

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя  
(повне найменування вищого навчального закладу)

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(назва факультету)

Кафедра комп'ютерних систем та мереж  
(повна назва кафедри)

## КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістр

(назва освітнього ступеня)

на тему: Математичне та програмне забезпечення технологій  
проектування кабельних мереж

Виконав: студент 6 курсу групи СІм-61

спеціальності \_\_\_\_\_

123 «Комп'ютерна інженерія»

(шифр і назва спеціальності (напряму підготовки))

Митчик І.М.

(підпис)

(прізвище та ініціали)

Керівник

Баран І.О.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Тиш Є.В.

(підпис)

(прізвище та ініціали)

Завідувач

кафедри

Осухівська Г.М.

(підпис)

(прізвище та ініціали)

Рецензент

Дячук С.Ф.

(підпис)

(прізвище та ініціали)

Міністерство освіти і науки України  
**Тернопільський національний технічний університет імені Івана Пулюя**

Факультет комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету)

Кафедра комп'ютерних систем та мереж

(повна назва кафедри)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

доц. Осухіська Г.М.

(підпис)

(прізвище та ініціали)

«    »

20\_\_ р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

за спеціальністю

123 Комп'ютерна інженерія

студенту

Митчику Іллі Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема роботи

Математичне та програмне забезпечення технологій

проекування кабельних мереж

Керівник роботи

Баран Ігор Олегович, к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом по університету від «28» жовтня 2021 року № 4/7-916

2. Термін подання студентом роботи 21.12.2021

3. Вихідні дані до роботи

наукові літературні джерела

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1 Аналітична частина. 2 Теоретична частина. 3. Практична частина.

4 Охорона праці та безпека в надзвичайних ситуаціях

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Тема, мета, задачі, об'єкт, предмет дослідження, актуальність. 2. Аспекти проектування

електричних кабельних мереж. 3 Фрагменти плану кабельної розкладки та кабельного журналу

4. Етапи трасування. 5. Визначення відстані від вершини до фрагмента, побудова шляху

6. Побудова дерев Штейнера. 7. Блок-схема алгоритму побудови дерева Штейнера

8. Порівняння можливостей досліджених САПР. 9. Програмна реалізація та результати роботи.

10. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Осухівська Г.М., доцент	01.11.21	07.12.21
Безпека в НС	Стадник І. Я., професор	01.11.21	09.12.21

7. Дата видачі завдання \_\_\_\_\_ 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Затвердження теми кваліфікаційної роботи	28.10.21	Виконано
2	Аналіз літературних джерел	29.10.21-18.11.21	Виконано
3	Обґрунтування актуальності дослідження	18.11-21.11.21	Виконано
4	Аналіз предмету дослідження та предметної області	21.11-26.11.21	Виконано
5	Проведення дослідження методів та засобів аналітичного опрацювання даних	22.11-30.11.21	Виконано
6	Оформлення розділу «Аналітична частина»	20.11-26.11.21	Виконано
7	Оформлення розділу «Теоретична частина»	27.11-02.12.21	Виконано
8	Оформлення розділу «Практична частина»	03.12-10.12.21	Виконано
9	Оформлення розділу «Охорона праці та безпека в надзвичайних ситуаціях»	26.11-12.12.21	Виконано
10	Нормоконтроль	11.12-14.12.21	
11	Попередній захист роботи	15.12.21	Виконано
12	Захист кваліфікаційної роботи	22.12.21	

Студент \_\_\_\_\_  
(підпис)

*Митчик І.М.*  
\_\_\_\_\_  
(прізвище та ініціали)

Керівник роботи \_\_\_\_\_  
(підпис)

*Баран І. О.*  
\_\_\_\_\_  
(прізвище та ініціали)

## АНОТАЦІЯ

Математичне та програмне забезпечення технологій проектування кабельних мереж // Кваліфікаційна робота за освітнім рівнем «магістр» // Митчик Ілля Миколайович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних систем та мереж, група СІм-61 // Тернопіль, 2021 // с. – 71, рис. – 34 , табл. – 1 , аркушів А1 – 10 , бібліогр. – 28.

Ключові слова: КАБЕЛЬНА МЕРЕЖА, ТРАСУВАННЯ, САПР, AUTOCAD, ДЕРЕВА ШТЕЙНЕРА

Кваліфікаційна робота присвячена дослідженню математичних основ і програмних засобів проектування кабельних мереж.

