системи, що дозволить різко скоротити сам процес розробки та зменшити затрати на нього. Огляд доступних джерел інформації показав, що системи, завданням яких є підтримка процесу розробки СТС, існують. Тому наступний параграф присвятчено аналізу таких систем.

Отже, в § 1.1 розглянуто місце СТС в охоронних системах, їх основні завдання, база виконання, узагальнена структура та можливі варіанти їх реалізації, що дозволило виявити протиріччя між якістю, функціональними можливостями та затратами на закупівлю, встановлення та експлуатацію, які виникають при виборі структури СТС і складу її компонентів. Це протиріччя поглиблюється дуже широкою номенклатурою компонентів, які випускаються сотнями фірм.

1.2 Аналіз засобів розробки і оцінювання якості систем тривожної сигналізації

Процес розробки ефективної СТС включає такі етапи [20]: (i) визначення цілей СТС, що вимагає збору інформації про характеристики об'єкту, визначення характеру загроз на основі характеристик потенційних порушників та можливих цілей нападу; (ii) проектування СТС, що включає оптимізацію розміщення персоналу, розробку процедур його дій при загрозах порушень і самих порушеннях, а також оптимізацію складу та розміщення засобів інженерного і технічного захисту для забезпечення основних функцій СТС, до котрих відносять виявлення (визначення ознак вторгнення сповіщувачем, отримання та оцінювання сигналу тривоги від нього), часову затримку порушника на шляху доступу до цілі та реагування на порушення (припинення дій порушника, склад сил реагування, зв'язок з ними); (iii) аналіз та оцінка проекту СТС (комп'ютерні моделі, діаграми послідовності дій порушника, оцінка ризиків); (iv) затвердження кінцевого проекту СТС або, при виявленні слабких місць – повторне проектування та аналіз СТС.

Серед відомих рішень, що частково забезпечують вказані вище елементи процесу розробки СТС, є системи: ASSESS, ATLAS, SAVI, EASI, VISA, ADKBS, JCATS, TAM, FOF, SAFE, SNAP (США) [20, 112], CLASP (Англія) [122], "Вега-2" (Росія) [45].

Аналітична система для оцінки засобів охорони та безпеки „ASSESS" визначає шляхи порушника та оцінює ефективність СТС відносно пограбування та саботажу сторонніми і внутрішніми особами, а також внутрішніми особами, що співпрацюють із сторонніми. Обчислює імовірність виявлення не насильного втручання внутрішнього порушника, імовірність сповільнення втручання стороннього порушника та імовірність затримки порушника [112, 20]. Таким чином, вона забезпечує тільки третій етап розробки.

Система часового аналізу ворога „ATLAS" визначає найбільш уразливі місця для атаки стороннім порушником та насильного втручання внутрішнього порушника. Мінімізує імовірність порушення та час нейтралізації порушника

після його виявлення і визначає критичні для ефективності всієї охоронної системи елементи [112]. Ця система теж забезпечує тільки третій етап розробки.

Система методичного аналізу вразливості проникнення „SAVI" проводить аналіз всіх можливих шляхів атаки порушника на відповідний об'єкт та визначає до десяти оптимальних. Складається з модуля для побудови діаграми послідовності дій порушника та модуля аналізу об'єкта [20, 112]. І ця система торкається третього етапу розробки.

Модель оцінки ефективності охоронної системи „EASI" дозволяє кількісно оцінити ефективність довільної охоронної системи щодо протидії заданому порушнику, який використовує визначений шлях та сценарій нападу. Аналітик вводить дані, що описують виявлення, часову затримку та реакцію системи на шляху порушника, після чого модель видає імовірності зупинки дій зловмисника [20, 112]. Система автоматизованої оцінки „SAFE" здійснює вибір найбільш вразливих маршрутів, використовуючи дані про об'єкт, параметри елементів захисту, шляхи порушника та сил реагування. Ця система може використовуватись разом з моделлю EASI для оцінки ефективності охоронної системи [20]. Ці системи теж стосуються тільки третього етапу розробки.

Методика аналізу вразливості СТС „VISA" - це технологія, що використовує експертні дані та всі наявні дані про ефективність СТС (імовірність виявлення і т.д.). Шлях проникнення порушника розбивається на окремі етапи, кожен з яких оцінюється за ймовірністю виявлення, оцінкою часової затримки та можливістю припинення його дій, потім оцінюється вся система загалом [112]. Таким чином, система „VISA" теж забезпечує тільки третій етап розробки.

Система комп'ютеризованого аналізу безпеки периметрів в'язниць "CLASP" використовує узагальнену математичну модель СТС периметру території, яка дозволяє оцінити шість критеріїв її якості (імовірність виявлення, імовірність вламування, гірший випадок вламування, діапазон місячних фальшивих тривог, річні витрати, загальні витрати), які базуються на введеній

інформації про параметри СТС, оцінює імовірності втеч та враховує можливі зміни форми периметру [122, 123, 124, 125]. Таким чином, система “CLASP”, хоча і є однією з найдосконаліших, також забезпечує тільки третій етап розробки.

Досить поширеними є системи оцінювання дій порушника та можливої протидії охорони. Зокрема, система, що використовує базу знань часових затримок, „ADKBS” призначена для оцінювання затримок (часу спрацювання) елементів захисту (бар’єрів) під час проникнення порушника, що використовує відповідні методи атаки [112]. Система оцінювання конфлікту та тактичної симуляції „JCATS” визначає ефективність дій сил реагування відповідно до можливого вторгнення порушників. Симулює всі можливі елементи середовища бою (територія об’єкту, тип охорони, автомобілі, зброя, і т.д.) [112]. Методологія поверхневого оцінювання (Tabletop Assesment Methodology, TAM) використовується для оцінювання ефективності охоронної системи та потенціалу придушення дій порушника [112]. Методика аналізу елементів фізичної протидії (Force-on-force, FOF) використовується для симуляції протидії злочинцям [112]. Процедура аналізу охоронної системи „SNAP” використовує апарат мережевого моделювання, де аналітик моделює об’єкт, сили захисту та сили порушника [20]. Всі перелічені системи дозволяють поглиблено аналізувати окремі аспекти функціонування охоронних систем в цілому і СТС зокрема. Однак вони знову стосуються тільки третього етапу розробки.

Російська фірма "Елерон" [75] розробила методичний апарат і спеціалізовану комп'ютерну програму "Вега-2" [45], в основу котрої покладено математичне та імітаційно-ігрове моделювання. В системі реалізовано аналітичний та імітаційний підходи до оцінювання ефективності охоронної системи, що визначається імовірністю припинення дій порушника. Але вона використовує віртуальну (ігрову) охоронну систему, тому комп'ютерну програму "Вега-2" можна використати тільки на першому етапі розробки.

В роботах [35, 36] запропоновано підходи до розробки СТС периметру

території, де використовуються наступні її тактико-технічні характеристики: імовірність виявлення, частота помилкових спрацьовувань, вразливість системи, універсальність і гнучкість засобу виявлення, надійність, довговічність, простота монтажу і експлуатації, вартість "погонного метра" зони безпеки, замаскованість (візуальна і технічна) засобів виявлення. Однак ці роботи мають описовий характер, є, в основному, оглядом відомих рішень в області СТС та описом їх компонентів. Описані підходи до розробки СТС периметру території виражають бажання розробника, а не пропонують конкретні рішення. Тому основні ідеї щодо вимог до процесу обробки слід використати в комп'ютерній системі підтримки процесу розробки СТС.

На основі проведеного аналізу можна виділити основні та найважливіші характеристики, котрі відіграють основну роль при оцінюванні ефективності охоронних систем: імовірність виявлення порушника, імовірність хибної тривоги, збільшення часу проникнення порушника (його затримка) та імовірність нейтралізації порушника. Виходячи з проведеного в §1.1 розподілу функцій між охоронною системою та СТС (див. рис. 1.2), можна вважати, що найважливішими характеристиками СТС будуть імовірність виявлення порушника та імовірність хибної тривоги.

Як це було відзначено в §1.1, велика різноманітність компонентів СТС на ринку дозволяє запропонувати безліч варіантів їх виконання. Тому оцінювання всіх цих варіантів з допомогою розглянутих вище систем вимагає абсолютно неприйнятного часу і не має ніякої практичної цінності через те, що при такому порядку розробки цілком не враховується один з найважливіших для споживача параметрів СТС – її вартість. Таким чином, з аналізу функцій відомих систем можна зробити наступні висновки: (і) ні одна з них не забезпечує підтримку всіх етапів розробки; (іі) ні одна з них не забезпечує підтримку другого етапу – розробки варіантів СТС. Вони призначені та придатні тільки для кінцевого оцінювання обмеженого числа кращих варіантів СТС, а не для генерування множини можливих варіантів, їх попереднього відбору та оптимізації. При цьому під час оптимізації обов'язково необхідно враховувати не тільки

функціональні параметри пропонованих СТС, а і їх вартісні параметри, як одні з найважливіших для користувача. Тому в наступному параграфі слід розглянути відомі методи проектування систем різного призначення.

Таким чином, в §1.2 проаналізовано основні етапи розробки СТС і особливості відомих засобів оцінювання ефективності СТС, що дозволило виділити основні характеристики СТС. Показано, що відомі системи оцінювання ефективності охоронних систем підтримують тільки окремі етапи розробки СТС і не підтримують процедури розробки варіантів СТС та їх оптимізації.

1.3. Методи оцінювання функціонально-вартісних характеристик систем тривожної сигналізації

Кількість охоронних систем та їхні функціональні можливості на сьогодні зростають дуже швидко. Сучасні високоефективні охоронні системи характеризуються [113, 115, 116]: (i) комплексністю – можливістю протистояти різним видам порушників, зокрема, добре навченим і оснащеним; (ii) багаторівневим захистом, який змушує порушника послідовно прориватися через окремі рубежі, що дозволяють краще його класифікувати, обґрунтовано вибрати метод його нейтралізації, зібрати відповідні сили нейтралізації; (iii) використанням методів штучного інтелекту для захисту від ще невідомих методів дії порушника; (iv) захистом від несанкціонованого доступу для захисту не тільки від зовнішніх, але і від внутрішніх порушників; (v) прихованою дією, для того, щоб порушник не міг наперед розробити методи подавлення охоронної системи. Очевидно, що це веде до росту ціни окремих реалізацій охоронних систем і СТС, як їх невід'ємної та важливої складової, тому актуальною задачею є оцінювання СТС за їх техніко-економічними показниками (функціонально-вартісними характеристиками).

Аналізуючи охоронні системи, слід врахувати ряд їх особливостей [59]: (i) наявність конфлікту інтересів "охорона-зловмисник", що відрізняє їх від інших людино-машинних систем; (ii) апріорна невизначеність вхідних даних для

проектування охоронних систем та СТС, як їх складової частини, що включає перелік загроз, модель порушника, сценарії розвитку конфліктної ситуації тощо (це означає, що такі системи слід віднести до слабоформалізованих); (iii) випадковий характер часових параметрів, що включає час руху порушника та охорони, час подолання порушником окремих рубежів охорони, час спрацювання сил реагування і т.д. (iv) трудомісткість проведення реального експерименту, що визначає доцільність використання математичного моделювання для аналізу ефективності охоронних систем. Виходячи з перелічених особливостей, в [20, 59] були запропоновані методи проектування охоронних систем, включаючи СТС. Найповніше вони викладені в [59], тому їх аналіз ведемо згідно [59]. Основою для запропонованих методів був підхід, який передбачає створення моделей.

Модель оцінювання ефективності СТС повинна базуватись на її структурі, цілях функціонування, критеріях ефективності та інструменті їх оцінювання, де під критеріями ефективності розуміють можливість СТС протистояти діям зловмисника при обмежених ресурсах на її створення. Зазвичай використовують критерії [59]: (i) типу “ефект-затрати” (економічна ефективність); (ii) елімінуючі (виключаючі) критерії, що дозволяють оцінити якість СТС по заданих показниках та виключити ті варіанти, що не задовільняють заданим обмеженням (наприклад, умови експлуатації); (iii) методи багатокритеріальної оптимізації; (iv) зважені критерії (наприклад, метод лінійної згортки часткових показників).

Згідно з критерієм “ефект-затрати” елементи охоронної системи характеризуються цільовими функціями, кожна з яких описується набором показників: виявлення (імовірність виявлення порушника, рівень хибних тривог, імовірність відмови сповіщувача, час передачі інформації на пульт охорони), затримка (час подолання фізичних бар’єрів, способи подолання фізичних бар’єрів, набір засобів для подолання бар’єрів, ефективність сил фізичного реагування), реагування (час оцінювання ситуації оператором пульта охорони, час збору сил охорони, час розгортання сил охорони), ресурсоємність

(затрати на придбання компонентів, монтаж та налагодження системи, її поточну експлуатацію, ремонт та реновацію) [20, 59]. На основі оцінювання таких показників можна зробити висновок про ефективність всієї охоронної системи та про ефективність СТС як її складової частини. Однак ми знову маємо справу з аналізом існуючих СТС, а не генеруванням оптимізованих. При цьому критерій економічної ефективності тільки декларується, пропонуються методи оцінювання такого критерію, однак не досліджується взаємодія з іншими критеріями при розробці СТС.

Згідно [59] оцінювання ефективності СТС можна здійснити на базі наступних методів аналізу: детерміністичного підходу, методами багатокритеріальної оптимізації, логіко-імовірнісним та імітаційним моделюванням.

Детерміністичний підхід [59] передбачає розподіл об'єктів по категоріях, для котрих необхідно спроектувати СТС в залежності від їх важливості / потенційної загрози, можливої і / або допустимої соціально-економічної шкоди від прогнозованих загроз, типу об'єкту та ін. Кожній категорії встановлюються диференційні вимоги щодо безпеки, таким чином рівень захисту буде відповідати значущості об'єкта. Однак для визначення стану СТС проводиться експертна оцінка, як засіб обробки слабо структурованих даних. Такий підхід не годиться для побудови комп'ютерних систем підтримки розробки СТС. Він придатний скоріше для першого етапу розробки СТС (див. §1.2).

Методи багатокритеріальної оптимізації основані на агрегуванні інформації про об'єкт, що описується множиною критеріїв, для його комплексного оцінювання [79]. Серед них виділяють: (і) методи лексикографічного впорядкування, основані на домінуванні критеріїв, які ранжуються по важливості, вибір оптимального об'єкта здійснюється по головному критерію, а на всі інші критерії накладаються обмеження; (іі) ітераційні методи преференційного вибору, основані на ітераційних процедурах вибору найбільш преференційного об'єкта; (ііі) аксіоматичний підхід з використанням теорії корисності, що передбачає визначення властивостей неявної функції преференцій, котрою оперує особа, що приймає рішення, при

виборі оптимального варіанту [59, 60].

Наприклад, один із ітераційних методів – метод “зміщеного ідеалу” передбачає: (i) моделювання двох варіантів охорони об’єкту – “умовно найкращий” (ідеальний) з максимальними значеннями корисних критеріїв та “найгірший ” з мінімальними значеннями корисних критеріїв; (ii) визначення вектора переваг особою, котра приймає рішення; (iii) ранжування кожного варіанту СТС шляхом порівняння його із ідеальним варіантом, обчислюючи “відстань” (метрику) до нього; (iv) найменш преференційний варіант викидається з розгляду, після чого процедура повторюється, поки не залишиться один – найкращий [60, 59]. Такий метод виходить з припущення, що існує один варіант, найкращий за всіма критеріями. Однак з точки зору розробки СТС така гіпотеза не доказана. Мало того, така гіпотеза видається хибною, найкраща за характеристиками якості СТС, очевидно, буде далеко не найдешевшою. Адже критерії якості та ресурсоємності за своєю природою протилежні.

Логіко-імовірнісні методи дозволяють отримати кількісну оцінку ризику, як міри небезпеки. Вони використовують ступінь ризику – як імовірність невиконання СТС своїх цільових функцій та рівень безпеки. Процедура аналізу складається з наступних кроків: (i) створення сценарію розвитку вторгнення (граф - “дерево”), що являє собою логіко-імовірнісну модель функціонування СТС; (ii) за допомогою функції безпеки системи описується аналітичний граф, що дозволяє визначити найкоротший шлях проникнення; (iii) за допомогою логіко-імовірнісних перетворень функція небезпеки системи приводиться до однієї з канонічних форм та замінюється імовірнісною функцією [59].

Імітаційне моделювання, як один із основних методів, що може бути використаний для оцінювання ймовірностей виявлення порушника, застосовують для моделювання можливості спрацювання СТС, часу руху сил реагування, часу руху порушника та інших випадкових процесів. Ефективність СТС визначається статистично, як відношення кількості захоплень порушника до загальної кількості випробувань [59].

Однак описані реалізації логіко-імовірнісних методів та методу імітаційного моделювання також відносяться до оцінки існуючих СТС і не передбачають прив'язки критеріїв якості до критерію ресурсоємності.

Зважені критерії [59] представляють собою спеціально сконструйовані залежності, які дозволяють надати процесу оптимізації бажаного напряму. Однак відома методологія подійно-часового аналізу охоронних систем, розроблена спеціалістами групи компаній "ИСТА" [24], яка реалізує цей метод, використовує методику оцінки ефективності СТС, що базується на зваженому критерії. Останній включає статистику загроз [81] та вартісну оцінку можливої завданої шкоди [1]. Така методологія також не пов'язує оцінку ефективності СТС із затратами на її створення.

До зважених критеріїв відноситься також метод лінійної згортки показників. Хоча він не відноситься до методів багатокритеріальної оптимізації (фактично він перетворює багатокритеріальну оптимізацію в однокритеріальну), проте широко застосовується через свою простоту. Слід зазначити, що для правильної роботи даного методу, показники, котрі оптимізуються, повинні мати однакову «фізичну природу» [59, 60]. Це означає, що коректно пов'язати взаємно-протилежні критерії, такі як якість (функціональні можливості) та ресурсоємність (затрати на закупівлю, встановлення, відлагодження, експлуатацію, ремонт та реновацію), довільної системи теоретично неможливо. Хоча на практиці такий підхід може створити можливість вирішення (хоча би попереднього) ряду задач. Розвитком і частковим випадком методу лінійної згортки слід вважати метод штрафних функцій [106, 127]. Його особливістю є накладання штрафів на проект СТС який володіє тими чи іншими недоліками. При цьому, значення штрафів усіх недоліків мають однакову шкалу та іноді й фізичну природу (або безрозмірні). Це дозволяє їх додавати з метою отримання сумарної штрафної функції проекту СТС. Штрафну функцію СТС можна записати, наприклад, як $F = P_1 Q + P_2 R + C$, де F - значення штрафної функції, P_1 - ціна хибної тривоги (в грошових одиницях) - виїзду служби охорони або підтримки , Q - ймовірність хибної

тривоги, P_2 - вартість підохоронного об'єкта, R - ймовірність невиявлення загрози, C - вартість придбання і монтажу СТС. В цьому прикладі усі штрафи записуються в кількості грошових одиниць. Оскільки цей метод є розвитком методу лінійної згортки, недоліки останнього притаманні також і йому.

Дещо інший підхід до оцінювання СТС запропоновано в [34]. Він базується на експертній оцінці відносних потенційних можливостей типів і видів засобів виявлення порушника, запропонована відповідна модель порушника, що характеризується 12 параметрами, та вказано відповідний відносний рівень здатності виявлення даного порушника відповідним засобом. Як класифікаційний критерій, вибрано тип взаємодії порушника з засобом виявлення, тобто яким способом об'єкт виявлення викликає у засобу виявлення відповідну зміну фізичної величини, котра ним реєструється. Цей підхід є оригінальним, хоча тісно пов'язаним з підходом, запропонованим в [50]. Ні в [34], ні в [50] не вирішується задача оптимізації функціонально-вартісних характеристик, але такий підхід можна використати у відповідній комп'ютерній системі підтримки процесу розробки СТС.

Таким чином, розглянуті вище відомі методи оцінки СТС в основному оцінюють СТС згідно критеріїв якості, практично не звертаючи увагу на критерій ресурсоємності. Тому вони не можуть бути основою для комп'ютерної системи підтримки процесу розробки СТС (включаючи її оптимізацію за заданими параметрами). Очевидно, ці методи послужили основою для створення розглянутих в §1.2 систем оцінки якості СТС. Але, як сказано в §1.2, ці системи придатні тільки для кінцевого оцінювання обмеженого числа кращих варіантів СТС, а не для генерування множини можливих варіантів, їх попереднього відбору та оптимізації. Тому необхідно розробити шляхи вдосконалення СТС, які передбачають саме підтримку другого етапу розробки СТС (згідно переліку етапів, представленого в §1.2). Після генерування множини можливих варіантів її можна обробляти різними методами оптимізації, зокрема, методами штрафних функцій. Отриманий результат еквівалентний тому, що дав би метод штрафних функцій в чистому вигляді.

Отже, в § 1.3 розглянуто методи оцінювання функціонально-вартісних (техніко-економічних) характеристик СТС. Показано, що відомі методи спрямовані на оцінку якості (функціональних характеристик) і не приділяють уваги аналізу вартісних характеристик, які є дуже важливими для замовника.

1.4. Шляхи вдосконалення комп'ютеризованих систем тривожної сигналізації

Проектувальнику СТС не важко застосувати певні набори компонентів СТС відповідно до їх функціонального призначення. Проте, із збільшенням кількості компонентів, важко оцінити можливість виконання задачі для даної зони периметру території всіма наявними на ринку компонентами, врахувати особливості кожного з них, оцінити ресурсоємність кожного варіанту, порівняти всі характеристики (або хоча би найважливіші з них) з іншими варіантами (якщо не всіма, то хоча би з кращими) та прийняти обґрунтоване рішення про те, який варіант для даного замовника найкращий. Особливо це важко зробити для багатозонного об'єкта охорони, ще і при використанні багатошарового захисту, де необхідно враховувати взаємодію різних видів захисту і засобів, які їх реалізують [51]. Тому, як це вже було зазначено в §1.1, зазвичай використовують вже готові шаблонні рішення, котрі не завжди оптимальні з точки зору функціонально-вартісних характеристик, оскільки не враховують особливості кожної із зон периметру території.

Однієї універсальної системи, яка була би оптимальною для всіх видів загроз, кліматичних і географічних умов та всіх умов експлуатації, не існує. Вибір ефективного варіанту СТС периметру підохоронної території залежить від великої кількості чинників – кліматичних і сезонних умов, моделі потенційного порушника, наявності пасивної огорожі, кількості та видів розривів в ній (автомобільні проїзди, ворота, хвіртки), обслуговування СТС, фінансових можливостей замовника та ін. [35].

Таким чином, як це видно з проведених в §§1.1…1.3 аналізів, при оптимізації СТС виникають протиріччя. Їх можна звести до глобального

протиріччя (характерного не тільки для СТС, а для всіх систем) та локального (характерного власне для СТС). Їх можна сформулювати наступним чином:

1. Протиріччя між ресурсоємністю СТС (затратами на її закупівлю, встановлення та експлуатацію) та якістю, тобто імовірністю виявлення порушника та імовірністю хибної тривоги. Це традиційне протиріччя для всіх технічних (і не тільки) систем. Воно не може бути радикально вирішене, його можна тільки перенести на інший рівень. Таке перенесення є технічно та економічно вигідним, власне воно і є суттю технічного прогресу. Методи вирішення такого протиріччя діляться на еволюційні та революційні. Революційний шлях вирішення протиріч, які виникають при рішенні традиційних задач, полягає у створенні нової науково-технічної бази для вирішення цієї задачі. Щодо СТС, революційний шлях можливий за рахунок розвитку штучного інтелекту, мікроелектронних технологій, успіхів біоніки та інших наук. Еволюційний шлях полягає в отриманні оптимальних щодо заданих критеріїв розв'язків задачі. Проведений в §1.2 і §1.3 аналіз показав, що не існує систем розробки СТС, зокрема, комп'ютерних, які би дозволяли оптимізувати їх щодо критеріїв ресурсоємності та якості, іншими словами, за функціонально-вартісними характеристиками. Такий шлях є перспективним, він дозволяє відносно швидко отримати результат – покращення функціонально-вартісних характеристик СТС, які будуть пропонуватися споживачу. Така комп'ютерна система повинна пропонувати свої оптимізовані рішення за прийнятний час, бути відносно дешевою (не вимагати значних ресурсів для свого функціонування) та не вимагати високої кваліфікації ані персоналу фірми, яка встановлює СТС, ані споживача (користувача).
2. Протиріччя між можливістю запропонувати безліч варіантів виконання СТС і малою пропускною здатністю існуючих систем оцінки їх якості. Запропонована в п.1 комп'ютерна система дозволить вирішити і це протиріччя. Вона повинна скоротити множину можливих варіантів виконання СТС до обмеженого числа оптимізованих за функціонально-

вартісними характеристиками варіантів. Таку обмежену множину (бажано до 10 варіантів) цілком можна детально проаналізувати з допомогою систем, розглянутих в §1.2.

Таким чином, необхідність створення комп'ютерної системи, яка могла би підтримувати власне другий етап процесу розробки СТС, є очевидною. Така система має пропонувати обмежений набір оптимізованих за функціонально-вартісними характеристиками СТС для оцінки з допомогою спеціалізованих систем, описаних в §1.2. При цьому слід враховувати неможливість створення однієї, "абсолютно оптимальної" СТС. Тому не слід використовувати метод лінійної згортки показників так, як це описано в §1.3. Його слід використати як один з декількох методів під час оцінки та кінцевого вибору серед альтернативних оптимальних СТС, які пройшли всі інші методи оцінки.

Тому базою для створюваної комп'ютерної системи повинні бути методи багатокритеріальної оптимізації. Однак вони теж не повинні зводити результат своєї роботи до пошуку одного оптимального рішення – в більшості випадків така реалізація процедури оптимізації означає перехід до однокритеріальної оптимізації (в явному або неявному виді). Найдоцільнішим є пошук негірших рішень серед всієї множини можливих СТС. Такий алгоритм відповідає пошуку лівої нижньої границі за Парето [57]. При цьому доцільний окремий пошук лівої нижньої границі для всіх основних функціонально-вартісних характеристик СТС, зокрема, для обґрунтованих в §1.2 імовірності виявлення порушника та імовірності хибної тривоги. Очевидно, оптимізовані за різними критеріями варіанти СТС не обов'язково будуть співпадати. Однак наявність на виході обмеженого числа кращих варіантів СТС не є значним недоліком такого підходу. В подальшому ці варіанти повинні пройти тестування з допомогою систем, описаних в §1.2, при яких частина варіантів буде визнана непридатною або гіршою інших. Кінцевий відбір варіанту СТС, яка буде реалізована, повинен провести замовник. При цьому, як це було вказано вище, доцільно використати метод лінійної згортки показників якості.

Для функціонування методів багатокритеріальної оптимізації необхідна генерація множини всіх можливих варіантів СТС. Така задача може бути вирішена з допомогою методу морфологічного синтезу [44, 49, 57, 72,]. Він полягає в побудові морфологічної таблиці, лінійки якої відповідають функціональним вузлам розроблюваної СТС (відповідно до узагальненої структури СТС, представленої на рис. 1.2), а стовпці – альтернативним варіантам їх виконання. Генерація варіантів СТС полягає в послідовному переборі всіх можливих шляхів проходження таблиці послідовно по лінійках, що відповідає агрегуванню (поєднанню) окремих компонентів у єдину СТС. При цьому допустимі тільки такі шляхи проходження матриці, в яких в одну систему поєднуються сумісні компоненти, тобто, наприклад, підключення сповіщувача з аналоговим виходом до ПКП з відповідним аналоговим входом. Для забезпечення сумісності слід використати додаткові ознаки. При їх неспів-паданні варіант відкидається, як це передбачено згідно елімінуючих критеріїв. Ці критерії слід теж застосовувати, коли отриманий варіант СТС не відповідає заданим обмеженням. Наприклад, якщо допустима робоча температура сповіщувача не охоплює температурний діапазон його дійсної експлуатації.

Реалізація методів багатокритеріальної оптимізації вимагає застосування комбінаторних алгоритмів пошуку оптимальних рішень. Основними серед методів комбінаторної оптимізації є повний перебір, випадковий пошук, метод віток і границь, метод еврістик та еволюційні методи, до яких відносяться генетичні алгоритми та мурашині алгоритми. При створенні пропонованої комп'ютерної системи необхідний відбір найбільш перспективних методів.

Таким чином, метою дисертаційної роботи є створення методів і засобів оптимізації функціонально-вартісних характеристик комп'ютеризованих систем тривожної сигналізації на основі автоматизованого синтезу та відбору кращих рішень, що базуються на використанні генетичного алгоритму.

Для досягнення мети необхідно вирішити наступні завдання:

1. проаналізувати та оцінити функціонально-вартісні характеристики типових СТС з метою виявлення їх основних недоліків і шляхів їх покращення;

2. розробити методи оптимального відбору кращих варіантів серед множини можливих структур СТС;
3. розробити методику застосування запропонованих методів при розробці комп'ютеризованих СТС;
4. розробити автоматизовану систему оптимального відбору кращих варіантів серед множини можливих структур СТС для представлення користувачеві;
5. розробити типові комп'ютернизовані СТС та виконати їх функціонально-вартісний аналіз;
6. вдосконалити функціонально-вартісні характеристики компонентів КСТС для забезпечення кращих функціонально-вартісних характеристик мережі;
7. розробити методи забезпечення захисту зв'язку в запропонованих КСТС від втручання в їх роботу порушників;
8. дослідити зразки впроваджених КСТС на базі відібраних кращих варіантів та проаналізувати їх функціонально-вартісні характеристики.

Таким чином, у § 1.4 сформульовані протиріччя, які виникають при оптимізації СТС. Показано, що вони можуть бути вирішені шляхом створення комп'ютерної системи, яка могла би підтримувати процес розробки СТС та КТС. Розглянуто шляхи створення такої комп'ютерної системи сформульовано мету дисертації та задачі дослідження, які дозволяють її досягнути.

Висновки до розділу 1

1. Розглянуто місце СТС в охоронних системах, їх основні завдання, база виконання, узагальнені структури та можливі варіанти їх реалізації, що дозволило виявити протиріччя між функціональними можливостями та затратами на закупівлю, встановлення та експлуатацію, які виникають при виборі структури СТС і складу її компонентів. Це протиріччя поглиблюється широкою номенклатурою компонентів, які випускаються сотнями фірм.
2. Розглянуто основні етапи процесу розробки СТС, проаналізовано особливості відомих засобів оцінювання ефективності СТС, що дозволило виділити основні характеристики проектованих СТС. Показано, що відомі системи оцінювання ефективності охоронних систем підтримують тільки окремі етапи розробки СТС і не підтримують процедури розробки варіантів СТС та їх оптимізації.
3. Розглянуто методи оцінювання функціонально-вартісних характеристик СТС. Показано, що відомі методи спрямовані на оцінювання якості, тобто функціональних характеристик, і практично не приділяють уваги аналізу вартісних характеристик, які є дуже важливими для замовника.
4. Сформульовано протиріччя, які виникають при оптимізації СТС. Показано, що ці протиріччя можуть бути вирішені шляхом створення комп'ютерної системи, яка могла би підтримувати процес розробки СТС. Розглянуто шляхи створення такої комп'ютерної системи, сформульовано мету дисертації та задачі дослідження, які дозволяють її досягнути.

РОЗДІЛ 2
МЕТОДИ І ЗАСОБИ АВТОМАТИЗОВАНОГО ПРОЕКТУВАННЯ КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ТРИВОЖНОЇ СИГНАЛІЗАЦІЇ

2.1. Постановка задачі оптимізації функціонально-вартісних характеристик систем тривожної сигналізації

Задачі проектування і реінженірингу СТС ускладнюються тим, що кожен компонент таких систем описується великою кількістю різнорідних параметрів, окремі з яких можуть мати імовірнісну природу. Класифікація параметрів на однорідні групи дозволить формалізувати їх обробку та створити методи розв'язання задач проектування. В результаті аналізу фізичної природи параметрів вдалося виділити наступні три групи характеристик: обмежень використання, ефективності та ресурсоємності [9]. До групи характеристик обмежень використання належать ті, котрі обмежують область його використання згідно документів сертифікації, наприклад, область дії компонента (область простору, в межах якої компонент здатен виявити загрозу), діапазон температур, в яких він зберігає працездатність, стійкість до природних, випадкових і цілеспрямованих (створених порушником) завад та пошкоджень, час напрацювання на відмову тощо. До групи характеристик ефективності відносяться ті, що забезпечують виконання компонентом його безпосередніх функцій по виявленню відповідного типу порушника, зокрема, імовірності виявлення порушника та хибного спрацювання. На ці імовірності безпосередній вплив мають характеристики чутливості і селективності давача (захищеності від хибного спрацьовування) [106].

До групи характеристик ресурсоємності відносяться ті, що визначають вартість придбання, монтажу та обслуговування компонентів, наприклад: зручність монтажу компонента, гарантійний термін експлуатації та престиж фірми виробника, від якого часто залежить ціна.

Найпростіша СТС складається з одного компонента виявлення, у цьому випадку ефективність системи дорівнює ефективності цього компонента. Для оцінювання ефективності складніших систем прийнято користуватись підходами на основі теорії імовірності [31], що описані нижче.

Під ефективною СТС слід розуміти систему, всі компоненти якої працездатні та виявляють всі загрози. Нехай $A_{Відмова}$ – подія „Порушник проник на територію у зв'язку з тим, що хоча б один елемент системи непрацездатний і потребує ремонту", $A_{Невиявлення}$ – подія „Порушник проник на територію у зв'язку з тим, що хоча б один із працездатних елементів не виявив загрозу", $A_{Виявлення}$ – подія „Порушник проник на територію, але система його виявила". Результатом цілеспрямованих дій порушника є проникнення на територію, а отже $P(A_{Відмова}) + P(A_{Невиявлення}) + P(A_{Виявлення}) = 1$, звідки імовірність невиявлення порушника системою $R^{sys} = 1 - P(A_{Виявлення})$ можна оцінити як

$$R^{sys} = P(A_{Відмова}) + P(A_{Невиявлення}), \tag{2.1}$$

де $P(A_{Відмова}), P(A_{Невиявлення})$ – імовірності подій $A_{Відмова}$ і $A_{Невиявлення}$.

Ресурсоємність СТС формується ресурсоємністю придбання, монтажу і експлуатації системи. Перші дві характеристики визначаються згідно прейскурантів, тоді як характеристикою ресурсоємності експлуатації СТС є деяка функція від оцінки частоти хибних спрацьовувань Q^{sys} [106]. Наприклад, хибний виїзд служби безпеки на об'єкт тягне за собою додаткові витрати. При цьому, важливо мати на увазі, що ці витрати повинні бути меншими за збитки, завдані порушником.

В загальному випадку об'єкти охорони мають складну геометричну форму, яку можна представити у вигляді графа, кожне ребро якого називатимемо ділянкою. Кожна ділянка характеризується умовами, що обмежують множину придатних для її охорони сповіщувачів [50], а саме: набором загроз, які треба виявляти, і специфікою завад, що виникають на ній (рух дрібних тварин,

близькість джерел електромагнітного чи іншого випромінювання):

$$\underbrace{id_1^1 \;\; id_2^1 \ldots \;\; id_{n_1}^1}_{\text{«ID сповіщувачів» для відслідковування першої загрози в першій ділянці}} \quad \underbrace{id_1^2 \;\; id_2^2 \ldots \;\; id_{n_2}^2}_{\text{«ID сповіщувачів» для відслідковування другої загрози в першій ділянці}} \bullet\bullet\bullet \underbrace{id_1^Z \;\; id_2^Z \ldots \;\; id_{n_Z}^Z}_{\text{«ID сповіщувачів» для відслідковування Z-тої загрози в останній ділянці}}.$$

Довжини кожної ділянки S_i та згадані умови утворюють вхідні дані задачі проектування СТС.

Нехай $N_{i,j}$ – кількість компонентів j -тої моделі, що призначені для охорони i -тої ділянки, тоді проект охоронної системи можна представити матрицею $Z \times n$

$$N = \begin{pmatrix} N_{1,1} & \ldots & N_{1,n} \\ \vdots & \ddots & \vdots \\ N_{Z,1} & \ldots & N_{Z,n} \end{pmatrix}, \tag{2.2}$$

де Z – кількість ділянок периметру території, n – кількість альтернативних варіантів сповіщувачів: $n = \sum_{i=1}^{Z} n_i$, n_i – кількість моделей сповіщувачів, які згідно умов i -тої ділянки придатні для її охорони.

Для обчислення імовірності невиявлення порушника за формулою (2.1) необхідно оцінити значення імовірності подій $A_{Відмова}$ і $A_{Невиявлення}$. Згідно теорії надійності [31], імовірність того, що хоча б один сповіщувач СТС непрацездатний і потребує ремонту, можна оцінити за формулою

$$P(A_{Відмова}) = 1 - F^{sys}, \quad F^{sys} = \prod_{i=1}^{Z} F_i^{zone}, \quad F_i^{zone} = \prod_{j=1}^{n_i} (p_j)^{N_{ij}}, \tag{2.3}$$

де n_i – кількість моделей компонентів, придатних для охорони і-тої ділянки; N_{ij} – кількість компонентів j-тої моделі, що встановлені для охорони і-тої ділянки; p_j – імовірність того, що компонент j-тої моделі перебуває в працездатному стані; F^{sys} – імовірність того, що всі елементи системи у

працездатному стані; F_i^{zone} – імовірність того, що всі елементи, призначені для охорони і-тої ділянки у працездатному стані; Z – кількість ділянок.

Імовірність того, що хоча б один із працездатних елементів не виявить загрозу, по аналогії з [31], обчислюється так:

$$P_{Невиявлення} = \max_{i=1,..,Z} R_i^{zone}, \quad R_i^{zone} = \max_{j=1,..n_i} \begin{cases} r_j, if\, N_{ij} > 0, \\ 0, if\, N_{ij} = 0, \end{cases} \tag{2.4}$$

де r_j – значення імовірності невиявлення загрози компонентом j-тої моделі за умови, що він перебуває у працездатному стані; n_i – кількість моделей компонентів, придатних для охорони і-тої ділянки; R_i^{zone} – імовірність невиявлення загрози працездатними сповіщувачами і-тої ділянки.

Підставляючи (2.3) та (2.4) в (2.1), отримаємо імовірність R^{sys} невиявлення системою порушника

$$R^{sys} = 1 - F^{sys} + \max_{i=1,..,Z} R_i^{zone}. \tag{2.5}$$

Оцінювання ресурсоємності придбання та монтажу системи з використанням комбінації підходів [31] і [86] можна виконати за формулою

$$C^{sys} = \sum_{i=1}^{Z} C_i^{zone}, \quad C_i^{zone} = \sum_{j=1}^{n_i} c_j N_{ij}, \tag{2.6}$$

де c_j – вартість придбання, монтажу та гарантійного обслуговування сповіщувача j-тої моделі; n_i – кількість моделей компонентів, придатних для охорони і-тої ділянки.

Оцінювання ресурсоємності експлуатації СТС за аналогією до теорії масового обслуговування, можна здійснити за формулою середньої імовірності хибних спрацьовувань [22]:

$$Q^{sys} = \frac{1}{Z}\sum_{i=1}^{Z} Q_i^{zone}, \qquad Q_i^{zone} = \frac{\sum_{j=1}^{n_i} N_{ij} q_j}{\sum_{j=1}^{n_i} N_{ij}}, \tag{2.7}$$

де q_j – імовірність хибного спрацювання сповіщувача j-тої моделі; Q_i^{zone} – імовірність хибної тривоги з вини сповіщувача, що знаходиться на і-тій ділянці; N_{ij} – кількість компонентів j-тої моделі, що встановлені для охорони і-тої ділянки; Z – кількість ділянок.

Для оцінювання імовірностей працездатності сповіщувачів p_j, невиявлення порушника r_j та хибної тривоги q_j можна скористатись одним з трьох підходів: (і) статистичні експерименти, (іі) експертне оцінювання або (ііі) моделювання. Межі застосування, переваги і недоліки кожного з цих підходів визначаються доступністю і достовірністю необхідної інформації.

Статистичне оцінювання ймовірностей виконується згідно співвідношення [65]

$$p_j = \frac{N_0 - N_{fail}}{N_0}, \tag{2.8}$$

де N_0 – загальна кількість компонентів в експерименті, N_{fail} – кількість пошкоджених компонентів протягом гарантійного терміну, p_j – статистична оцінка ймовірності безвідмовної роботи компонента j-тої моделі.

Проте, застосування даного підходу для оцінювання значень імовірностей p_j вимагає збору даних про тривалість безвідмовної роботи компонентів, що у зв'язку із витратністю і довготривалістю таких експериментів утруднює його використання.

Статистичний підхід успішно застосовується для оцінювання імовірностей r_j та q_j в [106]. Розглянемо розвинутий там підхід детальніше. Користуючись

умовними ймовірностями того, що компонент не реагує на k-ту загрозу можна обчислити ймовірність невиявлення загрози компонентом j-тої моделі

$$r_j = \sum_{k>0} P\{Y_j^0 \mid Y_j^k\},$$

де $P\{Y_j^0 \mid Y_j^k\}$ – ймовірність того, що компонент j-тої моделі не зреагує на k-ту загрозу.

Ймовірність хибного спрацьовування j-го компонента дорівнює

$$q_j = \sum_{k>0} P\{Y_j^k \mid Y_j^0\}.$$

Самі умовні ймовірності можна оцінити так: $P\{Y_j^0 \mid Y_j^k\} = \frac{m_{k,j}}{M_{k,j}}$, де $m_{k,j}$ – загальна кількість неспрацювань сповіщувача j-тої моделі на k-ту загрозу, $M_{k,j}$ – загальна кількість експериментів з сповіщувачем j-тої моделі на виявлення k-тої загрози.

Частина виробників компонентів надають експертні дані про імовірність виявлення загроз тим чи іншим компонентом. Згідно цих даних, величина імовірності невиявлення загрози r_j переважної їх більшості не перевищує 0.1 [37, 38, 39]. Крім того, слід зауважити, що ні статистичний ні експертний підходи не спроможні пояснити залежність виявлення загроз компонентом від умов навколишнього середовища [34, 95, 100].

В цьому випадку доцільно побудувати математичні моделі, які апроксимують значення імовірностей у залежності від температури, тиску чи інших параметрів навколишнього середовища, розглянутих нижче. Наприклад, для мікрохвильових сповіщувачів імовірності невиявлення загроз r_j і хибних спрацьовувань q_j суттєво залежать від діапазону допустимих температур функціонування компонентів та ширини пучка променів (рис. 2.1), утворюваного передавачем і прийнятого приймачем [117]. Характеристика ширини пучка променів задається діаметром утворюваного ними еліпса у метрах. Характеристики нижніх і верхніх меж діапазону робочих температур

подається фірмами-виробниками у градусах Цельсія.

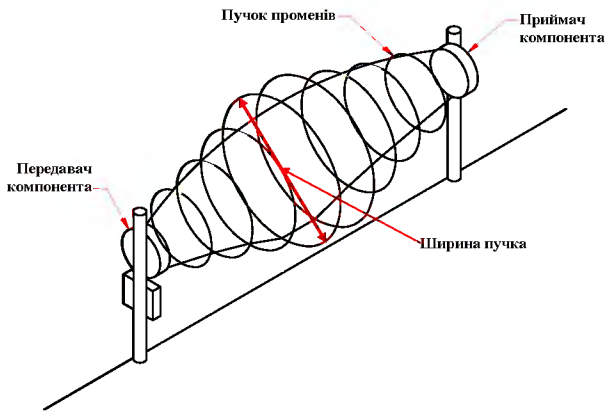


Рис.2.1. Пучок променів між передавачем і приймачем для виявлення загроз

Значення цих характеристик, на відміну від оцінок імовірності виявлення, суттєво відрізняються у компонентах різних виробників і різного принципу дії. Збільшення відстані між передавачем і приймачем неодмінно призведе до розширення пучка променів. Виходячи з цього, дані характеристики можуть бути кращим підгрунтям для оцінювання імовірнісних характеристик сповіщувачів в різних умовах, порівняно із конкретним значенням імовірності виявлення порушника, яке задекларував виробник.

Наближення температури середовища до крайніх меж робочого діапазону температур компонента може призвести до зміни параметрів напівпровідникових радіоелементів, з яких він складається [71], що може спричинити підвищення імовірностей невиявлення загроз і хибних тривог.

З використанням формули статистичного нормування виборок [22] можна прошкалювати значення ширини пучка променів для усіх відомих компонентів відносно їхнього мінімального і максимального значень

$$E_j = \frac{f_j - f_{\min}}{f_{\max} - f_{\min}},$$

де f_j – значення ширини пучка для j-го компонента, $f_{\min}$, $f_{\max}$ – мінімальне і максимальне значення ширини пучка по множині всіх компонентів.

З [117] відомо, що більша ширина пучка призводить до зростання частоти хибних спрацьовувань, те саме стосується і наближення температури середовища до екстремально високої або низької. Звідси можна припустити, що модельна залежність ймовірності хибної тривоги j-го компонента має вигляд

$$q_j(T) = q_{\min} + E_j q_{width} + \mu_q(T) q_{temperature},$$

де $q_{\min}$ – найменша можлива імовірність хибного спрацьовування, можлива в умовах оптимальної температури середовища ($T = T_{opt}$) і найменшої ширини пучка променів ($E_j = 0$); E_j – нормоване значення ширини пучка j-го компонента, q_{width} – складова ймовірності хибного спрацьовування, що може бути досягнута при найбільшій ширині пучка, $\mu_q(T)$ – деяка функція, що описує залежність впливу другої складової ймовірності хибної тривоги ($q_{temperature}$) від температури середовища T.