Проведений аналіз процесу проектування електричних кабельних мереж, виконано програмна реалізація алгоритму трасування і інтеграція отриманих рішень в графічну САПР для побудови плану кабельної розкладки. Тим самим розроблено методичний підхід для забезпечення інформаційного супроводу процесу проектування електричних кабельних мереж. Запропоновано принципово новий підхід, який передбачає мінімізацію довжини з'єднань при побудові плану кабельної розкладки на основі алгоритму побудови мінімальних дерев Штейнера.

Запропонований підхід дозволить оптимізувати протяжність кабелю і кабельних конструкцій, тим самим знизивши матеріальні витрати на будівельно-монтажні роботи; підвищити продуктивність праці проектувальника; зменшити кількість помилок при проектуванні; підвищити якість документації продукції, що випускається.

## ANNOTATION

Mathematical and software means for cable network design // Master thesis // Mytchyk Illia // Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Systems and Nets, group CIm - 61 // Ternopil, 2021 // p. – 71, fig. – 34 , table. – 1, Sheets A1 - 10 , Ref. - 28.

Keywords: CABLE NETWORK, TRACE, CAD, AUTOCAD, STEINER TREES

The thesis deals with the research of mathematical bases and software of designing of cable networks.

The analysis of the process of designing electric cable networks is carried out, the software implementation of the tracing algorithm and the integration of the obtained solutions into the graphic CAD for the construction of the cable layout plan are performed. Thus, a methodological approach has been developed to provide information support for the process of designing electrical cable networks. A fundamentally new approach is proposed, which involves minimizing the length of connections when constructing a cable layout plan based on the algorithm for constructing minimal Steiner trees.

The proposed approach will optimize the length of the cable and cable structures, thereby reducing material costs for construction and installation work; increase the productivity of the designer; reduce the number of design errors; to improve the quality of product documentation.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ

CAD (САПР) - Computer-Aided Design (система автоматизованого проектування)

LISP (LISt Processing – обробка списків) – мова програмування загального призначення з підтримкою парадигм функціонального та процедурного програмування

AIBC КОЕ - автоматизована інформаційно-вимірювальна система комерційного обліку електроенергії

АСУ ТП - автоматизована система управління технологічним процесом

БД – база даних

ВРП - відкритий розподільний пристрій

ГЕК - граф елементних комплексів

ГКС - граф комутаційної схеми

ЗГС - зважений граф схеми

ЗРУ - закритий розподільний пристрій

РЗА - релейний захист і автоматика

КЖ – кабельний журнал

КМ – кабельна мережа

МС – монтажна схема

НКП - низьковольтний комплектний пристрій

ОПС - охоронно-пожежна сигналізація

ПА - протиаварійна автоматика

ПЕКМ - проектування електричних кабельних мереж

ПС - підстанція

ПУЕ - правила улаштування електроустановок

СЕП – схема електрична принципова

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ СКОРОЧЕНЬ І ТЕРМІНІВ .....	6
ВСТУП.....	9
РОЗДІЛ 1. АНАЛІТИЧНА ЧАСТИНА .....	11
1.1. Проблематика процесу ПЕКМ.....	11
1.2. Маршрут проектування КМ .....	12
1.3. Методика ПЕКМ .....	15
1.4. Висновки до розділу .....	22
РОЗДІЛ 2. ТЕОРЕТИЧНА ЧАСТИНА .....	23
2.1. Алгоритмізація задачі ПЕКМ .....	23
2.2. Трасування дротових з'єднань .....	26
2.3. Побудова дерев Штейнера .....	30
2.4. Хвильовий алгоритм .....	39
2.5. Алгоритм Дейкстри.....	41
2.6. Висновки до розділу .....	42
РОЗДІЛ 3. ПРАКТИЧНА ЧАСТИНА. ....	43
3.1 Інструментальні засоби проектування КМ.....	43
3.1.1. Model Studio CS Кабельне господарство .....	43
3.1.2. САПР ЦВК .....	44
3.1.3. Eplan FieldSys.....	46
3.1.4. КОМПАС-Графік .....	47
3.1.5. E3.Series .....	50
3.1.6. AutoCAD Electrical .....	51
3.1.7. ADVJToolkit.....	51
3.2. Порівняльний аналіз САПР .....	52
3.3. Програмна реалізація .....	54
3.3.1. Модуль трасування .....	56