Для функції ймовірності невиявлення загроз ширший пучок зменшує ймовірність невиявлення. Таким чином дана ймовірність обчислюється за формулою:

$$r_j(T) = r_{\min} + (1 - E_j) r_{width} + \mu_r(T) r_{temperature},$$

де $r_{\min}$ – найменша можлива ймовірність невиявлення загрози, що досягається в умовах оптимальної температури середовища ($T = T_{opt}$) і найбільшої ширини пучка променів ($E_j = 1$); E_j – нормоване значення ширини пучка j-го компонента, r_{width} – складова ймовірності невиявлення загрози, що може бути досягнута при найменшій ширині пучка, $\mu_r(T)$ – деяка функція, що описує залежність впливу другої складової ймовірності невиявлення ($r_{temperature}$) від температури середовища T.

Функції $\mu_q(T)$ та $\mu_r(T)$ у випадку обмеженості експериментальних чи експертних даних можна наблизити залежностями трикутної форми, графіки яких наведені на рис. 2.2. Проте їхній вигляд можна додатково уточнювати

появою нових статистичних спостережень функціонування компонента в тих чи інших температурних умовах або методом експертного оцінювання.

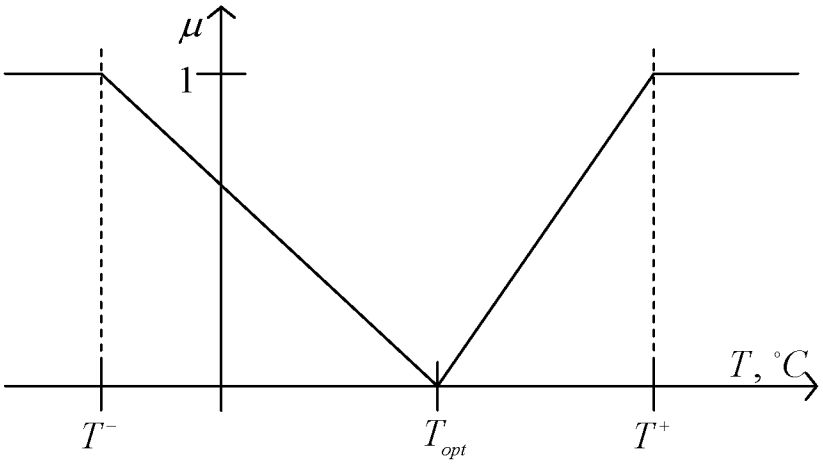


Рис.2.2. Функція для моделювання залежностей ймовірностей невиявлення загроз і хибних спрацьовувань від температури середовища

Користуючись розробленими функціонально-вартісними показниками (2.5), (2.6) та (2.7) задачу їх оптимізації можна представити у наступному виді

$$\left(Q_{sys}(\vec{N}), R_{sys}(\vec{N}), C_{sys}(\vec{N})\right) \xrightarrow{\vec{N}} \min, \tag{2.9}$$

при виконанні умов покриття периметру території областями дії сповіщувачів

$$\begin{cases} L_1 N_{1,1} + L_2 N_{1,2} + \cdots + L_n N_{1,n} \geq S_1, \\ L_1 N_{2,1} + L_2 N_{2,2} + \cdots + L_n N_{2,n} \geq S_2, \\ \quad \vdots \qquad \vdots \qquad \vdots \\ L_1 N_{Z,1} + L_2 N_{Z,2} + \cdots + L_n N_{Z,n} \geq S_Z, \end{cases} \tag{2.10}$$

де $\vec{N}$ – матриця (2.2), що представлена стрічкою

$$\vec{N} = \left(N_{1,1}, N_{1,2} \ldots N_{Z,n}\right) \tag{2.11}$$

S_i – протяжність і-тої ділянки периметру; L_j – область дії компонента j-тої моделі.

Отже, в § 2.1 обґрунтовано вибір характеристик компонентів СТС та їх розподіл на групи обмежень, ефективності та ресурсоємності. Описано підходи для оцінювання імовірностей працездатності, хибної тривоги і невиявлення загрози, а також їхні переваги, недоліки та межі застосування. Виведено

формули обчислення даних характеристик для СТС в цілому, що дозволило побудувати функції мети, необхідні при розв'язку задач розробки СТС. Отримані на їх основі числові значення характеристик систем дозволять порівняти оптимальність розв'язків цих задач. Даний підхід апробовано в [9, 74].

2.2. Вдосконалення методів знаходження оптимальних рішень

В більшості випадків СТС периметру території призначені для охорони об'єктів, що вписані в багатокутник із Z сторін (ділянок). Кожна з ділянок має свою специфіку охорони відносно загроз, котрі потрібно виявляти на ній, та кількості наявних для її захисту компонентів (див. § 2.1). При цьому охорона кожної ділянки може бути реалізована за допомогою великої множини компонентів, що випускаються промисловістю та характеризуються широким набором різноманітних характеристик [100, 117], тому кількість варіантів реалізації всієї СТС є достатньо великою.

Формування множини альтернативних реалізацій компонентів, тобто виконання вимог щодо характеристик обмеження, відбувається шляхом фільтрування всієї множини доступних компонентів, в результаті чого відкидаються всі компоненти, які строго не підходять для експлуатації в умовах заданих на ділянці природних (температурний режим, вологість та ін.), випадкових і цілеспрямованих завад, наприклад, сильного вітру, руху дрібних або великих тварин, близькість ліній електропередач і т.д.

2.2.1. Оптимізація на основі методу морфологічних таблиць

Морфологічна таблиця (матриця) представляє собою узагальнену структуру у вигляді множини алтернативних реалізацій компонентів, з яких компонуються СТС [56, 57]. Морфологічна таблиця складається з m стрічок, що дорівнює кількості компонентів в узагальненій структурі СТС, і n_i стовпчиків, що дорівнює кількості альтернативних варіантів реалізації кожного

з них. Альтернативні варіанти, у свою чергу, вибираються відповідно до заданих на ділянках загроз.

Кількість стовпчиків морфологічної таблиці є індивідуальною для кожної стрічки. Вона описує кількість альтернативних варіантів реалізації складової частини СТС і відповідає кількості наявних компонентів даного функціонального призначення [25, 57,]. Так, кожному функціональному компоненту можна поставити у відповідність одну стрічку таблиці, а кожному способу її реалізації – одну клітинку в даній стрічці. Множину альтернатив можна представити у вигляді відношення:

$$M =< X, R >, \tag{2.12}$$

де X – множина типів комонентів $X =< x_1, x_2, ..., x_m >$, де x_i - тип компоненту (сповіщувач, кабель, приймально-контрольний пристрій, сирена, рідкокристалічна (ЖКІ) панель, і т.д.); R - система множин способів реалізації компонентів кожного типу $R =< R_1, R_2, ..., R_m >$, де R_i – множина способів реалізації i-того типу компонента, m – кількість типів компонентів.

Множина альтернатив СТС представлена композицією елементів системи R, тобто кожна альтернатива включає по одному елементу $r_{i,j}$ з множини способів реалізації R_i ($r_{i,j} \in R_i$), $i = 1,...,m$, $j = 1,...,n_i$ (стрічки та стовпці морфологічної таблиці), n_i - потужність множини R_i. Таким чином, загальне число альтернатив k, представлених морфологічною таблицею, дорівнює:

$$k = \prod_{i=1}^{m} n_i , \tag{2.13}$$

де n_i - потужність множини R_i, m – кількість типів компонентів.

З формули (2.13) очевидною є експоненціальна залежність кількості варіантів перебору від кількості типів компонентів m і їх поліноміальна залежність від кількості альтернативних реалізацій n_i.

Перед виконанням процедури формування морфологічної таблиці та вибору оптимальних систем вводяться дані про загальні вимоги до СТС

периметру території (діапазон робочих температур, параметри завад, які можуть виникати при експлуатації системи та інші обмеження), розміри ділянок периметру та загрози, які необхідно виявляти на них. Після вводу кожного компонента та його параметрів в базу даних (БД), автоматично обчислюються їхні функціонально-вартісні характеристики [9], котрі будуть використовуватись при обчисленні цільових функцій при оптимізації структури СТС [74].

Загальна структура морфологічної таблиці наведена в табл. 2.1, де у стрічках знаходяться назви типів компонентів (x_1 ... x_m), що є елементами множини X, а в стовпцях – варіанти реалізації кожного компонента ($r_{1,1}...r_{m,n_m}$), де $r_{i,j}$ - булеві змінні, що набувають значень 0 або 1. Альтернативний варіант СТС формується вибором реалізації кожного компонента, при цьому, $\sum_{j=1}^{n_i} r_{i,j} = 1$ для всіх $i = 1...m$, тобто для кожного компонента можна вибрати лише одну його реалізацію. Далі у таблицях реалізації, що формують одну з альтернатив, виділятимемо сірим кольором.

Структуру морфологічної таблиці можна організувати двома способами: з гомогенним або гетерогенним покриттям ділянок периметру. Гомогенне покриття ділянок означає, що для охорони кожної ділянки периметру використовується деяка кількість сповіщувачів однакової моделі, а встановлювати на одній ділянці сповіщувачі різних моделей недопустимо. Гетерогенне покриття ділянок передбачає можливість одночасного застосування сповіщувачів різних моделей на одній і тій же самій ділянці. Приклад гомогенної організації морфогічної таблиці приведено в табл. 2.2.

Таблиця. 2.1

Морфологічна таблиця

Тип компонента	Реалізації компонента			
x_1	$r_{1,1}$	$r_{1,2}$	. . .	$r_{1,n1}$
x_2	$r_{2,1}$	$r_{2,2}$	. . .	$r_{2,n2}$
. . .	. . .	. . .	. . .	. . .
x_m	$r_{m,1}$	$r_{m,2}$	. . .	$r_{1,nm}$

Таблиця 2.2

Гомогенна морфогічна таблиця

Тип компонента	Реалізації компонента			
Сповіщувач	FMW-3	Лінар	. . .	Barrier-300
Кабель	CV-K 402	…	CV-K 404	-
ПКП	CA-10	….	Integra	-
Клавіатура ЖКІ	INT-KLCDS-BL	Pyronix MX-ICON	-	-
Сирена	SP-4006 R	SOW-300 R	SOW-100	-

Розглянемо складність задачі оптимізації за гомогенною морфологічною таблицею. Нехай охоронна територія складається з чотирьох ділянок, кожна з них може охоронятися одним з десяти альтернативних компонентів. Тоді кількість ітерацій повного перебору при гомогенному покритті ділянок дорівнює згідно формули (2.13)

$$k = 10^4 = 10000.$$

Це не надто велике число, і повний перебір такої кількості варіантів може бути здійснений у межах прийнятного часу навіть на комп'ютерах незначної потужності. Проте зі збільшенням кількості зон тривалість повного перебору відчутно збільшується. Так, вже для шістнадцяти зон тривалість перебору з використанням сучасного комп'ютера Intel Core i7 Extreme Edition i980EE (продуктивністю 147600 MIPS) [108], що дозволяє опрацьовувати приблизно 100 варіантів за одну наносекунду (10^{-9} секунди) або 1 варіант за 10^{-11} секунди, складатиме $10^{16} \cdot 10^{-11} = 10^5$ секунд або, приблизно, 27 годин 30 хвилин.

Звідси видно, що повний перебір при розв'язанні поставленої задачі не може скласти конкуренцію комбінаторним методам. Але оцінка кількості допустимих розв'язків задачі дискретного програмування має важливе значення для їх порівняльного аналізу. Зокрема, це дозволить встановити, яка частка допустимих розв'язків була оброблена тим чи іншим алгоритмом і яка їх частина була відкинута при формуванні множини оптимальних розв'язків.

Недоліком організації записів у наведеному прикладі морфологічної таблиці є неможливість одночасного використання різнотипних компонентів

при побудові СТС. Тобто, вибір того чи іншого сповіщувача зумовлює його використання на усіх ділянках периметру підохоронного об'єкту. Альтернативою до цього підходу є гетерогенна організація морфологічної таблиці шляхом деталізації інформації про використання сповіщувачів у вигляді їх комбінацій для охорони кожної ділянки.

Розглянемо складність задачі оптимізації СТС на основі гетерогенної морфологічної матриці. Для цього приймемо наступні припущення. Нехай компоненти, що призначені для виявлення загроз на j-тій ділянці, поділені на t груп за межами дії, так, що

$$\sum_{i=1}^{t} a_{i,j} = n_j ,$$

де $a_{i,j}$ – кількість сповіщувачів, що розміщені на j-тій ділянці і належать до i-тої групи меж дії; n_j – кількість моделей сповіщувачів, які згідно умов i-тої ділянки придатні для її охорони.

Посортуємо компонети в кожній групі в порядку зростання їхніх меж дії

$$L_{i,j} > L_{i-1,j},\ i = 1,2,...,t ,$$

де $L_{i,j} = L_j$ – межі дії сповіщувачів j-тої моделі, які вміщено в i-ту групу областей дії.

Нехай межі дії компонентів кратні одні одним, тобто

$$\frac{L_{i,j}}{L_{i-1,j}} = \Pi_{i-1,j},\ i = 1,2,...,t ,$$

де Π_{i-1} – ціле число, що визначає, яку кількість компонентів рівня i–1 потрібно для покриття меж дії компонента рівня i. Наприклад, число Π_0 визначає, скільки потрібно компонетів нульового рівня для покриття ділянки, яку охоплює один компонент першого рівня.

Оцінка кількості альтернативних варіантів проектування СТС може здійснюватися шляхом розбиття її ділянок на множину теоретичних ділянок, довжини яких дорівнюють найдовшим межам дії компонента:

$$c_{i,j} = \begin{cases} a_{i,j} + c_{i-1,j}^{\Pi i-1,j}, \ якщо\ i > 0, \\ a_{0,j}, \ якщо\ i \le 0, \end{cases} \tag{2.14}$$

де $a_{i,j}$ – кількість компонентів j-того типу, що належать до i-тої групи меж дії, $c_{t,j}$ – кількість альтернативних варіантів покриття ділянки j-того типу.

Оцінку для теоретичної ділянки (2.14) можна використати для обчислення кількості варіантів покриття справжніх ділянок, протяжність яких визначається значеннями $S_1, S_1, ..., S_Z$:

$$k \ge c_{t,1}^{|S_1 / L_{t,1}|} c_{t,2}^{|S_2 / L_{t,2}|} ... c_{t,Z}^{|S_Z / L_{t,Z}|}, \tag{2.15}$$

де $L_{i,j}$ – межі дії компонетів j-того типу, i-тої групи, S_j – протяжність j-тої ділянки, $c_{t,j}$ – кількість варіантів покриття ділянки j-того типу.

Порівнюючи оцінку складності гомогенного (2.13) та гетерогенного (2.15) варіантів, видно, що окрім добутку кількостей алтернативних компонентів, остання має ще й експоненціальну залежність від протяжності ділянок. Незважаючи на те, що складність задачі проектування у гомогенній постановці нижча, ніж у гетерогенній, ресурсоємність гомогенного підходу більша, в чому можна переконатись за допомогою наступного прикладу.

Нехай компонент А має ресурсоємість ρ_A і межі дії α, компонент В має ресурсоємність ρ_B і межі дії β, при чому $\rho_A > \rho_B$, $\alpha > \beta$, $\beta = \frac{1}{2}\alpha$, $\rho_B = \frac{1}{2}\rho_A$. Тоді для покриття ділянки протяжністю $\alpha + \beta$ з використанням гомогенного підходу потрібно 2 одиниці компонента А або 3 одиниці компонента В, що виливається у ресурсоємності $2\rho_A$ або $3\rho_B$. При використанні гетерогенного підходу ресурсоємність дорівнює $\rho_A + \rho_B$, що менше або дорівнює ресурсоємності гомогенних покриттів: $\rho_A + \rho_B < 2\rho_A$ або $\rho_A + \rho_B = 3\rho_B$ відповідно. Отже, гетерогенний підхід має потенційну можливість у зменшенні ресурсоємності, тому саме він буде детально розглядатися в даній роботі.

Наведемо приклад гетерогенної організації морфологічної матриці для периметру з 4-ох ділянок, довжиною 124, 298, 299, 299 м (рис. 2.3), на котрих

необхідно виявляти рух при наявних для використання 6-ти типів мікрохвильових сповіщувачів із зонами дії 200, 100, 75, 75, 50, 50 м.

У таблиці 2.3 сірим позначені комірки морфологічної матриці, що описують одну із можливих алтернатив реалізації СТС. Слід зазначити, що даний підхід може бути використаний не лише для сповіщувачів, а й також для інших компонентів (кабель, ПКП, панель ЖКІ та ін.). Перша ділянка периметру, довжиною 124 м, має 17 варіантів сполук сповіщувачів, що наведені у таблиці 2.4, де кожна стрічка таблиці формує варіант СТС, в якому: кожна цифра у стрічці відповідає кількості штук сповіщувачів з межами дії 200, 100, 75, 75, 50, 50 м, відповідно, для повного покриття ділянки. Згідно (2.13) кількість альтернативних варіантів, що описуються морфологічною таблицею 2.4, становитиме $N_{\text{var}} = 17*94*94*94 = 14119928 \approx 14*10^6$.

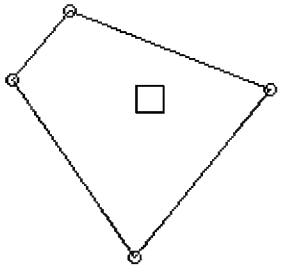


Рис. 2.3. Приклад периметру з 4-ох ділянок

Таблиця 2.3

Морфологічна таблиця

Ділянки периметру	Варіанти охорони дінянок (реалізації комбінацій сповіщувачів)			
1	варіант 1	варіант 2	. . .	варіант 17
2	варіант 1	варіант 2	. . .	варіант 94
3	варіант 1	варіант 2	. . .	варіант 94
4	варіант 1	варіант 2	. . .	варіант 94
Тип компонента	Реалізації компонента			
Кабель	CV-K 402	…	CV-K 404	-
ПКП	CA-10	….	Integra 64	-
Клавіатура ЖКІ	INT-KLCDS-BL	Pyronix MX-ICON	….	INT-KLCD-GR
Сирена	SP-4006 R	SOW-300 R	SOW-100	-

Формування морфологічних таблиць для об'єктів класу складності СТС важко робити вручну, доцільно скористатися комп'ютерними алгоритмами.

Алгоритми формування морфологічних таблиць визначаються специфікою задачі – стрічки $R_i, i = 1,...,n$ для СТС формуються множинами можливих варіантів покриття сповіщувачами усіх n ділянок підохоронного об'єкта.

Таблиця 2.4

Варіанти охорони ділянки 1

№ варіанту	200	100	75	75	50	50
1	1	0	0	0	0	0
2	0	2	0	0	0	0
3	0	1	1	0	0	0
4	0	1	0	1	0	0
5	0	1	0	0	1	0
6	0	1	0	0	0	1
7	0	0	2	0	0	0
8	0	0	1	1	0	0
9	0	0	1	0	1	0
10	0	0	1	0	0	1
11	0	0	0	2	0	0
12	0	0	0	1	1	0
13	0	0	0	1	0	1
14	0	0	0	0	3	0
15	0	0	0	0	2	1
16	0	0	0	0	1	2
17	0	0	0	0	0	3

Розглянемо одну із реалізацій алгоритму формування множини можливих варіантів покриття сповіщувачами однієї ділянки протяжністю L (рис.2.4). Він базується на рекурсивній функції *rec*, що на вхід отримує 2 аргументи (індекс поточного компонента та довжину ділянки). Якщо індекс поточної моделі компонента вихдоть за межі загальної кількості моделей компонентів, то здійснюється вихід з алгоритму. Функція $sum(i)$ обчилюється за формулою $sum(i) = \sum_{k=i}^{M} rang_k \cdot count_k$.

Зважаючи на значну кількість стовпчиків (варіантів покриття ділянок) у морфологічній таблиці описаної вище структури, доцільно використати підходи для мінімізації їх кількості через усунення апріорі неоптимальних варіантів покриття кожної ділянки. З цією метою спочатку проводимо оптимізацію

кожної ділянки зокрема, після цього складаємо оптимізовані СТС з цих оптимальних ділянок. Хоча такий підхід зменшує обчислювальну складність, але виникає можливість невиявлення деяких оптимальних СТС.

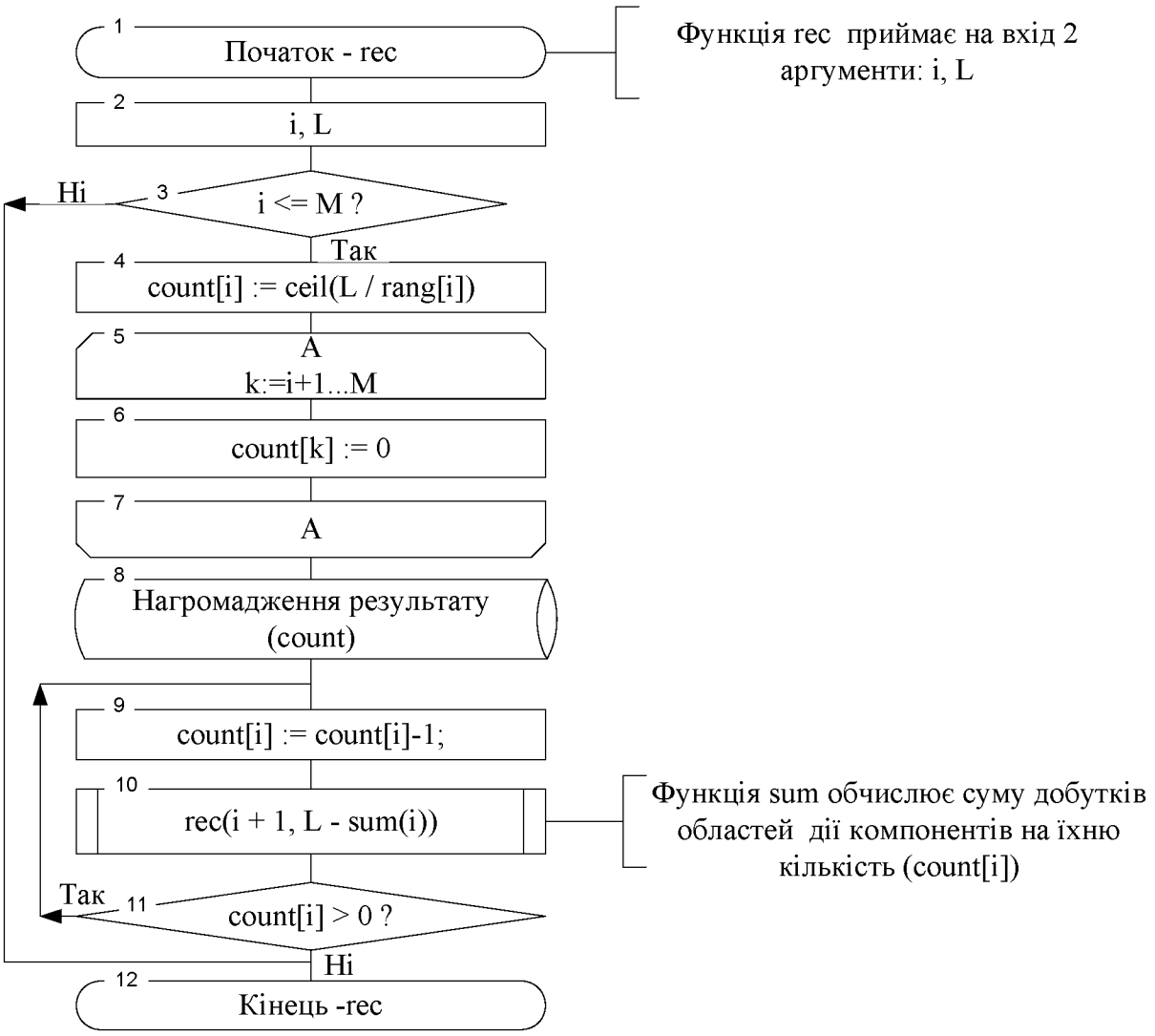


Рис. 2.4. Алгоритм формування гетерогенної морфологічної матриці СТС

Формування мінімізованої морфологічної таблиці виконується за допомогою алгоритму [97]. В алгоритмі критерії Q_i^{zone}, R_i^{zone}, F_i^{zone}, C_i^{zone} дозволяють сформулювати векторну цільову функцію для оцінювання функціонально-вартісної характеристики покриття ділянки у такому вигляді:

$$f_1 = Q_i^{zone}, f_2 = R_i^{zone}, f_3 = 1 - F_i^{zone}, f_4 = C_i^{zone},$$

$$f_j \xrightarrow{x^{(k)}} \min, j = 1,2...,M; \qquad M = 4,$$

де $x^{(k)}$ – k-тий варіант покриття і-тої ділянки.

Згідно [97] розв'язок $x^{(1)}$ мажорує $x^{(2)}$, при виконанні двох умов: (i) стверджується, що розв'язок $x^{(1)}$ не гірший ніж $x^{(2)}$ по всіх критеріях, якщо $f_j(x^{(1)}) \prec f_j(x^{(2)})$ для всіх $j = 1,2,...,M$, де оператор $\prec$ означає, що ліва частина гірша за праву, а оператор $\succ$ – навпаки; (ii) розв'язок $x^{(1)}$ строго кращий від $x^{(2)}$ хоча би за одним критерієм, або $f_j(x^{(1)}) \succ f_j(x^{(2)})$ хоча б для одного $j = 1,2,...,M$. У випадку, коли всі критерії $f_j(x^{(i)})$, $j = 1,2,..M$ є цільовими функціями на мінімум, умови запишуться таким чином: (i) стверджується, що розв'язок $x^{(1)}$ не гірший ніж $x^{(2)}$ по всіх критеріях, якщо $f_j(x^{(1)}) \le f_j(x^{(2)})$ для всіх $j = 1,2,...,M$; (ii) розв'язок $x^{(1)}$ строго кращий від $x^{(2)}$ хоча би за одним критерієм, або $f_j(x^{(1)}) < f_j(x^{(2)})$ хоча б для одного $j = 1,2,...,M$.

Формальний опис алгоритму створення мінімальної морфологічної матриці (маркування мажорних розв'язків) [97]:

Step 0: Begin with $i = 1$.

Step 1: For all $j \neq i$, compare solutions $x^{(i)}$ and $x^{(j)}$ for domination using the above two conditions for all M objectives.

Step 2: If for any j, $x^{(i)}$ is dominated by $x^{(j)}$, mark $x^{(i)}$ as 'dominated'.

Step 3: If all solutions (that is, when $i = N$ is reached) in the set are considered, Go to Step 4, else increment i by one and Go to Step 1.

Step 4: All solutions that are not marked 'dominated' are non-dominated solutions.

Після завершення алгоритму частина розв'язків позначені як ті, над якими мажорують інші розв'язки. Усі немарковані розв'язки утворюють множину Парето-оптимальних варіантів покриття ділянки. Комбінування різних співвідношень Парето-оптимальних варіантів покриття кожної зони дають множину варіантів СТС, що утворюють множину претендентів на Парето-оптимальні розв'язки задачі (2.9–2.10).

Процедура оптимізації є багатопрохідною. Під час кожного проходу з кожної стрічки морфологічної матриці почергово вибирається лише один із елементів-стовпчиків, таким чином, кожен прохід формує один альтернативний варіант побудови СТС. Потім альтернативні варіанти оцінюються за критеріями

ефективності та ресурсоємності системи в цілому. Кожен крок кожного проходу починається з вибору ключових показників компонента з бази даних. При цьому виконуються наступні операції [74]:

1. Перевірка ключових показників обмежень. Якщо ці показники для компонента не відповідають обмеженням для системи в цілому, то поточний варіант відкидається, зокрема:

- перевіряється врахування конкретних умов експлуатації даного компонента в зоні його розміщення. Якщо компоненти поточного варіанту системи не витримують необхідних умов експлуатації, то варіант відкидається;
- перевіряється сумісність компонентів у поточному варіанті системи. Компонент, який вибирається з бази даних першим при формуванні нового варіанту системи (у нашому випадку сповіщувач), вважається наперед сумісним. Надалі наступні компоненти перевіряються на сумісність з попередньо вибраними компонентами. Якщо компоненти поточного варіанту системи є несумісними, то варіант відкидається;
- перевіряється взаємозамінність компонентів – визначається відсутність якісних відмінностей у ключових функціональних показниках та параметрах підключення елементів. Наприклад, сповіщувачі, що відрізняються тільки зоною дії, є взаємозамінними, тому що для захисту однакового периметру потрібна різна кількість цих сповіщувачів. Сповіщувачі, які мають різний вихідний сигнал, не є взаємозамінними.

2. Визначення кількості компонентів СТС, які забезпечують захист заданого периметру території. Наприклад, підраховується кількість потрібних сповіщувачів, виходячи з меж їхньої дії і довжини охоронної ділянки згідно (2.4) з врахуванням заданого коефіцієнта перекриття; для кабелів визначається сумарна довжина (з врахуванням можливості підключення декількох сповіщувачів до одного багатожильного кабеля); для ПКП визначається необхідна кількість входів, яка, в свою чергу, визначається кількістю сповіщувачів.

3. Проводиться обчислення критеріїв (2.5)–(2.8) для кожного альтернативного варіанту проектованої системи.
4. Здійснюється відбір тих варіантів СТС, які створюють Паретові границі всіх альтернативних варіантів за двома ключовими функціональними характеристиками [73, 74]. Ці границі створюють для пар показників якість-затрати Q_k^{sys} / C_k^{sys} і надійність-затрати R_k^{sys} / C_k^{sys}. Якщо деякий варіант побудови СТС периметру території повторюється як оптимальний в Паретових границях обох ключових функціональних показників, то він є, безумовно, оптимальним. Якщо ж такого варіанту немає, то умовно-оптимальний варіант можна виявити шляхом пошуку тих варіантів, які знаходяться найближче до Паретових границь цих двох ключових функціональних показників при поступовому збільшенні затрат C_k^{sys} на реалізацію СТС.

Даний метод ефективний для оптимізації СТС з обмеженою кількістю альтернативних варіантів, множина яких доступна на початку розв'язання задачі оптимізації. Проте даний метод характеризується значною часовою складністю, тому ефективний для задач малої розмірності.

Оптимізацію функціонально-вартісних характеристик СТС в залежності від вимог ефективності та ресурсоємності можна організувати у вигляді одношарових та багатошарових мереж. Ефективність багатошарових мереж зростає зі збільшенням кількості шарів, проте призводить до гірших показників ресурсоємності. При цьому, ресурсоємність зростає як одноразово, у зв'язку з більшою кількістю компонентів, так і в режимі експлуатації СТС у зв'язку зі збільшенням кількості хибних тривог через їх накладання. Розглянемо особливості оцінювання функцій мети одношарової та багатошарової СТС.

Одношаровою мережею будемо вважати мережу, в якій порушнику для проникнення необхідно перетнути межі дії сповіщувачів одного принципу дії. Оцінювання ресурсоємності придбання і монтажу одношарової СТС здійснюється за описаними вище формулами (2.5), (2.6) та (2.7).

Багатошаровою мережею вважається така мережа, в якій порушнику для проникнення необхідно перетнути межі дії сповіщувачів багатьох принципів дії. В [106] стверджується, що з метою ймовірнішого виявлення порушника слід використати більше сенсорів на тій же самій ділянці. Однак, використання більшої кількості сенсорів підвищує не лише витрати на придбання СТС, але і ймовірність її хибного спрацьовування. Тому кількість сенсорів має відповідати рівню, при якому імовірність ідентифікації порушника достатня.

Розглянемо приклад (рис. 2.5) комбінування компонентів на двох шарах однієї ділянки з метою зменшення імовірності невиявлення загроз.

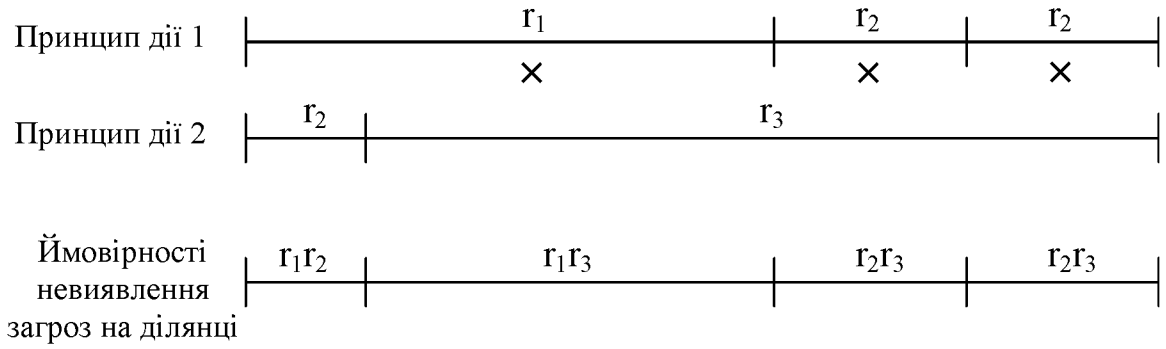


Рис. 2.5. Добутки імовірностей невиявлення у двошаровій СТС

Таким чином, ймовірність невиявлення загрози на ділянці, представленій на рис. 2.7, обчислюється за формулою

$$R = 1 - \Big((p_1 + p_2 - p_1 p_2)(p_1 + p_3 - p_1 p_3)(p_2 + p_3 - p_2 p_3)(p_2 + p_3 - p_2 p_3)\Big) + \\ + \max\{r_1 r_2, r_1 r_3, r_2 r_3, r_2 r_3\},$$

де R – ефективність СТС, p_j – імовірність безвідмовної роботи сповіщувача j-тої моделі, r_k – імовірність невиявлення загроз сповіщувачем k-тої моделі.

Однак, при збільшенні кількості шарів, імовірності хибного спрацювання їхніх сповіщувачів додаються [106], (рис. 2.6).

Імовірність хибного спрацювання є характеристикою як ефективності СТС, так і її ресурсоємності при експлуатації СТС. Згідно з [22] її оцінюють як середній потік хибних спрацьовувань по всіх шарах ділянки (див. рис.2.6)

$$Q = \frac{1}{4}\big(q_1 + q_2 - q_1 q_2 + q_1 + q_3 - q_1 q_3 + q_2 + q_3 - q_2 q_3 + q_2 + q_3 - q_2 q_3\big),$$

де Q – інтенсивність потоку хибних спрацьовувань, q_j – імовірність хибного спрацьовування сповіщувача j-тої моделі.

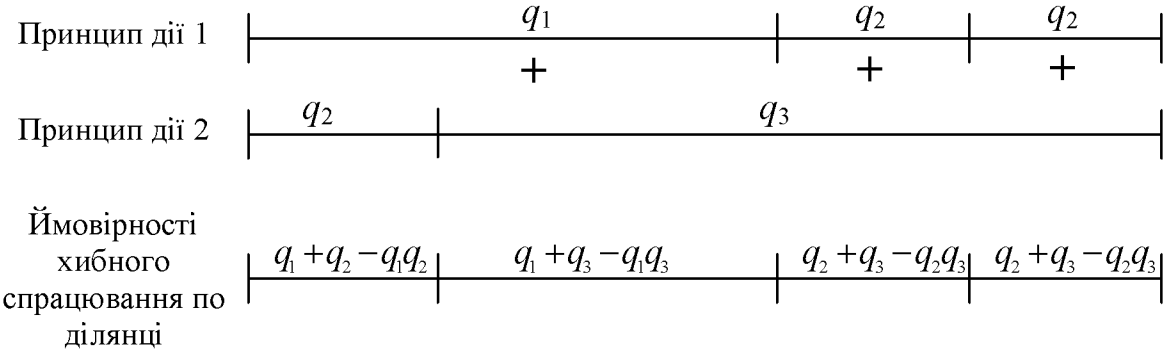


Рис. 2.6. Комбінування імовірностей хибного спрацювання двошарової СТС

Отже, постановка задачі проектування багатошарової СТС записується як

$$C_{sys,m}(\vec{N}) = \sum_{i=1}^{Z}\sum_{j=1}^{n}\sum_{l=1}^{\lambda}\left(C_j + W_j + K_j\right)N_{l,i,j}\,, \qquad (2.16)$$

де Z – кількість ділянок охоронної мережі; n – кількість альтернативних компонентів; λ - кількість шарів охоронної мережі; C_j – вартість придбання і монтажу j-го компонента; W_j – вартість гарантійного обслуговування j-го компонента; K_j – вартість кабельних сполучень до j-го компонента; $N_{l,i,j}$ – кількість компонент j-го типу, що згідно проекту системи, встановлені на l-тому шарі i-тої ділянки; $\vec{N} = (N_{1,1,1}, N_{1,1,2}, ..., N_{\lambda,Z,n})$ - вектор кількостей компонентів на кожній ділянці.

Для забезпечення належної ефективності СТС слід проводити сервісні обслуговування компонентів, проте надто часте обслуговування підвищує не лише вартість цих заходів, але й збільшує час неможливості користування системою [106]. Крім того, персонал, під час виконання обслуговуючих заходів, може встановити пристрої несанкціонованого втручання в мережу безпеки, що може негативно вплинути на якість її функціонування [43]. Ресурсоємність експлуатації двошарової СТС оцінюється із співвідношення

$$Q_{sys,m}(\vec{N}) = \frac{1}{Z}\sum_{i=1}^{Z}\frac{1}{\left|B(\vec{N},i)\right|}\sum_{(j,k)\in B(\vec{N},i)} q_j + q_k - q_j q_k, \qquad (2.17)$$

де q_j – імовірність хибного спрацьовування компонента j-го типу, $B(\vec{N},i)$ – множина пар індексів компонент двох рівнів, межі дії яких перетинаються на i-тій ділянці.

Ефективність двошарової СТС обчислюється за формулою

$$R_{sys,m}(\vec{N}) = \left(1 - \prod_{i=1,\ldots,Z}\prod_{(j,k)\in B(\vec{N},i)} (p_j + p_k - p_j p_k)\right) + \max_{i=1,\ldots,Z}\max_{(j,k)\in B(\vec{N},i)} r_j r_k, \qquad (2.18)$$

де p_j – імовірність безвідмовної роботи j – того типу компоненту, $p_k^{невиявлен.}$ – імовірність невиявлення загроз компонентом k – того типу, $B(\vec{N},i)$ – множина пар індексів компонент, межі дії яких перетинаються на i-тій ділянці.

Виходячи з цього, слід додати критерій, що мінімізує кількість обслуговуючих заходів, і застосувати інженерні рішення криптостійкого зв'язку між компонентами, що описані в третьому розділі.

Розв'язання задачі (2.9)–(2.10) може бути здійснене різними методами дискретного програмування: повним перебором, випадковим пошуком, евристиками, методами віток і границь, динамічного програмування, еволюційними методами (генетичними алгоритмами і алгоритмами мурашиної колонії). Крім того, оскільки задача є багатокритеріальною, то при її розв'язанні потрібно використати один із прийомів обробки векторної функції мети (2.9): агрегування (лінійна згортка), метод послідовних уступок або метод Парето-множин. Проаналізуємо перелічені методи розв'язання цієї задачі.

Для оцінювання можливостей використання різних алгоритмів розв'язання задачі слід враховувати її властивості, а саме дискретність та перервність області шуканих розв'язків, а також багатокритеріальність, мультимодальність, нелінійність. Задачі виду (2.9)–(2.10) є:

- багатокритеріальними, їх розв'язання класичними однокритеріальними оптимізаційними алгоритмами неможливе без методів агрегування;

- мультимодальними – містять локальні екстремуми, тобто проектовані системи можуть мати велику кількість схожих за значеннями цільової функції комбінацій, які різко відрізняються набором компонентів;
- нелінійними, адже характеристика ефективності (2.18) – нелінійна, кожен додатковий компонент має складний вплив на кінцеві характеристики $Q^{sys}, R^{sys}, C^{sys}$ системи, бо він характеризується цілком індивідуальним набором значень характеристики ефективності та ресурсоємності;
- перервними на просторі пошуку в зв'язку з тим, що співвідношення області дії сповіщувачів $\vec{X}$ не обов'язково кратні довжині ділянок, тому в цих місцях виникає перервність області пошуку екстремуму.

Ці особливості унеможливлюють застосування класичних градієнтних методів для її розв'язання. Тому було проведено аналіз методів комбінаторної оптимізації: повний та обмежений перебір, випадковий пошук, метод віток і границь, еволюційні методи. Найкращими щодо повноти результуючої множин Парето-оптимальних розв'язків виявилися метод повного та обмеженого перебору морфологічних таблиць, а також багатокритеріальні ГА.

Розв'язання задачі методом повного перебору полягає у послідовній підстановці у обмеження (2.10) та функцій мети (2.5), (2.6), (2.7) всіх допустимих значень індексів компонентів.

Розв'язання задачі за допомогою випадкового пошуку має ряд переваг і недоліків у порівнянні з усіма іншими методами. Перевага випадкового пошуку у порівнянні з повним перебором полягає у ширшому покритті області пошуку за одиницю часу. Недоліком є сильна залежність збіжності алгоритму від параметрів генератора псевдовипадкових чисел. Основними параметрами, що впливають на спроможність методу знайти оптимальний розв'язок, є рівномірність розподілу і повторюваність ланцюжка значень у псевдовипадковій послідовності. При опрацюванні достатньо великої кількості варіантів випадковий перебір буде зациклюватися на деякій невеликій підобласті можливих розв'язків, не маючи змоги вийти за її межі. Цей недолік не дозволяє обійти області, що потенційно можуть містити шуканий

оптимальний розв'язок. Однак перевагою випадкового перебору є простота реалізації та гнучкість при додаванні або зміні цільових функцій.

Згаданий недолік випадкового пошуку характерний також і для методу еврістик. Еврістики формують розв'язки на основі емпіричних правил. Прикладом еврістики може слугувати правило: взяти компонент з мінімальною ціною, і, якщо надійність СТС не буде нижча заданого значення, вважати СТС оптимальною. Інший варіант: взяти найнадійніший компонент (але ціна СТС не повинна перевищувати задану). Недоліком методу є залежність розв'язків від особливостей еврістичного правила. Тому область пошуку еврістичних методів досить вузька, що може призвести до втрати оптимального розв'язку. Ще одним недоліком еврістик є неможливість прозорої зміни або додавання нових цільових функцій без відповідної модифікації його емпіричних правил.

Ефективним методом розв'язання задач цілочисельного і дискретного програмування є метод віток і границь [61]. Він передбачає формалізацію методу оцінки нижньої границі функції мети і розбиття множини розв'язків на підмножини, що попарно не перетинаються (розгалуження). Класичний варіант не можна використати для розв'язку багатокритеріальних задач (з векторною функцією мети), а багатокритеріальні модифікації призначені для розв'язку задач лінійного програмування на бінарній області параметрів [121]. Тому методом віток і границь не можна розв'язати задачу (2.9)–(2.10), що є багатокритеріальною і нелінійною, а її параметри приймають різні значення. Результат аналізу методів комбінаторної оптимізації приведено у табл. 2.5.

Таблиця 2.5

Порівняльний аналіз комбінаторних методів

№	Назва методу	Багато-критері-альність	Швидкість збіжності	Ширина області пошуку	Гнучкість до моди-фікації	Пристосу-вання до задачі
1	Повний перебір	– / +	низька	широка	–	+
2	Випадковий пошук	+	висока	вузька	+	+
3	Метод віток і границь	–	висока	широка	–	–
4	Метод еврістик	+	середня	вузька	–	+
5	Еволюційні методи	+	висока	широка	+	+

Як видно з табл. 2.5, найкращими є еволюційні методи, які будуть в подальшому використані для рішення задачі (2.9)–(2.10).

2.2.2. Оптимізація детермінованими еволюційними методами

Згідно з (2.13), при ускладненні СТС (збільшенні кількості ділянок і їх розмірів, збільшенні кількості компонентів) кількість варіантів, які необхідно проаналізувати, різко зростає за експоненціальним законом. При цьому час проектування СТС неприйнятно зростає навіть при великій продуктивності обчислювальних ресурсів. Тому повний перебір варіантів СТС слід замінити алгоритмом, який більш цілеспрямовано шукає оптимальні рішення. Як зазначалося в § 2.2, цільова функція задачі оптимізації СТС має характеристики, які можуть бути адекватно враховані методами еволюційного пошуку [68].

У зв'язку з тим, що задачі (2.9)–(2.10) та (2.16)–(2.18) містять три суперечливі критерії, доцільно застосувати еволюційні алгоритми на базі підходів Фонцези-Флемінга [101]. Для їх використання необхідно формалізувати метод відображення хромосом у варіанти побудови СТС. Як показує досвід застосування еволюційних методів [83, 94, 98, 111, 102], ефективність оптимізації ГА суттєво залежить від методів відображення хромосом, що представляються стрічками чисел (генів), на область рішень задачі та від параметрів ініціалізації генератора псевдовипадкових чисел, селекції та мутації.

Досліджено альтернативні методи відображення хромосом на множину структур СТС, де послідовність генів: (i) описує кількості компонентів того чи іншого виду, що задовільняють вимоги обмежень використання; (ii) описує залишкові кількості компонентів, що задовольняють вимоги обмежень використання; (iii) описує індекси, що формуються під час попереднього пошуку Парето-оптимальних варіантів покриття ділянок спеціальним методом, що є елементом новизни даної роботи.

Найпростіший метод відображення полягає у кодуванні структури системи $\vec{X} = (N_{1,1}, N_{1,2}, ..., N_{z,n})$ у вигляді послідовності з $K = Z \times n$ цілих чисел, що описують кількості сповіщувачів кожного виду:

$$\vec{g} = [g_1, g_2, ..., g_K],\ g_1, g_2, ..., g_K \geq 0,\ K = Z \times n \tag{2.19}$$

де g_k – кількість одиниць k-тої моделі сповіщувача; Z – кількість ділянок периметру; n – кількість варіантів сповіщувачів, причому $n = \sum_{i=1}^{Z} M_i$; M_i - кількість моделей придатних для охорони i-тої ділянки сповіщувачів.

В термінах еволюційного програмування стрічка елементів g_k називається стрічкою генів. Цей метод запропоновано та апробовано в [8, 13, 92].