3.3.2. Інтеграція з графічною САПР.....	58
3.3. Результати роботи .....	60
3.4. Висновки до розділу .....	61
РОЗДІЛ 4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	62
4.1. Охорона праці.....	62
4.2. Комп'ютерне забезпечення процесу оцінки радіаційної та хімічної обстановки.....	65
4.3. Висновки до розділу .....	67
ВИСНОВКИ.....	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	69
ДОДАТОК А. Тези конференції	
ДОДАТОК Б. Програмний код реалізації алгоритму побудови мінімальних дерев Штейнера	



## ВСТУП

**Актуальність теми.** ПЕКМ, їх розкладка і трасування є однією з найбільш важливих і трудомістких завдань при проектуванні об'єктів електроенергетики, зокрема електричних ПС. Як правило, ці роботи виконуються вручну фахівцем сектора первинних з'єднань проектної організації. Підрахунок довжин кабелів і траси розкладки вибираються з великим наближенням, що часто призводить до необгрунтованих перевитрат і здорожчення кабельної продукції на етапах закупівлі і будівництва об'єкта.

Кабельне господарство ПС об'єднує в собі силові кабелі, контрольні кабелі систем телемеханіки, РЗА, ОПС, зв'язку та ін. З урахуванням вимог ПУЕ, заповнюваності кабельних каналів, розташування обладнання, що накладають деякі обмеження на можливі варіанти розкладки кожного кабелю, завдання значно ускладнюється. Час розробки документації збільшується прямо пропорційно кількості кабелів на об'єкті.

**Метою дослідження** є розробка методичного підходу для забезпечення інформаційного супроводу процесу ПЕКМ.

В роботі поставлено та розв'язано **наступні задачі**:

- аналіз процесу проектування електричних ПС в цілому і електричних КМ з метою оцінки ступеня автоматизації проектних робіт;
- алгоритмічна реалізація завдання трасування КМ, як електричних з'єднань;
- програмна реалізація завдання трасування КМ, як електричних з'єднань;
- інтеграція рішення задачі трасування дротових з'єднань з завданням побудови плану КМ.

**Об'єкт дослідження:** процес проектування кабельних мереж.

**Предмет дослідження:** математичне та програмне забезпечення для проектування кабельних мереж.

**Методи дослідження:** Метод теоретичного дослідження та експериментальний з використання персонального комп'ютера. Методологічну основу дослідження становлять фундаментальні положення комп'ютерної та програмної інженерії, наукові дослідження вітчизняних і зарубіжних компаній та вчених у сфері трасування кабельних мереж.

**Наукова новизна отриманих результатів.**

- реалізовано об'єкт з новими властивостями;
- проведена інтеграція професійного інструменту проектування в вигляді графічної САПР з моделями і методом побудови мінімальних дерев дротових з'єднань для отримання оптимальної КМ.

**Практичне значення одержаних результатів.** Розробка і впровадження методики інформаційного супроводу процесу ПЕКМ дозволить оптимізувати протяжність кабелю і кабельних конструкцій, тим самим знизивши матеріальні витрати на будівельно-монтажні роботи; підвищити продуктивність праці проектувальника; зменшити кількість помилок; підвищити якість документації, котра випускається.

**Публікації.** Результати дослідження апробовано на ІХ науково-технічній конференції «Інформаційні моделі, системи та технології» у вигляді опублікованих тез:

Митчик І.М. До проблеми трасування проводового монтажу. Інформаційні моделі, системи та технології: Праці ІХ наук.-техн. конф. (Тернопіль, 08-09 грудня 2021 р.), Тернопіль, 2021. – С. 138.

**Структура роботи.** Робота складається з пояснювальної записки та графічної частини. Пояснювальна записка складається з вступу, 4 розділів, висновків, списку використаної літератури та додатків. Обсяг роботи: пояснювальна записка – 71 арк. формату А4, графічна частина – 10 аркушів формату А1.

# РОЗДІЛ 1

## АНАЛІТИЧНА ЧАСТИНА

### 1.1. Проблематика процесу ПЕКМ

Електричні КМ служать для передачі електроенергії на відстань. Вони використовуються як безпосередні силові магістралі енергетичних потоків (силові кабелі), або для роботи систем управління, захистів, автоматики, сигналізації і т. д. (контрольні кабелі).