В еволюційному підході обмеження багатокритеріальної задачі побудови охоронної системи (2.10) набуває вигляду

$$\begin{cases} g_1 L_1 + g_2 L_2 + \cdots + g_{n_1} L_n \geq S_1, \\ g_{n_1+1} L_1 + g_{n_1+2} L_2 + \cdots + g_{n_1+n_2} L_n \geq S_2, \\ \vdots \\ g_{n_1+\cdots+n_{M-1}+1} L_1 + g_{n_1+\cdots+n_{M-1}+2} L_2 + \cdots + g_{n_1+\cdots+n_{M-1}+n_M} L_n \geq S_z. \end{cases} \tag{2.20}$$

$$\vec{f}(g_1, g_2, ..., g_K) = [Q(g_1, g_2, ..., g_K), R(g_1, g_2, ..., g_K), C(g_1, g_2, ..., g_K)] \to \min, \tag{2.21}$$

де Q, R, C — критерії ефективності та ресурсоємності СТС, обчислені за формулами (2.5)–(2.7), параметри якої закодовані в хромосомі $g_1, g_2, ..., g_K$.

Недоліком цього методу відображення є можливість генерування ГА таких послідовностей стрічки генів, що не задовільняють обмеженням (2.20) або формують СТС з надлишковою кількістю сповіщувачів (необгрунтовано збільшують ресурсоємність). Вилучити надлишкові сповіщувачі запропоновано за допомогою алгоритму (рис. 2.7), де вдосконалено метод відображення таким чином, щоб кожен ген кодував необхідну залишкову кількість компонентів, що дозволить забезпечити генерування СТС, які гарантовано покривають периметр об’єкта. Вдосконалений метод працює за схемою

$$\vec{g} = [g_{11}, g_{12}, ..., g_{ij}, ..., g_{ZM}], \tag{2.22}$$

$$0 \le g_{ik} \le \left\lceil \frac{S_i - \sum_{j=1}^{k-1} g_{ij} L_j}{L_k} \right\rceil, \; k = 1,...,M_i, \;\; i = 1,...,Z, \qquad (2.23)$$

де g_{ij} - залишкова кількість сповіщувачів моделі j (ген хромосоми) розміщених на i-ій ділянці, S_i - протяжність i-тої ділянки, L_k-область дії сповіщувача k-тої моделі, M_i- кількість моделей придатних для охорони i-тої ділянки сповіщувачів.

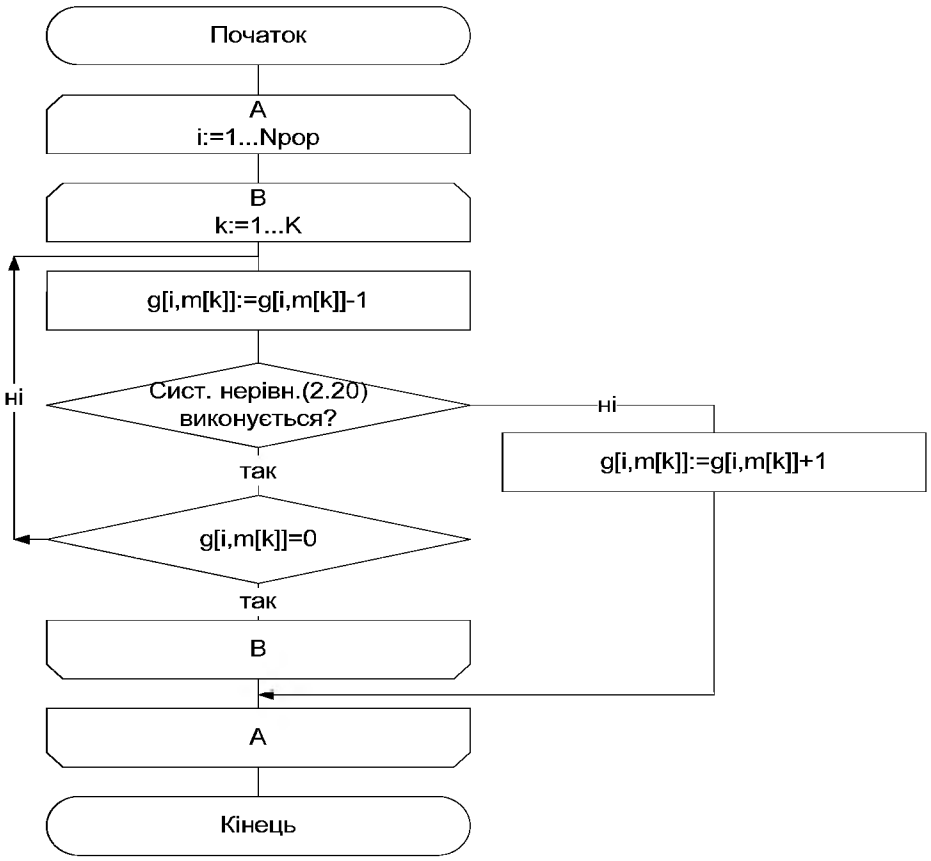


Рис. 2.7. Алгоритм усунення надлишкових сповіщувачів

Отже, метод відображення (2.22) – (2.23) формує СТС, які задовільняють обмеження (2.20) – гарантовано покривають всі ділянки периметру.

Третій метод відображення, що складає елемент новизни даної роботи, полягає у кодуванні стрічкою генів варіантів Парето-оптимальних співвідношень покриття ділянок підохоронного об'єкту

$$\vec{g} = [g_1, g_2, ..., g_i, ..., g_Z], \;\; 1 \le g_i \le M_i, \;\; i = 1,...,Z, \qquad (2.24)$$

де M_i – кількість Парето-оптимальних варіантів покриття i -тої ділянки. Визначення M_i та характеристик кожного варіанту здійснюється за допомогою алгоритмів, що розроблені в §2.2.1 і наведені на рис. 2.4 та рис.2.5.

Задача (2.20), (2.21) багатокритеріальна, тому не має єдиного розв'язку. Натомість існує множина з N_{pop} її Парето-оптимальних розв'язків, яку можна представити матрицею

$$\mathbf{G} = \begin{bmatrix} g_1^1, g_2^1, \ldots, g_K^1 \\ g_1^2, g_2^2, \ldots, g_K^2 \\ \vdots \\ g_1^{N_{pop}}, g_2^{N_{pop}}, \ldots, g_K^{N_{pop}} \end{bmatrix}. \qquad (2.25)$$

Для знаходження множини (2.25) пропонується використати генетичний алгоритм, описаний нижче. В термінах генетичного програмування (2.25) — це популяція розв'язків задачі (2.20), (2.21). При цьому вектор $\vec{g}$ утворює хромосому, що представляє варіант СТС.

Генетичний алгоритм складається з наступних кроків:

1. Позначимо: N_{init} – об'єм початкової популяції розв'язків; p_{cross} – ймовірність схрещування; N_{pairs} – кількість пар хромосом, що на кожній генерації вибираються для схрещування; p_{mut} – ймовірність мутації, N_{cross} – кількість точок схрещування, $N_{MAX.ITER}$ – максимальна кількість ітерацій генетичного алгоритму, $N_{immigrants}$ – кількість хромосом-емігрантів на кожній генерації.
2. Формуємо вектор індексів компонентів в порядку спадання їхньої межі дії: $m_1 = \arg\max\{\rho(1)...\rho(K)\}$, де функція $\arg\max\{\bullet\}$ визначає індекс найбільшого значення у множині величин межі дії компонентів, $\rho(k)$ – межі дії компонента з індексом k ; $m_2 = \arg\max_{k \neq m_1}\{\rho(1)...\rho(K)\}$, де умова $k \neq m_1$ означає, що серед індексів компонентів функція $\arg\max\{\bullet\}$ обирає будь-який за виключенням m_1 ; наступні кроки відбуваються аналогічно. Формально цю послідовність дій можна описати, як

$$m_j = \underset{\forall k, k \neq \{m_1,...,m_{j-1}\}}{\arg\max} \rho(k),\ j = 1,...,K, \qquad (2.26)$$

де умова $\forall k, k \neq \{m_1,...,m_{j-1}\}$ означає, що серед усіх індексів k обираються лише ті, що ще не були позначені як максимальні.

3. Ініціалізуємо генератор випадкових чисел в початковий стан, щоб можна було досліджувати різні схеми генерування початкової популяції та селекції при ідентичних послідовностях псевдовипадкових чисел.
4. Створюємо початкову популяцію розв'язків

$$\mathbf{G} \leftarrow \begin{bmatrix} rnd(0, N_{count})^{(1)(1)}, rnd(0, N_{count})^{(1)(2)}, rnd(0, N_{count})^{(1)(K)} \\ \vdots \\ rnd(0, N_{count})^{(N_{init})(1)}, rnd(0, N_{count})^{(N_{init})(2)}, rnd(0, N_{count})^{(N_{init})(K)} \end{bmatrix},$$

де $rnd(0, N_{count})$ — ціле випадкове число в діапазоні від 0 до N_{count}.

Можливі також інші варіанти створення початкової популяції розв'язків, наприклад, встановленням $g_k = \text{round}[L_j / \rho(id_i^j)]$, де round() — функція заокруглення дійсного числа до найближчого більшого цілого. Індекс альтернативного сповіщувача при цьому можна встановити випадковим чином $i = rnd(1, n_j)$.

5. Перевірка об'єму популяції на рівність початковому значенню $N_{pop} \leftarrow N_{init}$.
6. Проводимо першу генерацію $I \leftarrow 1$.
7. Виключаємо з популяції (матриці $\mathbf{G}$) ті хромосоми (стрічки), що не задовільняють обмеженням задачі (2.10).
8. Видаляємо надлишкову кількість сповіщувачів, якщо використовується метод відображення (2.19), якщо ж використовуються методи (2.22), (2.23) або (2.24), то даний крок пропускається. Видалення здійснюється в кожній ділянці СТС, що кодується стрічками матриці $\mathbf{G}$, починаючи з сповіщувачів з найбільшим радіусом дії (при $k = 1$) і поступово доходить до сповіщувачів з найменшим радіусом дії (при $k = K$). Видалення відбувається так: кількість g_k^i ($i = 1,...N_{pop}$) сповіщувачів відповідного типу зменшується на одиницю $g_k^i \leftarrow g_k^i - 1$, $i = 1,...N_{pop}$ і перевіряється виконання обмеження

задачі (2.20), якщо воно виконується, то ця кількість зменшується ще на одиницю і т.д., поки не порушиться виконання умови (2.20) або g_k^i досягне нуля. Коли відбулося останнє, тобто $g_k^i = 0$, то переходимо до сповіщувачів з меншим радіусом дії $k \leftarrow k+1$ і повторюємо описаний алгоритм видалення. При досягненні $k = K$ переходимо на наступний розв'язок у популяції розв'язків: $i \leftarrow i+1$. Цей крок у генетичному алгоритмі становить один з елементів наукової новизни дисертаційного дослідження (детальніше його описно нижче), а блок-схему алгоритму наведено на рис. 2.7.

9. Оцінюємо значення векторної функції мети (2.21) для всієї популяції хромосом **G**, що утворює матрицю функцій мети популяції

$$\mathbf{F} = \begin{bmatrix} \vec{f}(g_1^1, g_2^1, ..., g_K^1) \\ \vec{f}(g_1^2, g_2^2, ..., g_K^2) \\ \vdots \\ \vec{f}(g_1^{N_{pop}}, g_2^{N_{pop}}, ..., g_K^{N_{pop}}) \end{bmatrix} =$$

$$= \begin{bmatrix} Q(g_1^1, g_2^1, ..., g_K^1) & R(g_1^1, g_2^1, ..., g_K^1) & C(g_1^1, g_2^1, ..., g_K^1) \\ Q(g_1^2, g_2^2, ..., g_K^2) & R(g_1^2, g_2^2, ..., g_K^2) & C(g_1^2, g_2^2, ..., g_K^2) \\ & \vdots & \\ Q(g_1^{N_{pop}}, g_2^{N_{pop}}, ..., g_K^{N_{pop}}) & R(g_1^{N_{pop}}, g_2^{N_{pop}}, ..., g_K^{N_{pop}}) & C(g_1^{N_{pop}}, g_2^{N_{pop}}, ..., g_K^{N_{pop}}) \end{bmatrix}.$$

10. Якщо $I >= N_{MAX.ITER}$, то переходимо на крок 15.

11. Оцінюємо пристосованості хромосом за матрицею **F** за допомогою підходів на основі властивостей передування елементів Парето-множини, згідно Фонцези-Флемінга [101], що утворить вектор пристосованості

$$\vec{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N_{pop}} \end{bmatrix}.$$

12. Додаємов нову популяцію $N_{immigrants}$ емігрантів.

13. Ймовірність селекції k-тої хромосоми обчислюємо за формулою

$$p_k = \frac{v_k + 1}{\sum_{j=1}^{N_{pop}} [v_j + 1]},$$

де v_k — оцінка пристосованості k-тої хромосоми.

14. Проводимо селекцію N_{pairs} пар хромосом за правилом "колеса рулетки" і виконуємо їх N_{cross}-точкове схрещування з ймовірністю p_{cross}, здійснюючи при цьому мутацію генів з ймовірністю p_{mut}. Утворені дочірні хромосоми розташовуються в нові стрічки матриці $\mathbf{G}$.
15. Проводимо наступну генерацію $I \leftarrow I + 1$, для чого переходимо на крок 6.
16. Після завершення роботи генетичного алгоритму здійснюємо відбір множини Парето-оптимальних розв'язків задачі (2.20), (2.21) шляхом маркування домінантних (мажорних) розв'язків.

Запропоновані методи пошуку оптимальних рішень СТС за допомогою морфологічних таблиць та ГА мають свої переваги і недоліки [14]. Алгоритм повного перебору з обмеженнями доцільно використовувати для оптимізації простих СТС з обмеженою кількістю компонентів, оскільки в даному випадку будуть розглядатись усі варіанти при виборі оптимальної СТС. При використанні великої кількості компонентів вступають в дію обмеження кількості варіантів захисту ділянок, що зменшить в загальному кількість СТС, котрі розглядаються під час вибору оптимальних. В цьому випадку, для оптимізації складних СТС, доцільніше використовувати генетичний алгоритм, що забезпечує виявлення варіантів СТС, які мають кращі параметри, при цьому досягається суттєве скорочення часу роботи алгоритму.

Відбір кращих СТС реалізується за допомогою спеціальної процедури нормування трьохвимірного метричного простору можливих значень цільових функцій (2.5)-(2.7) Парето-оптимальних розв'язків задачі (2.9)–(2.10)

$$\min_{\vec{N}} \sqrt{\left(Q_{sys}(\vec{N})\beta_1\right)^2 + \left(R_{sys}(\vec{N})\beta_2\right)^2 + \left(C'_{sys}(\vec{N})\beta_3\right)^2}, \qquad (2.27)$$

де $\beta_1, \beta_2, \beta_3$ – ваги метрики; $\vec{N}$ – варіант СТС (див. (2.11)); $Q_{sys}(\vec{N}), R_{sys}(\vec{N})$ обчислюються за формулами (2.5), (2.7); $C'_{sys}(\vec{N})$ – нормований критерій вартості СТС:

$$C'_{sys}(\vec{N}) = \frac{C_{sys}(\vec{N}) - C_{sys,\min}}{C_{sys,\min} - C_{sys,\max}}, \tag{2.28}$$

де $C_{sys,\min}$, $C_{sys,\max}$ – найменша та найбільша вартість СТС по можині всіх знайдених розв'язків; значення критеріїв; $C_{sys}(\vec{N})$ обчислюється за формулою (2.6).

Найкращим розв'язком вважається той, норма якого мінімальна. Отримана множина оптимальних СТС надається замовнику для остаточного відбору. Аналіз множини розробником дозволяє виявити "вузькі місця" в структурах СТС і стимулює відповідне удосконалення елементної бази.

2.2.3. Оцінка систем тривожної сигналізації нечіткими еволюційними методами

Однак, розроблений в §2.1 метод формування критеріїв оцінки компонентів СТС не враховує впливи середовища на надійність виявлення порушника (ризик споживача) та імовірність хибного спрацювання (ризик виробника). Це пояснюється тим, що розроблений метод базується на детермінованих показниках якості, а надійність виявлення порушника та імовірність хибних спрацювань носять принципово недетермінований характер і, зазвичай, відомі з обмеженою точністю (виробники якщо і надають відповідні імовірності, то не надають їх залежності від впливаючих величин). Тому врахувати їх можна тільки на базі результатів експериментальних досліджень і статистики їх експлуатації [34]. Такі результати носять в більшості не кількісний, а якісний характер. Тому детермінований підхід до них принципово малопридатний. Хороші результати в таких випадках дають методи нечіткого виводу.

З метою вдосконалення процедури оптимізації СТС вже при оцінюванні імовірнісних характеристик компонентів СТС скористаємося апаратом нечітких множин. Врахування невизначених впливів завад та вразливості до них складових СТС ґрунтується на нечітких множинах інтенсивності завади того чи іншого виду в межах кожної ділянки. Імовірності невиявлення загроз, хибного спрацювання та безвідмовної роботи компонента *j*-тої моделі залежать від інтенсивностей електромагнітних, кліматичних та механічних завад, зокрема, описаних в [34], на *i*-тій ділянці та вразливості до них компонентів *j*-тої моделі.

Діапазони фізичних величин, що описують інтенсивність завад, коли вони починають негативно впливати на сповіщувачі, невизначені або невідомі. Проте, за допомогою експертів можна отримати їхні оцінки у вигляді нечітких функцій і отримати наближені значення ризику невиявлення загрози, як

$$r_{ij} = r_j + \sum_{d=1}^{D} a\left(\tilde{E}_{id} \cap \tilde{V}_{jd}\right) w_d , \tag{2.29}$$

$$q_{ij} = q_j + \sum_{d=1}^{D} a\left(\tilde{E}_{id} \cap \tilde{V}'_{jd}\right) w'_d , \tag{2.30}$$

$$p_{ij} = p_j + \sum_{d=1}^{D} a\left(\tilde{E}_{id} \cap \tilde{V}''_{jd}\right) w''_d , \tag{2.31}$$

де r_{ij} , q_{ij} , p_{ij} – імовірності невиявлення загроз, хибного спрацювання та безвідмовної роботи, відповідно, компонентом j-тої моделі з врахуванням впливу завад на і-тій ділянці, де його розміщено; r_j – базова оцінка імовірності невиявлення загроз компонентом j-тої моделі при забезпеченні придатних умов експлуатації (без завад); d – індекс завади; D – загальна кількість відомих завад; a – функція дефазифікації нечіткої множини; $\tilde{E}_{id}$ – нечітка множина, що описує особливості периметру території, а саме інтенсивність завади d на і-тій ділянці; $\tilde{V}_{jd}, \tilde{V}'_{jd}, \tilde{V}''_{jd}$ – нечіткі множини, що описують вразливості компонентів j-тої моделі до завади d ; $\cap$ – оператор перетину нечітких множин; w_d, w'_d, w''_d – додатки до імовірнісних характеристик комонентів при максимальному впливі

загрози d, що встановлюються експертами з врахуванням умови $0 \le r_j + \sum_{d=1}^{D} w_d \le 1$, де $w_d \in \{w_d, w'_d, w''_d\}$.

Враховуючи відмінність імовірнісних характеристик (2.29)–(2.31) компонента j-тої моделі, залежно від властивостей завад на ділянці, формули (2.5)–(2.7) набудуть вигляду

$$Q_i^{zone} = \frac{\sum_{j=1}^{n_i} N_{ij} q_{ij}}{\sum_{j=1}^{n_i} N_{ij}},\ R_i^{zone} = \max_{j=1,..n_i} \begin{cases} r_{ij}, if\ N_{ij} > 0, \\ 0, if\ N_{ij} = 0, \end{cases}\ F_i^{zone} = \prod_{j=1}^{n_i} (p_{ij})^{N_{ij}}. \qquad (2.32)$$

Використання формул (2.29)–(2.32) дозволяє покращити СТС шляхом врахування особливостей периметру території, що описуються множинами $\widetilde{E}_{id}$, $d = \overline{1, D}$, і зменшити на 16-22% Парето-оптимальну множину шляхом відсіювання СТС, що через погодні умови значно збільшують частоту хибних спрацювань. Для реалізації методу введено сімейство функцій належності, реалізації яких вибирають методом експертних оцінок, на основі статистичних даних і результатів відповідних досліджень. Тоді задача синтезу в нечіткій постановці записується аналогічно до неї у детермінованій постановці (2.9)–(2.10), (2.5)–(2.7), за виключенням того, що формули оцінки імовірнісних характеристик корегуються процедурою нечіткого виводу (2.29)–(2.31). Також доцільно ввести критерій мінімізації невизначеності отриманих результатів

$$f_{uncert} = \sum_{d=1}^{D} w\left(\widetilde{E}_{id} \cap V_{jd}\right) \longrightarrow \min, \qquad (2.33)$$

де $w(\bullet)$ – функція ширини нечіткої множини, що дорівнює різниці між крайньою правою та крайньою лівою межами перетину функцій належності.

Для розв'язку задачі (2.9 – 2.10), (2.32), (2.33) можна скористатися методами комбінаторної оптимізації, що були використані для розв'язання задачі у детермінованій постановці (2.5 – 2.7), (2.9 – 2.10) з врахуванням (2.32), (2.33) за допомогою розробленого ГА. В [60] розроблено метод пошуку найбільш прийнятного варіанту СТС серед заданої множини варіантів, але він

не генерує Парето-оптимальних структур. Пропонована оптимізація з врахуванням (2.32), (2.33) буде генерувати множину рівноцінних Парето-оптимальних СТС.

При обчисленнях функції $a(...)$ слід застосувати одну з формул дефазифікації, що повертала б точкову оцінку інтенсивності завад по ділянці. Розглянемо альтернативні підходи для виконання дефазифікації оцінок завад: за максимумом вірогідності, за центром ваги (інша назва – метод центроїду) та за верхнім межовим значенням інтенсивності завади [67, 118]

$$a\left(\tilde{E}_{id} \cap \tilde{V}_{jd}\right) = \begin{cases} r \mid \max\limits_{r} \mu(r), & C = 1, \\ \dfrac{\int\limits_{-\infty}^{\infty} r\mu(r)\,dr}{\int\limits_{-\infty}^{\infty} \mu(r)\,dr}, & C = 2, \\ \max r, & C = 3, \end{cases} \tag{2.34}$$

де r – точкове значення оцінок ризику умов по ділянці, $\mu(r)$ – функція належності його нечіткої оцінки, при $C = 1$ - найвірогідніша оцінка, при $C = 2$ - оцінка за центром ваги, при $C = 3$ - оцінка за верхнім межовим значенням.

Згідно [34] до природно-кліматичних завад належать: сильний вітер, дощ, калюжі, сніг або туман, до механічних завад: рух малих тварин, великих тварин, птахів. Електромагнітні завади створюють розряди блискавки, повітряні та підземні високовольтні силові лінії. Проте в роботі [34] не наведено значень показників перелічених факторів, тобто відсутні об'єктивні означення фізичних величин, що описують силу вітру, рівень опадів і т.п.

У випадку, коли відомі лише нижні та верхні значення параметрів можна скористатися інтервальними числами [3]. Проте, коли присутня незначна за об'ємом вибірка експериментальних або експертних даних, можна описати параметр одною з простих (трикутна, s-подібна, z-подібна) нечіткою функцією належності: нижня, верхня межі та найвірогідніше значення [68]. Застосування складних функцій належності недоцільне при обмеженій вибірці.

Нечітку множину інтенсивності завад на периметрі території доцільно описати функціями належності трапецевидного класу [4]

$$\mu(x,a,b,c,d)=\begin{Bmatrix}0, x\le a\\ \dfrac{x-a}{b-a}, a\le x\le b\\ 1, b\le x\le c\\ \dfrac{d-x}{d-c}, c\le x\le d\\ 0, d\le x\end{Bmatrix}, \tag{2.35}$$

де a,b – параметри S-сегменту, b,c – параметри інтервал-сегменту, c,d – параметри Z-сегменту.

Точки цієї функції отримують з бази метеорологічних спостережень, їх можуть додатково корегують експерти. Форма функції визначає вірогідність даного значення інтенсивності завади. Через обмеженість експериментальних даних про вразливість сповіщувачів до кожного виду завад доцільно обрати для опису нечіткої множини вірогідності ураження сповіщувача заданою інтенсивністю завад функцію належності класу S . У цих функціях слід встановити нижню та верхню межу мінімальних і максимальних інтенсивностей завад, які впливають на роботу сповіщувача

$$\mu(x,a,b)=\begin{Bmatrix}0, x\le a\\ \dfrac{x-a}{b-a}, a\le x\le b\\ 1, b\le x\end{Bmatrix}, \tag{2.36}$$

де a,b – нижня та верхня межі функії належності, x - значення аргументу функції ($a\le x\le b$).

Розглянемо приклад оптимізації СТС із нечітким виводом. Завадою виступатиме швидкість вітру. Параметри нечітких множин, що описують вірогідність швидкості вітру на периметрі підохоронного об’єкту та вразливості до нього мікрохвильових сповіщувачів визначені за даними [34]. Ілюстрації цих нечітких множин наведені на рис. 2.8, де множина вірогідностей швидкості

вітру на ділянці $\widetilde{E}_{id}$ описана трапецевидною функцією, а множина вірогідності враження сповіщувача $\widetilde{V}_{jd}$ позначена функцією класу S. Перетин множин $\widetilde{E}_{id} \cap \widetilde{V}_{jd}$ виділено заштрихованою областю – він дозволяє обчислити найвірогідніше значення швидкості вітру, що може вразити сповіщувач.

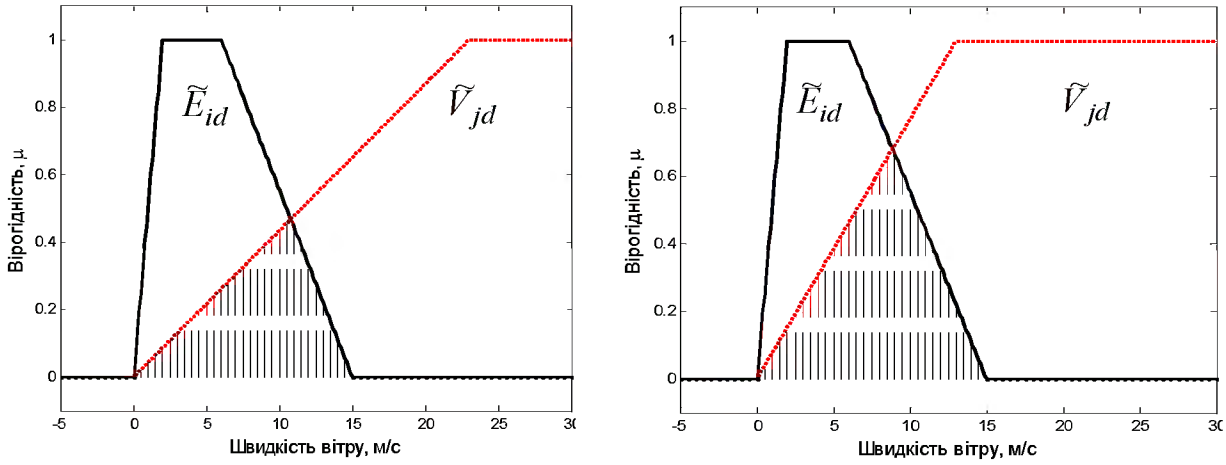


Рис. 2.8. Ілюстрація нечітких функцій швидкості вітру та вразливості до нього сповіщувачів двох моделей: (а) та (б)

Розроблений метод оптимізації структури СТС із нечітким виводом апробовано в [16, 18, 93] і описано в § 4.1 та а результати отримали практичне впровадження (Додаток Ґ). Він враховує невизначеності оцінок інтенсивностей завад і вразливості до них компонентів СТС, що дало змогу виявити напрямки покращення СТС за рахунок врахування особливостей периметру території.

Отже, в § 2.2 розроблено та удосконалено наступні неградієнтні методи:

1. Метод морфологічних таблиць, який шляхом ітераційного опрацювання стрічок морфологічної таблиці формує множину СТС.
2. Метод відображення хромосом генетичного алгоритму в область аргументів комбінаторної задачі багатокритеріальної оптимізації функціонально-вартісних характеристик СТС.
3. Метод оцінки функціональних характеристик СТС, що враховує невизначеності оцінок інтенсивності завад і вразливість до них компонентів

шляхом використання нечітких множин та їх дефазифікації при визначенні критерію ризику.

2.3. Створення бази даних компонентів

Як показано в §1.1, СТС складається з визначеного набору компонентів (сповіщувачі, кабелі, ПКП, розширювачі, ітд.), кожен з яких реалізується великою кількістю моделей, які в свою чергу мають власний набір параметрів. Зрозуміло, що дана предметна область містить великий набір даних, тому доцільно використати базу даних, як механізм, який би забезпечив їх взаємопов'язаність, мінімальну надлишковість, незалежність від програм, цілісність та захист від неавторизованого доступу [42, 62]. Предметну область СТС можна легко представити в табличному вигляді [34, 100, 117] тому доцільно використати найбільш розповсюджену реляційну модель БД, з підтримкою систем управління БД [46], зокрема, MS Access [23], котра забезпечить максимальну ефективність роботи з даними [74, 90, 126].

На першому етапі розробки БД проводиться системний аналіз та словесний опис інформаційних об'єктів предметної області [42]. Тому, виходячи з § 1.1, доцільно розділити всі компоненти СТС по класах на сповіщувачі, кабелі, ПКП, розширювачі, засоби оповіщення, ЖКІ панелі, домомофони, зчитувачі, контролери, батареї живлення, кожен з яких має власні параметри та взаємодіє з іншими класами. Наприклад, сповіщувачі СТС розділяються по типу взаємодії з порушником (принцип дії) [34, 50], а тип, в свою чергу, має набір параметрів, що характеризують його властивості [100]. Інші класи характеризуються параметрами та типом взаємодії між собою.

Другим етапом є проектування інфологічної моделі предметної області за допомогою графічної мови ER-моделювання [62]. Розроблено опис розподілу класів (рис. 2.9), де у таблиці «Довідник класів» описано класи компонентів, у таблиці «Довідник принципів дії» - принципи дії компонентів, у таблиці «Довідник загроз» - опис загроз, котрі можна виявляти наявною множиною

компонентів [74], у таблиці «Принципи дії класу», - принципи дії компонентів, котрі будуть належати до даного класу, у таблиці «Загрози принципу дії» - типи загроз, які має виявляти компонент даного принципу дії.

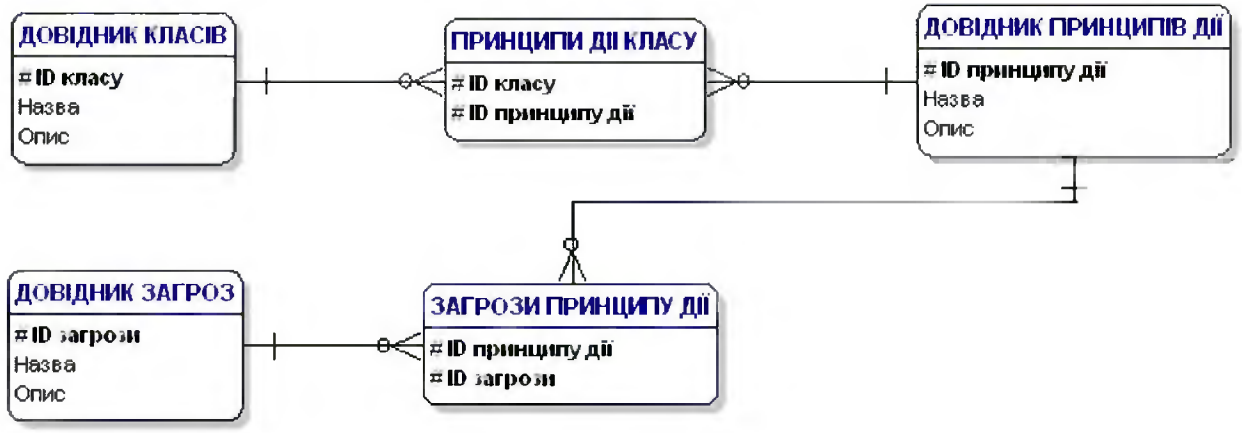


Рис. 2.9. Схема розподілу класів, принципів дії компонентів та загроз

Запропонований опис компонентів СТС представлено на рис. 2.10, де таблиця «Довідник компонентів» містить інформацію про їх клас, принцип дії, тип, якість, надійність і ціну. Кожен компонент характеризується набором параметрів, описаних у таблиці «Параметри компонента», де їм присвоєно значення, нормалізоване значення та одиниця вимірювання. Інформацію про класи, принципи дії та одиниці вимірювання параметрів компонентів представлено відповідно у таблицях «Довідник класів», «Довідник принципів дії» та “Довідник одиниць вимірювання ”.

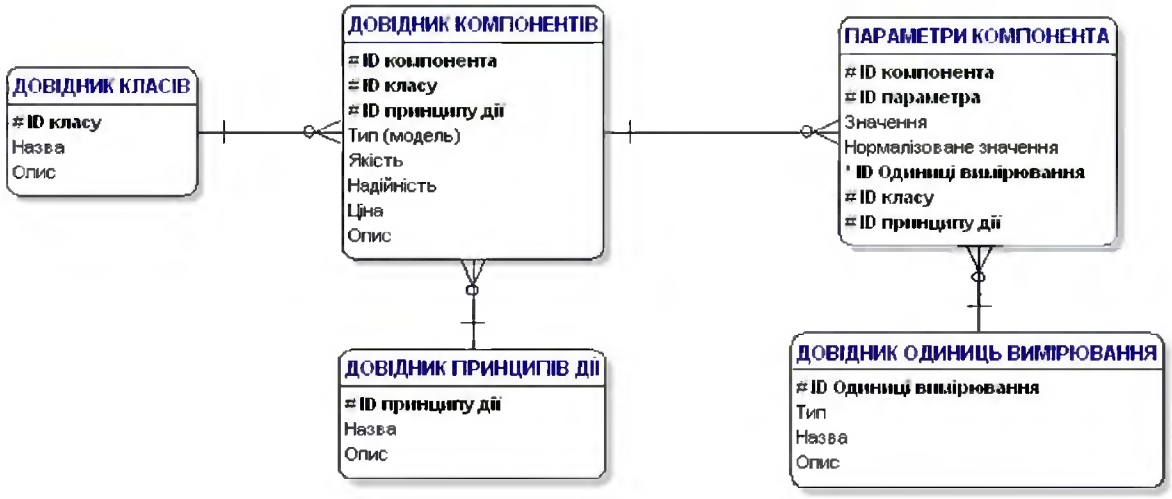


Рис. 2.10 Схема опису компонентів

На рис. 2.11 представлено опис структури використання параметрів в загальній системі [74]. Так, параметри відносять до класів і принципів дії та розподіляють по критеріях (таблиця «Довідник визначених параметрів»). Опис критеріїв знаходиться у таблиці «Довідник критеріїв», а опис самих параметрів - у таблиці «Довідник параметрів», причому одиниці вимірювання описані у таблиці «Одиниці вимірювання параметрів». Кожна окрема одиниця вимірювання описується у таблиці «Довідник одиниць вимірювання».

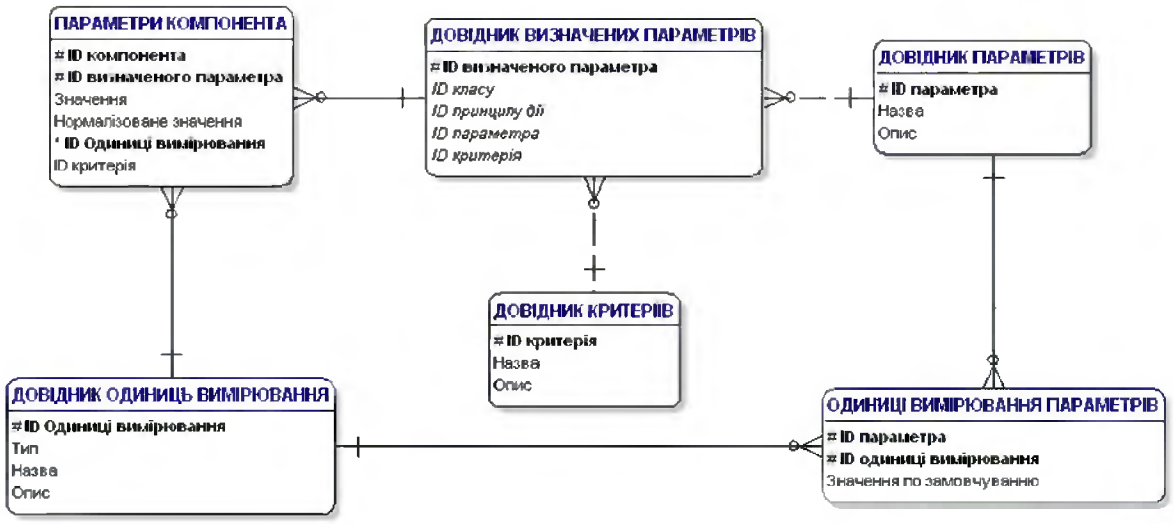


Рис. 2.11 Опис схеми параметрів компонентів

Окрім компонентів СТС, їх параметрів та інформації про розподіл класів, принципів дії компонентів та загроз запропонована структура (рис. 2.12), що містить інформацію про периметри територій для СТС, які розробляються. Таблиця «Довідник об'єктів» містить назву об'єкта, для якого створюється СТС, і координати розміщення ПКП. Елементи об'єкту описуються у таблиці «Елементи об'єкту», де кожному об'єкту ставляться у відповідність зони та їх характеристики, описані у таблиці «Довідник зон», та типи загроз, які необхідно виявляти на даних зонах, описані у таблиці «Загрози елементів».

Третім етапом є перевірка на нормалізацію даних (дозволяє мінімізувати дублювання даних) [42, 62]. Розроблена модель даних відповідає першій нормальній формі (не містить атрибутів і груп атрибутів, що повторюються або таких, що мають більше одного значення). Також вона відповідає другій нормальній формі (не містить атрибутів, залежних тільки від частини

унікального ідентифікатора) та третій нормальній формі (не містить атрибутів, залежних від тих атрибутів, які не є частиною унікального ідентифікатора).

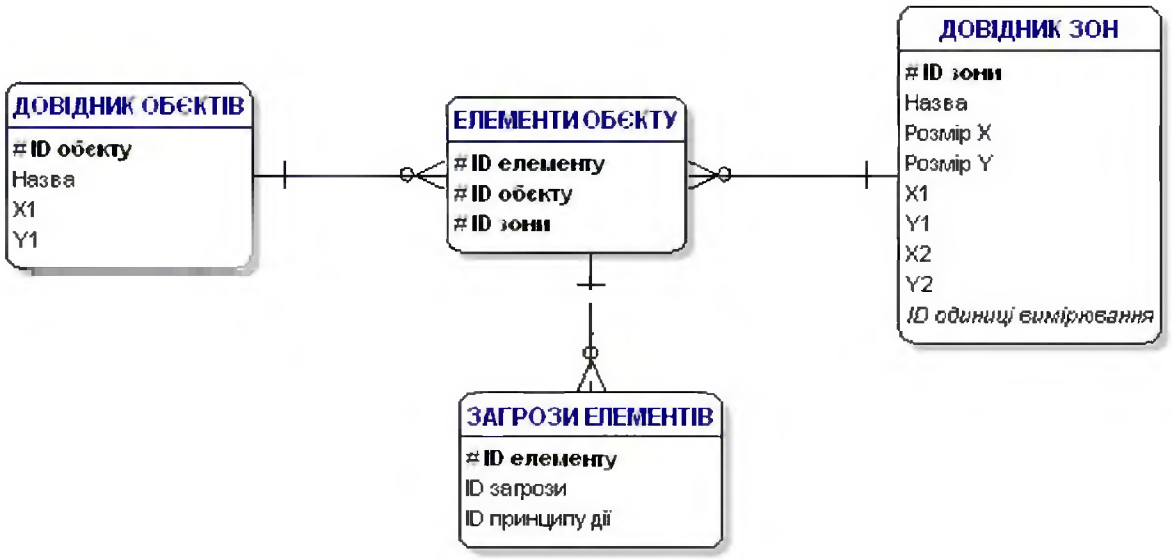


Рис. 2.12 Схема опису зон периметрів територій

Четвертим етапом є даталогічне проектування, що дозволяє на базі ER-моделі спроектувати реляційну БД компонентів СТС на основі СУБД MS Access [42, 62], яка забезпечує гнучкий ввід та заміну параметрів компонентів (Додаток И). В БД спочатку вводять загальні дані про класи, їх принципи дії, параметри компонентів тощо, які будуть використовуватися при вводі інформації про самі компоненти СТС та створенні запитів до БД [90, 126].

Отже, у § 2.3 на базі СУБД MS Access створено БД, що забезпечило гнучкий доступ та управління даними в процесі розробки СТС (див § 2.4).

2.4. Комп'ютерна система підтримки процесу розробки систем тривожної сигналізації

Створення комп'ютерної системи підтримки процесу розробки СТС складається з наступних етапів: (i) розробка БД, що містить інформацію про компоненти СТС (див. § 2.3); (ii) розробка форм вводу та редагування даних в БД; (iii) розробка вікна для вводу опису периметру території; (iv) розробка

методів агрегування та відбору оптимальних СТС (див. §2.2); (v) розробка вікна графічного виводу результатів. Взаємозв'язок між розробленими елементами та дані, що передаються між ними, представлено на узагальненій схемі рис. 2.13. Запропонована структура забезпечує виконання вимог щодо розробки СТС, представлених на рис. 2.14.

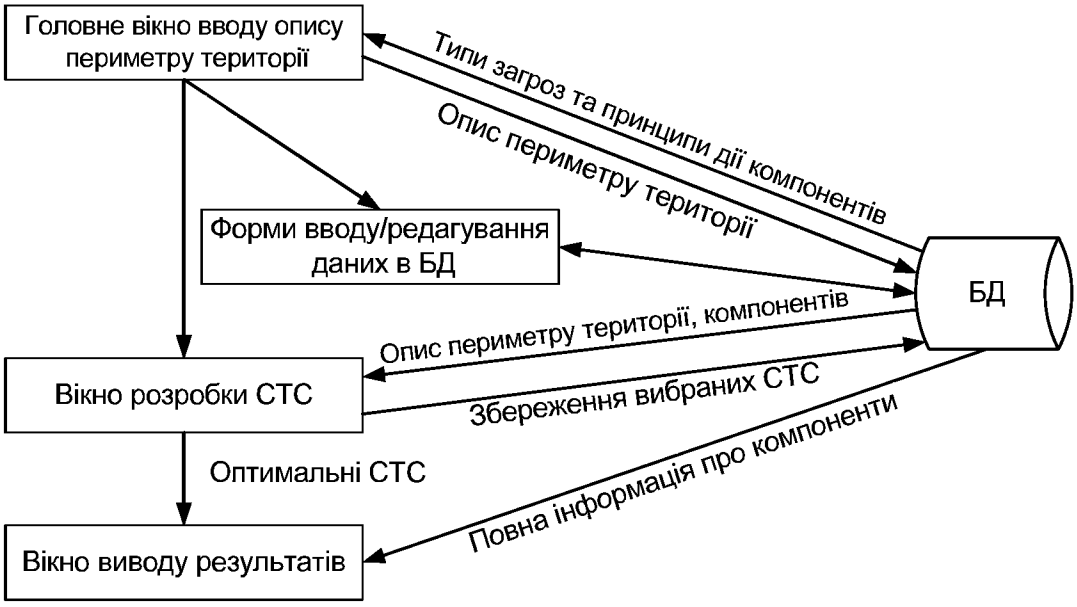


Рис. 2.13. Загальна структура комп'ютерної системи підтримки процесу розробки СТС

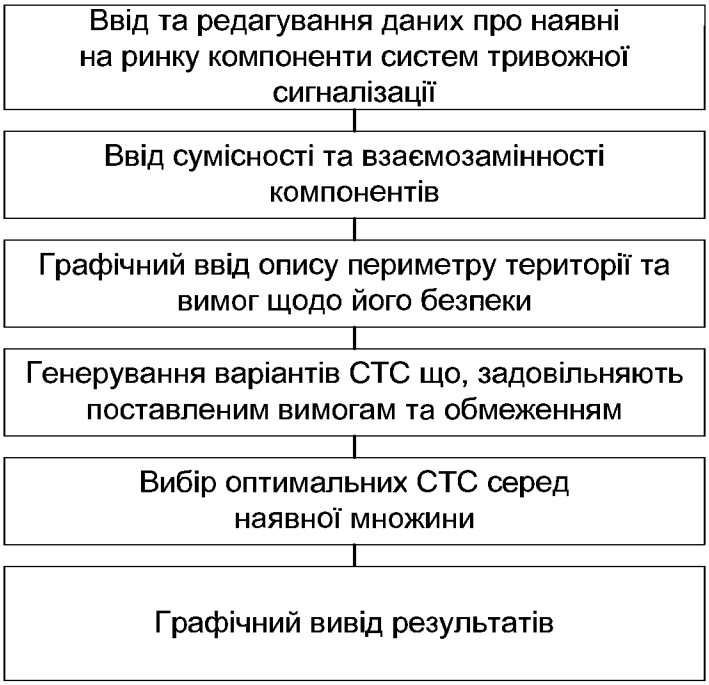


Рис. 2.14. Вимоги до процесу розробки СТС

Як видно з рис. 2.13, головне діалогове вікно вводу опису периметру території отримує з БД інформацію про компоненти і загрози, які вони можуть виявляти, і дозволяє ввести опис периметру території в БД. За його допомогою викликають форми вводу/редагування БД і вікно розробки СТС. Останнє отримує опис периметру з БД і наявні компоненти, використовує модулі для розробки та вибору оптимальних СТС і передає інформацію вікну виводу результатів.

Згідно пропонованої структури (рис. 2.13) і встановлених вимог (рис 2.14), розроблено комп'ютерну систему підтримки процесу розробки СТС [13, 87, 92] в середовищі Microsoft Visual Studio 2008, що дозволяє створювати СТС за 4 етапи. На етапі №1 (підготовчий), здійснюють ввід опису класів, принципів дії компонентів СТС, їх параметрів і функціонально-вартісних характеристик (обмежень, ефективності та ресурсоємності) згідно §2.1 та типів загроз в БД, описану в §2.3. Їх використовують у формах вводу значень характеристик моделей компонентів, встановлення їх сумісності та взаємозамінності. На рис. 2.15 схематично представлені форми та послідовність вводу даних в БД.

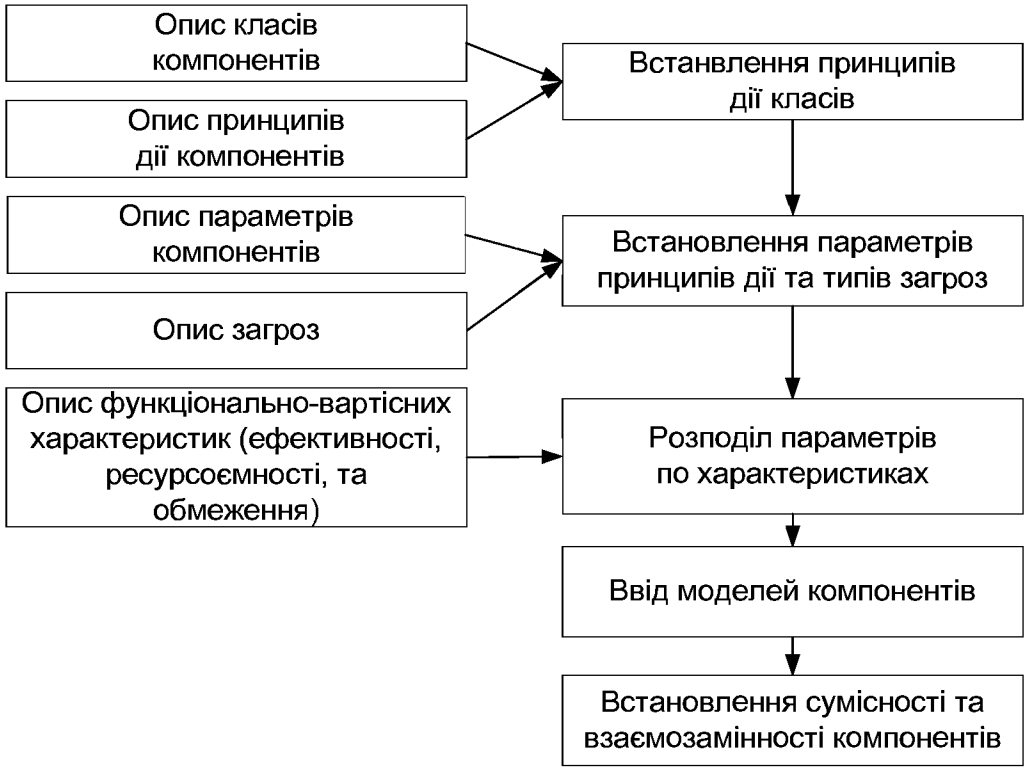


Рис. 2.15. Схематичне представлення форм та послідовності вводу даних в БД

Ввід моделі компонента в БД представлений на рис. 2.16, де в діалоговому вікні “Components” вибирають клас компонента (серед наперед визначених, наприклад, “Sensor”), вибирають принцип дії (знаходяться в списку визначених для даного класу принципів дії), вводять назву (тип) компонента та його опис. Після натиснення кнопки “ADD Component” опис вводиться в БД і активізується рамка вводу значень параметрів компонента, що містить список параметрів згідно вибраного раніше принципу дії компонента. Далі кожен параметр вибирають із списку, встановлюють відповідне значення та одиниці вимірювання. Після натиснення кнопки “ADD” компонент додається у список введених компонентів, який, після натиснення кнопки “OK”, запишеться в БД.

Далі встановлюється сумісність і взаємозамінність компонента з іншими за допомогою діалогового вікна рис. 2.17. В його лівій частині міститься інформація про компонент, а в правій можна вибрати клас компонентів, принцип дії та відповідно – компонент, з яким після натиснення кнопки “SET Compatibility” або “SET Exchangeability” буде встановлена сумісність або взаємозамінність.

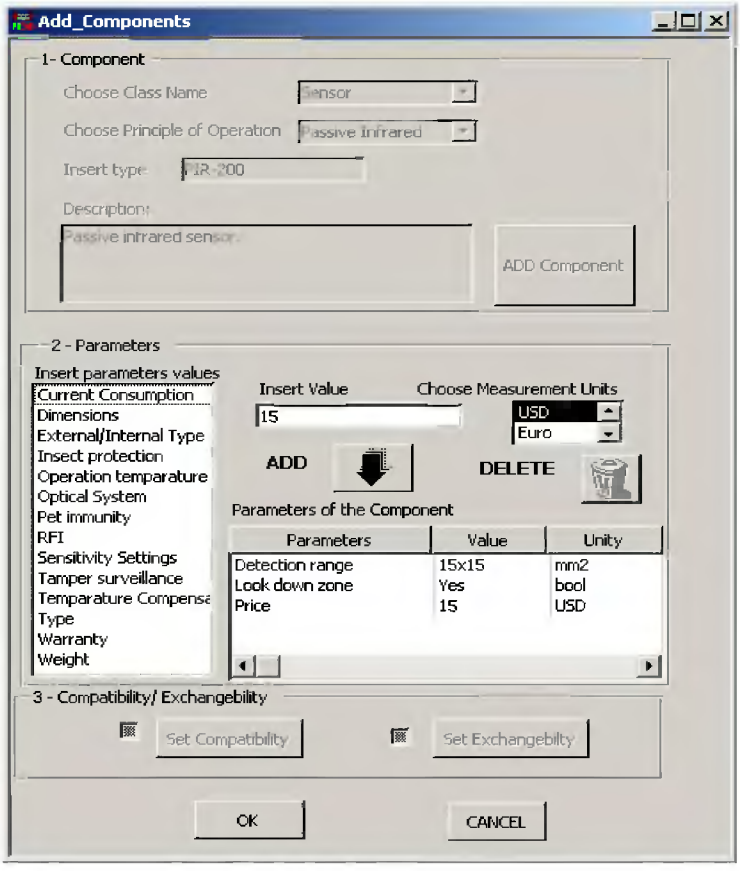


Рис. 2.16. Діалогове вікно вводу компонентів

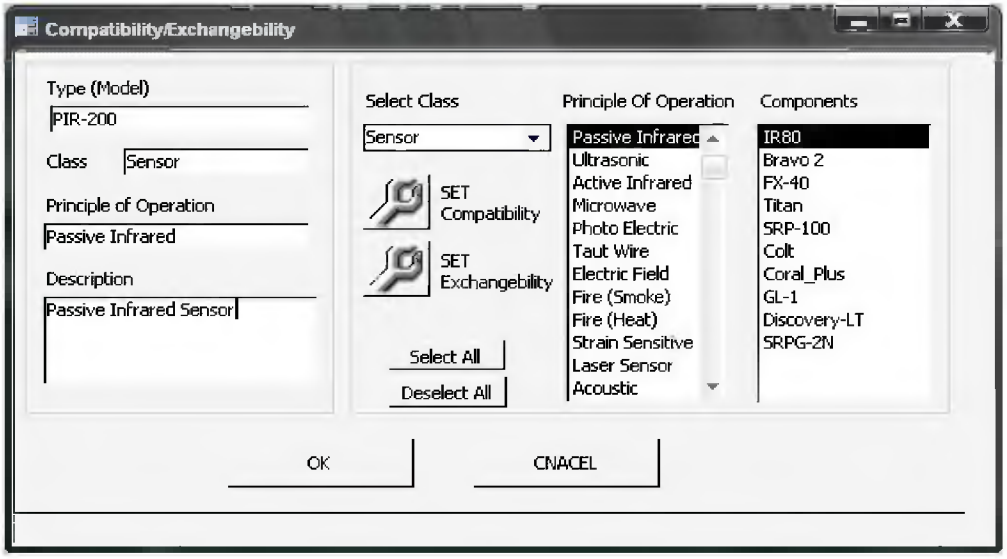


Рис. 2.17. Діалогове вікно вводу сумісності та взаємозамінності компонентів

Перевірити, відредагувати та встановити нові класи компонентів, їх принципи дії, параметри та загрози відповідно до їх класу, одиниці вимірювання параметру та критерії, до яких відносять даний параметр, можна за допомогою діалогового вікна редагування параметрів (рис. 2.18).

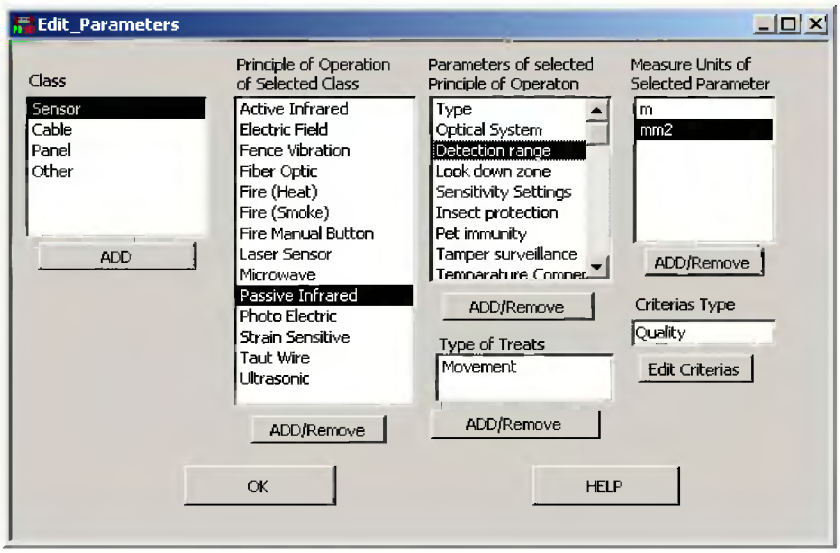


Рис. 2.18. Діалогове вікно редагування параметрів

На другому етапі за допомогою діалогового вікна рис. 2.19 вводять опис периметру території - завантажують її зображення 1, графічно розташовують зони периметру 2, встановлюють типи загроз ділянок 3, встановлюють принципи дії сповіщувачів, які будуть використовуватись для захисту території

4, масштаб 5 і назву об'єкту 6. При цьому можна вивести вже введений периметр території 7 або додати введений опис периметру в БД системи проектування 8 [87].

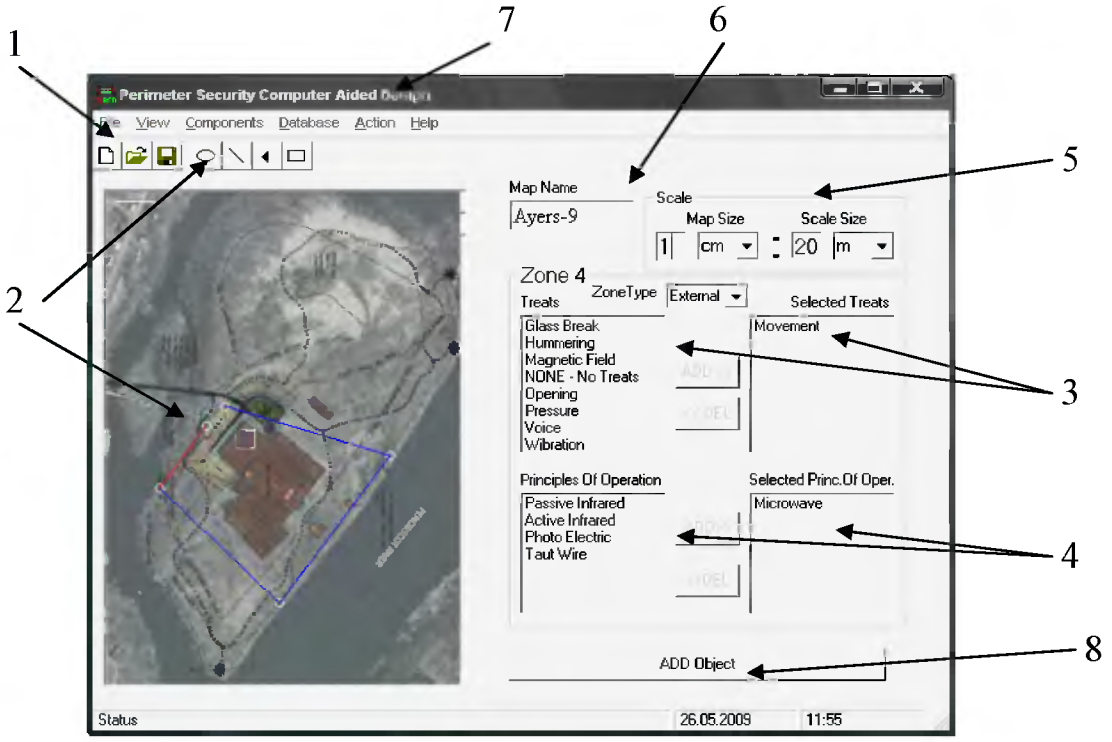


Рис. 2.19. Вікно вводу опису периметру

На третьому етапі розробляють СТС, що включає створення варіантів СТС, оптимізованих за функціонально-вартісними характеристиками одним із запропонованих в §2.2 алгоритмів, які можна вибрати у вікні проектування (рис. 2.20). Також можна вибрати введений периметр території 1 з відображенням відповідних ділянок та їх довжини, загрози і принципи дії компонентів, які мають забезпечити захист, а також метод розробки та оптимізації 10 (повний / неповний перебір, або ГА). Також вводиться обмеження 2 на кількість оптимальних варіантів ділянок при переборі. Виділивши один з варіантів 4, можна отримати інформацію про використані в зоні компоненти 5. Після розробки всього периметру СТС 6 виводиться список оптимальних СТС 7 з можливістю отримання інформації про склад виділеної системи 8 і графічне відображення оптимальних СТС 9 за їх функціонально-вартісними характеристиками (частоти хибних спрацьовувань "Quality",

імовірності невиявлення системою порушника “Reliability” та ресурсоємності придбання та монтажу системи “Cost”) (рис 2.21).

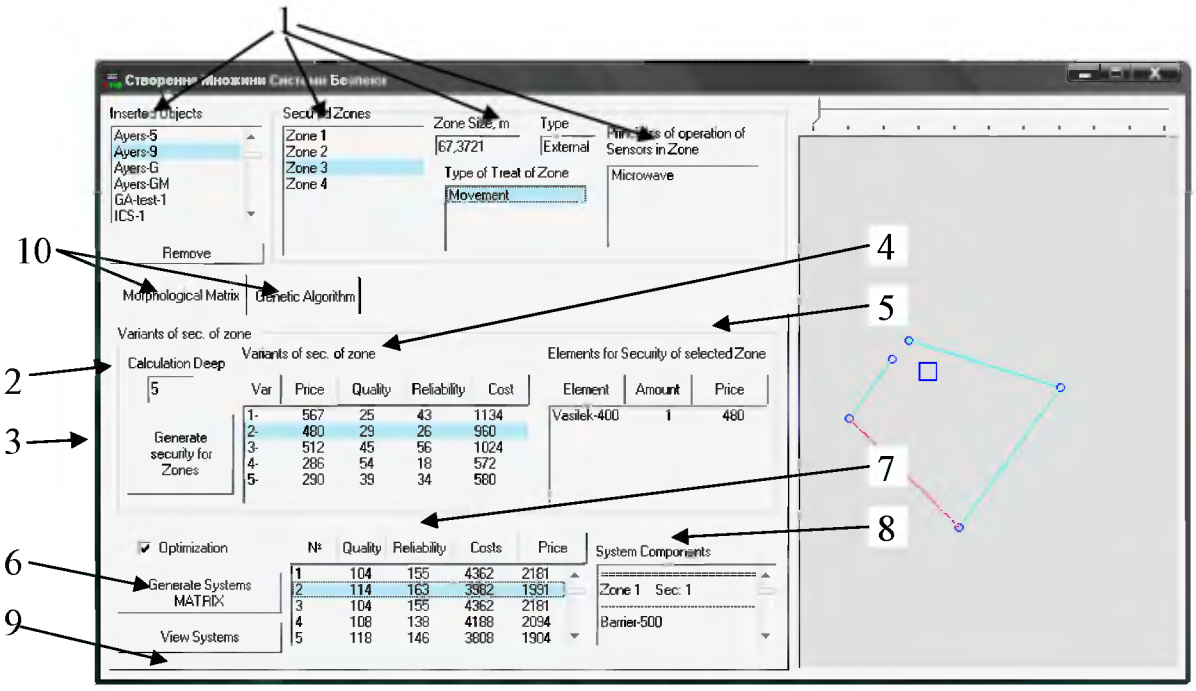


Рис. 2.20. Вікно розробки СТС

На четвертому етапі оцінюють оптимальні СТС на виведених вікнах (рис. 2.23), де окрема точка 1 координат Quality/Cost, Reliability/Cost означає окрему СТС, вибравши котру можна отримати інформацію про даний варіант СТС 2.

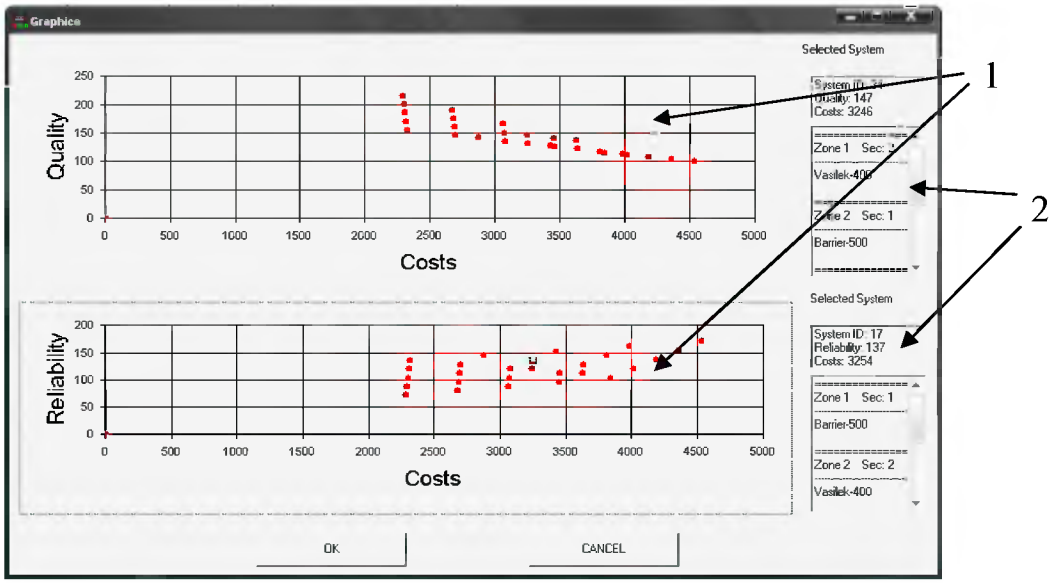


Рис. 2.21. Графічний вивід результатів

На рис. 2.22 представлено в графічному вигляді місце пропонованих методів та засобів в розробленій комп'ютерній системи підтримки процесу розробки СТС та взаємодію її програмних модулів між собою. Структура рис. 2.22 наочно представляє процес оптимізації СТС і засоби, що використовуються на кожно-му етапі розробки. Показано також місце запропонованих в результаті аналізу оптимізованих СТС і розроблених в розділі 3 мікроконтролерних вузлів, які дозволяють додатково покращити ресурсоємність оптимізованих СТС.

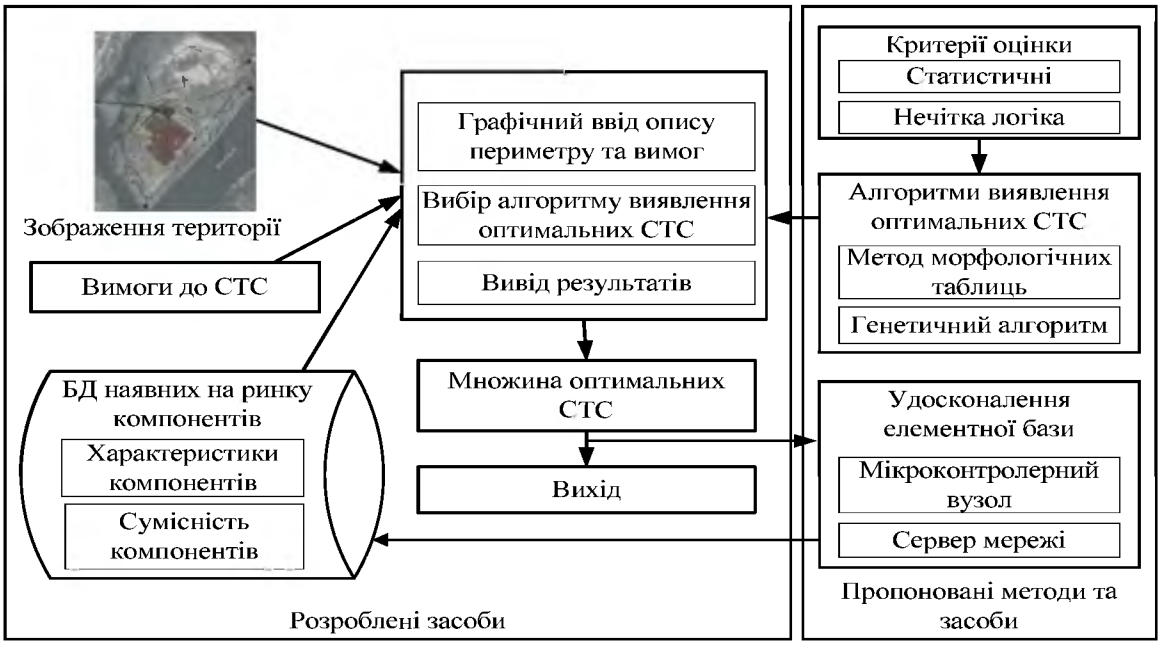


Рис. 2.22. Графічна інтерпретація методу розробки оптимізованих СТС

Отже, в § 2.4 оаисано створено комп'ютерну систему підтримки процесу розробки СТС згідно заданої структури та поставлених вимог, що дало змогу розробляти оптимальні за функціонально-вартісними характеристиками (частоти хибних спрацьовувань, імовірності невиявлення системою порушника та ресурсоємності придбання та монтажу системи) СТС, використовуючи запропоновані в § 2.2 методи знаходження оптимальних рішень та базу даних наявних компонентів (див. § 2.3).

Висновки до розділу 2

1. Обґрунтовано вибір критеріїв оцінки якості СТС та створено алгоритм обробки множини наявних технічних характеристик компонентів СТС, що дало змогу здійснити оцінку компонентів СТС за вибраними функціонально-вартісними характеристиками. Дана оцінка дасть змогу зменшити та деталізувати область пошуку оптимальних рішень під час розробки СТС.
2. Запропоновано набір методів знаходження оптимальних рішень, що включають методи повного і неповного перебору морфологічних матриць та генетичний алгоритм, які дозволили виявляти оптимальні рішення при розробці СТС.
3. Створено базу даних на основі створеної ER- моделі даних компонентів СТС, реалізованій за допомогою СУБД MS Access, що дозволило забезпечити гнучкий доступ та управління даними в процесі розробки СТС.
4. Створено комп'ютерну систему підтримки процесу розробки СТС згідно заданої структури та поставлених вимог, що дало змогу пропонувати замовнику набір Парето-оптимальних СТС за функціонально-вартісними характеристиками, використовуючи запропоновані автором методи знаходження оптимальних рішень та базу даних наявних компонентів.

РОЗДІЛ 3
ВДОСКОНАЛЕННЯ КОМПОНЕНТІВ КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ТРИВОЖНОЇ СИГНАЛІЗАЦІЇ

3.1. Аналіз результатів розробки комп'ютеризованих систем тривожної сигналізації

Досвід розробки різноманітних СТС периметру території з використанням запропонованої комп'ютерної системи підтримки розробки СТС дозволив виявити недолік, притаманний традиційним структурам СТС та КСТС. Він полягає в тому, що в таких системах відносно велику вагу мають затрати на кабель підключення сповіщувачів.

Наприклад, розглянемо СТС периметру прямокутної території розмірами 100×100 м (рис. 3.1). Нехай перший варіант такої СТС базується на традиційних сповіщувачах з радіусом дії 5 м (на кожну сторону території потрібно 10 сповіщувачів) та аналоговим інтерфейсом, які мають на виході нормально розімкнуті контакти по трьох каналах (наприклад, активного та пасивного інфрачервоних каналів, а також тампер – сенсор порушення цілісності корпусу сповіщувача). Таким чином, враховуючи необхідність живлення сповіщувача, для зв'язку використовуємо кабель, який містить чотири пари провідників, його ціна в Україні 28 центів США за погонний метр. При цьому приймально-контрольний пристрій розміщено в одному з кутків території, що охороняється (додатковими затратами кабелю на лінії зв'язку між місцем розміщення ПКП та периметром території нехтуємо).

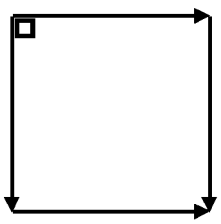


Рис. 3.1. Топологія СТС периметру прямокутної території

Загальну довжину кабелю такої СТС з топологією "зірка" можна оцінити методом сумування. Сумарна довжина кабелю L для підключення сповіщувачів сторони, яка примикає до ПКП, може бути визначена як сума членів прогресії

$$L = \frac{L\max + L\min}{2} K, \tag{3.1}$$

де $L\max$ і $L\min$ – довжина кабелів до найбільш і найменш віддалених сповіщувачів; $K = 10$ – кількість сповіщувачів, розміщених на одній стороні.

Згідно прийнятих розмірів $L = 500$ м. Для економії кабелю сповіщувачі під'єднуємо до ПКП з обох сторін периметру, як це показано стрілками на рис. 3.1. Тоді сумарна довжина кабелю для підключення сповіщувачів зі сторони, яка не примикає до ПКП, може бути визначена як

$$L1 = L + KLst, \tag{3.2}$$

де Lst – довжина сторони периметру території, що охороняється.

Згідно прийнятих розмірів $Lst = 100$ м, тоді $L1 = 1500$ м. Загальна довжина кабелю такої СТС з топологією "зірка" буде становити 4000 м, а ціна – 1120 доларів США. При цьому використані сповіщувачі мають ціну в Україні 27…30 доларів США, відповідно ціна повного комплекту сповіщувачів для такої СТС буде становити 1080…1200 доларів США. Як бачимо, ціна кабелю підключення сповіщувачів практично рівна ціні сповіщувачів, хоча функціональні задачі сповіщувачів і кабелю в складі СТС не співмірні – сповіщувач виявляє порушника, приймає рішення про подачу сигналу тривоги, формує сигнал тривоги, а кабель лише передає цей сигнал до ПКП.

При використанні мережевих сповіщувачів, наприклад, типу DSC, ціна яких складає 80…85 доларів США, ціна повного комплекту сповіщувачів для такої СТС буде становити 3200…3400 доларів США, тобто буде суттєво перевищувати сумарну ціну сповіщувачів і кабелю для традиційних систем – 2200…2320 доларів США (при цьому ще не врахована ціна кабелю для мережевих сповіщувачів).

Ціна сповіщувачів для СТС з безпровідним зв'язком буде ще вищою. Тому розроблена комп'ютерна система підтримки розробки СТС не видає, як оптимальні, СТС, які використовують мережеві та безпровідні сповіщувачі.

Можливим шляхом вдосконалення СТС є зменшення затрат на них за рахунок оснащення традиційних дешевих сповіщувачів спеціалізованим мережевим контролером, який буде виконувати роль "мережевої карти" для сповіщувача. При цьому доцільно забезпечити живлення сповіщувачів через мережу, подібно до того, як це виконано в інтерфейсі 1-Wire [41]. Однак інтерфейс 1-Wire є дуже малопотужний, може бути використаний тільки в мережах, де всі пристрої мають такий інтерфейс, при цьому довжина таких мереж не перевищує декількох метрів. Але відповідні схемотехнічні рішення дозволяють забезпечити живлення сповіщувачів через мережу.

При цьому мікроконтролер, що ввійде в склад спеціалізованого мережевого контролера такої КСТС, може мати малі обчислювальні ресурси. Адже його завданнями будуть тільки опитування станів аналогового виходу традиційного сповіщувача (його нормально замкнуті або розімкнуті вихідні контакти створюють логічні нулі або одиниці на входах мікроконтролера) та організація взаємодії з ПКП в складі мережі. Тому можна використати недорогий мікроконтролер, ціна якого в Україні, навіть без оптових знижок, не перевищуватиме 1,5 долара США. Тоді ціна мережевого контролера може не перевищувати 10 доларів США.

В такому випадку довжина кабелю для КСТС, представленої на рис. 3.1, буде не перевищувати 400 м (якщо від однієї вітки мережі можна живити до 20 сповіщувачів) або 600 м (якщо від однієї вітки мережі можна живити до 10 сповіщувачів). Таким чином, затрати на кабель будуть становити 112 або 168 доларів США, а сумарна ціна сповіщувачів і кабелю – 1192…1368 доларів США, що суттєво менше, ніж всі розглянуті вище варіанти. Слід відзначити, що оцінка затрат на кабель в останньому випадку зроблена для кабелю, який містить чотири пари провідників. Однак двопровідна мережа може

використовувати дешевший кабель, що додатково покращить вартісні характеристики пропонованої КСТС.

Таким чином, в даному параграфі, на основі аналізу затрат на лінії зв'язку СТС периметру території з топологією "зірка", обґрунтовано доцільність розробки спеціалізованого мережевого контролера на базі дешевого мікроконтролера, який дозволив би перейти до топології "спільна шина" і забезпечив би живлення сповіщувачів від мережі зв'язку.

3.2. Комп'ютеризована система тривожної сигналізації без підтримки захисту зв'язку

3.2.1. Інтерфейсний контролер сповіщувачів

У §3.1 було обґрунтовано доцільність розробки спеціалізованого мережевого контролера на базі недорогого мікроконтролера, який забезпечив би роботу сповіщувачів в мережі з топологією "спільна шина" і їх живлення від цієї ж мережі. Узагальнена структура такого спеціалізованого мережевого контролера UNC і його взаємозв'язки зі сповіщувачем SU представлена на рис. 3.2. Сповіщувач SU включає сенсори $S_1 \ldots Sn$, з'єднані із схемою аналогової обробки та прийняття рішень APCP, яка передає рішення (виявлення / не виявлення порушення) вихідній схемі сповіщувача OCSU. Мікроконтролер MC опитує виходи OCSU та взаємодіє з мережею, використовуючи апаратні засоби послідовного двопровідного інтерфейсу IF HW. Джерело живлення PS забезпечує живлення усіх елементів.

Принципово-структурна схема апаратних засобів послідовного двопровідного інтерфейсу IF HW і блока живлення PS розробленого мережевого контролера представлені на рис. 3.3, де блок живлення складається з вхідної розв'язки V1, C1 і стабілізатора живлення VR з вихідним конденсатором C2. Логічні нулі, які приходять по мережі на базі модифікованого послідовного інтерфейсу RS-232C, мають амплітуду +12 В і

заряджають конденсатор C2. Він підтримує напругу на вході стабілізатора VR достатньою, щоби на його виході формувалася напруга +5 В для живлення мікроконтролера MC і приймача інтерфейсу на транзисторі V2. Сам сповіщувач живиться напругою +12 В від конденсатора C1.

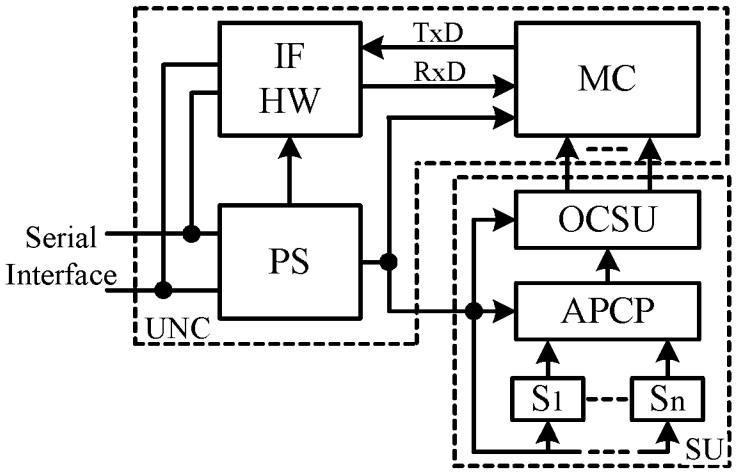


Рис. 3.2. Узагальнена структура спеціалізованого мережевого контролера і його взаємозв'язки зі сповіщувачем

Апаратні засоби послідовного інтерфейсу IF HW складаються з:

- приймача на транзисторі V2 і резисторах R1, R2, що представляє собою інвертор, вихідні сигнали якого узгоджені по напрузі з допустимим мікроконтролером MC рівнем;
- передавача на транзисторі V4, стабілітроні V3 і резисторах R3, R4, який представляє собою підсилювач-інвертор вихідних сигналів 0 і +5 В мікроконтролера MC до рівня, який відповідає сигналам мережі (+12 В і –12В відповідно). Надійне запирання транзистора V4 досягається наявністю стабілітрона V3 з напругою стабілізації, близькою до 8 В.

Важливою перевагою запропонованої схеми блоку живлення PS та апаратних засобів інтерфейсу IF HW є автоматичне відстеження амплітуди відповіді контролера за амплітудою імпульсів заряду конденсаторів C1 контролерів в мережі. Різниця амплітуд, яка відповідає спаду напруги на діоді V1 і транзисторі V4, не дозволяє вихідним імпульсам інтерфейсу контролера заряджати конденсатори C1 інших контролерів. В той же час, імпульси +12 В

звертання ПКП підзаряджають конденсатори С1 всіх контролерів в мережі незалежно від того, до якого контролера сервер звертається з запитом. Тому в мережі контролерів не виникає перерозподіл заряду і всі контролери, з точки зору живлення, працюють індивідуально.

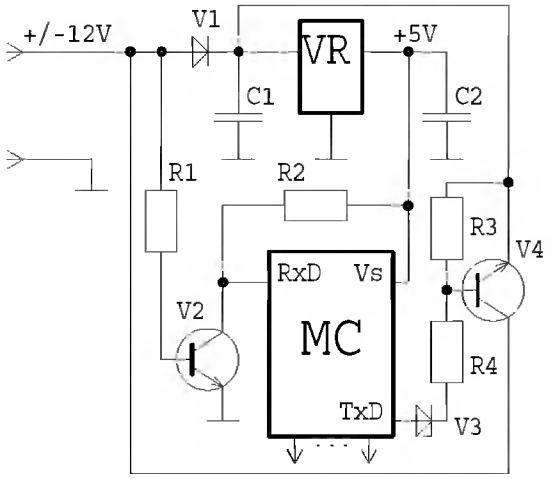


Рис. 3.3. Принципово-структурна схема блока живлення і апаратних засобів інтерфейсу контролера

Важливу роль в пропонованій мережі на базі модифікованого послідовного інтерфейсу RS-232C відіграє адаптер мережі, який підключається до ПКП. Його принципова схема представлена на рис. 3.4, основними вузлами адаптера є:

- блок живлення, який складається з трансформатора напруги мережі Т1, і двохполярного випрямляча на діодах V1, V2 і конденсаторах фільтрів С1, С2;
- двокаскадний підсилювач на транзисторах V3, V5 і резисторах R1…R3, який служить для формування потужних вихідних сигналів інтерфейсу RS-232C приймально-контрольного пристрою для заряду конденсаторів С1 універсального мережевого контролера (див. рис. 3.3);
- стабілізатор струму розряду лінії (встановлення логічної 1) на польовому транзисторі V6 і резисторі R5;
- схема захисту від коротких замикань на транзисторі V4 і резисторі R4;
- схема захисту від перенапружень і завад в лінії на симетричному стабілітроні V7 і резисторі R6.

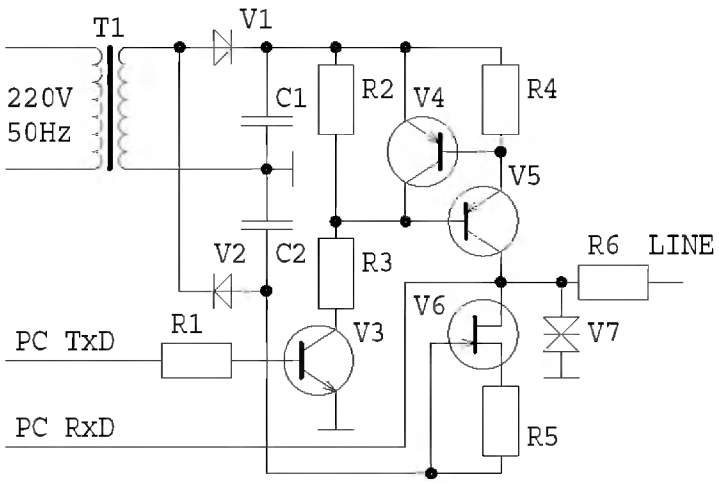


Рис. 3.4. Принципова схема адаптера мережі

Для оптимізації використання потужності живлення в якості МС використано мікроконтролер типу ATtiny2313 фірми Atmel [76], в якому оптимально для даного випадку поєднуються необхідні функціональні можливості, відносно мале споживання і низька ціна. Всі операції, які виконує МС, відповідають традиційним завданням мікроконтролерів. Для зменшення струму споживання МС понижено частоту кварцового резонатора – 4 МГц. Допустима кількість сповіщувачів в мережі – 64, з врахуванням того, що кожен сповіщувач може мати декілька чутливих елементів. Однак відносно великий струм споживання традиційними сповіщувачами не дозволяє включати у вітку мережі більше 16 сповіщувачів (при використанні найдешевшої лінії зв'язку – телефонного кабелю) через спад напруги у тонких провідниках лінії.

3.2.2. Приймально-контрольний прилад

Як ПКП (сервер), в такій мережі доцільно використати персональний комп'ютер ПК. В процесі функціонування мережі сервер почергово опитує сповіщувачі СТС, посилаючи відповідний запит з допомогою описаного адаптера мережі (див. рис. 3.4). Кожен логічний 0, який для інтерфейсу RS-232C відповідає напрузі +12 В, підзаряджає конденсатор С1 в блоці живлення мережевого контролера (див. рис. 3.3). Крім того, запит сервера, через приймач на транзисторі V2, поступає на мікроконтролер МС (див. рис. 3.3). МС опитує чутливі елементи $S_1 \ldots Sn$ (див. рис. 3.2), формує байт відповіді та передає її на

сервер, використовуючи передавач на транзисторі V4 (див. рис. 3.3).

В такій КСТС, як пристрій візуалізації, прийняття і виконання рішень щодо заходів безпеки (ввімкнення сирени, виклик охорони), використовується ПК. Його обчислювальні ресурси можуть бути мінімальні, можна використати старі ПК, що мають низьку ціну. Основною вимогою до нього є наявність COM-порта. Сучасні ПК і нотбуки не мають COM-портів, але можна використати перехідник USB-COM, який не збільшує суттєво вартість такої КСТС.

Оригінальними вузлами в розробленому мережевому контролері є IF HW і PS, які забезпечують функціонування контролера при живленні від мережі, а також адаптер мережі, який підключається до ПК. Розглянемо детальніше їх побудову. Для зменшення споживання енергії конденсатора C1 (див. рис. 3.3) в розробленому мережевому контролері вжиті наступні заходи:

- Передача запитів ПК здійснюється при швидкодії інтерфейсу RS-232C 1200 Бод, а передача повідомлень контролера здійснюється при швидкодії 9600 Бод. Це дозволяє збільшити відносний час заряду конденсатора C1 блока живлення контролера, але вимагає постійної зміни програмним шляхом налаштувань COM-портів комп'ютера та мікроконтролерів;
- ПК починає запит стартовим імпульсом (згідно протоколу інтерфейсу RS-232C лог. нуль, напруга якого відповідає +12 В), далі передається імпульс, що відповідає молодшому розряду байта. Тому, для покращення заряду конденсатора C1 блоку живлення (див. рис. 3.3), програмним шляхом перші два розряди встановлюють в нуль. Це обмежує кількість під'єднаних до одного адаптера контролерів мережі (тобто до одного COM-порту) до 64. Однак така кількість цілком прийнятна. Хоча спад напруги на опорі лінії не дозволяє підключити до одної двопровідної лінії більше 10-16 сповіщувачів, можна до виходу адаптера (див. рис. 3.4) підключити чотири вітки;
- Струм розряду лінії (визначається опором резистора R5, рис. 3.4) вибраний мінімальним для даної лінії і вибраної швидкодії інтерфейсу – 1,5 мА.

Таким чином, в даному параграфі розроблено спеціалізований мережевий контролер, який дозволяє перейти в КСТС до топології “спільна шина” і забезпечує живлення сповіщувачів від мережі зв’язку. Слід відзначити, що цей контролер не має захисту повідомлень в мережі від дій зловмисника. Зокрема, зловмисник може легко дослідити протоколи обміну та імітувати відповіді одного або декількох сповіщувачів. Тому такі контролери можна використати лише тоді, коли зловмисник не може мати доступу до мережі (наприклад, кабелі прокладено в металевих трубах, цілісність яких контролюється давачем тиску) або самі сповіщувачі такої КСТС охороняють свою лінію зв’язку (наприклад, кабелі прокладено в коридорі, який охороняється даною СТС).

3.3. Структура мережі із захистом зв’язку між сповіщувачами

Основними недоліками мережі сповіщувачів КСТС на базі контролерів, розроблених в §3.2, є відсутність захисту запитів ПКП і відповідей контролерів від імітації зловмисником та необхідність постійного формування ПК запитів (інакше припиниться живлення сповіщувачів). Остання вимога дуже навантажує ПК. Хоча швидкодії навіть застарілих ПК вистачає для постійного обміну даними, доцільно використати для реалізації ПКП мікроконтролер, який не буде мати жодних інших задач. Таке рішення також піднімає захищеність КСТС від блокування ПКП комп’ютерними вірусами загального поширення або спеціальних програм, спрямованих на порушення роботи даної КСТС.

На рис. 3.5 представлена узагальнена структура запропонованої КСТС, що складається з набору сповіщувачів Спов. 1 … Спов. n, виходи яких під’єднані до спільної двопровідної шини з допомогою мережевих контролерів МК1..МКn на базі мікроконтролерів [89]. Така структура теж, як і розроблена в §3.2 (див. рис. 3.2…3.4), скорочує довжину необхідного кабелю за рахунок заміни топології “зірка” на топологію “спільна шина” з розгалуженнями по 10…20 сповіщувачів на вітку. Функції сервера мережі, що підтримує її функціонування та безпеку, в структурі рис. 3.5 виконує мікроконтролер МКПКП. Він також

забезпечує функціонування сповіщувачів шляхом постійного посилання у мережу сигналу, що заряджає конденсатори, які живлять чутливі елементи сповіщувачів. Такий нестандартний інтерфейс доцільно реалізувати програмно. Таке рішення хоч і навантажує МКПКП, але воно не суперечить його функціональним обов'язкам, адже імпульси живлення сповіщувачів будуть формуватися саме при виконанні інтерфейсних функцій.

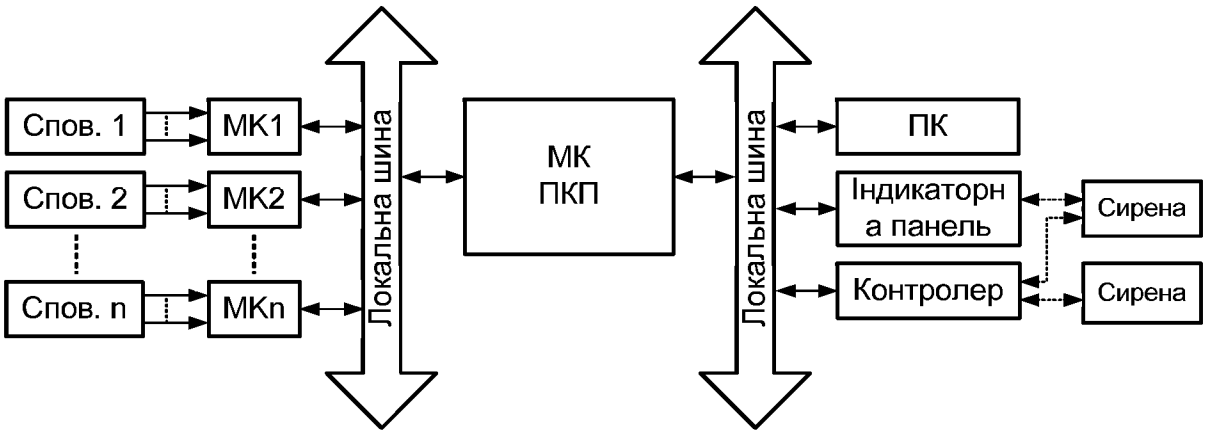


Рис. 3.5. Узагальнена структура пропонованої КСТС із захистом зв'язку

Вихід МКПКП доцільно реалізувати за допомогою вбудованого в МКПКП інтерфейсу RS232. Його завданням буде передача тривожного повідомлення на рівень охоронної системи (див. рис. 1.2), в центр прийняття рішень. Для універсальності (можливості використання в автономних СТС) доцільно передбачити можливість підключення до МКПКП стандартних елементів СТС (звукову сигналізацію, табло індикації зони порушення, пульт керування для постановки і зняття з охорони окремих сповіщувачів).

Для такої КСТС необхідний захист протоколу обміну зі сповіщувачами для виключення можливості імітації зловмисником відключених ним одного або декількох сповіщувачів мережі. При цьому слід відзначити, що інформаційний потік даних від сповіщувачів дуже малий. Стан кожного сенсора сповіщувача можна відобразити одним бітом. Тому сповіщувач, куди входить три-чотири сенсори (рідко зустрічаються складніші сповіщувачі) видає три-чотири біти. Такі короткі повідомлення неможливо зашифрувати. Тому, хоча слід збільшити довжину повідомлень, але використати відомі методи забезпечення захисту

повідомлень в комп'ютерних мережах недоцільно, бо: (і) для забезпечення високого ступеня криптостійкості слід різко збільшити обсяг повідомлень, що, в свою чергу, вимагає або збільшення швидкодії мережі (що встановлює більш жорсткі вимоги до типу і довжини кабелю), або збільшення часу реакції КСТС на дії порушника (що погіршує функціональні характеристики КСТС); (іі) забезпечення високого ступеня криптостійкості відомими методами вимагає застосування генераторів псевдовипадкових чисел високої якості, які важко реалізувати дешевим мікроконтролером через його невисоку обчислювальну потужність і малий об'єм оперативної пам'яті. Тому доцільна розробка спеціалізованого послідовного мережевого інтерфейсу, який забезпечить високу стійкість мережі до зламу, тобто до імітації сповіщувачів за рахунок відносно невеликого збільшення обсягу повідомлень та використання елементів захисту інформації, які властиві самому інтерфейсу.

Таким чином, в §3.3 розроблено структуру КСТС, де ПКП і контролери мережі (якими оснащені традиційні сповіщувачі) виконано на базі мікроконтролерів, а також обґрунтовано доцільність розробки спеціального послідовного мережевого інтерфейсу, який забезпечить виявлення заміни сповіщувачів імітаторами, навіть при наявності доступу зловмисника до лінії зв'язку.

3.4. Формування комплексу елементів захисту інформації для мережі сповіщувачів

Для забезпечення захисту мережі слід максимально утруднити можливість імітації зловмисником роботи одного або кількох сповіщувачів (або цілої мережі), для чого йому необхідно розшифрувати протокол обміну даними. Всі методи дешифрування ґрунтуються на аналізі окремих повідомлень, тому для захисту мережі слід: (і) приховати в мережі джерела сигналів, тобто запити ПКП і відповіді сповіщувачів повинні мати однакові параметри, щоби змусити зловмисника для ідентифікації кожного джерела виконати щомога більше операцій, коли СТС видасть сигнал тривоги; (іі) приховати структуру самих

запитів і повідомлень; (iii) досить часто змінювати параметри захисту в on-line режимі (з невідомим зловмиснику періодом декількох хвилин); (iv) не дати зловмиснику вивчати реакцію сповіщувачів на спроби штучно створених атак, навіть при легальному доступі зловмисника на територію, що знаходиться під охороною КСТС, і можливості безперешкодної дії на сенсори сповіщувачів.

Інтерфейси всіх мікроконтролерів стандартизовані [7, 114, 119], але вони не виконують поставлені вимоги. Як показав аналіз широко вживаних послідовних інтерфейсів, вони також не виконують поставлені вимоги, зокрема, RS232 і SPI не мережеві, RS485 і CAN вимагають дорогої витої пари, I^2C розрахований на малі відстані, LIN (а також всі попередні інтерфейси) не забезпечує живлення сповіщувачів [114, 119]. Найкращою базою був би інтерфейс 1-Wire, проте він забезпечує живлення лише спеціальних малопотужних пристроїв. Тому, для захисту зв'язку між компонентами мережі, розроблено аналогічний до 1-Wire [15, 17] спеціалізований інтерфейс з вбудованим захистом. Для максимального використання перелічених методів захисту, а також можливості гнучкої зміни як параметрів окремих елементів інтерфейсу, так і їх набору вибрана програмна реалізація інтерфейсу. Для його реалізації слід проаналізувати елементи послі-довних інтерфейсів, які могли б бути використані для захисту повідомлень.

Узагальнено послідовний інтерфейс можна представити, з точки зору зловмисника, як потік бітів в мережі, що ідуть один за одним в часі. При зламі необхідно: (i) розрізнити біти окремих пристроїв (приписати конкретні біти до відповідних сповіщувачів або ПКП); (ii) скласти окремі повідомлення (кожен біт встановити на своє місце); (iii) зуміти імітувати відповіді сповіщувачів, які не містять сигналу тривоги; (iv) зуміти імітувати роботу сповіщувача в мережі (правильно реагувати на запити ПКП); (v) зуміти розрізняти момент необхідної зміни параметрів інтерфейсу; (vi) зуміти міняти ці параметри потрібним чином.

Для захисту мережі слід створити зловмиснику труднощі на всіх етапах зламу. Як основу захисту, доцільно використати генератори псевдовипадкових чисел з рівномірним законом розподілу, які легко реалізувати програмно. При

цьому для окремих елементів захисту доцільно використати індивідуальні генератори, періоди повторення яких є взаємно простими.

Аналіз побудови та особливостей послідовних інтерфейсів показав, що елементи захисту можна поділити на три групи: 1) параметри імпульсів, що передаються по мережі; 2) варіант шифру при передачі повідомлень; 3) порядок видачі повідомлень. Кожна з цих груп включає цілий ряд елементів, які можна використати для створення труднощів зловмиснику. Перша група включає:

1. псевдовипадкову зміну частоти обміну даними;
2. псевдовипадкову зміну тривалості імпульсів;
3. псевдовипадкову зміну кількості бітів у запиті сервера;
4. псевдовипадкову зміну кількості бітів у відповіді сповіщувача;
5. псевдовипадкову зміну кількості інформаційних бітів у запиті сервера;
6. псевдовипадкову зміну кількості інформаційних бітів у відповіді сповіщувача;
7. псевдовипадкову зміну місця інформаційних бітів у запиті сервера;
8. псевдовипадкову зміну місця інформаційних бітів у відповіді сповіщувача;
9. стандартизацію амплітуди імпульсів.

Друга група включає:

1. псевдовипадкову зміну логічних номерів сповіщувачів;
2. багатоваріантність запиту сервера;
3. багатоваріантність відповіді сповіщувача;
4. введення відповідей, що не несуть інформації про стан сенсорів сповіщувача;
5. псевдовипадкове доповнення повідомлень при зміні кількості бітів в них;
6. заміну алгоритму шифрування інформації.

Третя група включає:

1. псевдовипадкову заміну почергового і групового опитування сповіщувачів;
2. псевдовипадкову заміну розміру груп;
3. псевдовипадкову заміну складу груп;
4. псевдовипадкове чергування окремих бітів повідомлень при відповіді групи.

Перелічені елементи захисту базуються на генераторах псевдовипадкових чисел з рівномірним законом розподілу, які повинні мати індивідуальні налаштування та взаємно прості періоди повторення. Ці налаштування і процедура їх заміни повинні встановлюватися тільки в процесі ініціалізації сповіщувачів при відлагодженні мережі уповноваженою особою, при цьому сповіщувач підключається безпосередньо до окремого виходу ПКП, доступ до якого захищено паролем. Це забезпечує індивідуальний характер захисту КСТС, незалежний від її виробника та фірми, яка встановлює СТС.

Таким чином, надійність захисту мережі сповіщувачів в різних умовах використання може бути забезпечена: (i) великою кількістю варіантів у всіх групах елементів захисту; (ii) використанням довільної кількості елементів захисту; (iii) псевдовипадковим характером реалізації переважної більшості елементів захисту; (iv) індивідуальними налаштуваннями елементів захисту; (iv) частою зміною налаштування елементів захисту; (v) встановленням параметрів захисту (конкретного набору елементів захисту для даної СТС, їх початкових налаштувань, зокрема, параметрів генераторів псевдовипадкових чисел) під час налаштування вже готової конкретної СТС; (vi) встановленням параметрів захисту тільки однією особою, уповноваженою замовником (користувачем) – ні фірма-виробник, ні фірма, що встановлює СТС, цих параметрів не повинні знати.

Крім того, доцільною є зміна параметрів елементів захисту при переході від режиму функціонування окремих сповіщувачів СТС “не під охороною” об’єкта (наприклад, під час обслуговування клієнтів у відповідному приміщенні банку) до режиму дійсної охорони. Коли традиційний сповіщувач перебуває в режимі “не під охороною”, то він все одно генерує сигнали тривоги по відповідних каналах, проте ПКП на них не реагує – тобто в мережі легко створити ситуації, які імітують спрацювання заданих сповіщувачів по заданих каналах (коли зловмисник має легальний доступ до ділянок території або приміщень, що охороняються) і збирати дані про роботу мережі з допомогою відповідного реєструючого пристрою (такий пристрій може не проявляти себе в

мережі, навіть бути безконтактним і його встановлення не можна виключити). Така ситуація, з точки зору криптографії, відповідає стану, при якому зловмисник знає частину повідомлень, що суттєво полегшує злам мережі.

Зміна параметрів елементів захисту при переході від одного до другого режиму роботи дезорієнтує зловмисника. В режимі роботи "не під охороною" можна передавати цілком випадкові повідомлення, не пов'язані з генерованими сповіщувачем сигналами тривоги, тоді з точки зору криптографії стан захисту мережі залишається таким, що зловмисник не знає ні одного повідомлення (хоча може думати, що їх знає). Остання особливість дозволяє створити пастку для зловмисника – під час роботи "не під охороною" використовувати достатньо прості засоби захисту, які не міняються, і які зловмисник відносно легко зламає, що спровокує його на активні дії та дозволить його знешкодити.

Для обчислення попередньої оцінки стійкості до зламу мережі КСТС обмежимось шістьма запропонованими елементами захисту групи параметрів імпульсів, що передаються по мережі: (iii) зміну кількості бітів у запиті сервера; (iv) зміну кількості бітів у відповіді сповіщувача; (v) зміну кількості інформаційних бітів у запиті сервера; (vi) зміну кількості інформаційних бітів у відповіді сповіщувача; (vii) зміну місця інформаційних бітів у запиті сервера; (viii) зміну місця інформаційних бітів у відповіді сповіщувача;.

Ця попередня оцінка показує верхню межу імовірності зламу протоколу і у реальній системі може бути нижчою за рахунок використання більшої кількості елементів захисту.

Нехай $n_{servbits}$ – кількість інформаційних бітів у запиті сервера; $n_{detecbits}$ – кількість інформаційних бітів у відповіді сповіщувача. Розрядність повідомлень сервера та сповіщувача однакова і дорівнює $n_{totalbits.}$

Імовірність виявлення зловмисником кількості та розміщення інформаційних бітів у повідомленні сервера чи сповіщувача обчислюється як обернена величина до біноміального відношення розрядності повідомлення та кількості інформаційних бітів у ньому:

$$p_{\text{hack}}(n_{\text{infobits}}) = 1/C_{n_{\text{infobits}}}^{n_{totalbits}} = \frac{n_{\text{infobits}}!(n_{totalbits} - n_{\text{infobits}})!}{n_{totalbits}!}, \quad (3.3)$$

де $p_{\text{hack}}(n_{\text{infobits}})$ - функція від кількості інформаційних біт n_{infobits}, що обчислює імовірність виявлення зловмисником їх місцезнаходження у повідомленні сервера або сповіщувача; $n_{totalbits}$ – розрядність повідомлень сервера та сповіщувача; n_{infobits} – кількість інформаційних бітів у повідомленні сервера чи сповіщувача; $C_{n_{\text{infobits}}}^{n_{totalbits}}$ - біноміальний коефіцієнт із $n_{totalbits}$ по n_{infobits} елементів.

Оскільки мінімальна довжина повідомлення з відповіддю сповіщувача дорівнює одному розряду (є або нема тривоги), а максимальна – задекларованій розрядності повідомлень сервера та сповіщувача – $n_{\text{totalbits}}$, то імовірність виявлення зловмисником інформаційних бітів у відповіді сповіщувача при зміні їхньої кількості і місцезнаходження обчислюється з формули

$$p_{\text{hack\_detect}} = \prod_{n_{\text{infobits}}=1}^{n_{\text{totalbits}}} p_{\text{hack}}(n_{\text{infobits}}) \quad (3.4)$$

За аналогією, імовірність виявлення зловмисником інформаційних розрядів у запиті сервера при зміні їхньої кількості і місцезнаходження обчислюэться за формулою

$$p_{\text{hack\_serv}} = \prod_{n_{\text{infobits}}=n_{\text{minservbits}}}^{n_{\text{totalbits}}} p_{\text{hack}}(n_{\text{infobits}}), \quad (3.5)$$

де $n_{\text{minservbits}}$ – мінімальна кількість розрядів, що необхідні для формування запиту сервера. Ця величина повинна бути достатньою для кодування індексів сповіщувачів у СТС, тобто тим більша, чим більшу кількість сповіщувачів дозволяє розмістити ПКП на одній шині. Наприклад, при $n_{\text{minservbits}}$=3 ПКП може опитувати СТС, що налічує до 8 сповіщувачів (2^3), при $n_{\text{minservbits}}$=6 ПКП може опитувати СТС, що налічує до 64 сповіщувачів (2^6).

У випадку охорони периметру об'єкта на острові Ayers (див. Додаток Б) потрібна досить масштабна СТС, тому встановлено $n_{\text{minservbits}}$=6, що дозволяє будувати системи, що налічують до 64 сповіщувачів. Для довжини повідомлень сервера та сповіщувача достатньо семи розрядів ($n_{totalbits}$=7) (наприклад, як американський стандартний код обміну повідомленнями ASCII, що широко використовується у більшості мережевих протоколів прикладного рівня: POP3, HTTP і т.д.).

Таким чином, підставляючи у формули (3.3)–(3.5) $n_{\text{minservbits}}$=6, $n_{totalbits}$=7 отримаємо, що імовірність імітації сповіщувачів (зламу СТС) не більша за

$$p_{\text{hack\_sys}} = p_{\text{hack\_detect}} \cdot p_{\text{hackserv}} \cdot 100\% = 5.3967 \cdot 10^{-7} \approx 10^{-6}\%.$$

Отже, значення імовірності зламу СТС становить не більше 10^{-6}%, оскільки її стійкість до зламу додатково покращується й іншими, описаними у третьому розділі дисертації елементами захисту відкритих ліній зв'язку.

Таким чином, в § 3.4 розроблено та реалізовано засоби захисту мережі сповіщувачів шляхом використання 19 елементів захисту (здебільшого на базі генераторів псевдовипадкових чисел). Це дало змогу забезпечити високий рівень захисту обміну даними в мережі.

3.5. Інтерфейсний контролер сповіщувачів з підтримкою захисту зв'язку

Всі методи дешифрування ґрунтуються на аналізі окремих повідомлень, тому в §3.4 першою стоїть вимога приховати джерела сигналів в мережі сповіщувачів. Дана умова не виконана в мережі, розробленій в § 3.2 [10, 88]. Для її реалізації запропоновано підхід, що базується на розділенні імпульсів живлення сповіщувачів та імпульсів повідомлення [89].

Часова діаграма імпульсів в мережі сповіщувачів представлена на рис 3.6. Хоча імпульси живлення сповіщувачів явно виділяються з потоку бітів в мережі, біти запитів ПКП і всіх сповіщувачів мають однакові параметри. Тому

зловмиснику важко розрізнити (без змоги спеціального ввімкнення резисторів у різних місцях мережі та багатоканального зняття даних про амплітуду імпульсів, що вимагає тривалого вільного доступу до провідників мережі), які з імпульсів є командами ПКП, а які – відповідями сповіщувачів, а також який конкретно сповіщувач згенерував даний імпульс. Наявність імпульсів живлення дозволяє також додатково знизити ресурсоємність СТС за рахунок відмови від використання дорогих кварцових резонаторів, як елементів, що з достатньою точністю задають частоту тактових імпульсів мікроконтролерів сповіщувачів. Якщо синхронізувати таймери сповіщувачів по фронту імпульсу живлення, то, при використанні мікроконтролерів AVR, можна обійтися внутрішнім RC генератором. Його похибка не перевищує 3%, тому похибка формування 10-го імпульсу в серії не перевищить 0,6 мс, що, при довжині імпульсу 2 мс цілком допустимо (якщо ПКП буде опитувати імпульси відповіді сповіщувачів в моменти, які входять в інтервал від $((2n+1)-0{,}3)$ мс до $((2n+1)+0{,}3)$ мс, де $n=0...9$ – номер імпульсу, який приймає ПКП, то збоїв через відхилення частот тактових генераторів сповіщувачів не буде). При цьому тривалість імпульсів повідомлень сповіщувачів (при псевдовипадковій зміні тривалості - п. 3 першої групи елементів захисту) не можна використати як елемент ідентифікації сповіщувача, і використання RC генератора не знижує захищеність мережі.

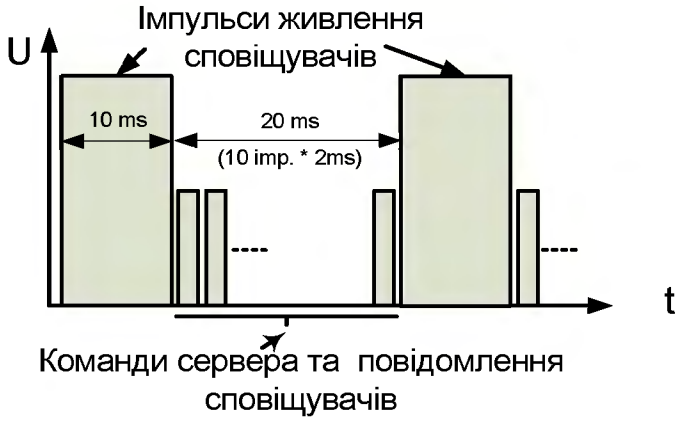


Рис. 3.6. Часова діаграма імпульсів в мережі сповіщувачів

Амплітуда імпульсів запитів ПКП і повідомлень сповіщувачів вибрана

близькою до 12 В, відповідно до напруги акумулятора резервного живлення. Амплітуду імпульсів живлення сповіщувачів доцільно вибрати подвійною до резервного живлення, для того, щоб їх генерувати з допомогою накопичувача, в якому конденсатор на час повідомлень сповіщувачів з допомогою MOS-ключів підключається до живлення ПКП, а на час генерування імпульсів живлення сповіщувачів з допомогою інших потужних MOS-ключів вмикається в мережу послідовно з напругою живлення ПКП. Тоді максимальна кількість сповіщувачів у вітці мережі визначається допустимим вихідним струмом, який видає ПКП. Дешеві MOS-ключі дозволяють підключати до 20 сповіщувачів із струмом живлення 30 мА на одну вітку, що задовільняє більшість споживачів.

В режимі групового опитування сповіщувачів при чергуванні окремих бітів сповіщувачів під час передачі повідомлень групи, порядок появи бітів в мережі вказано в таблиці 3.1.

Таблиця 3.1

Порядок появи бітів в мережі

Інтер-вал	Приймально-контрольний прилад	Сенсор 1	Сенсор 2	* * *	Сенсор n
1	Sync				
2	Cycl				
3	Sync				
4	Rec1				
5	Sync				
6	Rec2				
7	Sync				
8			Ans11		
9		Ans12			
* * *					
11					Ans1n
12	Sync				
13			Ans21		
14		Ans22			
* * *					
16					Ans2n
17	Sync				

При цьому прийняті позначення: 1) Sync – синхронізація таймера по фронту імпульсів живлення; 2) Cycl – початок циклу опитування сенсорів; 3)

Rec1 – перша половина запиту ПКП; 4) Rec2 – друга половина запиту ПКП; 5) Ans 11…Ans 1n – біти першої половини відповіді сповіщувачів; 6) Ans 21…Ans 2n – біти другої половини відповіді сповіщувачів.

Видача окремих бітів повідомлення в мережу сповіщувачем, який входить в групу, визначається однією з таблиць порядку відповіді, які записуються в пам'ять мікроконтролерів інтерфейсних контролерів при ініціалізації мережі. Вони несуперечливі (в один момент може відповідати лише один сповіщувач) та періодично міняються в групі згідно команди, що міститься в запиті ПКП.

Розроблена структурна схема інтерфейсного контролера сповіщувачів із захистом зв'язку представлена на рис. 3.7. В неї входять стабілізатори СТ1 і СТ2, мікроконтролер типу ATtiny2313 (містить вбудований компаратор), і вузли приймача і передавача [12]. Мікроконтролер опитує виходи традиційного сповіщувача та передає в мережу свою відповідь після поступлення запиту від ПКП, який містить адресу сповіщувача, встановлену під час його ініціалізації при налаштуванні мережі. Вузли приймача і передавача узгоджують рівні сигналів логічних нулів і одиниць мережі та мікроконтролера.

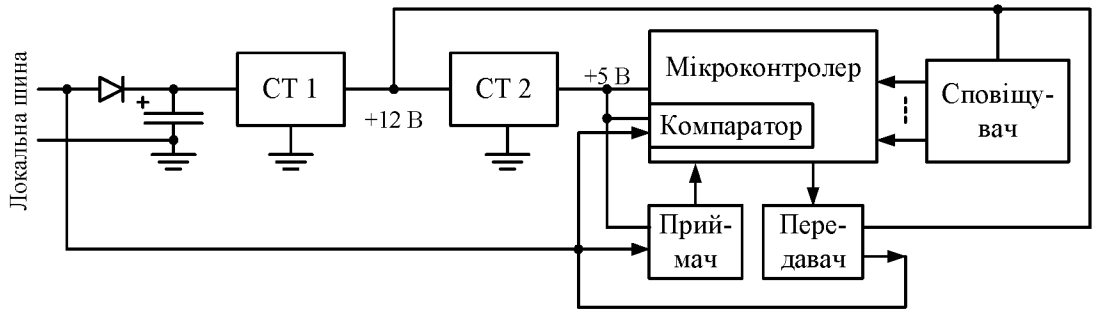


Рис. 3.7. Структурна схема сповіщувача із підтримкою захисту зв'язку

Основною особливістю схеми рис. 3.7 є система живлення. В її склад входять діод і конденсатор, які формують постійну напругу живлення шляхом періодичного підзаряду конденсатора імпульсами від ПКП (відповідно до рис. 3.6). Сповіщувач живиться нормальною для нього напругою +12 В з виходу стабілізатора СТ1. Ця ж напруга поступає на передавач сповіщувача і визначає амплітуду його вихідних імпульсів. Конденсатор живлення заряджений до напруги, вищої за напругу виходу передавача, і діод залишається закритим під

час передачі повідомлень сповіщувача. Таким чином, навантаження на мережу різко падає і спад напруги на проводах між сповіщувачем і сервером стає меншим розкиду напруги стабілізації СТ1. Це не дозволить визначити джерело повідомлення через аналіз амплітуди імпульсів. Мікроконтролер і приймач живляться напругою +5В від СТ2, яка служить і для формування опорної напруги компаратора, що виділяє імпульси живлення сповіщувачів для його синхронізації та виключення сприйняття їх як запиту ПКП.

На рис. 3.8 приведена принципова схема інтерфейсного контролера сповіщувачів із захистом зв'язку. В неї входять мікроконтролер D1 ATtiny2313, конденсатор скидання С3, блок живлення (V1, ST1, ST2, C2, C4, C5), приймач (V2, R1, R2, C1), передавач (V3, V4, R5, R8, R9), подільники амплітуди імпульсів мережі (R3, R4) та опорної напруги (R6, R7), а також захист від короткого замикання мережі (R10, R11). Сумарна ціна деталей інтерфейсного контролера сповіщувачів із підтримкою захисту зв'язку не перевищує 40 грн. Через використання малогабаритних деталей контролер розміщається в корпусі сповіщувача, йому не потрібен власний корпус, а сенсор захисту сповіщувача від порушення його цілісності (тампер) захищає також і контролер.

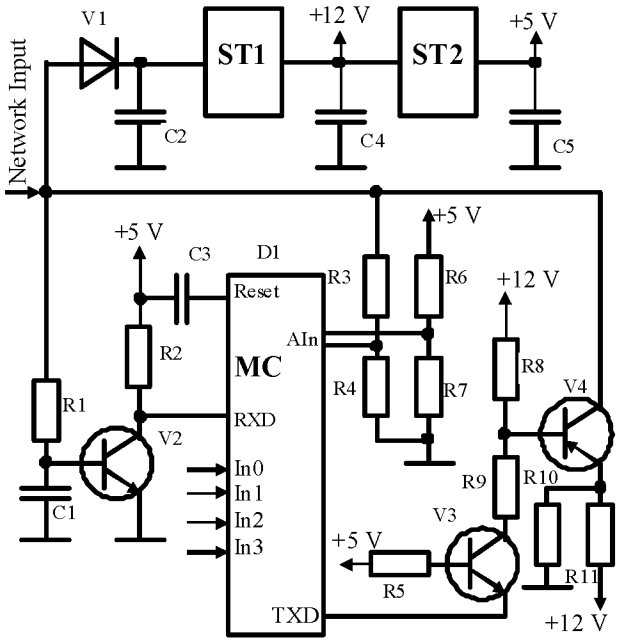


Рис. 3.8. Принципова схема інтерфейсного контролера сповіщувачів із підтримкою захисту зв'язку

Через те, що напруга імпульсів живлення в мережі рис. 3.8 вдвічі вища, ніж в схемі рис. 3.3, спад напруги на лінії не впливає на роботу сповіщувачів. Тому до однієї вітки мережі можна підключити до 20 сповіщувачів. Крім того, можна зменшити ємність (і ціну) конденсатора С2 (див. рис. 3.9) через те, що напруга пульсацій на ньому може бути суттєво вища.

3.6. Програмне забезпечення сповіщувачів мережі з підтримкою захисту зв'язку

Програма роботи мікроконтролерів сповіщувачів складається з чотирьох підпрограм – головної, обробки запитів переривань компаратора, приймання запитів ПКП та передачі повідомлень контролера. Головна підпрограма, після початкового налаштування мікроконтролера, реалізує тільки цикл очікування запитів переривань від компаратора. Алгоритм обробки запитів переривань компаратора подано на рис. 3.9. При його виконанні таймер Т0 синхронізується за фронтом імпульсу живлення (який розпізнається компаратором), генератори псевдовипадкових чисел переходять в наступний стан, опитуються сенсори і лічильник переривань компаратора. За станом останнього проходить прийом запиту ПКП (після запиту до даного сповіщувача готується його відповідь – шифрується стан сенсорів) або передача свого повідомлення (якщо перед тим був запит ПКП до даного сповіщувача). Для уникнення розсинхронізації ПКП що декілька хвилин розсилає запит "всі одиниці", за яким лічильник переривань компаратора примусово онулюється.

На рис. 3.10 представлено алгоритм приймання запитів ПКП, який зводиться до періодичного опитування виходу приймача згідно переривань попередньо налаштованого таймера Т0 та складання отриманих бітів в код.

На рис. 3.11 представлено алгоритм передачі повідомлень контролера, що зводиться до періодичної зміни стану виходу передавача згідно підготовленої відповіді за перериваннями попередньо налаштованого таймера Т0, прийому

переданих бітів, аналогічно попередньому алгоритму і контролю правильності передачі шляхом порівняння переданого і прийнятого повідомлень.

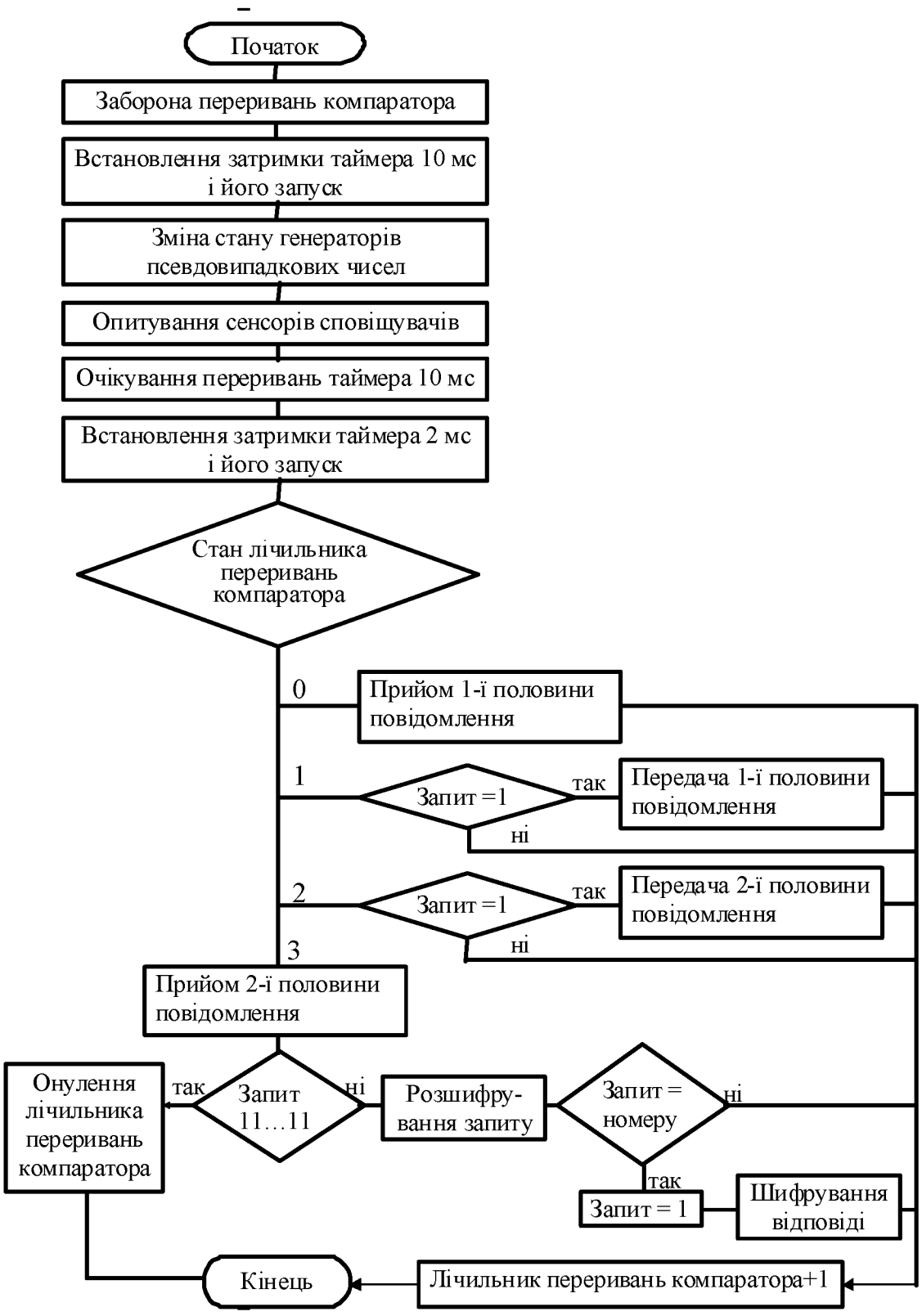


Рис. 3.9. Алгоритм обробки запитів переривань компаратора

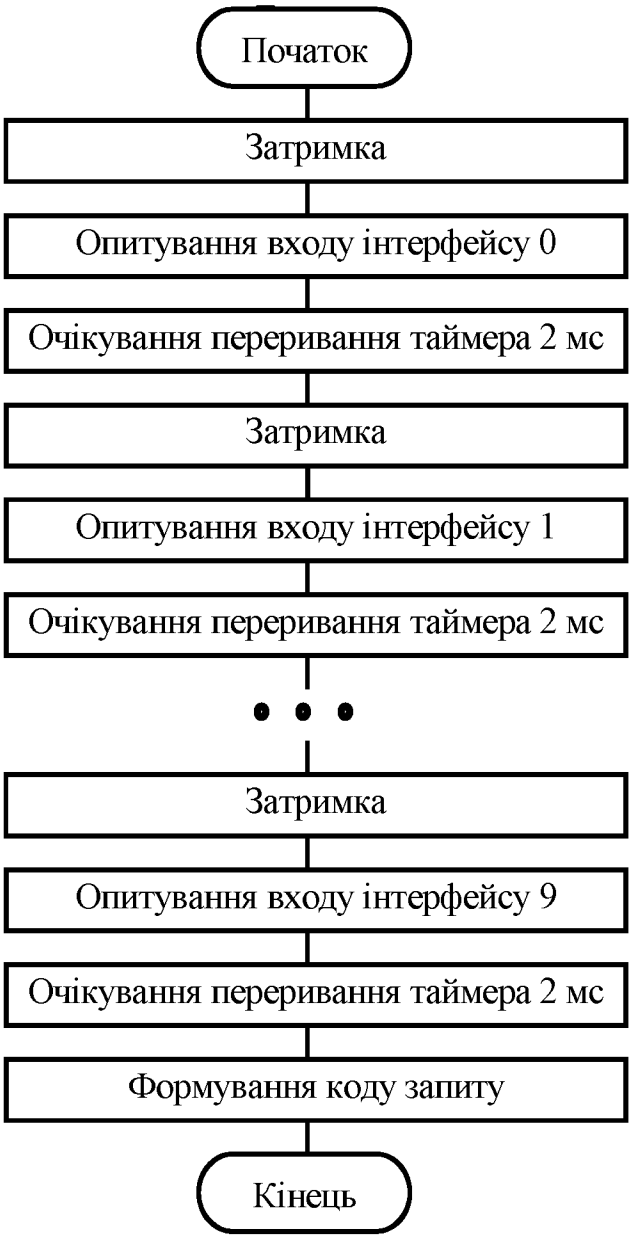


Рис. 3.10. Алгоритм приймання запитів приймально-контрольного приладу

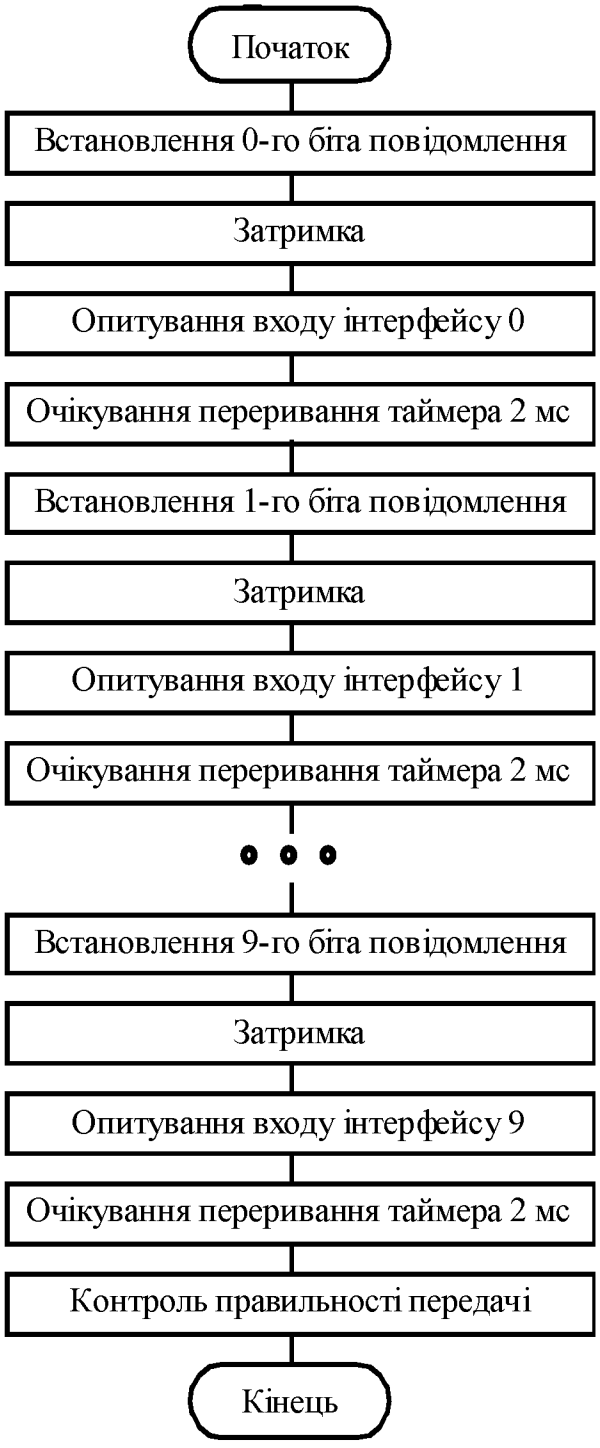


Рис. 3.11. Алгоритм передачі повідомлення контролера

Важливою є процедура ініціалізації мережі, метою якої є узгодження дії всіх мікроконтролерів в складі мережі. При її виконанні кожному МК1…МКn задаються: (і) індивідуальні номери в мережі (щоб не користуватися довгими заводськими номерами, що зменшує захищеність мережі); (іі) індивідуальні

початкові налаштування генераторів псевдовипадкових чисел; (iii) набір індивідуальних кодових таблиць, згідно яких кожен МК1…МКn повинен визначати номери бітів, які, при роботі в групі, повинен формувати саме він. Також під час ініціалізації записуються біти захисту пам'яті програм МК1…МКn.

Як видно, при ініціалізації на МК1…МКn поступають секретні дані. Тому доступ до процедури ініціалізації повинен бути строго обмежений. Для цього в пропонованій мережі використано наступні заходи: (i) алгоритми формування індивідуальних номерів МК1…МКn, параметрів генераторів псевдовипадкових чисел і кодових таблиць містяться в МКС і не можуть бути зчитані; (ii) перед початком ініціалізації проводиться аутентифікація адміністратора мережі; (iii) дійсний набір елементів захисту для даної мережі вибирається адміністратором мережі на основі загального списку; (iv) ініціалізація сповіщувачів проводиться окремим портом сервера, доступ до якого захищено паролем і пломбою адміністратора мережі; (v) процедура ініціалізації проводиться для всіх сповіщувачів мережі послідовно, без переривань; (vi) розширення мережі або її модернізація вимагає проведення процедури ініціалізації заново.

Останній пункт вимог забезпечує максимальний захист від можливості заміни сповіщувача імітатором. Однак він не вигідний, коли конфігурація мережі часто змінюється. Тому можна виділити спеціальний переносний блок-ініціалізатор, який, під час розширення мережі або її модернізації проводить ініціалізацію всіх сповіщувачів на місці експлуатації, а також ПКП. Такий блок адміністратор зберігає в сейфі, доступ до нього додатково обмежено паролем.

Можна також провести ініціалізацію запасних контролерів, які повинні зберігатися в корпусі ПКП (захищені його тампером) ввімкненими в мережу. Тоді їх крадіжка гарантовано буде виявлена мережею (як злам ПКП і відсутність відповіді на запит ПКП). При необхідності адміністратор підключає такий контролер до дійсного сповіщувача на місці його експлуатації, ігноруючи тривожні повідомлення за час цього підключення.

Отже, запропоноване програмне забезпечення інтерфейсних контролерів реалізує підтримку пропонованого протоколу обміну даними в мережі сповіщувачів і може бути реалізоване дешевими 8-ми бітними мікроконтролерами.

Висновки до розділу 3

1. Обґрунтовано використання мережевого контролера, який може суттєво зменшити довжину кабелю СТС, а значить, і його ціну за рахунок переходу від топології "зірка" до топології "спільна шина" та живлення сповіщувачів від мережі.
2. Розроблено структуру мережі без захисту повідомлень та її компоненти, які, за рахунок оснащення традиційних сповіщувачів контролером мережі на базі мікроконтролера та використання комп'ютера як приймально-контрольного приладу, забезпечують значне зниження ресурсоємності СТС.
3. Шляхом аналізу особливостей послідовних інтерфейсів виявлено максимальну кількість їх елементів (19), які використано для захисту повідомлень в мережі, що забезпечує надійний захист мережі від імітації сповіщувачів.
4. Розроблено структуру мережі, яка використовує виявлені елементи захисту повідомлень, розроблено структурну та принципову схеми інтерфейсного контролера.
5. Розроблено алгоритми роботи інтерфейсного контролера, які забезпечують його функціонування в мережі із захистом повідомлень.
6. Розроблено методику забезпечення максимального захисту, за рахунок процедури ініціалізації інтерфейсних контролерів під час налаштування СТС, диференціації захисту в режимах "під охороною" та "не під охороною", та створення пасток для зловмисника.

РОЗДІЛ 4
ВПРОВАДЖЕНІ КОМП'ЮТЕРИЗОВАНІ СИСТЕМИ ТРИВОЖНОЇ СИГНАЛІЗАЦІЇ

4.1. Комп'ютеризована система тривожної сигналізації периметру території острова Ayers

В рамках спільного американсько-українського науково-дослідного проекту "Проектування дистрибутивної сенсорної мережі для безпеки Ayers Island з використанням технології функціонально-вартісного аналізу", грант № CRDF FSTM UM2-5012-TE-03 (2003-2005 рр.), котрий виконувався згідно програми Фонду цивільних досліджень і розвитку США "Перші кроки до ринку" факультетом комп'ютерних інформаційних технологій, ТНЕУ спільно з корпорацією Trefoil, штат Мейн, США, автором розроблено:

- базу даних компонентів СТС (див. § 2.3) [90, 126];
- процедуру обробки множини наявних характеристик ефективності та ресурсоємності компонентів СТС (див § 2.1) [9];
- алгоритм оптимізації спроектованих СТС за функціонально-вартісними характеристиками (див. § 2.2.1) [73, 74];
- комп'ютерну систему підтримки процесу розробки СТС (див. § 2.4) [87].

Для острова Ayers встановлено вимоги по покритюю периметру, що складається з 8-ми ділянок по 356, 286, 449, 375, 111, 264, 266, 59 м (рис. 4.1), де необхідно виявляти рух при використанні 6-ти типів мікрохвильових сповіщувачів із зонами дії 200, 100, 75, 75, 50, 50 м. Для першої ділянки комп'ютерна система підтримки процесу розробки СТС згенерувала 175 варіантів розміщення сповіщувачів, частково представлених у таблиці 4.1, де кожна стрічка таблиці є варіантом СТС, а цифри у стрічках означають кількості відповідних компонентів, потрібних для повного покриття ділянки. З врахуванням варіантів інших ділянок кількість комбінацій повного перебору становитиме $175*94*305*175*17*76*76*7 \approx 6 \cdot 10^{14}$.

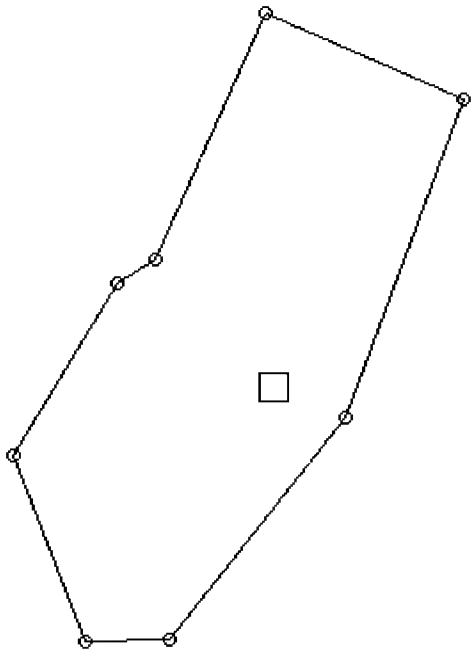


Рис. 4.1. Конфігурація периметру з 8-ми ділянок

Таблиця 4.1

Варіанти покриття ділянки довжиною 356 метрів

	200	100	75	75	50	50
1	2	0	0	0	0	0
2	1	2	0	0	0	0
3	1	1	1	0	0	0
4	1	1	0	1	0	0
5	1	1	0	0	2	0
…						
172	0	0	0	0	3	5
173	0	0	0	0	2	6
174	0	0	0	0	1	7
175	0	0	0	0	0	8

Перебір комбінацій (повний або обмежений) складається з генерування множини комбінацій і вибір серед них Парето-оптимальних. При використанні однопроцесорного комп'ютера Genuine Intel T1350 з тактовою частотою 1.86 ГГц і 1 ГБ оперативної пам'яті тривалість оптимізації складає близько 740405066.5 хв або 1382 роки. Об'єм оперативної пам'яті, необхідний для розв'язання задачі, складає не менше $Z \cdot n$ байт, де Z – кількість ділянок, n -

кількість альтернативних компонентів покриття кожної ділянки, якщо для її покриття потрібно не більше 255 компонентів. Таким чином, для розв'язання поставленої задачі необхідно 6*8*603500919700000 ≈ 26346 ТБ. Звідси очевидна непридатність повного перебору для розв'язання такого типу задач.

Гетерогенна морфологічна матриця з Парето-оптимальними варіантами покриття ділянок має лише 10*10*13*10*6*7*7*4 = 15288000 варіантів СТС, які не обов'язково оптимальні. Для вибору Парето-оптимальної множини розв'язків використано алгоритм їх маркування [97] за критеріями (2.5)-(2.7). Об'єм оптимальної множини склав 178 розв'язків (див. Додаток К). Кожна стрічка є варіантом СТС і містить 6x8 елементів, що описують число сповіщувачів кожної з 6 моделей на 8-ох ділянках периметру.

В табл. 4.2 наведено чотири Парето-оптимальні варіанти побудови СТС, які вибрано за допомогою процедури нормування (2.27), описаної в § 2.2.2. Перший та другий варіанти – з найменшими імовірностями хибного спрацювання (при $\beta_1 = 1, \beta_2 = \beta_3 = 0$) та невиявлення (при $\beta_1 = \beta_3 = 0, \beta_2 = 1$), третій – з найменшою ресурсоємністю (при $\beta_1 = \beta_2 = 0, \beta_3 = 1$), четвертий – розв'язок, критерії котрого найближчі до точки утопії (при $\beta_1 = \beta_2 = \beta_3 = 1$). Кінцевий варіант вибирає замовник за критерієм, який для нього має більший пріоритет у порівнянні з іншими.

Таблиця 4.2

Оптимальні варіанти СТС острова Ayers

№ Варі-анту	Критерії			Сповіщувачі	Централь	Візуалізація
	Q_{sys}	R_{sys}	C_{sys}, дол. США			
41	0.03	0.171	7750	31 шт. моделі №3	Integra 32	ЖК Панель
83	0.25	0.035	6750	15 шт. моделі №1	SA-64	ЖК Панель
27	0.25	0.218	5388	8 шт. моделі №1 8 шт. моделі №4 2 шт. моделі №6	SA-64	ЖК Панель
84	0.12	0.159	6200	7 шт. моделі №1 1 шт. моделі №2 11 шт. моделі №3	SA-64	ЖК Панель

Розглянемо вплив завад не чітко визначеного типу та інтенсивності на результати оптимізації СТС охорони периметру острова Ayers. Врахування їх впливу здійснимо за допомогою розробленого в §2.2.3 методу нечіткого виводу. Як впливаючу величину, розглянемо швидкість вітру. Як зазначалося в §2.2.3, параметри нечітких множин визначені на основі даних [34]. Імовірнісні характеристики сповіщувачів, що використовуються під час розробки СТС, параметри нечітких функцій вразливості та вартості наведені в таблиці 4.3.

Таблиця 4.3.

Характеристики сповіщувачів

Тип сповіщувача, j	Область дії, м	Ймовірність функціонування p_j	Ймовірність хибної тривоги q_j	Ймовірність невиявлення r_j	Параметри вразливості, м/с, див. (2.36)	Вартість, дол. США
1	200	0.999	0.25	0.02	[0 23.0]	450
2	100	0.999	0.25	0.11	[0 23.0]	300
3	75	0.999	0.03	0.14	[0 21.5]	250
4	75	0.999	0.25	0.15	[0 23.0]	186
5	50	0.999	0.25	0.15	[0 25.0]	200
6	50	0.999	0.25	0.20	[0 13.0]	150

Характеристики інтенсивності впливу завади “сильний вітер” описуються трапецевидною нечіткою функцією (2.35) із параметрами $a=0$ $b=2$ $c=6$ $d=15$. Зображення нечітких множин, що характеризують перший та шостий сповіщувачі з таблиці 4.2, та їх перетин з нечіткою множиною “сильний вітер” наведено на рис. 2.8. Згідно (2.30), скореговані значення імовірностей хибного спрацювання з врахуванням впливів завад складають: $q_1=0.5302$, $q_2=0.5302$, $q_3=0.3274$, $q_4=0.5302$, $q_5=0.5103$, $q_6=0.7089$. Їх отримано шляхом дефазифікації перетинів нечітких множин вразливості кожного сповіщувача з множиною “сильний вітер” і встановленням значення $w_d=1-\max\limits_j q_j=0.75$.

Оптимізація з нечітким виводом, що враховує вплив “сильного вітру” на сповіщувачі, скорочує обсяг Парето-оптимальних структур до 138 варіантів СТС, що складає приблизно 78% від початкової множини 178 варіантів (див. Додаток К).

Отже, при виконанні проекту “Проектування дистрибутивної сенсорної мережі для безпеки Ayers Island з використанням технології функціонально-вартісного аналізу” апробовано розроблені в розділі 2 методи оптимізації функціонально-вартісних характеристик СТС на базі ГА та нечітких множин і впроваджена комп’ютерна система підтримки процесу розробки СТС (див. Додаток Б). Використання неповного перебору дозволило вирішувати такі задачі оптимізації за прийнятний час 24 хв., а використання методів нечіткого виводу дозволило скоротити кількість Парето-оптимальних структур СТС на 22% шляхом відсіювання СТС, що збільшують частоту хибних спрацювань.

Другий із запропонованих оптимальних варіантів СТС (див. табл. 4.1) впроваджено на острові Ayers. Замовник вважав, що підвищення ціни на 1360 доларів США є виправданим через меншу інтенсивність хибних тривог для цього варіанту.

4.2. Комп’ютеризована система тривожної сигналізації периметру території за українсько-турецьким проектом

В рамках спільного українсько-турецького науково-дослідного проекту “Розробка методів проектування та оптимізації систем виявлення порушників безпеки”, згідно договору № М/47-2008 від 27.03.08, котрий виконувався за підтримки Міністерства освіти і науки України Науково-дослідним інститутом інтелектуальних комп’ютерних систем ТНЕУ спільно з Інститутом технологій, м.Гебзе, Республіка Туреччина, автором:

- розроблено метод відображення хромосом ГА в область аргументів комбінаторної задачі багатокритеріальної оптимізації функціонально - вартісних характеристик СТС [8, 93], (див. § 2.2);
- проведено порівняльний аналіз алгоритмів багатокритеріальної комбінаторної оптимізації [14];

- програмно реалізовано метод розробки оптимізованих СТС на основі генетичного алгоритму окремим модулем комп'ютерної системи підтримки процесу розробки СТС [13, 92];
- розроблено експериментальний зразок варіанту КСТС, отриманого комп'ютерною системою підтримки процесу розробки СТС (див. Додаток Б).

Слід зазначити, що за основу даного проекту були взяті результати американсько-українського проекту (див. § 4.1). Апробація ГА з розробленими методами відображення (2.19)–(2.24) була здійснена на задачах оптимізації трьох периметрів різної конфігурації та розмірності. Їх детальний опис наведено в Додатку Л.

Конфігурацію першого периметру було наведено на рис. 2.3. Морфологічна матриця для оптимізації СТС даного периметру описує $\approx 14 * 10^6$ варіантів. На тестовій платформі (див. §4.1) повний перебір цієї множини варіантів триває 594156.25 мс або 9.9 хвилин. Дослідження динаміки часток оптимальних рішень різними методами для першого периметру представлено на рис. 4.2.

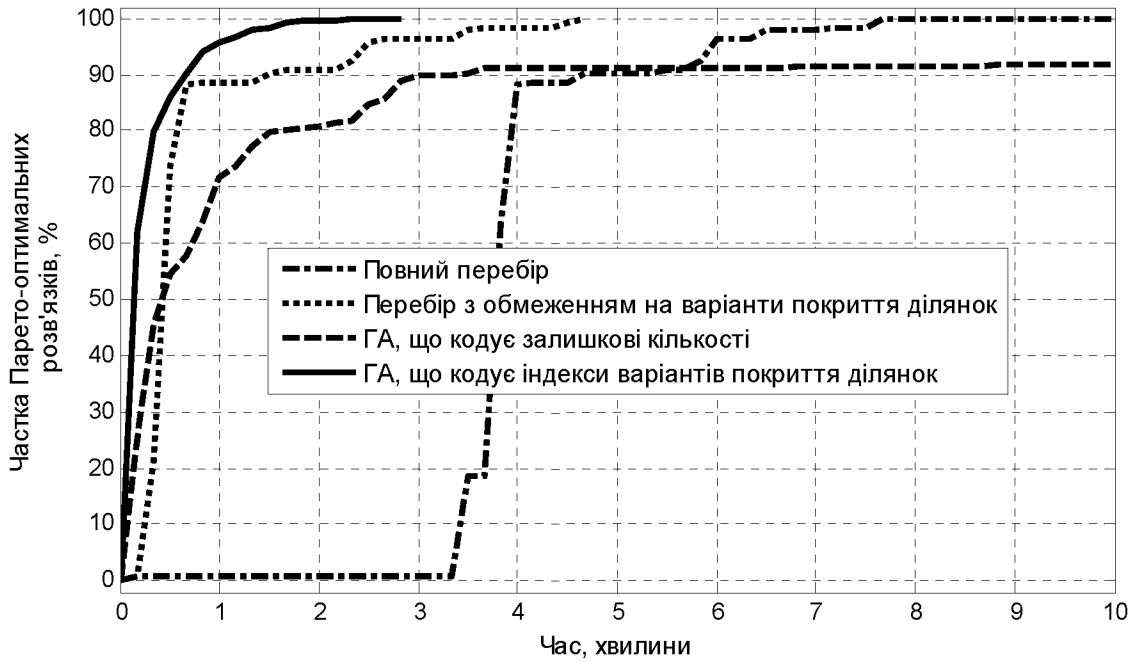


Рис. 4.2. Динаміка частки Парето-оптимальних результатів комбінаторних алгоритмів при оптимізації першого периметру

Методом повного перебору вибрано 236 Парето-оптимальних рішень, що утворюють еталонну множину, яка використовується для проведення аналізу. Як видно з рис. 4.2, ГА, що кодує індекси, за 2.33 хв. знаходить всі Парето-оптимальні рішення, незважаючи на те, що метод повного перебору (який знайшов її за 7.67 хв.), за рахунок відсутності процедур формування хромосом, повинен мати перевагу на малих вибірках. Виграш швидкості знаходження еталонної множини ГА складає 69% навіть для малої вибірки.

Для порівняння отриманих результатів оптимізації доцільно розробити критерії кількісного оцінювання ефективності алгоритмів формування Парето-множини варіантів реалізації СТС. Перший критерій – тривалість виконання алгоритму на одному і тому ж комп'ютері. Другий критерій – частка Парето-оптимальних розв'язків у загальній множині, сформованій об'єднанням результатів всіх алгоритмів.

На другому периметрі (див. рис. 4.1) досліджено динаміку частки Парето-оптимальних результатів комбінаторних алгоритмів за дуже обмежений час – 1 хв. У порівнянні з найкращою схемою перебору (рис. 4.3) ГА до часу 0,5 хв. має перевагу, хоча він виконує процедури формування хромосом. Максимально ГА формує на 15 % більше розв'язків порівняно з найкращою схемою перебору (для часу 0,2 хв.). Ці дослідження показують, що розроблений метод оптимізації на основі ГА може бути використаний і для вирішення задач у інших системах, наприклад, керування відносно швидкоплинними процесами.

Проведені дослідження алгоритмів перебору показали, що алгоритм з обмеженням вибору (вибір лише з Парето-оптимальних ділянок) дає за одну хв. в середньому на 15% більше Парето-оптимальних результатів, ніж ГА, який кодує залишкові кількості компонентів, тобто вимагає процедури усунення надлишкових сповіщувачів.

Як вже було відзначено, ускладнення структури СТС, зокрема збільшення кількості ділянок, їх розмірів, кількості компонентів, типів сповіщувачів веде до експоненційного росту кількості допустимих розв'язків задачі (2.9)–(2.10) та (2.16)–(2.18). При цьому тривалість повного перебору неприйнятно зростає.

Обмеження кількості перебраних варіантів СТС (алгоритм неповного перебору) хоч і дає можливість працювати зі складними периметрами, але не проаналізовані варіанти можуть містити рішення, кращі порівняно з виявленими. В цьому випадку ГА знаходить кращі рішення, ніж неповний перебір. Час оптимізації для ГА зі зростанням складності задачі оптимізації (наприклад, збільшенні кількості типів сповіщувачів СТС Ayers Island) стабілізується на цілком прийнятному рівні – 3 хвилини (табл. 4.3).

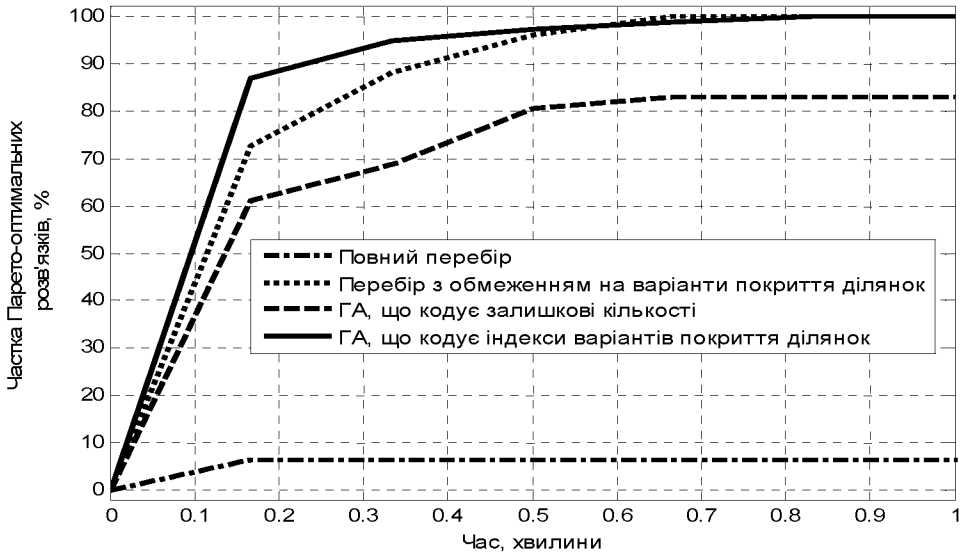


Рис. 4.3. Фрагмент динаміки частки Парето-оптимальних результатів комбінаторних алгоритмів при оптимізації другого периметру

Таблиця 4.3.

Результати порівняльного аналізу схем перебору і ГА при оптимізації периметру №3 (див. Додаток Л)

К-сть спов.	Кількість ітерацій		К-сть опт. систем	Час виконання			% співпадань за 3 хв	
	Повного перебору	Обмеже-ного перебору		повного перебору	Обмеже-ного перебору	ГА	Пере-бору	ГА
1	1	1	1	15 мс.	15 мс.	1 сек.	100	100
2	$\approx 3*10^6$	2916	12	7 хв.	422 мс.	20 сек.	100	100
3	$\approx 196*10^{12}$	8	2	4.6 доби	15 мс.	1 хв.	100	100
4	$\approx 3*10^{12}$	$\approx 10^6$	509	6 років	7.8 хв.	3 хв.	29	64
5	$\approx 196*10^{12}$	$\approx 15\cdot 10^6$	903	433 роки	4 год. 5 хв.	3 хв.	7	39
6	$\approx 603*10^{12}$	$\approx 15\cdot 10^6$	178	1382 роки	24 хв.	3 хв.	47	53

Як видно з табл. 4.3, кількість оптимальних рішень не обов'язково зростає при зростанні кількості типів сповіщувачів. Це означає, що використання нового типу сповіщувача може радикально міняти оптимальні рішення. Тому слід проводити оптимізацію СТС, використовуючи множину всіх компонентів бази даних, що підтверджує доцільність використання ГА. Для ГА час оптимізації задається кількістю наступних популяцій, що обмежують процес оптимізації, тому він практично постійний. З табл. 4.3 видно, що ГА недоцільно використовувати для оптимізації простих СТС. ГА забезпечують суттєве зменшення кількості циклів обчислень, однак кожен цикл є значно складнішим порівняно з методом повного перебору. Тут перевага ГА проявляється тільки для СТС, які використовують 4 і більше типів сповіщувачів. Повний перебір знаходить всі оптимальні СТС. Але, як видно з табл. 4.3, частка знайдених ГА оптимальних СТС за заданий час при їх ускладненні починає зменшуватися. Однак це зменшення плавніше в порівнянні з неповним перебором. Для 4 і 6 сповіщувачів ГА сформував 64% і 53% оптимальних СТС, але обмежений перебір за той самий час сформував лише 29% і 47% оптимальних СТС. Тому очевидною є ефективність ГА за швидкодією.

Периметр, описаний в § 4.1, був оптимізований за допомогою ГА з розробленим новим методом відображення (2.24). Результати оптимізації наведені на рисунках 4.4 та 4.5. Ріст частки оптимальних СТС, знайдених ГА та неповним перебором при використанні 4 типів сповіщувачів від часу оптимізації показаний на рис. 4.4. Як видно, для малих обсягів вибірки неповний перебір при великому часі оптимізації знаходить 100% оптимальних рішень швидше за ГА. Однак для малих часів оптимізації ГА значно обганяє неповний перебір.

Крім того, для прикладу рис. 4.4, на рис. 4.5 показано структуру оптимальних рішень знайдених з допомогою ГА (○) та обмеженого перебору (◊) для часу оптимізації 50 с. відносно еталонної множини (•), знайденої повним перебором. Як видно, неповний перебір спочатку виявляє Парето-

оптимальні варіанти високої якості та ціни, а ГА спершу виявляє варіанти, ближчі до початку координат.

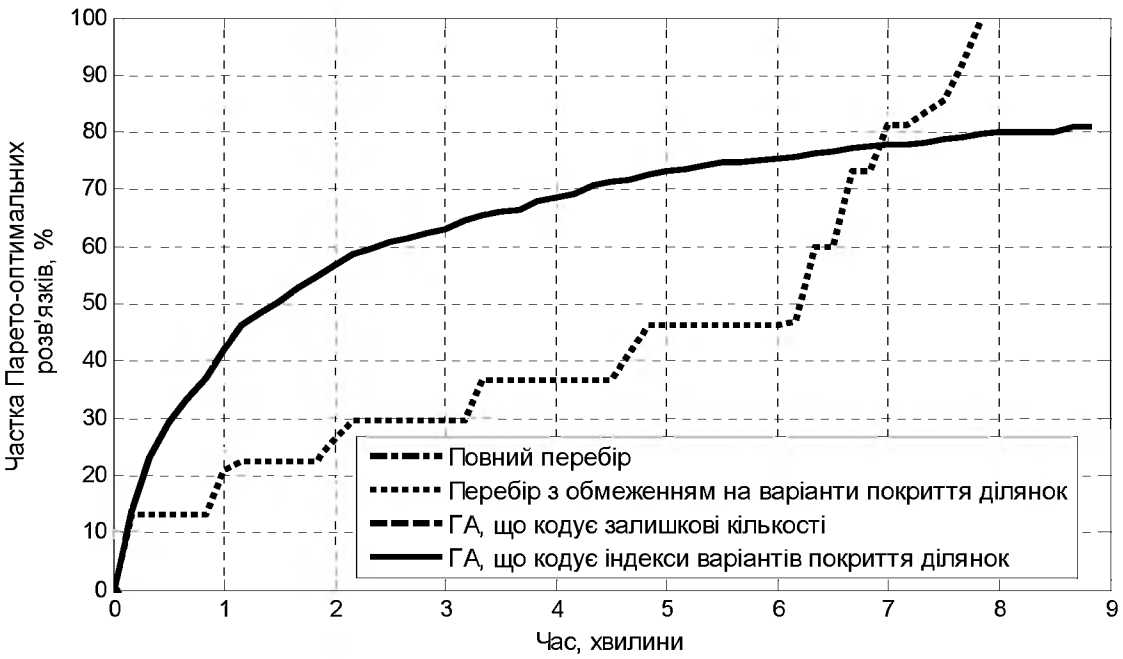


Рис. 4.4. Ріст частки оптимальних СТС, знайдених ГА та неповним перебором

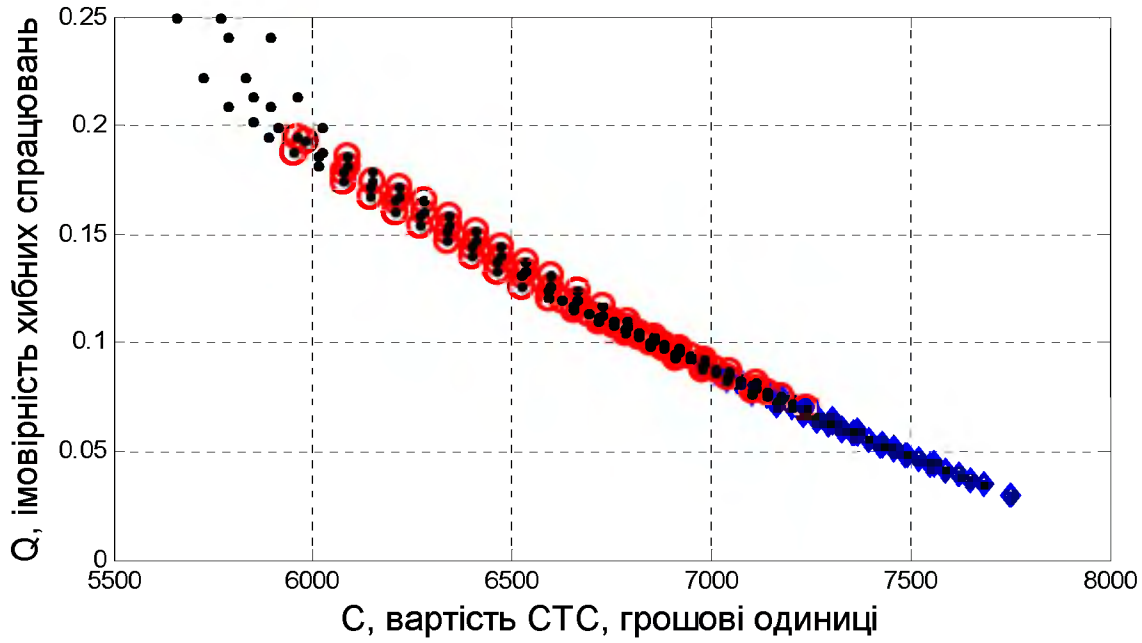


Рис. 4.5. Порівняння результатів ГА (○) та обмеженого перебору (◊) з еталонною множиною (●), обчислених за 50 секунд

На рис. 4.6 проілюстровано динаміку частки Парето-оптимальних розв'язків у агрегованій множині, знайдених досліджуваними алгоритмами для чотирьох типів сповіщувачів.

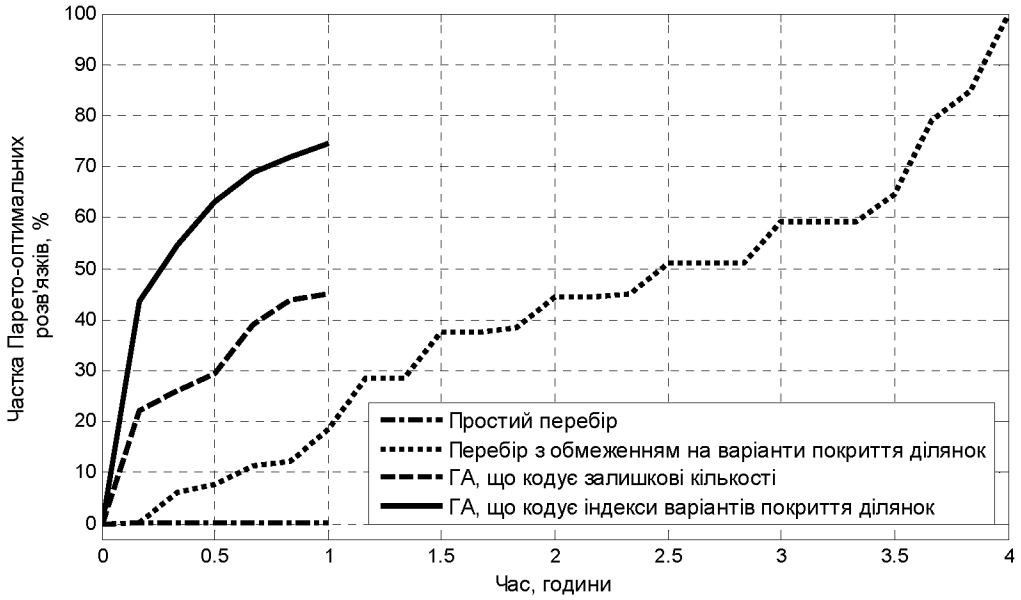


Рис.4.6. Динаміка частки результатів у агрегованій Парето-множині для 4-ох типів сповіщувачів

Таким чином можна ствердити, що при рішенні задач оптимізації ГА мають безперечну перевагу при великих вибірках. Однак він має також переваги в деяких режимах роботи і при малих вибірках. ГА знаходить всі Парето-оптимальні рішення, працює в деяких випадках краще інших алгоритмів і, при обмеженні часу роботи, знаходить рішення, кращі за інші алгоритми.

Для дослідження роботи оптимального варіанту КСТС розроблено її експериментальний зразок (рис. 4.7) із розробленим в розділі 3 захистом лінії зв'язку між ПКП та інтерфейсними контролерами сповіщувачів. Запропонована і реалізована структура цієї КСТС включає:

1. приймально-контрольний прилад (сервер мережі), реалізований на мікроконтролері ATmega128 з оперативною пам'яттю об'ємом 32 КБ (мікросхема UM61256), яка ініціалізує контролери сповіщувачів (див. рис.

3.4), живить їх через мережу, керує ними, періодично опитує їх стан і приймає рішення про порушення безпеки;

2. індикатор на рідких кристалах, підключений до приймально-контрольного приладу для відображення інформації про стан сповіщувачів КСТС;
3. мембранну клавіатуру для керування та налаштування режимів роботи системи, встановлення або зняття з охорони відповідних зон КСТС;
4. блок живлення, що формує напругу +12 В для живлення приймально-контрольного приладу і сповіщувачів через двопровідну мережу;
5. сповіщувачі (пасивний інфрачервоний, розбиття скла, димовий), що забезпечують виявлення руху та акустичних сигналів відповідної форми;
6. двопровідну лінію зв'язку (мережу «спільна шина») для передачі даних та живлення сповіщувачів;
7. контролери сповіщувачів (див. рис. 3.4), які розміщуються в корпусі сповіщувачів і передають стан їх виходів на приймально-контрольний прилад.

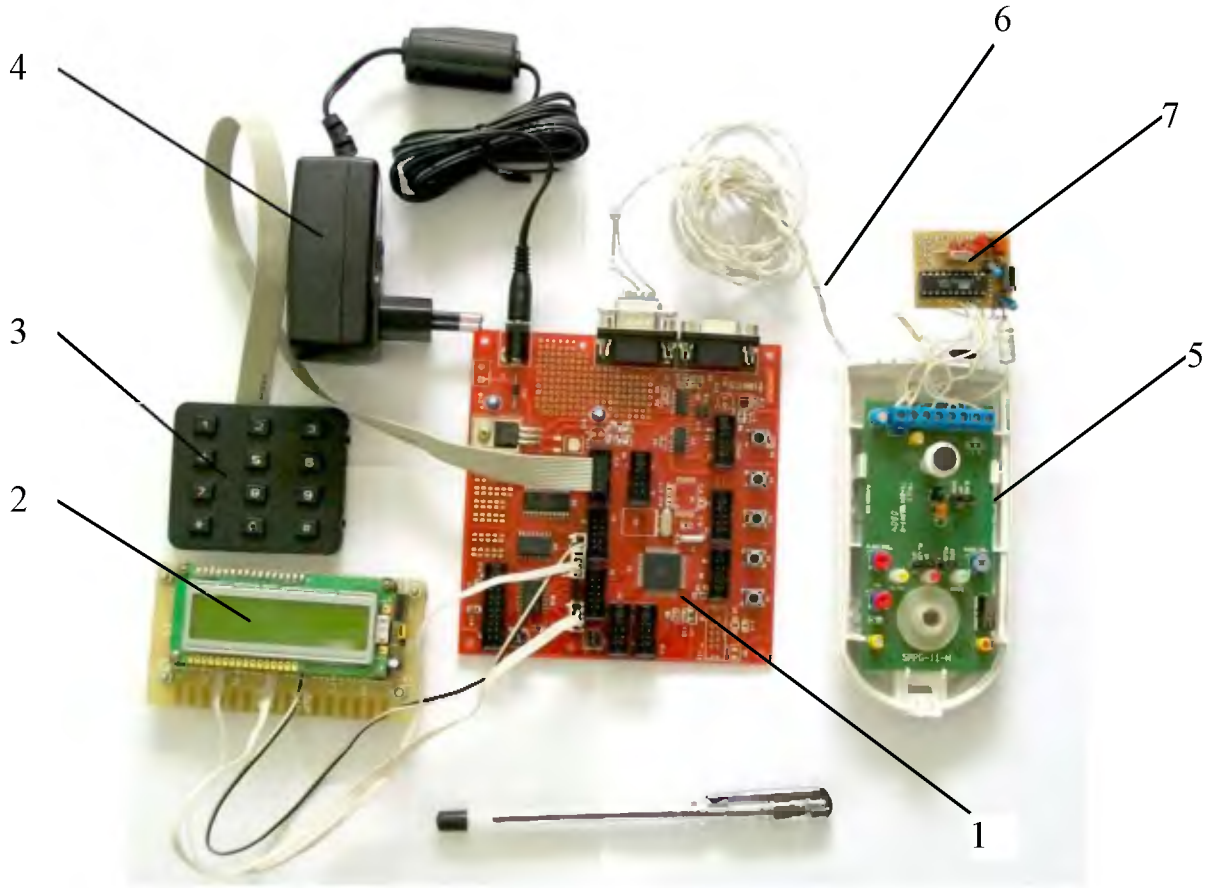


Рис. 4.7 Експериментальний зразок КСТС

Дослідження експериментального зразка КСТС показали його працездатність і високу стійкість запропонованого інтерфейсу до зламу - ні одна спроба зламу не вдалася.

Отже, під час виконання спільного українсько-турецького науково-дослідного проекту “Розробка методів проектування та оптимізації систем виявлення порушників безпеки”, на основі запропонованих в § 2.3 методів оптимізації на основі ГА, що програмно реалізовані автором як окремий модуль оптимізації в комп’ютерній системі підтримки процесу розробки СТС, доведено доцільність використання ГА для даного класу задач. Їх використання в розробленій комп’ютерній системі підтримки процесу розробки СТС дозволяє зменшити часову складність розробки СТС та підвищити збіжності ГА за рахунок використання запропонованого методу відображення. Також розроблено і досліджено експериментальний зразок КСТС. Результати досліджень апробовано в роботах [8, 13, 92, 93].

4.3. Комп’ютеризована система тривожної сигналізації НДІ інтелектуальних комп’ютерних систем

В рамках теми «Розробка теоретичних основ підтримки прийняття рішень для синтезу розподілених систем безпеки», державний реєстраційний номер 0106U010731, застосовано розроблену автором комп’ютерну систему підтримки процесу розробки СТС (див. § 2.5) [87] до розробки не СТС периметру територій, а до розробки СТС безпеки приміщень. З її допомогою створено Парето-оптимальні за критеріями якість/вартість, надійність/вартість структури КСТС НДІ інтелектуальних комп’ютерних систем, з яких вибрано один з найдешевших варіантів.

НДІ Інтелектуальних комп’ютерних систем (ІКС) Тернопільського національного економічного університету є хорошою базою для реалізації даного дослідження. Він містить 19 кімнат, більшість з яких обладнана комп’ютерною технікою, тому потребує посиленої охорони. Географічно НДІ

розміщений у напівпідвальному приміщенні II корпусу ТНЕУ. Частина вікон НДІ виходить на зовнішню сторону території ТНЕУ, де немає огорожі, поряд із тротуаром - тому є вільний доступ до вікон. Хоча у проміжку між вікнами є грати, охоронець може не чути звук розбитого скла та виламування грат, оскільки він знаходиться на першому поверсі. Також в приміщеннях НДІ встановлені прості картонно-дерев'яні двері, які не складають великої перешкоди зловмиснику в доступі до кімнати. Враховуючи сказане, було запропоновано спроектувати та встановити КСТС НДІ ІКС, котра має мати максимальну інформативність: охоронець має знати, який з сповіщувачів спрацював (розбиття скла, руху, відкривання дверей) і в якому приміщенні, що дасть йому змогу швидше реагувати на порушення.

Процес проектування КСТС НДІ ІКС з використанням комп'ютерної системи підтримки процесу розробки СТС [87] містить декілька етапів. Першим є ввід опису території НДІ ІКС. Для цього завантажується зображення плану периметру, встановлюється розміщення зон (по діагоналі кожної кімнати проводиться окрема лінія - зона), встановлюються типи загроз кожної зони (в даному випадку – рух, розбиття скла, відкриття дверей), встановлюється місце розміщення ПКП та світлового табло з клавіатурою (рис. 4.8).

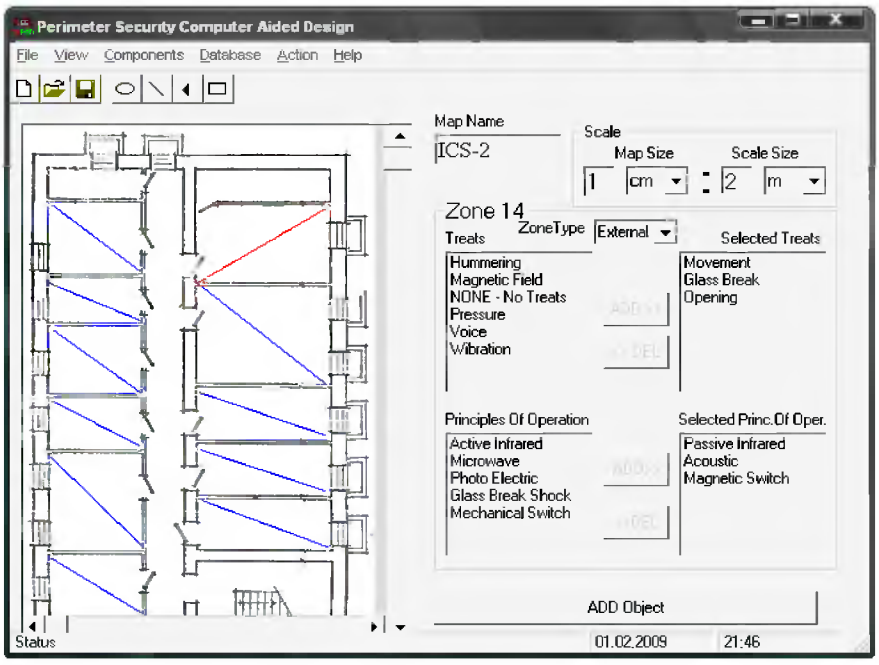


Рис. 4.8. Вікно вводу опису периметру території НДІ ІКС

На наступному кроці комп'ютерна система підтримки процесу розробки СТС проектує та вибирає ряд оптимальних варіантів КСТС НДІ ІКС (рис. 4.9).

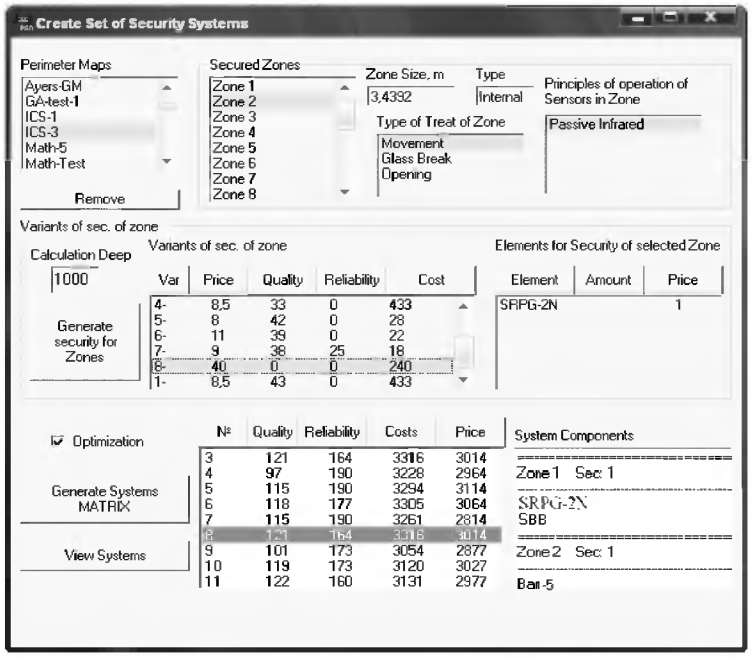


Рис. 4.9. Вікно вибору введеного периметру та проектування СТС НДІ ІКС

Для порівняння варіантів СТС без інтерфейсних контролерів та з ними, було розроблено, з використанням комп'ютерної системи підтримки процесу розробки СТС, дві системи – без інтерфейсних контролерів та з ними. З врахуванням цінових обмежень вибрано одну із оптимальних СТС (табл. 4.4.) та отримано її кошторис. Як видно з табл. 4.4, основну вартість системи складають сповіщувачі, блоки розширення, кабель та охоронна панель (ПКП).

Також була розроблена КСТС, яка використовує інтерфейсні контролери. Її кошторис представлено в табл. 4.5. Як видно з табл. 4.5, основну вартість системи складають сповіщувачі та інтерфейсні контролери. Затрати на кабель в 27 разів менші, ніж на сповіщувачі.

Порівняння кошторисів табл. 4.4 і табл. 4.5 показує, що навіть в приміщенні, тобто при малій довжині лінії зв'язку, використання інтерфейсного контролера вигідне. Хоча економія отримана не за рахунок проводів (сумарна вартість кабелю і контролерів другого варіанту більша за вартість кабелю

першого), але використання наявного в НДІ ІКС комп'ютера перекриває втрати. Хоча, навіть при затратах на вживаний комп'ютер на рівні 100-150 доларів США, використання інтерфейсного контролера все одно вигідне.

Таблиця 4.4

Кошторис СТС НДІ ІКС без інтерфейсних контролерів

№	Обладнання	Тип	К-сть	Ціна, дол. США	Сума, дол. США
1	Панель охоронна (ПКП)	CA-64	1	130,00	130,00
2	Корпус панелі	PC-500	1	8,00	8,00
3	Блок живлення	DSC 12/15	1	35,00	35,00
4	Блок розширення	EXP-764	6	30,00	180,00
5	Клавіатура + ЖКІ	LCD-764	1	60,00	60,00
6	Табло світлодіодне	LD-112	1	37,00	37,00
7	Сирена	SA-252T	1	15,00	15,00
8	Давач комбінований: пасивний інфрачервоний + розбиття скла	SRPG-2N	18	27,00	486,00
9	Давач пасивний інфрачервоний (штора)	Visu 99-L	1	23,00	23,00
10	Геркон (Двері)	СМК 1-1	20	0,60	12,00
11	Короб монтажний	КМ-1	80 м	0,20	16,00
12	Кабель	8*0,22	500 м	0,28	140,00
				Всього	**1142**

Таблиця 4.5

Перелік обладнання КСТС НДІ ІКС з інтерфейсним контролером

№	Обладнання	Тип	К-сть	Ціна, дол. США	Сума, дол. США
1	Сповіщувач комбінований: пасивний інфрачервоний + розбиття скла	SRPG-2N	18	27,00	486,00
2	Сповіщувач пасивний інфрачервоний (штора)	Visu 99-L	1	23,00	23,00
3	Геркон (Двері)	СМК 1-1	20	0,60	12,00
4	Короб монтажний	КМ-1	80 м	0,20	16,00
5	Кабель	2*0,22	90 м	0,20	18,00
6	Контролер мережі		19	9,00	171,00
7	Адаптер мережі		1	8,00	8,00
8	ПК, наявний в НДІ ІКС		1	0	0
				Всього	**734**

Структурна схема КСТС НДІ ІКС представлена на рис 4.10. До мережі сповіщувачів (пасивних інфрачервоних і розбиття скла), оснащених інтерфейсними контролерами (див. рис. 3.3, 3.4 і 4.11) [10, 88], підключені з допомогою таких же контролерів сповіщувачі пожежної сигналізації. Як ПКП служить файловий сервер, який вже є в НДІ ІКС. Додаткове навантаження на нього незначно збільшує час доступу до нього.

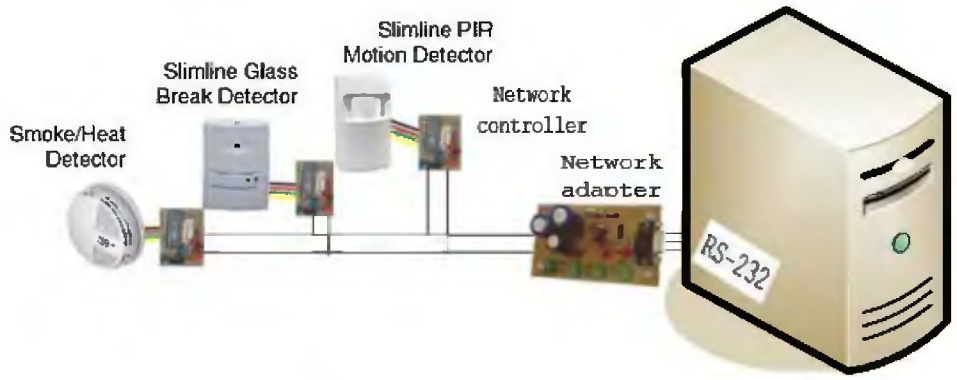


Рис. 4.10. Структура КСТС НДІ ІКС

Рис. 4.11. Сповіщувач SRPG-2N з приєднаним мережевим контролером

Комп'ютер в КСТС (див. рис. 4.10) відіграє також роль пристрою візуалізації. За допомогою програмного пакету Guard64 фірми Satel [103] було створено карту розміщення сповіщувачів на території НДІ ІКС (рис. 4.12), причому зеленим кольором висвітлюються зони, котрі на даний час знаходяться під охороною (1); білим кольором з косою штриховкою – зони, зняті з охорони, наприклад (2); сірим квадратом з цифрою – сповіщувач, який

не видає сигнал тривоги, в даному випадку – сповіщувачі руху, розбиття скла або відкривання дверей, наприклад (3). Заміна кольору сірого квадрату на зелений означає спрацювання даного сповіщувача, наприклад, в кімнатах 2014 і 2017 (4) спрацювали сповіщувачі 7 і 28, що відповідають за стан дверей (двері відкриті). У випадку, коли спрацьовує один із сповіщувачів зони, котра знаходиться під охороною, подається сигнал тривоги, зона виділяється червоним кольором, і на екран виводиться повідомлення про номер сповіщувача та місце його розміщення.

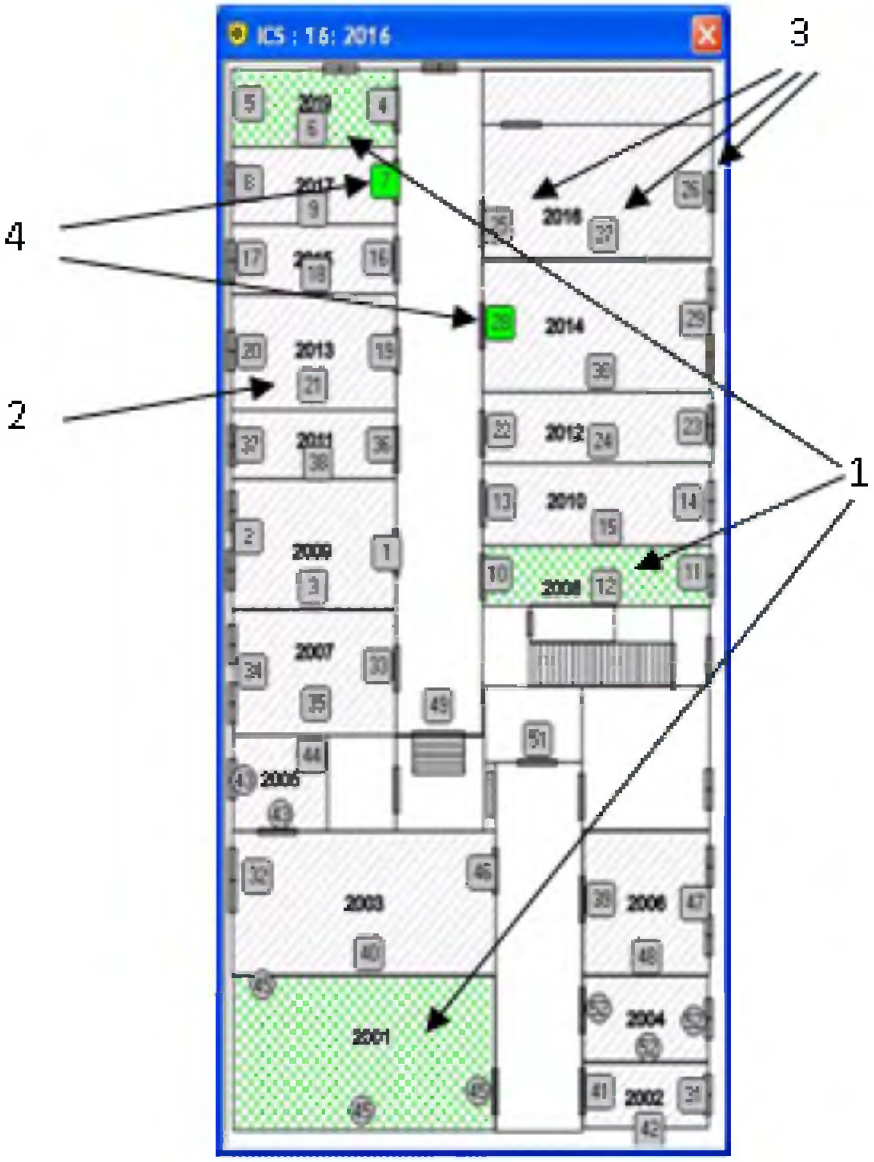


Рис. 4.12. Карта розміщення сповіщувачів НДІ ІКС в програмі Guard64.

Таким чинов, в §4.3 з допомогою комп'ютерної системи підтримки процесу розробки СТС розроблено структуру КСТС приміщення НДІ ІКС на базі запропонованого мережевого контролера та комп'ютера в ролі ПКП. Аналіз кошторисів двох структур такої КСТС (з використанням інтерфейсного контролера і без нього) показав, що навіть при малих довжинах кабелю використання інтерфейсного контролера дає економічний ефект.

Висновки до розділу 4

1. В результаті виконання спільного американсько-українського науково-дослідного проекту “Проектування дистрибутивної сенсорної мережі для безпеки Ayers Island з використанням технології функціонально-вартісного аналізу”, грант № CRDF FSTM UM2-5012-TE-03, було створено декілька варіантів КСТС периметру території острова Ayers, використовуючи впроваджену комп’ютерну систему підтримки процесу розробки СТС, яка використовує запропоновані автором (§2.2) методи знаходження оптимальних рішень та БД наявних компонентів (§2.3). Використання одного із запропонованих варіантів СТС на острові Ayers дало змогу вибрати найбільш оптимальний склад компонентів за заданими критеріями відповідно до поставлених вимог, що в кінцевому випадку зменшило вартість обладнання без зменшення його функціональних властивостей.
2. В цьому ж проекті реалізовано запропонований метод оцінювання вразливості СТС периметру території з врахуванням неповної інформації про характеристики сповіщувачів та охоронних зон (§2.2.3). Розроблений метод використано для оцінки оптимальних СТС. Це дало змогу створювати найменш вразливі СТС з мінімізацією коштів, затрачених на їх побудову.
3. Під час виконання спільного українсько-турецького науково-дослідного проекту “Розробка методів проектування та оптимізації систем виявлення порушників безпеки”, № M/47-2008, на основі запропонованої в § 2.3 моделі оптимізації на основі ГА, що програмно реалізована автором, як окремий модуль оптимізації в комп’ютерній системі підтримки процесу розробки СТС, доведено доцільність використання ГА для даного класу задач, показано його переваги в різних умовах дослідження.

   4. За допомогою розробленої комп’ютерної системи підтримки процесу розробки СТС розроблена КСТС приміщення НДІ ІКС, ТНЕУ. Аналіз затрат на її обладнання показав рентабельність використання запропонованого мережевого контролера та комп’ютера в ролі ПКП.

ВИСНОВКИ

У дисертаційній роботі розв'язано наукову задачу з розробки методів і засобів оптимізації функціонально-вартісних характеристик комп'ютеризованих систем тривожної сигналізації. При цьому отримано наступні основні результати:

1. Аналіз ринку систем тривожної сигналізації показав, що переважна частина СТС спроектована за шаблонами і мало враховує специфіку периметру території, вимоги споживача та нову елементну базу. Запропоновано шляхи подолання цих недоліків шляхом створення комп'ютерної системи підтримки процесу розробки систем тривожної сигналізації та вдосконаленням компонентів.
2. Розроблено метод відображення хромосом генетичного алгоритму в область аргументів комбінаторної задачі багатокритеріальної оптимізації функціонально-вартісних характеристик комп'ютеризованих систем тривожної сигналізації за допомогою спеціальної процедури формування множини всіх можливих варіантів покриття даної ділянки та маркування мажоруючих варіантів покриття ділянок. Це дозволило зменшити часову складність алгоритму до 69 % і збільшити частку згенерованих Парето-оптимальних СТС на 15 %.
3. Для підвищення достовірності критеріїв відбору удосконалено метод оцінки функціональних характеристик систем тривожної сигналізації, який враховує невизначеності оцінок інтенсивності завад і вразливість до них компонентів цих систем шляхом використання нечітких множин та їх дефазифікації при визначенні критерію ризику проникнення порушника, що дозволило при розробленні програмного забезпечення формалізувати ризики невиявлення загроз сповіщувачами в конкретних умовах їх роботи.
4. Для реалізації запропонованих методів створено комп'ютерну систему, яка підтримує процес розробки систем тривожної сигналізації, яка базується на розвинутих і удосконалених методах їх структурної оптимізації та оцінки функціональних характеристик під час автоматизованого агрегування та

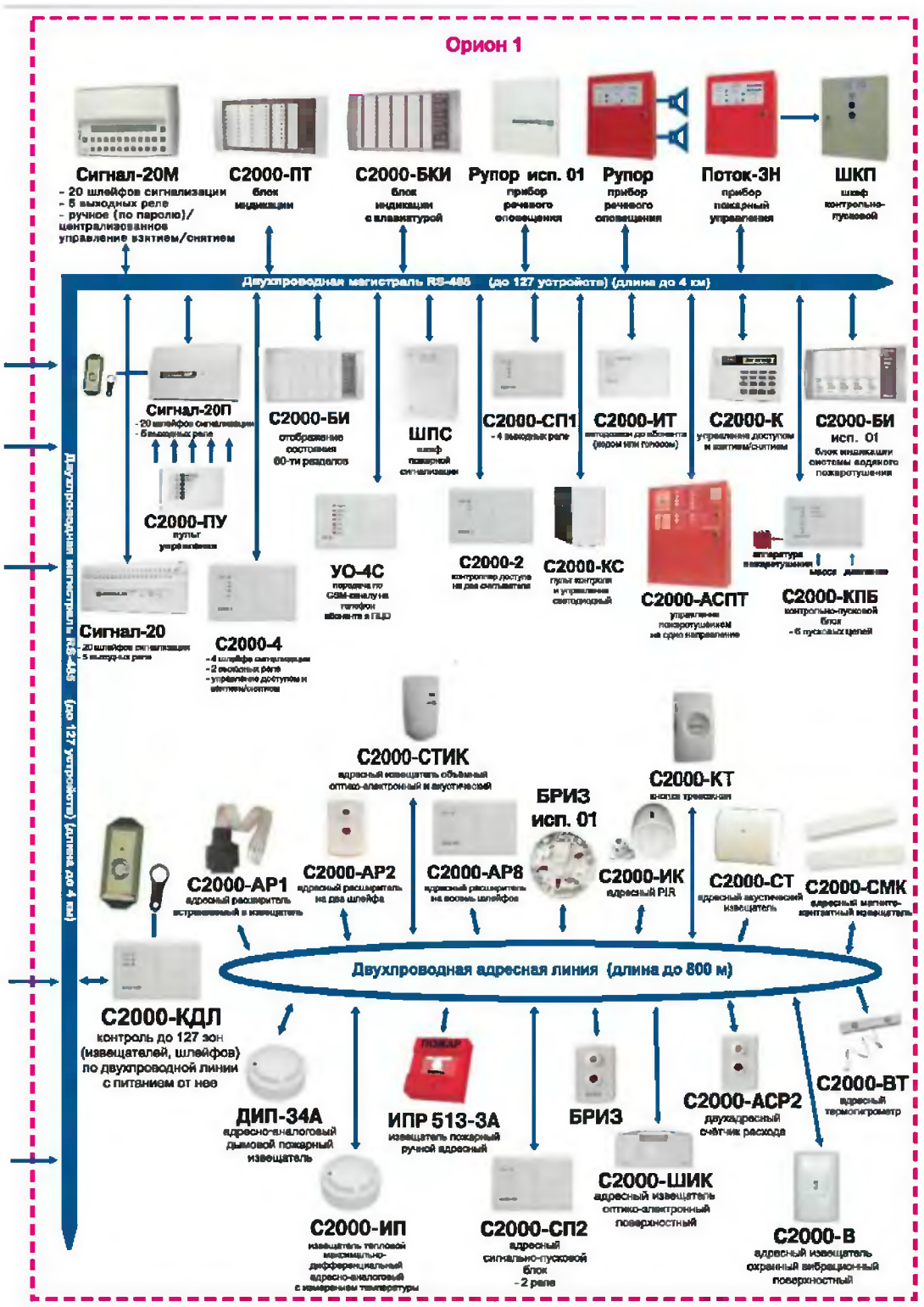
відбору кращих рішень, що дозволило покращити функціонально-вартісні характеристики таких систем.

5. Аналіз вартості компонентів розроблених систем тривожної сигналізації показав, що (і) традиційні сповіщувачі таких систем мають відносно низьку ціну; (іі) в таких системах затрати на зв'язок між сповіщувачами і приймально-контрольним приладом високі через топологію "зірка"; (ііі) мережеві сповіщувачі мають значно вищі ціни, через що системи на їх базі не потрапляють в множину Парето-оптимальних рішень. Така ситуація дозволяє покращити функціонально-вартісні характеристики систем тривожної сигналізації шляхом розробки спеціалізованого контролера, що забезпечує зв'язок сповіщувача з приймально-контрольним приладом по двохпровідній мережі та живлення великої групи сповіщувачів по цій же мережі. Розроблено дві модифікації такого контролера:
   - для систем тривожної сигналізації без підтримки захисту зв'язку (мережа захищена), що базується на недорогому мікроконтролері, забезпечує обмін даними і живлення від мережі для груп 10-15 традиційних сповіщувачів.
   - для систем тривожної сигналізації із захистом зв'язку (мережа незахищена), який використовує запропонований послідовний інтерфейс, що базується на виявленій максимальній кількості його елементів (19), які використано для захисту повідомлень, їх комбінованому застосуванні, а також прихованій детермінованій та псевдовипадковій їх заміні, що забезпечує надійний захист мережі від імітації сповіщувачів (імовірність зламу не більше 10^{-6} %) та індивідуальний захист систем (за рахунок встановлення параметрів захисту тільки уповноваженою особою, незалежно від виробника системи).
6. Запропоновано підвищити ступінь захисту мереж від зламу шляхом переводу (за командою сервера мережі) сповіщувачів, які перебувають в режимі "не під охороною", в режим роботи, відмінний від режиму "під охороною", при якому більшість повідомлень (80…90%) є повністю

випадковими, що не дозволяє зловмиснику, який має легальний доступ до підохоронної зони в режимі “не під охороною”, вивчати повідомлення сповіщувачів при штучно створених небезпеках, а також дозволяє створювати пастки в пропонованій мережі за рахунок спрощення захисту в режимі “не під охороною”, що створює ілюзію можливості заміни одного або декількох сповіщувачів імітаторами та сприяє передчасним діям зловмисника, які легко виявити в режимі “під охороною”.

Додаток **А**

Узагальнена структура системи охорони “Оріон”



Додаток **Б**

Акт про впровадження системи підтримки процесу розробки систем тривожної сигналізації на острові Ayers, США

Trefoil CORPORATION

CERTIFICATE OF IMPLEMENTATION

November 10, 2009

This certifies that Mr. Pavlo Bykovyy has worked well in the project "Design of Distributed Sensor Network for Ayers Island Security Using Value Analysis Technology" funded by a "First Steps to Market" grant of the U.S Civilian Research and Development Foundation (#CRDF FSTM UM2-5012-TE-03). The Research Institute of Intelligent Computer Systems at the Ternopil National Economic University together with Trefoil Corporation worked on this project.

During this project Mr. Pavlo Bykovyy developed the algorithms and automated design tools for designing security sensor networks, and implemented them in a software package called PSCAD (Perimeter Security Computer Aided Design), which was developed jointly with Trefoil Corporation.

PSCAD uses various attributes associated with components together with cost constraints to produce high quality systems for specific situations. It also can take into account quality, reliability, and cost constraints and requirements. As part of his implementation, Mr. Pavlo Bykovyy developed a database to store the many important attributes of components and security systems including cost requirements, performance criteria, and the many requirements and properties of the components. The software is very flexible and very complete, and works in a highly automated manner. It produces a Pareto-optimal set of security networks that satisfy given constraints as long as the constraints permit some solution.

Mr. Pavlo Bykovyy performed very well in the project and produced a good piece of software.

Prof. George Markowsky,
President of Trefoil Corporation
17 Oak Street
P.O. Box 127
Orono, Maine 04473
USA
markov@Trefoil.com

Преклад АКТУ

ПЕРЕКЛАД З АНГЛІЙСЬКОЇ НА УКРАЇНСЬКУ МОВУ

Корпорація Trefoil
АКТ ВПРОВАДЖЕННЯ
10 Листопада 2009

Даний акт засвідчує, що Павло Биковий брав активну участь у виконанні проекту "Проектування дистрибутивної сенсорної мережі для безпеки острова Ayers з використанням технології функціонально-вартісного аналізу" за підтримки гранту "Перші кроки до ринку" Фонду цивільних досліджень і розвитку США, № CRDF FSTM UM2-5012-TE-03. В даному проекті приймали участь Науково-дослідний інститут інтелектуальних комп'ютерних систем, Тернопільський національний економічний університет, спільно з корпорацією Trefoil.

Під час виконання проекту Павлом Биковим розроблено алгоритми та засоби автоматизованого проектування систем тривожної сигналізації, котрі були реалізовані ним спільно з корпорацією Trefoil в комп'ютерній системі підтримки процесу проектування - PSCAD.

Комп'ютерна система PSCAD використовує різні функціональні та вартісні характеристики компонентів для створення якісних систем тривожної сигналізації у кожному випадку. Вона також може враховувати обмеження на якість, надійність та вартість. Як частина свого впровадження, Биковий Павло розробив базу даних для збереження варіантів систем тривожної сигналізації, параметрів та характеристик їх компонентів, включаючи характеристики вартості, ефективності, та ін.. Комп'ютерна система є гнучкою та автоматизованою. Вона пропонує Парето-оптимальну множину систем тривожної сигналізації які задовольняють поставленим користувачем вимогам.

Павло Биковий сумлінно виконував поставлені перед ним завдання та розробив якісну комп'ютерну систему підтримки процесу проектування.

проф. Джордж Марковський,
президент корпорації Trefoil,
вул. 17 Оак
пош. скринька 127
Ороно, Мейн 04473
США

Переклад автентичний

Начальник відділу перекладачів
Тернопільського національного
економічного університету

Демченко Г.М.

Додаток **В**

Акт про впровадження результатів дисертаційної роботи по українсько-турецькому проекту

GEBZE INSTITUTE of HIGH TECHNOLOGY
ELECTRONICS ENGINEERING DEPARTMENT
Çayırova Kampusü, 41400, Gebze, Kocaeli, Turkey
Tel: +90 262 605 24 14

22 June 2010

CERTIFICATE OF IMPLEMENTATION

This document certifies that Mr. Pavlo Bykovyy has worked well on the research project "*Design and Optimization Methods of Physical Intrusion Detection for Security Systems*" funded by The Scientific and Technological Research Council of Turkey (TÜBİTAK) and The Ministry of Education and Science of Ukraine according to the agreements (#M/47-2008 from 27 March 2008). The Research Institute of Intelligent Computer Systems, Ternopil National Economic University, Ukraine and Gebze Institute of Technology, Turkey worked as partners in this project.

In the project, Mr. Pavlo Bykovyy has developed the optimization method of the computerized alarm systems cost-functional characteristics using the proposed type of selection in Genetic algorithm during the solution process of an extreme combinatorics task.

In a test example with the limited set of components, Mr. Pavlo Bykovyy improved the advantages of the proposed method, which allows to reduce the algorithm time-complexity to 69 % and increase the part of generated Pareto-optimal alarm systems for 15 %. Also it was suggested to use this method for a large alarm system's components set.

As a result of this project, Mr. Pavlo Bykovyy has implemented the proposed methods in the computerized system which supports the process of alarm system design.

Assoc. Prof. Dr. Serkan Aksoy
Gebze Institute of Technology
Department of Electronics Engineering
Gebze, Kocaeli
TURKEY

Переклад АКТУ

ПЕРЕКЛАД З АНГЛІЙСЬКОЇ НА УКРАЇНСЬКУ МОВУ

ІНСТИТУТ ТЕХНОЛОГІЙ ГЕБЗЕ
КАФЕДРА ЕЛЕКТРОННОЇ ІНЖЕНЕРІЇ

22 червня 2010р.

АКТ ВПРОВАДЖЕННЯ

Даний акт засвідчує, що Павло Биковий брав активну участь у виконанні проекту "Розробка методів проектування та оптимізації систем виявлення порушників безпеки". Він фінансувався Радою з питань науково-технічних досліджень Турецької Республіки (ТЮБІТАК) та МОН України (договір № М/47-2008 від 27.03.08). В даному проекті брали участь: Науково-дослідний інститут інтелектуальних комп'ютерних систем, Тернопільський національний економічний університет та Інститут технологій, м.Гебзе, Республіка Туреччина.

В даному проекті Павло Биковй розробив метод оптимізації функціонально-вартісних характеристик комп'ютеризованих систем тривожної сигналізації шляхом запропонованого способу відображення хромосом в генетичному алгоритмі під час розв'язання задачі екстремальної комбінаторики.

На тестовому прикладі з обмеженим набором компонентів систем тривожної сигналізації Павло Биковий обгрунтував переваги запропонованого методу, що дозволило зменшити часову складність алгоритму до 69 % і збільшити частку згенерованих Парето-оптимальних систем тривожної сигналізації на 15 %. Також обгрунтовано доцільність використання даного методу для великого набору компонентів систем тривожної сигналізації.

В результаті виконання проекту Павло Биковий впровадив запропоновані ним алгоритми пошуку оптимальних рішень в комп'ютеризованій системі підтримки процесу проектування систем тривожної сигналізації.

к.т.н., доц. Серкан Аксой
Кафедра електронної інженерії, Інститут технологій
м.Гебзе, Коацелі, Республіка Туреччина

Переклад автентичний

Начальник відділу перекладачів
Тернопільського національного
економічного університету

Демченко Г.М.

Додаток Ґ

Акт впровадження результатів по темі НДІ ІКС

Науково-дослідний інститут інтелектуальних комп'ютерних систем

Research Institute of Intelligent Computer Systems

Ternopil National Economic University
Ministry of Education and Science of Ukraine
Glushkov Institute of Cybernetics
National Academy of Sciences of Ukraine

Тернопільський національний економічний університет
Міністерство освіти і науки України
Інститут кібернетики ім. В.М. Глушкова
Національна академія наук України

3 Peremoga Square, Ternopil, 46009, Ukraine
Tel: +380 (352) 43-6038 Fax: +380 (352) 43-6354(24 hrs.)
http://www.tanet.edu.te.ua/ics/ ics@tneu.edu.ua

площа Перемоги 3, Тернопіль, 46009, Україна
Тел: (0352) 43-6038 Факс: (0352) 43-6354 (24 hrs)
http://www.tanet.edu.te.ua/ics/ ics@tneu.edu.ua

АКТ
про впровадження результатів
спільного українсько-турецького науково-дослідного проекту "Розробка методів проектування та оптимізації систем виявлення порушників безпеки", згідно договору № М/47-2008 від 27.03.08

Цим актом нижчепідписаний к.т.н., доц. кафедри інформаційно-обчислювальних систем та управління, директор Науково-дослідного інституту інтелектуальних комп'ютерних систем Кочан В.В. підтверджує впровадження результатів проекту "Розробка методів проектування та оптимізації систем виявлення порушників безпеки", згідно договору № М/47-2008 від 27.03.08, котрий виконувався за підтримки Міністерства освіти і науки України Науково-дослідним інститутом інтелектуальних комп'ютерних систем, ТНЕУ спільно з Інститутом технологій, м.Гебзе, Республіка Туреччина.

В результаті виконання міжнародного проекту були отримані наступні наукові та практичні результати:

1. Отримав подальший розвиток метод початкового наближення Парето-множини в генетичному алгоритмі розв'язання багатокритеріальної задачі екстремальної комбінаторики.
2. Запропоновано нову структуру мережі сенсорів безпеки.
3. Запропоновано новий метод забезпечення криптостійкості мережі сенсорів безпеки на базі дешевих мікроконтролерів.
4. Створено спеціалізовану систему проектування систем безпеки.
5. Розроблено спеціалізований інтерфейсний контролер сенсорів.
6. Розроблено приймально-контрольний пристрій мережі сенсорів безпеки.

Розроблена спеціалізована система проектування систем безпеки та експериментальний зразок системи безпеки на базі запропонованих інтерфейсних контролерів сенсорів та приймально-контрольного пристрою успішно використовуються в Науково-дослідному інституті інтелектуальних комп'ютерних систем.

Директор НДІ інтелектуальних
комп'ютерних систем,
к.т.н., доцент

Кочан В.В.

Додаток Д

Акт впровадження

СТС НДІ ІКС по темі НДІ ІКС 0106U010731

«ЗАТВЕРДЖУЮ»
Проректор з наукової роботи
Тернопільського національного
економічного університету
проф. Запорожний З.В.

«__» 06 2010 р.

АКТ
про впровадження результатів дисертаційної роботи
аспіранта кафедри інформаційно-обчислювальних систем та управління
Бикового Павла Євгеновича "Методи і засоби оптимізації функціонально-вартісних характеристик комп'ютеризованих систем сигналізації на основі генетичного алгоритму"
у науково-дослідній роботі на тему:
"Розробка теоретичних основ підтримки прийняття рішень для синтезу розподілених систем безпеки "
(державний реєстраційний номер 0106U010731)

Ми, комісія у складі директора Науково-дослідного інституту інтелектуальних комп'ютерних систем Тернопільського національного економічного університету (ТНЕУ), наукового керівника науково-дослідної роботи к.т.н., доцента Кочана В.В. та начальника науково-дослідної частини ТНЕУ Письменного В.І., створена для приймання роботи, виконаної в рамках тематичного плану науково-дослідних робіт ТНЕУ на тему "Розробка теоретичних основ підтримки прийняття рішень для синтезу розподілених систем безпеки" (державний реєстраційний номер 0106U010731), встановила:

1. Розроблений Биковим П.С. в дисертаційній роботі метод відображення хромосом генетичного алгоритму в область аргументів комбінаторної задачі багатокритеріальної оптимізації функціонально-вартісних характеристик комп`ютеризованих систем тривожної сигналізації, що зосереджує пошук на множині лише допустимих розв'язків задачі, дав змогу зменшити часову складність алгоритму оптимізації систем тривожної сигналізації і збільшити кількість отриманих за одиницю часу Парето-оптимальних систем тривожної сигналізації.
2. Запропонована Биковим П.Є. організація послідовного інтерфейсу, що базується на виявленій максимальній кількості елементів послідовного інтерфейсу, які можна використати для захисту повідомлень, їх комбінованому застосуванні, а також прихованій детермінованій та псевдовипадковій їх

заміні, забезпечила надійний захист мережі від імітації сповіщувачів та індивідуальний захист систем, що перешкоджає зловмиснику вивчати повідомлення сповіщувачів при штучно створених атаках і дає можливість створення пасток для зловмисника.

3. Запропонований Биковим П.Є. удосконалений метод оцінювання функціональних характеристик комп'ютеризованих систем тривожної сигналізації, який враховує невизначеності інтенсивностей завад і вразливість до них компонентів систем шляхом використання нечітких множин та їх дефазифікації при оцінюванні ризику проникнення порушника, дозволив формалізувати ризики невиявлення загроз сповіщувачами в конкретних умовах їх роботи та зменшити Парето-оптимальну множину шляхом відсіювання систем тривожної сигналізації, що збільшують частоту хибних спрацювань.
4. Запропонований Биковим П.Є. метод розробки комп'ютеризованих систем тривожної сигналізації, який базується на запропонованих методах їх структурної оптимізації та оцінки функціональних характеристик при автоматизованому агрегуванні і відборі кращих рішень, дозволив покращити функціонально-вартісні характеристики таких систем.

Науковий керівник
науково-дослідної роботи,
директор НДІ інтелектуальних
комп'ютерних систем,
к.т.н., доцент Кочан В.В.

Начальник науково-дослідної
частини Письменний В.І.

Додаток **Е**

Акт впровадження про впровадження у навчальний процес Тернопільського національного економічного університету (ТНЕУ)

«ЗАТВЕРДЖУЮ»
Проректор з науково-
педагогічної роботи
Тернопільського національного
економічного університету
_____ Шинкарик М.І.
«05» _____ 2010 р.

АКТ

про впровадження у навчальний процес Тернопільського національного економічного університету (ТНЕУ) результатів дисертаційної роботи аспіранта кафедри інформаційно-обчислювальних систем та управління Бикового Павла Євгеновича "Методи і засоби оптимізації функціонально-вартісних характеристик комп'ютеризованих систем сигналізації на основі генетичного алгоритму", що отримані в рамках виконання держбюджетної науково-дослідної теми НДІ ІКС-30-06 „К": "Розробка теоретичних основ підтримки прийняття рішень для синтезу розподілених систем безпеки" (державний реєстраційний номер 0106U010731).

Ми, комісія у складі: декана факультету комп'ютерних інформаційних технологій, д.т.н., проф. Дивака М.П., завідувача кафедри інформаційно-обчислювальних систем та управління, д.т.н., проф. Саченка А.О. та завідувача кафедри спеціалізованих комп'ютерних систем, д.т.н., проф. Николайчука Я.М. склали цей акт про те, що в навчальному процесі факультету комп'ютерних інформаційних технологій ТНЕУ для студентів напряму підготовки "Комп'ютерна інженерія" впроваджені та використовуються наступні результати дисертаційної роботи Бикового П.Є. при викладанні дисциплін:

1. "Мікроконтролери"
   - Проектування простих мікроконтролерних пристроїв
   - Проектування спеціалізованих послідовних інтерфейсів
2. "Мікроконтролери і спецпроцессори"
   - Проектування пристроїв на базі 8-ми бітних мікроконтролерів
   - Проектування спеціалізованих комп'ютерних мереж на базі 8-ми бітних мікроконтролерів
   - Шифрування даних в комп'ютерних мережах та базі 8-ми бітних мікроконтролерів
3. "Системи передавання даних"
   - Методи забезпечення стійкості послідовних інтерфейсів до зламу
   - Засоби створення комп'ютерних мереж на базі немережевих інтерфейсів

Ефект від використання результатів дисертаційної роботи Бикового П.Є. полягає у підвищенні якості вивчення майбутніми фахівцями сучасних методів проектування простих мікроконтролерних пристроїв, спеціалізованих комп'ютерних мереж та методів шифрування даних, покращенні якості курсового та дипломного проектування, що в результаті забезпечує підвищення рівня підготовки спеціалістів з напрямку "Комп'ютерна інженерія" професійного спрямування «Спеціалізовані комп'ютерні системи».

Декан факультету комп'ютерних
інформаційних технологій, д.т.н., професор — Дивак М.П.

Завідувач кафедри
інформаційно-обчислювальних
систем та управління, д.т.н., професор — Саченко А.О.

Завідувач кафедри
спеціалізованих комп'ютерних систем,
д.т.н., професор — Николайчук Я.М.

Додаток З
Модулі розв'язання багатокритеріальних задач оптимізації структури систем тривожної сигналізації

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Genetic_Algorithm
{
    [Serializable]
    public abstract class Paretable: IComparable
    {
        // Пристосованість даного розв'язку
        protected double fitness;

        public bool isBad(List<Paretable> list)
        {
            for (int i = 0; i < list.Count; i++)
                if (this.isBad(list[i]))
                    return true;
            return false;
        }

        public double[] normed_target_value;

        public double get_fitness()
        {
            return fitness;
        }

        public string target_string()
        {
            string s = "";
            for (int i=0; i<this.target().Length; i++)
                s += this.target()[i] + "\t";
            return s;
        }

        public int CompareTo(object other)
        {
            return this.get_fitness().CompareTo((other as Paretable).
get_fitness());
        }

        public void set_fitness(double r)
        {
            fitness = r;
        }
        public void inc_fitness()
        {
            fitness++;
        }

        // Значення векторної функції мети
        protected double[] tar;
        // Функція оцінює векторну цільову функцію хромосоми, що в той
        // чи інший спосіб характеризує її ефективність щодо розв'язання
        // даної задачі
        public abstract double[] target();
        public override string ToString()
        {
```

```
            return "target: Q=" + (this.target())[0] + ",R=" +
(this.target())[1] + ",C=" + (this.target())[2] + ", Fitness=" +
this.get_fitness();
        }

        static public bool operator <=(Paretable a, Paretable b)
        {
            if (a == null || b == null) return false;
            double[] av = a.target(), bv = b.target();

            for (int i = 0; i < av.Length; i++)
                if (av[i] > bv[i]) return false;

            return true;

        }
        static public bool operator >=(Paretable a, Paretable b)
        {
            double[] av = a.target(), bv = b.target();

            for (int i = 0; i < av.Length; i++)
                if (av[i] < bv[i]) return false;

            return true;

        }


        public bool isBad(Paretable b)
        {
            for (int i = 0; i < b.target().Length; i++)
                if (b.target()[i] > this.target()[i])
                    return false;
            for (int i = 0; i < b.target().Length; i++)
                if (b.target()[i] < this.target()[i])
                    return true;
            return false;
        }

    }

    [Serializable]
    public abstract class Chromosome : Paretable
    {
        abstract public Security_system getSecurity_system();
        abstract public string Short_system_description();
        abstract public string Long_system_description();
        // Булева змінна, що визначає чи обчислювалася функція мети
        // для даної хромосоми, щоб користуватися кешованим значенням
        protected bool evaluated;

        abstract public void GenRandomly(Random rnd);

        // Функція ініціалізації хромосоми
        //public abstract void init(int chrom_len);
        // Функція створення генів хромосоми за допомогою датчика
        // псевдовипадкових послідовностей
        //public abstract void create(Random rnd);
        // Функція схрещування, де батьками виступають дана хромосома
        // і хромосома задана в аргументі
        // parent, а на виході отримуємо хромосому-нащадка
        public abstract Chromosome crossover(Chromosome parent, bool[]
cross_points);
```

```
        public abstract int length();

        public Chromosome()
        {
            evaluated = false;
        }
    }


    [Serializable]
    class Variant_of_security : Chromosome
    {
        GA_PerimeterParetoZones g;

        public override Chromosome crossover(Chromosome parent, bool[]
cross_points)
        {
            // Генетичний код батьків записується в об'єктах масиву zones.
            // Тобто кожна зона з сенсорами, що її покривають -- це ген
            // хромосоми-
            // системи безпеки (Security system)

            // Створюємо генетичний код хромосоми-нащадка, що
            // спочатку містить порожню множину зон
            Variant_of_security child_cv =  new Variant_of_security(g);

            bool oc = false; // Порядок копіювання генів
            // Цикл по ділянках(зонах):
            for (int i = 0; i < var_indexes.Length; i++)
            {
                if (cross_points[i]) oc = !oc; // Коли точка схрещування -
                                               // змінюємо порядок копіювання
                if (oc)
                {
                    child_cv.var_indexes[i] = var_indexes[i];
                    //child_cv[1].var_indexes[i] = parent.var_indexes[i];
                }
                else
                {
                    //child_cv[1].var_indexes[i] = var_indexes[i];
                    child_cv.var_indexes[i] = (parent as
Variant_of_security).var_indexes[i];
                }
            }
            return child_cv;
        }

        public override double[] target()
        {
            return this.getSecurity_system().target();
        }

        public override string Long_system_description()
        {
            return Short_system_description();
        }

        public override string Short_system_description()
        {
            return getSecurity_system().Short_system_description();
        }

        public override int length()
        {
```

```
            return var_indexes.Length;
        }

        public override void GenRandomly(Random rnd)
        {
            flag_computed = false;
            for (int i = 0; i < var_indexes.Length; i++)
                var_indexes[i] = rnd.Next(0,
g.list_zone_variants_Pareto[i].Count);
        }

        Security_system Security_system_template;

        private int[] var_indexes; // Закодована система безпеки


        public void set_var_indexes(int i, int v)
        {
            flag_computed = false;
            var_indexes[i] = v;
        }

        bool flag_computed = false;

        Security_system internal_security_system;

        public override Security_system getSecurity_system()
        {
            if (flag_computed) return internal_security_system;

            Security_system ss = new Security_system(Security_system_template);

            for (int i = 0; i < var_indexes.Length; i++)
                ss.SetZone(i, g.list_zone_variants_Pareto[i][var_indexes[i]] as
Zone);
            internal_security_system = ss;
            flag_computed = true;
            return ss;
        }


        public Variant_of_security(GA_PerimeterParetoZones g)
        {
            this.g = g;
            Security_system_template = g.GetSecuritySystem();
            var_indexes = new int[Security_system_template.GetZonesCount()];
        }
    }


    [Serializable]
    public class Security_system : Chromosome
    {
        public override Chromosome crossover(Chromosome parent, bool[]
cross_points)
        {
            return this.crossover_per_zone(parent as Security_system,
cross_points);
        }

        public override bool Equals(object obj)
        {
            int[] a = this.getZoneConfigurationCode();
            int[] b = (obj as Security_system).getZoneConfigurationCode();
```

```
    if (a.Length != b.Length) return false;

    bool f = true;
    for (int i=0; i < a.Length; i++)
        if (a[i] != b[i])
        {
            f = false; break;
        }
    return f;
}

public override Security_system getSecurity_system()
{
    return this;
}

// Система безпеки периметру території - це сукупність всіх зон, що
// обладнані сенсорами, які відслідковують загрози в них.
ArrayList zones; // Сукупність зон разом з сенсорами,
                 // якими вони обладнані


public void AddZone(Zone z)
{
    zones.Add(z);
}

public void FillFromMarket(Sensor[] sensors_at_market)
{
    // Заповнити ділянки (зони) сенсорами, що доступні на ринку
    // і покликані відслідковувати загрози, що можуть
    // виникати на ділянках
    for (int i = 0; i < zones.Count; i++)
        (zones[i] as Zone).FillFromMarket(sensors_at_market);
}

public int[] getZoneConfigurationCode()
{
    // Описує стрічкою інтежерів систему безпеки
    int length = 0;
    for (int i = 0; i < this.GetZonesCount(); i++)
        length += this.GetZone(i).GetSensors_in_use_Count();

    int [] res = new int[length];

    int k =0;

    for (int i = 0; i < this.GetZonesCount(); i++)
        for(int j=0; j< this.GetZone(i).GetSensors_in_use_Count(); j++)
            res[k++] = this.GetZone(i).GetSensor_in_use(j).GetCount();

    return res;
}


public override string Short_system_description()
{
    string s = "";

    int[] rr = getZoneConfigurationCode();

    for (int i = 0; i < rr.Length; i++)
        s += rr[i] + "\t";
```

```
            s += "% target: ";
            double []tt= target();
            for (int i = 0; i < tt.Length; i++)
                s += tt[i] + ", ";
            return s + "\r\n";
        }

        public override string Long_system_description()
        {
            string str = ""; int i;
            for (i = 0; i < this.GetZonesCount(); i++)
                str += "" + i + ". " + this.GetZone(i).ToString();
            str += "\r\n";

            for (i = 0; i < this.GetZonesCount(); i++)
                str += this.GetZone(i).SensorsTableCell();

            str += "\r\n Perevirka: \r\n";
            int[] rr = getZoneConfigurationCode();

            for (i = 0; i < rr.Length; i++)
                str += rr[i] + "\t";
            str += "\r\n";

            return str;
        }



        public Security_system(Zone[] zz) : this()
        {
            // Створити систему безпеки, що складається з заданого масиву зон
            for (int i = 0; i < zz.Length; i++)
                zones.Add(zz[i]);
        }

        public Security_system()
        {
            // Створюємо порожню множину ділянок (зон):
            zones = new ArrayList();
        }

        public int GetZonesCount() { return zones.Count; /* Кількість зон */ }

        public override int length()
        {
            return GetZonesCount();
        }

        public Security_system crossover_per_zone(Security_system parent, bool[]
cross_points)
        {
            // Генетичний код батьків записується в об'єктах масиву zones.
            // Тобто кожна зона з сенсорами, що її покривають -- це ген
            //  хромосоми-
            // системи безпеки (Security system)
            // Створюємо генетичний код хромосоми-нащадка, що
            // спочатку містить порожню множину зон
            Security_system child_cv = new Security_system();

            bool oc = false; // Порядок копіювання генів
            // Цикл по ділянках(зонах):
            for (int i = 0; i < this.length(); i++)
            {
```

```
            if (cross_points[i]) oc = !oc; // Коли точка схрещування -
                                           // змінюємо порядок копіювання
            if (oc)
                child_cv.AddZone(this.GetZone(i));
            else
                child_cv.AddZone(parent.GetZone(i));
        }
        return child_cv;
    }

    public double GetProbabilityOfIncorrectAlarm()
    {
        // Імовірність хибного спрацьовування дорівнює
        // середньому імовірностей хибного спрацювання по зонах
        double q = 0;
        // Цикл по ділянках(зонах):
        for (int i = 0; i < this.GetZonesCount(); i++)
            q += this.GetZone(i).GetProbabilityOfIncorrectAlarm();
        return q / this.GetZonesCount();
    }

    public double GetProbabilityOfIndetection()
    {
        /* Імовірність виявлення загрози R

        double r_add = 0;

        for (int i = 0; i < this.GetZonesCount(); i++)
        {
            if ( this.GetZone(i).GetProbabilityOfIndetection() > r_add )
                r_add = this.GetZone(i).GetProbabilityOfIndetection();
        }

        return 1 - this.GetProbabilityOfFunctioning() + r_add;
    }

    public double GetProbabilityOfFunctioning()
    {
        double r = 1;
        for (int i = 0; i < this.GetZonesCount(); i++)
        {
            r*=this.GetZone(i).GetProbabilityOfFunctioning();
        }

        return r;
    }


    public double GetCost()
    {
        double c = 0;
        for (int i = 0; i < this.GetZonesCount(); i++)
            c += this.GetZone(i).GetCost();
        return c;
    }

    public override double[] target()
    {
        if (evaluated == false)
        {
            if (tar== null)
                tar = new double[3];
            tar[0] = this.GetProbabilityOfIncorrectAlarm();
            tar[1] = this.GetProbabilityOfIndetection();
```

```
                tar[2] = this.GetCost();


                evaluated =true;
            }
            return tar;
        }

        public Zone GetZone(int i) { return zones[i] as Zone; /* Отримати i-ту
зону */ }

        public void SetZone(int i, Zone z)
        {
            zones[i] = z;
        }

        public Security_system(Security_system s) : this()
        {
            // Створити систему безпеки, що є точною копією заданої
            for (int i = 0; i < s.GetZonesCount(); i++)
                zones.Add(new Zone(s.GetZone(i)));
        }

        public void SortSensorsInZonesByRange()
        {
            // Сортувати сенсори, що можуть бути використані
            // для відслідковування загрози на кожній зоні по
            // радіусу дії
            for (int i = 0; i < zones.Count; i++)
                (zones[i] as Zone).SortSensorsByRange();
        }

        public override void GenRandomly(Random rnd)
        {
            // Згенерувати випадково систему безпеки,
            // що покриває всі наявні на ній ділянки
            for (int i = 0; i < zones.Count; i++)
                (zones[i] as Zone).Gen_Randomly(rnd);
        }
    }
}


namespace Genetic_Algorithm
{
    [Serializable]
    public class GA_Perimeter_security : CAlgorithm
    {

        protected int
         N_immigrants, // кількість хромосом-емігрантів на кожній генерації
         N_init; // об’єм початкової популяції розв’язків.

        protected Random rnd; // Об'єкт для генерації псевдовипадкових
                              // послідовностей


        public Population get_population() { return pop; }

        [NonSerializedAttribute]
        protected TextBox debug;
        [NonSerializedAttribute]
        protected TextBox tb;
```

```
        [NonSerializedAttribute]
        protected ListBox lb;
        [NonSerializedAttribute]
        protected ListBox lb2;


        void norm(int j, double[,] a, double fmax, double fmin)
        {
            int i;
            norm(j, a);
            for (i = 0; i < a.GetLength(0); i++)
            {
                a[i, j] *= (fmax-fmin);
                a[i, j] += fmin;
            }

            FileStream fo = new FileStream("t.txt",
FileMode.OpenOrCreate,FileAccess.Write,FileShare.ReadWrite);

            StreamWriter bw = new StreamWriter(fo);

            String longStr ="";
            for (i = 0; i < a.GetLength(1); i++)
                longStr += i + "\t";
            longStr+="\r\n";

            for (i = 0; i < a.GetLength(0); i++)
            {
                for (int k = 0; k < a.GetLength(1); k++)
                {
                    longStr+=a[i, k] + "\t";
                }
                longStr+= "\r\n";
            }
            bw.WriteLine(longStr);
            fo.Close();
        }

        public void norm(int j, double[,] a)
        {
            double []acol = new double[a.GetLength(0)];
            for (int i =0; i<a.GetLength(0);i++)
                acol[i]=a[i,j];

            double amin=acol.Min(), amax=acol.Max();
            for (int i = 0; i < a.GetLength(0); i++)
            {
                a[i, j] -= amin;
                a[i, j] /= amax - amin;
            }
        }

        Security_system SS_TEMPLATE;

        public Security_system GetSecuritySystem()
        {
            return SS_TEMPLATE;
        }

        protected override void start()
        {
            int i, After_remove = 0; // Змінна After_remove визначає скільки
            // ітерацій пройшло з часу останнього видалення надлишкових
            // розв'язків
```

```
            // з метою пришвидшення швидкодії ГА.
            /* Ініціалізуємо генератор випадкових чисел в
             * початковий стан, щоб можна було легко оцінювати
             * різні вдосконалення генетичного алгоритму не залежачи
             * від "випадковостей". */
            rnd = new Random(0);

            // Створюємо порожню популяцію розв'язків
            pop = new Population();

            // Зв'язуємо генератор псевдовипадкових чисел з
            // об'єктом популяції
            pop.set_random(rnd);

            // Створюємо початкову популяцію розв'язків,
            // де N_init - об'єм початкової популяції розв'язків.
            for (i = 0; i < N_init; i++)
            {
                Security_system ss = new Security_system(SS_TEMPLATE);
                ss.GenRandomly(rnd);
                pop.Add(ss);
            }

            // Основний цикл ГА
            for (CURRENT_ITERATION = 1; !isItTimeToFinish();
CURRENT_ITERATION++)
            {
                //lb.Items.Add(i.ToString());
                pop.evaluation(); // Оцінюємо пристосованість початкової
                                  // популяції
                // pop.Exclude_bad(); // Виключаємо з популяції всі хромосоми,
                // Ймовірність вибору яких менша за 0.001
                pop.crossover(); // Створюємо нові хромосоми через кросовер
                // Добавляємо в популяцію імігрантів
                for (int j = 0; j < N_immigrants; j++)
                {
                    Security_system ss = new Security_system(SS_TEMPLATE);
                    ss.GenRandomly(rnd);
                    pop.Add(ss);
                }
                // Посортувати популяцію по пристосованості в порядку її
               // спадання
                pop.Sort(); pop.Reverse();
                Security_system bc = pop[0] as Security_system;
                // Виводимо дані про "найкращий" розв'язок-хромосому,
                // що знаходиться на вершині посортованої популяції
                if (lb != null)
                {
                    lb.Items.Add(CURRENT_ITERATION.ToString() + ". " +
bc.ToString() + ", Population value=" + pop.Count);
lb.SetSelected(lb.Items.Count - 1, true);
                }

                updateProgressBar();

                // Кожні 20 кроків видаляємо надлишкові розв'язки
                if (After_remove > 20)
                {
                    //pop.RemoveRange(200, pop.Count - 200);
                    pop.OnlyPareto(null);
                    After_remove = 0;
                }

                After_remove++;
```

```
            }
        }
    }
}

namespace Genetic_Algorithm
{
    [Serializable]
    public class GA_PerimeterParetoZones : GA_Perimeter_security
    {

        public long NUMBER_OF_ALL_ITERATIONS()
        {
            long NAI = 1;
            for (int i = 0; i < this.list_zone_variants_Pareto.Length; i++)
                NAI *= this.list_zone_variants_Pareto[i].Count;

            return NAI;
        }

        public string NUMBER_OF_ALL_ITERATIONS_formula()
        {
            string NAI = "NUMBER_OF_ALL_ITERATIONS_formula=";
            for (int i = 0; i < this.list_zone_variants_Pareto.Length; i++)
                NAI += this.list_zone_variants_Pareto[i].Count + "*";

            return NAI+"\r\n";
        }

        bool b_avoid_the_same_results_at_every_iteration = false;

        public void setB_avoid_the_same_results_at_every_iteration(bool b)
        {
            b_avoid_the_same_results_at_every_iteration = true;
        }

        protected override  void  start()
        {
            int i, After_remove = 0; // Змінна After_remove визначає скільки
            // ітерацій пройшло з часу останнього видалення надлишкових
           // розв'язків
            // з метою пришвидшення швидкодії ГА
            //DateTime strt_time = DateTime.Now;
            /* Ініціалізуємо генератор випадкових чисел в
             * початковий стан, щоб можна було легко оцінювати
             * різні вдосконалення генетичного алгоритму не залежачи
             * від "випадковостей". */
            rnd = new Random(0);

            // Створюємо порожню популяцію розв'язків
            pop = new Population();

            // Зв'язуємо генератор псевдовипадкових чисел з
            // об'єктом популяції
            pop.set_random(rnd);

            // Створюємо початкову популяцію розв'язків,
            // де N_init - об'єм початкової популяції розв'язків.
            for (i = 0; i < N_init; i++)
            {
                Variant_of_security ss = new Variant_of_security(this);
                ss.GenRandomly(rnd);
                pop.Add(ss);
            }
```

```
            // Основний цикл ГА
            for (CURRENT_ITERATION = 1; !isItTimeToFinish();
CURRENT_ITERATION++)
            {
                if (b_avoid_the_same_results_at_every_iteration)
                    pop = pop.AvoidTheSameRes();
                pop.evaluation(); // Оцінюємо пристосованість початкової
                                  // популяції
                pop.crossover(); // Створюємо нові хромосоми через кросовер

                // Добавляємо в популяцію імігрантів
                for (int j = 0; j < N_immigrants; j++)
                {
                    Variant_of_security ss = new Variant_of_security(this);
                    ss.GenRandomly(rnd);
                    pop.Add(ss);
                }
                // Посортувати популяцію по пристосованості в порядку її
                // спадання
                pop.Sort(); pop.Reverse();

                Variant_of_security bc = pop[0] as Variant_of_security;
                // Виводимо дані про "найкращий" розв'язок-хромосому,
                // що знаходиться на вершині посортованої популяції
                if (lb != null)
                {
                    lb.Items.Add(this.CURRENT_ITERATION.ToString() + ". " +
bc.ToString() + ", Population value=" + pop.Count);
lb.SetSelected(lb.Items.Count - 1, true);
                }

                updateProgressBar();
                // Кожні 20 кроків видаляємо надлишкові розв'язки
                if (After_remove > 20)
                {
                    pop.OnlyPareto(null);
                    After_remove = 0;
                }

                After_remove++;
            }
        }

        [NonSerializedAttribute] TextBox textBox2;

        static void rec(int i, double sizeX, int[] count, double[] ranges,
TextBox tb, List<int[]> list_counts)
        {
            if (i > count.Length - 1) return;
            count[i] = (int)Math.Ceiling(sizeX / ranges[i]);
            for (int k = i + 1; k < count.Length; k++)
                count[k] = 0;
            print(ranges, count, tb, list_counts);
            do
            {
                count[i]--;
                rec(i + 1, sizeX - sum(i, ranges, count), count, ranges, tb,
list_counts);
            } while (count[i] > 0);
        }

        static double sum(double[] a, int[] count)
        {
```

```
            return sum(0, a, count);
        }

        static double sum(int s, double[] a, int[] count)
        {
            double r = 0;
            for (int i = s; i < a.Length; i++)
                r += a[i] * count[i];
            return r;
        }

        static void print(double[] ranges, int[] a, TextBox tb, List<int[]>
list_counts)
        {
            list_counts.Add((int[])a.Clone());
        }

        public List<Paretable>[] list_zone_variants_Pareto;

        public GA_PerimeterParetoZones(TextBox h, bool improved_exuhasive,
TextBox tb, ListBox lb, ListBox lb2)
            : base(h, tb, lb, lb2)
        {
            textBox2 = h;

            bool debuging_info = false;
            if (debuging_info)
                this.textBox2.Text = "";


            Security_system s = this.GetSecuritySystem();

            list_zone_variants_Pareto = new List<Paretable>[s.GetZonesCount()];

            for (int i = 0; i < s.GetZonesCount(); i++)
            {
                Zone z = s.GetZone(i);

                if (debuging_info)
                    this.textBox2.Text += "Zone " + i + " has " +
z.GetZoneLength() + " meters\r\n";

                double[] ranges = new double[z.GetSensors_in_use_Count()];
                for (int j = 0; j < ranges.Length; j++)
                {
                    ranges[j] =
z.GetSensor_in_use(j).GetSensorObject().GetRange();
                    if (debuging_info)
                        this.textBox2.Text += "" + ranges[j] + "\t";
                }
                if (debuging_info)
                    this.textBox2.Text += "\r\n";

                List<int[]> list_counts = new List<int[]>();
                List<Paretable> list_zone_variants = new List<Paretable>();
                rec(0, z.GetZoneLength(), new int[ranges.Length], ranges,
this.textBox2, list_counts);

                foreach (int[] count in list_counts)
                {
                    Zone za = new Zone(z);
                    for (int j = 0; j < count.Length; j++)
                        za.GetSensor_in_use(j).SetCount(count[j]);
```

```
                    list_zone_variants.Add(za);
                }

                list_zone_variants_Pareto[i] = new List<Paretable>();

                if (improved_exuhasive)
                    foreach (Zone za in list_zone_variants)
                    {

                        if (za.isBad(list_zone_variants))
                            continue;

                        list_zone_variants_Pareto[i].Add(za);
                        if (debuging_info)
                        {
                            textBox2.Text += "" + " count=[";
                            for (int j = 0; j < za.GetSensors_in_use_Count();
j++)
                                textBox2.Text += "" +
za.GetSensor_in_use(j).GetCount() + "\t";
                            textBox2.Text += "] sum = " + za.TotallyCatching() +
"\t";
                            textBox2.Text += "ProbabilityOfIndetection = " +
za.GetProbabilityOfIndetection() +
                                " ProbabilityOfBreak = " + (1 -
za.GetProbabilityOfFunctioning()) +
                                " ProbabilityOfIncorrectAlarm=" +
za.GetProbabilityOfIncorrectAlarm() +
                                " Cost= " + za.GetCost() + "\r\n";


                            this.textBox2.SelectionStart = textBox2.Text.Length;
                            this.textBox2.ScrollToCaret();
                        }
                    }
                else
                    list_zone_variants_Pareto[i] = list_zone_variants;
            }

            var_count = new int[s.GetZonesCount()];
            for (int i = 0; i < s.GetZonesCount(); i++)
                var_count[i] = list_zone_variants_Pareto[i].Count;
        }

        public int[] var_count;

    }
}


namespace Genetic_Algorithm
{
    [Serializable]
    public abstract class CAlgorithm
    {

        public bool F_COMPUTATION_TIME_CONSTRAINT = false;
        public double TIMECONSTRAINT = .3;

        public bool isItTimeToFinish()
        {
            if (!F_COMPUTATION_TIME_CONSTRAINT)
            {
                    return CURRENT_ITERATION > MAX_ITERARIONS;
```

```
            }
            else
                return (DateTime.Now - strt_time).TotalMinutes > TIMECONSTRAINT;
        }

        [NonSerializedAttribute] static BinaryFormatter bt = new
BinaryFormatter();
        public bool full_search; // Прапорець повного перебору: якщо ввімкнено,
            // то алгоритм переберe всю множину варіантів
        [NonSerializedAttribute] ProgressBar pb;
        void setProgressBar (ProgressBar v) { pb = v; }
        public Population pop;

        public void writePop(string fn)
        {
            FileStream fs = new FileStream(fn,
                FileMode.OpenOrCreate, FileAccess.Write);
            bt.Serialize(fs, this);
            fs.Close();
        }

        static public CAlgorithm readPop(string fn)
        {
            try
            {
                FileStream fs = new FileStream(fn,
                    FileMode.Open, FileAccess.Read);
                return (bt.Deserialize(fs) as CAlgorithm);
            }
            catch (Exception e)
            {
                return null;
            }
        }
        [NonSerializedAttribute] public Thread th;
        [NonSerializedAttribute]  protected CProgressForm frm;

        protected bool stop_is_pressed = false,
            progress_must_be_updated = true;
        public void setStop_is_pressed(bool b)
        {
            stop_is_pressed = b;
        }
        protected abstract void start();
        DateTime progressBar_last_updated;
        protected void updateProgressBar()
        {
            DateTime now = DateTime.Now;
            if ( (now - progressBar_last_updated).TotalSeconds < 0.5 )
                return;
            progressBar_last_updated = now;

            double time_elapsed = (DateTime.Now - strt_time).TotalMinutes;
            if (F_COMPUTATION_TIME_CONSTRAINT)
            {
                frm.setPercentage((int)((time_elapsed / TIMECONSTRAINT) * 100));
                frm.setLabel(""+time_elapsed);
            }
            else
            {
                frm.setLabel("" + CURRENT_ITERATION + " за " + time_elapsed + "
хв.");
                frm.setPercentage((int)(((double)CURRENT_ITERATION /
(double)MAX_ITERARIONS) * 100));
```

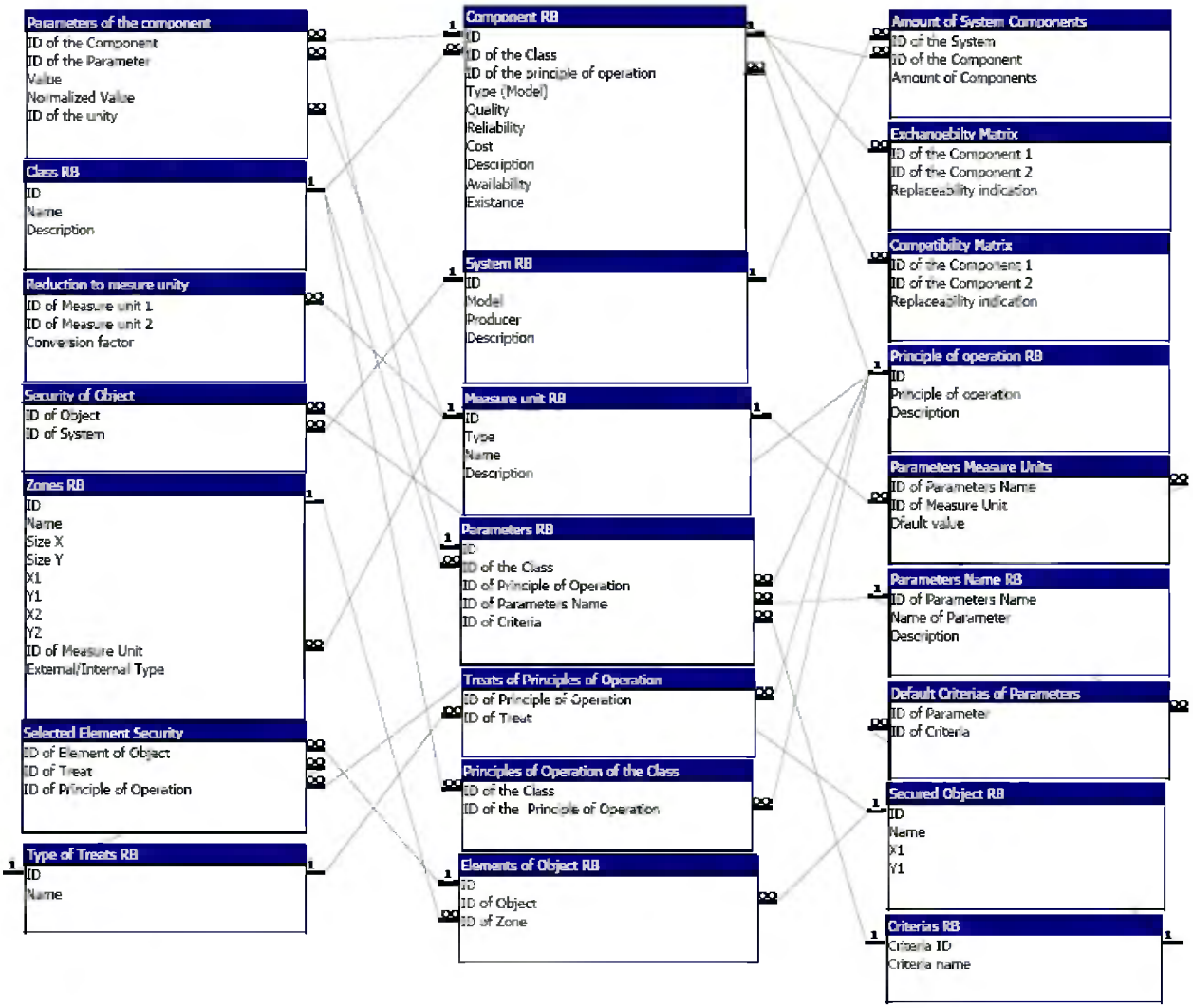
```
                frm.setOfLabel("Всього ітерацій: "+ MAX_ITERARIONS +
                    ". До завершення обчислень залишилось: " + ((MAX_ITERARIONS-
CURRENT_ITERATION)*time_elapsed/CURRENT_ITERATION) + "хв.");
            }
            doSave();
        }

        public long CURRENT_ITERATION,
                    MAX_ITERARIONS;// максимальна кількість ітерацій (генерацій)
                                   // генетичного алгоритму

        void start_thread()
        {

        }

        public void run(CProgressForm pf)
        {
            frm = pf;
            pb = frm.getProgressBar();
            if (F_COMPUTATION_TIME_CONSTRAINT)
                frm.setOfLabel("" + TIMECONSTRAINT);
            else
                frm.setOfLabel(""+MAX_ITERARIONS);

            strt_time = DateTime.Now;
            last_save = strt_time;
            progressBar_last_updated = strt_time;
            this.start();
            StopTimer();
            frm.setPercentage(100); // Finalize progress bar.
        }
        protected DateTime strt_time;
        protected bool AvoidTheSameRes_necessary = false;
        DateTime last_save;
        public List<Population> saves = new List<Population>();
        protected bool forming_Pareto = true;
        public void doSave()
        {
            if ((DateTime.Now - last_save).TotalMinutes < 10)
                return;
            last_save = DateTime.Now;

            if (forming_Pareto)
                pop.OnlyPareto(null); // Avoid the same results and forming
                                      // Pareto-set

            Population pop_copy = new Population();
            foreach (Paretable p in pop)
                pop_copy.Add(p);
            saves.Add(pop_copy);
        }

        public void StopTimer()
        {
            DateTime s0 = DateTime.Now;
            TimeElapsed_variants_generated = (s0 - strt_time).TotalMilliseconds;
            if (AvoidTheSameRes_necessary)
                pop = pop.AvoidTheSameRes();
            DateTime s1 = DateTime.Now;
            TimeElapsed_AvoidTheSameRes = (s1 - s0).TotalMilliseconds;
            pop.OnlyPareto(frm);
            DateTime s2 = DateTime.Now;
            TimeElapsed_FiltratingPareto = (s2 - s1).TotalMilliseconds;
```

```
            TimeElapsed_Total = (s2 - strt_time).TotalMilliseconds;
        }
        public double TimeElapsed_variants_generated,
TimeElapsed_AvoidTheSameRes,
            TimeElapsed_FiltratingPareto, TimeElapsed_Total;
    }
    [Serializable]
    public class ExuhasiveAlgorithm : CAlgorithm
    {
        public GA_PerimeterParetoZones ga; int[] var_indexes;
        bool debuging_info = false;
        [NonSerializedAttribute] TextBox textBox2, textBox3;
        bool cbParetoZones_EXUHASIVE_Checked, cb_leavePareto_EXUHASIVE_Checked;
        [NonSerializedAttribute] ListBox listBox2;
        public bool OPTIMIZED = true;
        public ExuhasiveAlgorithm(TextBox textBox1, TextBox textBox2, TextBox
textBox3, bool cbParetoZones_EXUHASIVE_Checked, bool
cb_leavePareto_EXUHASIVE_Checked, ListBox listBox1, ListBox listBox2, bool
full_search)

        {
            forming_Pareto = false; // Because exuhasive will add only Pareto-
            // optimal and unique results without need of any addtional effort.

            ga = new GA_PerimeterParetoZones(textBox2,
cbParetoZones_EXUHASIVE_Checked, textBox1, listBox1, listBox2);
            this.full_search = full_search;


            this.textBox2 = textBox2;
            this.textBox3 = textBox3;
            this.cbParetoZones_EXUHASIVE_Checked =
cbParetoZones_EXUHASIVE_Checked;

            if (cbParetoZones_EXUHASIVE_Checked)
                NUM_OF_IT_TO_MAKE_PARETO = 10;

            this.cb_leavePareto_EXUHASIVE_Checked =
cb_leavePareto_EXUHASIVE_Checked;
            this.listBox2 = listBox2;

            s = ga.GetSecuritySystem();

            var_indexes = new int[s.GetZonesCount()];

            CURRENT_ITERATION = 1;

            MAX_ITERARIONS = ga.NUMBER_OF_ALL_ITERATIONS();

            if (!full_search)
                MAX_ITERARIONS = Int64.Parse(textBox3.Text); // Змінна для
            //лімітування кількості варіантів системи
            // безпеки при повному переборі
            // Створюємо порожню популяцію розв'язків
            this.pop = new Population();
        }

        Security_system s;

        long NUM_OF_IT_TO_MAKE_PARETO = 10000;
        long NUM_OF_IT_BETWEEN_FORMING_TWO_PARETOS = 0;

        protected override void  start()
        {
```

```
            for(CURRENT_ITERATION=1;  !isItTimeToFinish(); CURRENT_ITERATION++)
            {
                if (CURRENT_ITERATION > MAX_ITERARIONS)
                    break;
                Security_system ss = new Security_system(s);
                for (int i = 0; i < var_indexes.Length; i++)
                    ss.SetZone(i,
ga.list_zone_variants_Pareto[i][var_indexes[i]] as Zone);
                if (OPTIMIZED)
                    pop.addIfNotBad(ss);
                else
                    pop.Add(ss);
                if (debuging_info)
                {
                    for (int j = 0; j < var_indexes.Length; j++)
                        textBox2.Text += var_indexes[j] + " ";
                    textBox2.Text += "\r\n";


                    textBox2.SelectionStart = textBox2.Text.Length;
                    textBox2.ScrollToCaret();
                }
                updateProgressBar();



                if (!OPTIMIZED && NUM_OF_IT_BETWEEN_FORMING_TWO_PARETOS >
NUM_OF_IT_TO_MAKE_PARETO)
                {
                    NUM_OF_IT_BETWEEN_FORMING_TWO_PARETOS = 0;
                    pop.OnlyPareto(null);
                }

                if (!OPTIMIZED)
                    NUM_OF_IT_BETWEEN_FORMING_TWO_PARETOS++;

                inc_var_indexes(var_indexes, ga.var_count, ga.var_count.Length -
1);
            }
        }

        // Processing a Pareto-set only if set is greater than 100 elements:
        public long MIN_SET_TO_PROCESS_PARETO = 100;

        bool inc_var_indexes(int[] var_indexes, int[] var_count, int i)
        {
            if (i < 0)
                return false;
            if (var_indexes[i] < var_count[i] - 1)
            {
                var_indexes[i]++;
                return true;
            }
            var_indexes[i] = 0;
            return inc_var_indexes(var_indexes, var_count, i - 1);
        }
    }
}
```

Додаток **И**

Загальна структура розробленої БД компонентів та СТС периметру території



Додаток **К**

Опис еталонної множини Парето-оптимальних СТС

№	К-сть сповіщувачів відповідного типу						Q_{sys}	R_{sys}	C_{sys}
	Тип 1	Тип 2	Тип 3	Тип 4	Тип 5	Тип 6			
1	10	5	0	0	0	0	0.250	0.125	6000.000
2	10	1	0	4	0	0	0.250	0.165	5544.000
3	9	1	2	4	0	0	0.209	0.166	5594.000
4	9	1	0	6	0	0	0.250	0.166	5466.000
...									
26	8	0	1	7	0	2	0.223	0.218	5452.000
27	8	0	0	8	0	2	0.250	0.218	5388.000
28	4	0	21	0	0	0	0.069	0.165	7050.000
29	3	0	24	0	0	0	0.055	0.167	7350.000
30	2	0	26	0	0	0	0.048	0.168	7400.000
...									
40	1	0	28	0	0	0	0.044	0.169	7450.000
41	0	0	31	0	0	0	0.030	0.171	7750.000
42	0	0	30	0	0	1	0.037	0.231	7650.000
43	1	0	27	0	0	1	0.051	0.229	7350.000
44	0	0	30	0	0	1	0.037	0.231	7650.000
45	2	0	24	1	0	0	0.062	0.177	7086.000
...									
80	8	0	3	6	0	1	0.195	0.218	5616.000
81	3	0	23	0	0	0	0.062	0.166	7100.000
82	2	0	25	0	0	0	0.058	0.167	7150.000
83	15	0	0	0	0	0	0.250	0.035	6750.000
84	7	1	11	0	0	0	0.122	0.159	6200.000
85	7	1	11	0	0	0	0.122	0.159	6200.000
86	6	2	12	0	0	0	0.117	0.160	6300.000
87	6	2	12	0	0	0	0.117	0.160	6300.000
88	7	0	13	0	0	0	0.110	0.160	6400.000
89	7	0	13	0	0	0	0.110	0.160	6400.000
90	6	0	15	0	0	0	0.103	0.161	6450.000
...									
174	3	0	23	0	0	0	0.062	0.166	7100.000
175	2	0	25	1	0	0	0.054	0.178	7336.000
176	7	1	11	0	0	0	0.126	0.159	6200.000
177	9	0	8	0	0	0	0.140	0.157	6050.000
178	6	2	12	0	0	0	0.113	0.160	6300.000

Додаток **Л**

Описи покриття трьох периметрів

	Периметр 1	Периметр 2	Периметр 3
Кількість сповіщувачів	6 (200, 100,75,75,50,50)	5 (200, 100, 75, 75, 50)	6 (200, 100,75,75,50,50)
Кількість ділянок	4	5	8
Загальна протяжність ділянок	(124 + 298 + 299 + 299)	316+200+448 + 180 + 220	356 + 286 + 449 + 375 + 111 + 264 + 266 + 59
Кількість ітерацій повного перебору	$14119928 \approx 14 * 10^6$	43804800 $\approx 43 * 10^6$	603500919700000 $\approx 603 * 10^{12}$
Кількість ітерацій повного перебору оптимальних варіантів покриття ділянок	6000=6*10*10*10	9856=7*4*11*4*8	15288000 = 10*10*13*10*6*7*7*4
Оцінка часу виконання повного перебору	594156.25 мс = 9.9 хв.	1905968.75 мс = 31.8 хв.	740405066.5 хв = 1382 роки
Оцінка часу виконання повного перебору оптимальних варіантів покриття ділянок	281343.75 мс = 4.7 хв.	57312.5 мс. = 0.9 хв.	1447046.875 мс. = 24 хв.
Кількість Парето-оптимальних результатів в еталонній (або агрегованій) множині	236	77	178

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Абалмазов Э.И. Концепция безопасности: тактика высокоэффективной зашиты. Стоимость стратеги, стратегические ресурсы, тактика защиты, сопоставимость тактических решений / Э.И. Абалмазов // Системы безопасности. – №4. – М., Июнь-июль, 1995.
2. Адресний модуль “CA-64 ADR-MOD” [Електронний ресурс]. – Режим доступу http://www.satel.pl/ru/product/104/CA-64%20ADR-MOD
3. Алефельд Г. Введение в интервальные вычисления / Г. Алефельд, Ю. Херцбергер. – М. : Мир, 1987. – 356 с.
4. Алиев Р. А. Управление производством при нечеткой исходной информации / Р. А. Алиев, А. Э. Церковный, Г. А. Мамедова. – М. : Энергоатомиздат, 1991. – 240 с.
5. Андрианов В. И. Охранные устройства для дома и офиса / В. И. Андрианов, А. В Соколов. – СПб. : Изд. «Полигон», 2000. – 312 с.
6. Ашихмин В. Н. Введение в математическое моделирование / [В. Н. Ашихмин, М. Б. Гитман, И. Э. Келлер, О. Б. Неймарк, В. Ю. Столбов, П. В. Трусов, П. Г. Фрик] ; под ред. П. В. Трусова. – Москва : Университетская книга, Логос, 2007. – 440 с.
7. Белов А.В. Конструирование устройств на микроконтроллерах. – СПб.: Наука и Техника, 2005. – 256 с.
8. Биковий П. Застосування генетичних алгоритмів для оптимізації дистрибутивних систем технічної безпеки / П.Биковий // Матеріали IX Міжнародної конференції “Контроль і управління в складних системах (КУСС-2008)”. – Вінниця: Вінницький національний технічний університет, 2008. – С.6.
9. Биковий П.Є. Вибір техніко-економічних показників компонентів дистрибутивних систем безпеки периметра територій / П.Є. Биковий, В.В. Кочан, А.О. Саченко, В.О. Турченко // Вісник технологічного

університету Поділля. – Технічні науки. – 2004, Ч.1, Т.2, №2. – С. 82-85.

10. Биковий П. Апаратні засоби мережі сенсорів систем безпеки / П. Биковий // Збірник тез міжнародної науково-технічної конференції "Комп'ютерні системи та мережні технології". – Київ: Національний авіаційний університет, 2008. – С. 44-48.
11. Биковий П. Апаратні засоби мережі сенсорів систем безпеки / П. Биковий // Збірник тез міжнародної науково-технічної конференції "Комп'ютерні системи та мережні технології", Національний авіаційний університет. – Київ, 17-19 березня, 2008. – С.44-48.
12. Биковий П. Дистрибутивна сенсорна мережа для систем безпеки / П.Биковий // Міжнародний науково-технічний журнал "Комп'ютинг". – 2009. – Т.8. – Випуск 2. – С. 157-164.
13. Биковий П. Оптимізація проектування дистрибутивних систем технічної безпеки за допомогою генетичного алгоритму / П. Биковий // Вісник Вінницького політехнічного інституту. – Вінниця, 2008. – №6. – С. 28-34.
14. Биковий П. Порівняльний аналіз алгоритмів виявлення оптимальних рішень у системах безпеки / П. Биковий // Вісник Національного університету "Львівська політехніка" «Комп'ютерні системи та мережі». – 2008. – №630. – С. 17-23.
15. Биковий П. Розробка мережевого протоколу для сенсорів систем безпеки / П. Биковий; В. Кочан // Тези доповіді Всеукраїнської наукової конференції Тернопільського державного технічного університету імені Івана Пулюя. – Тернопіль, 2009. – С.102.
16. Биковий П.Є. Багатокритеріальний синтез систем безпеки периметру території в умовах невизначеного впливу завад / П. Є. Биковий, Ю. Р. Піговський // Труди XI міжнародної науково-практичної конференції «Сучасні інформаційні та електронні технології». – Одеса, 2010. – Т.1. – С. 88.
17. Биковий П.Є. Криптостійкий протокол для мереж сенсорів безпеки / П.Є. Биковий, В.В. Кочан // Збірник тез десятої міжнародної науково-

практичної конференції «Сучасні інформаційні і електронні технології ». – Одеса, 2009. – Т.1. – С.189.

18. Биковий П.Є. Система нечіткого виводу для оцінки ризиків вразливості систем безпеки периметру території / П.Є. Биковий // Вісник Хмельницького національного університету. Технічні науки. 2010. –№2 (146). – С.174-180.
19. Введенский Б.С. Современные системы охраны периметров / Б. С. Введенский [Електронний ресурс] // Специальная техника. – М. – 1999. – Ч.1. – №3. – С.24-29 с.; Ч.2. – №4, 34-39 с. – Режим доступу http://ess.ru/publications/ articles/vvedenskiy/perimetr.htm
20. Гарсиа М. Проектирование и оценка систем физической защиты / Гарсиа М. ; [Пер. с англ. В.И. Воропаева, Е.Е. Зудина, К.А. Костылева, Н. И. Баяндина под ред. Р. Г. Магауенова]. – М. : Мир и ООО «Издательство АСТ». – 2002. – 386 с. – (Серия: Технологии безопасности).
21. Гіпермаркет засобів безпеки [Електронний ресурс]. – Режим доступу http://www.bezpeka-shop.com/catalog/ moduli_rasshirenija/ca-64_adr-mod-p-7617.html, http://www.bezpeka-shop.com/catalog/moduli_rasshirenija/ca-64_adr-p-7616.html
22. Гмурман В. Е. Теория вероятностей и математическая статистика. – Изд. 4-е, доп. – М. : Высш. школа, 1972. – 368 с.
23. Грох М. Microsoft Office Access 2007. Библия пользователя / Грох М., Стокман Д., Пауэлл Г. – Київ : Діалектика, 2008. – 1200 с.
24. Группа компаний ИСТА [Електронний ресурс]. – Режим доступу http://www.ista.ru
25. Гуткин Л.С. Оптимизация радиоэлектронных устройств по совокупности показателей качества / Гуткин Л.С. – М. : Советское радио, 1975. – 368 с.
26. Дамьяновски В. CCTV. Библия охранного телевидения / Дамьяновски В. ; [Пер. с англ.]. – М. : ООО «ИСС», 2002. – 352 с.
27. Дейт К. Дж.. Введение в системы баз данных / Дейт К. Дж.; Пер. с англ. –

8-е изд. – М. : Вильямс, 2006. – 1328 с.

28. Джонс Дж. К. Методы проектирования / Джонс Дж. К. ; [Пер с англ.]. – М. : Мир, 1986. – 326 с. – (2е. изд., доп.).
29. Долгов В.И. Принципы защиты алгоритма DES от атак дифференциального криптоанализа / В. И. Долгов, И. В. Лисицкая, С. А. Головашич, Р. В. Олейников // Радиотехника. – М. – 2000. – № 113. – С. 148-157.
30. Домарев В. В. Безопасность информационных технологий. Методология создания систем защиты / Домарев В. В. – К. : ООО "ТИД "ДС", 2002. – 688 с.
31. Дружинин Г.В. Надежность автоматизированных систем. – Изд. 3-е, перераб. и доп. – М. : Энергия, 1977. – 536 с.
32. ДСТУ 3960-2000 Системи тривожної сигналізації. Системи охоронної і охоронно-пожежної сигналізації. Терміни та визначення. – Київ: Державний комітет стандартизації метрології та сертифікації України, 2000. – 43 с.
33. ДСТУ IEC 60839-1-1-2001. Системи тривожної сигналізації. Частина 1. Загальні вимоги. Розділ 1. Загальні принципи. Держстандарт України, Київ. 2002. – 18 с.
34. Звежинский С. С. Классификации, особенности и информационно-измерительные модели средств обнаружения / С. С. Звежинский, В. А. Иванов, В. А. Рудниченко // Специальная техника. – М. – № 6. – 2007. – с. 26-33.
35. Иванов И.В. Охрана периметров / И. В. Иванов – М. : Радио и связь, 1997. – 120 с.
36. Иванов И.В. Охрана периметров-2 / И. В. Иванов – М. : Паритет Граф, 2000. – 196 с.
37. Извещатели охранные радиоволновые линейные “FMW-3” [Електронний ресурс]. – Режим доступу http://www.forteza.ru/files/index.php?id=84
38. Извещатели радиоволновые двухпозиционные “БАРЬЕР-300/500”

[Електронний ресурс]. – Режим доступу http://www.forteza.ru/catalogue/index.php?id=6

39. Извещатель охранный линейный радиоволновый “ЛИНАР 200” [Електронний ресурс]. – Режим доступу http://idsas.ru/pozharnaja-bezopasnost/pasport/linar_200_instrukcija.pdf
40. Интегрированная система охраны “Орион” [Електронний ресурс]. – Режим доступу http://www.bolid.ru/production/devices/
41. Інтерфейс 1-Wire фірми MicroLan [Електронний ресурс]. – Режим доступу ftp://ftp.elin.ru/pdf/1-Wire/standard.pdf
42. Карпова Т. С. Базы данных: модели, разработка, реализация / Т. С. Карпова – СПб. : Питер, 2002. – 304 с.
43. Каторин Ю. Ф. Большая энциклопедия промышленного шпионажа / Ю. Ф. Каторин, Е. В. Куренков, А. В. Лысов, А. Н. Остапенко. – СПб. : Изд. «Полигон». – 2000. – 629 с.
44. Колдасов Г.Д. Оптимизация решений при морфологическом синтезе технических систем / Г. Д. Колдасов // Механизация и автоматизация управления. – Киев: Украинские республиканские правления ВНТО приборостроителей им. С. И. Вавилова и ВНТО радиотехники, электроники и связи им. А. С. Попова, 1987. – № 3, – С. 3-6.
45. Комплекс “ВЕГА – 2” // ФГУП СНПО “Елерон” [Електронний ресурс]. – Режим доступу http://www.eleron.ru/vega2.html
46. Коннолли Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг ; Пер. с англ. – 3-е изд. – М. : Издательский дом “Вильямс”, 2003. – 1440 с.
47. Контроллер двухпроводной линии связи “С2000-КДЛ” [Електронний ресурс]. – Режим доступу http://www.bolid.ru/production/devices/devices_40.html
48. Кофейников Ю. К. Методы и анализ уязвимости объекта (текущее состояние) // Сборник материалов 1-го отраслевого совещания руководителей подразделений безопасности нефтеперерабатывающих и

химических предприятий России и СНГ, 2000. – С. 34-38.

49. Кочан В. В. Электрические измерители температуры повышенной точности со встроенными калибраторами: Автореферат диссертации на соискание ученой степени кандидата технических наук: 05.11.05, Киевский политехнический институт. – Киев, 1989. – 18 с.

50. Магауенов Р. Г. Системы охранной сигнализации: основы теории и принципы построения: учебное пособие / Рауль Габиденович Магауенов. – М. : Горячая линия, 2004. – 367 с.

51. Магауенов Р. Г. Системы охранной сигнализации: основы теории и принципы построения: учебное пособие / Р. Г. Магауенов. – 2-е изд., перераб. и. доп. – М. : Горячая линия, Телеком, 2008. – 496 с.

52. Майків І. М. Оптимізація послідовних інтерфейсів реалізованих на ПЛМ / І. М. Майків, В. В. Кочан // Вісник Хмельницького національного університету. – №2. – Том 1. Хмельницький, 2007. – С.117-122.

53. Модуль розширення адресних зон “CA-64 ADR” [Електронний ресурс]. – Режим доступу http://www.satel.pl/ru/product/103/CA-64%20ADR

54. Надежность в технике. Термины. ГОСТ 13377-75. – М. : Изд-во стандартов, 1975. – 21 с.

55. Николайчук Я. М. Теорія моделей руху даних розподілених комп’ютерних систем / Я. М. Николайчук, І. Р. Пітух, Н. Я. Возна. – Тернопіль: ТзОВ “Терно-граф”, 2008. – 216 с.

56. Норенков И. П. Основы автоматизированного проектирования / И. П. Норенков. – 2-е изд., перераб. и доп. – М. : Изд-во МГТУ им. Н.Э.Баумана, 2002. – 336 с.

57. Одрин В. М. Метод морфологического анализа технических систем. Курс лекций по программе «совершенствование творческой деятельности в процессе создания новых технических решений» / В. М. Одрин – М. : ВНИИПИ, 1989. – 312 с.

58. Одрин В. М. Метод морфологического анализа технических систем. – М.: ВНИИПИ, 1989. – 312 с.

59. Панин О. А. Анализ эффективности интегрированных систем безопасности: принципы, критерии, методы // Журнал “Системы безопасности”. – № 2. – М., 2006. – С. 60-62.

60. Панин О. А. Оптимизация параметров систем охранной сигнализации как задача многокритериального выбора / О. А. Панин, С. И. Журин // Защита информации. Конфидент. – №1. – СПб., 2004. – С. 84 – 87.

61. Пападимитриу Х. Комбинаторная оптимизация: алгоритмы и сложность / Х. Пападимитриу, К. Стайглиц. – М. : Мир, 1985. – С. 446-460.

62. Пасічник В. В. Організація баз даних та знань / В. В. Пасічник, В.А. Резніченко. – К. : Видавнича група BHV, 2006. – 384 с.

63. Патент 25609А України, МКІ G06F 15/00. Двопровідна локальна обчислювальна мережа, повторювач сигналу та інвертор для використання в ній / В. В. Кочан, В. О. Тимчишин (Україна); Заявл. 30.10.97 № 97105295; Видано 30.10.98.

64. Подиновский В.В. Парето-оптимальные решения многокритериальных задач / В. В. Подиновский, В. Д. Ногин. – М. : Наука, Гл. ред. физ.-мат. лит., 1982. – 256 с.

65. Половко А. М. Основы теории надежности. Практикум / А. М. Половко, С. В. Гуров. – СПб. : БХВ-Петербург, 2006. – 560 с.

66. Радиолокационная система охраны периметра и территории объектов (РЛС) “Orwell-R” [Електронний ресурс]. – Режим доступу http://elvees.ru/index.php?id=234&L=0

67. Ротштейн А. П. Нечеткий многокритериальный анализ вариантов с применением парных сравнений / А. П. Ротштейн, С. Д. Штовба // Известия РАН. Теория и системы управления. – 2001. – № 3. – С. 150-154.

68. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилинский, Л. Рутковский; пер. с польск. И.Д. Рудинского. – М.: Горячая линия – Телеком, 2006. – 452 с.

69. Седов В. В. Экономическая теория, Макроэкономика, Микроэкономика /

В. В. Седов. – Челябинск : ЧГУ, 2002. – 345 с.

70. Семенов А. Б. Проектирование и расчет структурированных кабельных систем и их компонентов / А. Б. Семенов. – М. : ДМК Пресс; М.: Компания АйТи. – 416 с.
71. Справочник конструктора РЭА: Общие принципы конструирования / под ред. Р. Г. Варламова. – М. : Сов. радио, 1980. – 480 с.
72. Тимчишин В. О. Підвищення ефективності проектування спеціалізованих комп'ютерних систем на базі типових мікропроцесорних платформ: Автореф. дис. канд. техн. наук: 05.13.13 [Електронний ресурс] / В. О. Тимчишин; Держ. ун-т "Львів. політехніка". – Л., 1999. – 19 с.
73. Турченко В. Оптимізація дистрибутивних сенсорних систем безпеки / В. Турченко, В. Кочан, П. Биковий, А. Саченко // Матеріали дев'ятої наукової конференції Тернопільського державного технічного університету імені Івана Пулюя. – Тернопіль, 2005. – С. 69.
74. Турченко В. Підхід до оптимізації дистрибутивних сенсорних систем безпеки / В. Турченко, В. Кочан, П. Биковий, А. Саченко, В. Коваль, Дж. Марковський // Вісник Тернопільського державного технічного університету імені Івана Пулюя. – 2005. – Т.10, №3. – С. 111-117.
75. ФГУП СНПО "ЕЛЕРОН" [Електронний ресурс]. – Режим доступу http://www.eleron.ru/development.html
76. Фірма Atmel / 2325 Orchard Parkway, San Jose, CA 95131 [Електронний ресурс]. – Режим доступу www.atmel.com.
77. Характеристики сповіщувача "Barrier-300" [Електронний ресурс]. – Режим доступу www.centers.ru/catalog/perimeter/russia/umirs/barier_300.htm
78. Характеристики сповіщувача "SPEC-5" [Електронний ресурс]. – Режим доступу http://spec.ru/product_about.php?p_s=1&&prod=sp5
79. Черноруцкий И. Г. Методы оптимизации и принятия решений / И. Г. Черноруцкий. – СПб. : Лань, 2001. – 384 с.
80. Чирков А. Ю. О многокритериальной задаче целочисленного линейного

программирования / А. Ю. Чирков, В. Н. Шевченко, Н. Ю. Золотых // Дискретный анализ и исследование операций. – Серия 2. – Том 12. – № 2. – Новосибирск: Изд. института математики им. С. Л. Соболева СО РАН, 2005. – С. 72-84.

81. Шепитько Г. Е. Проблемы охранной безопасности объектов / Г. Е. Шепитько. – М. : Русское право, 1995. – 352 с.

82. Alves M. J. Stability analysis of efficient solutions in multiobjective integer programming: A case study in load management / Maria João Alves, João Clímaco, Carlos Henggeler Antunes, Humberto Jorge, António G. Martins // Computers and Operations Research. – Vol. 35. – Iss. 1. – January 2008. – P. 186-197.

83. Andrés-Toro B. Multiobjective optimization and multivariable control of the beer fermentation process with the use of evolutionary algorithms / B. de Andrés-Toro, J. M. Girón-Sierra, P. Fernández-Blanco, J. A. López-Orozco, E. Besada-Portas // Journal of Zhejiang University Science. – 2004. – № 5 (4). – P. 378–389.

84. Application note AVR308. Software LIN slave. – Atmel, Rev. 1637B-AVR-05/02.

85. Application note AVR322: LIN v1.3 Protocol Implementation on Atmel AVR Microcontrollers. – Atmel, Rev. 7548A-AVR-12/05 [Електронний ресурс]. – Режим доступу http://www.atmel.com/dyn/resources/prod_documents/doc7548.pdf

86. Bohun Sean C. Modelling quality and warranty cost / C. Sean Bohun, Olivier Dubois, Hongbin Guo [et al.] // Canadian Applied Mathematics Quarterly. – Volume 12. – Number 1. – Spring 2004. – P. 37-66.

87. Bykovyy P. A CAD System That Optimizes Distributed Sensor Networks for Perimeter Security / P. Bykovyy, V. Kochan, A. Sachenko, G. Markowsky // Proceedings of the Second IEEE International Conference on Technologies for Homeland Security and Safety. – Istanbul, 2006. – P. 271-276.

88. Bykovyy P. A Low-Cost Network Controller for Security Systems /

P. Bykovyy, I. Maykiv, I. Turchenko, O. Kochan, V. Yatskiv, G. Markowsky // Proceedings of the Third IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'05). – Sofia, 2005. – P. 388-391.

89. Bykovyy P. Data Communication Crypto Protocol for Security Systems Sensor Networks / P. Bykovyy, V. Kochan, Y. Kinakh, A. Sachenko, O. Roshchupkin, S. Aksoy, G. Markowsky // Proceedings of the 5-th IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'09). – Rende (Cosenza), 2009. – P. 375-379.
90. Bykovyy P. Development of the Knowledge Base of Perimeter Security Systems / P. Bykovyy // Proceedings of the 2nd International IEEE Conference "Intelligent Systems". – Varna, 2004. – Vol. 3. – P. 54-57.
91. Bykovyy P. Fuzzy Inference System for Vulnerability Risk Estimation of Perimeter Security / P. Bykovyy, Y. Pigovsky, A. Sachenko, A. Banasik // Proceedings of the 5-th IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'09). – Rende (Cosenza), 2009. – P. 380-384.
92. Bykovyy P. Genetic Algorithm Implementation for Distributed Security Systems Optimization / P. Bykovyy, Y. Pigovsky, V. Kochan, A. Sachenko, G. Markowsky, S. Aksoy // IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA). – Istanbul, 2008. – P. 120-124.
93. Bykovyy P. Genetic Algorithm Implementation for Perimeter Security Systems CAD / P. Bykovyy, V. Kochan, A. Sachenko, G. Markowsky // 4-th IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'07). – Dortmund, 2007. – P. 634-638.
94. Carrillo-Ureta G. E. Genetic Algorithms for Optimal Control of Beer Fermentation / G. E. Carrillo-Ureta, P. D. Roberts, V. M. Becerra //

Proceedings of the IEEE International Symposium on Intelligent Control, September 5–7. – México City, 2001. – P. 391–396.

95. Crow Electronic Engineering Inc., NJ 07024, USA [Електронний ресурс]. – Режим доступу http://www.crowelec.com.

96. Daponte P. Artificial Neural Networks in Measurements / P. Daponte, D. Grimaldi // Measurement. – Vol. 23. – 1998. – P. 93-115.

97. Deb K. Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems // Evolutionary Computation. – USA, Massachusetts : Massachusetts Institute of Technology, 1999. – № 7 (3). – P. 205–230.

98. EVOCOM – Evolutionary Computation Matlab toolbox [Електронний ресурс]. – Режим доступу http://www.dacya.ucm.es/evocom/

99. FIPS PUB 140-1, Security Requirements for Cryptographic Modules, National Institute of Standards and Technology. – January 11, 1994.

100. Fire & Security Products // Siemens Building Technologies. – Edition: 10.2002 [Електронний ресурс]. – Режим доступу http://siemens.picon.ru/ir80/Product_release_pack_09-2002.doc

101. Fonseca C. M. Genetic algorithms for multiobjective optimization: formulation, discussion, and generalization / C. M. Fonseca, P. J. Fleming // Genetic Algorithms: Proceedings of the Fifth International Conference; [Forrest S. ed.]. – 1993. – P. 416-423.

102. Goldberg D. E. Genetic algorithms in search, optimisation, and machine learning / D. E. Goldberg. – MA (USA) : Addison-Wesley Publishing Company, 1989. – 432 p.

103. GUARD64 – програма для адміністрування системи безпеки на базі охоронної централі CA-64 фірми Satel [Електронний ресурс]. – Режим доступу http://www.satel.pl/ru/product/381/GUARD64

104. Kochan R. Development of a Dynamically Reprogrammable NCAP / R. Kochan, K. Lee, V. Kochan, A. Sachenko // Proceedings of the IEEE Instrumentation and Measurement Technology Conference IMTC/2004, May

18-20, Como, Italy, 2004. – P. 1188-1193.

105. Kochan V. Construction of Distributed Information Measurement Systems on the Basis of Modified RS-232C Interface / V. Kochan, V. Tymchyshyn // Proc.of the 10th IMEKO TC-4 Symposium on Development in Digital Measuring Instrumentation. – Naples, Italy, 1998. – P. 723-726.
106. Kohda T. Risk-based approach to design and maintenance of alarm systems for security / T. Kohda, K. Inoue // Proceedings of the International Carnahan Conference on Security Technology. – 1993. – P. 254-258.
107. LIN Specification Package, Revision 2.0: LIN Consorttium. – September 23, 2003 [Електронний ресурс]. – Режим доступу http://www7.informatik.uni-erlangen.de/~dulz/fkom/06/Material/3/LIN%20Specification%20Package.pdf
108. Logan T. Intel 980x Gulftown. Overclock3D LTD, March 2010 / Tom Logan [Електронний ресурс]. – Режим доступу http://www.overclock3d.net/reviews/cpu_mainboard/intel_980x_gulftown/4
109. Maners M. A. Design and maintenance of perimeter security cameras / M. A. Maners // Border & Transportation Security. – May 2003 [Електронний ресурс]. – Режим доступу http://www.gdsinternational.com
110. Maykiv I. The software-hardware method implementation of serial interfaces // Proceedings of 9-th International Conference Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET'2008). – Lviv-Slavsko, Ukraine, 19-23 February, 2008. – P. 439-441.
111. Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs / Z. Michalewicz. – New York : Springer-Verlag, 1992. – 387 p.
112. Nuclear Power Plant Security Assessment Technical Manual // Report of Sandia National Laboratories, Sept. 2007 [Електронний ресурс]. – Режим доступу http://www.prod.sandia.gov/cgi-bin/techlib/access-control.pl/2007/075591.pdf
113. Optex – manufacturer of infrared detection devices for the security industry [Електронний ресурс]. – Режим доступу http://www.optexeurope.com/
114. Paret D. Multiplexed Networks for Embedded Systems CAN, LIN, Flexray,

Safe-by-Wire... / Dominique Paret, Roderick Riesco. – John Wiley & Sons, 2007. – 434 p.

115. PELCO – manufacturer of video security systems [Електронний ресурс]. – Режим доступу http://www.surveillance-video.com/pelworleadma.html
116. Perimeter security [Електронний ресурс]. – Режим доступу http://www.perimetr.ru/secur200002.html
117. Perimeter Security Sensor Technologies Handbook [Електронний ресурс] / Electronic Security Systems Engineering Division. – North Charleson. – South Carolina, 1997. – 107 p. – Режим доступу http://www.nlectc.org/ perimetr/Hb-Word.doc.
118. Runkler T. A. Extended defuzzification methods and their properties / T. A. Runkler // Proceedings of the Fifth IEEE International Conference on Fuzzy Systems. – 1996. – P. 694-700.
119. Schleicher M. Digital Interfaces and Bus Systems for Communication / Manfred Schleicher, Blasinger Franc. – Publisher: M.K. JUCHHEIM GmbH & Co, 2005. – 153 p.
120. SMP Service [Електронний ресурс]. – Режим доступу http://www.tral.ru/
121. Sourd F. Multi-objective branch-and-bound. Application to the bi-objective spanning tree problem / F. Sourd, O. Spanjaard, P. Perny // Proceedings of 7th Int. Conf. on Multi-Objective Programming and Goal Programming. – Loire Valley (City of Tours), France, June 12-14, 2006. – 5 p.
122. Tarr C. J. CLASP: a computerised aid to cost effective perimeter security / C. J. Tarr // IEEE Proceedings of the International Carnahan Conference on Security Technology: Crime Countermeasures. – Atlanta, GA, USA. – Oct 1992. – P. 164-168.
123. Tarr C. J. Cost effective perimeter security / C. J. Tarr // IEEE Proceedings of the 28th International Carnahan Conference on Security Technology. – Albuquerque, NM, USA. – Oct. 1994. – P. 60-65.
124. Tarr C. J. Cost effective perimeter security. European Convention on Security and Detection / C. J. Tarr // Police Sci. Dev. Branch, Home Office, London

(Brighton). – May 1995. – P. 183-187.

125. Tarr C. J. The validity of security risk assessment / C. J. Tarr, P. Kinsman // IEEE Proceedings of the 30th International Carnahan Conference on Security Technology. – Oct. 1996. – P. 167-170.

126. Turchenko I. Database Design for CAD System Optimising Distributed Sensor Networks for Perimeter Security / I. Turchenko, V. Turchenko, V. Kochan, P. Bykovyy, A. Sachenko, G. Markowsky // Proceedings of the 8th IASTED International Conference Software Engineering and Applications. – Cambridge, 2004. – P. 59-64.

127. Xiao J. Ant colony system algorithm for the optimization of beer fermentation control / J. Xiao, Z.- K. Zhou, G.- X. Zhang // Journal of Zhejiang University Science. – № 5 (12). – 2004. – P. 1597-1603.