Основним завданням ПЕКМ є забезпечення високого ступеня їх надійності функціонування в процесі експлуатації, а також економічності на етапах будівництва.

КМ промислових підприємств і, зокрема, об'єктів електроенергетики мають надзвичайно велику протяжність і номенклатуру кабельної продукції, точок підключення і т. д. Тому процес проектування КМ є одним з найважливіших і трудомістких завдань.

Як правило, робота з вибору марок, перерізів кабелів, розробка схем з'єднань, КЖ, планів розкладки кабелю, виконується фахівцями проектних інститутів вручну, без використання спеціальних програмних засобів автоматизації проектування (за винятком графічних САПР). Наприклад, підрахунок довжин кабелів і траси розкладки вибираються з великим наближенням, що часто призводить до необґрунтованих перевитрат і подорожчання кабельної продукції на етапах закупівельних процедур і будівельно-монтажних робіт на об'єкті.

З урахуванням вимог правил улаштування електроустановок (ПУЕ) [1], заповнюваності кабельних конструкцій, розташування основного і допоміжного електротехнічного обладнання, що накладають деякі обмеження на можливі варіанти розкладки кожного кабелю, задача значно ускладнюється. Час розробки документації збільшується прямо пропорційно кількості кабелів на об'єкті.

Імовірність виникнення помилок при такому підході також збільшується. Виникає проблема і необхідність автоматизації конкретних проектних робіт, пов'язаних з побудовою плану кабельної розкладки. Для цього необхідно, в основному, вирішити задачу трасування з'єднань для отримання оптимальної КМ. Оптимальність в даному випадку буде полягати в мінімізації довжин кабелів і кабельних конструкцій [2].

В якості вихідних даних для розв'язання задачі трасування з'єднань повинні бути перелік ланцюгів (кабелів), особливості конструкції елементів та комутаційного поля, розташування елементів.

В алгоритмічному плані задача трасування - це побудова для всіх ланцюгів схеми реальних оптимальних монтажних з'єднань. Розрізняють метричний і топологічний аспекти завдання трасування. Метричний аспект передбачає врахування конструктивних розмірів елементів, з'єднань і комутаційного поля. Топологічний аспект пов'язаний з вибором допустимого просторового розташування окремих монтажних з'єднань на комутаційному полі при обмеженнях на кількість перетинів з'єднань, число шарів комутаційної схеми тощо.

Варто відмітити, що існує значна відмінність у алгоритмічних методах трасування для дротових і друкованих з'єднань. Зокрема для проведеного монтажу трасування виконується з використанням алгоритмів створення мінімальних дерев з'єднань. Повна МС (таблиця проводів) виходить при послідовному застосуванні зазначених алгоритмів для окремих ланцюгів схеми. Далі, на підставі аналізу паразитних зв'язків в отриманій МС траси окремі з'єднання можуть бути скориговані.

## 1.2. Маршрут проектування КМ

Проектування є набором технічних робіт для одержання описів нового чи вже модернізованого об'єкта, котрі будуть достатніми для реалізації або

виготовлення об'єкта у визначених умовах. Під час цього процесу з'являється потреба створення опису для побудови об'єкта, котрий ще не існує. Фінальні описи є комплектом завершеної конструкторсько-технологічної документації, котрий містить специфікації, креслення і т.д. В даному випадку, при ПЕКМ, вихідними документами є план розташування обладнання і проводок, а також КЖ.

Типовими аспектами в описах технічних об'єктів з точки зору теорії автоматизованого проектування є: функціональний, конструкторський та технологічний [2].

Таким чином, можна розділити процес проектування КМ за аспектами.

Функціональний - функціональні вимоги до окремих систем об'єкта, списки сигналів, вибір перетинів, типу, кількості кабелю, моделювання (перевірка і розрахунок струмів короткого замикання і т. д.);

Функціональні вимоги визначають набір функцій системи, від цього залежить склад пристроїв, отже, потреба в контрольних, інтерфейсних і кабелях живлення.

Згідно зі списком сигналів АСУ ТП, що передаються «сухим контактом» (безпосередньо від контрольованих пристроїв і апаратів до контролерів збору даних), визначаються вимоги до кількості жил кабелю (кожен сигнал передається окремою жилою) і точок підключення (шаф, збірок, апаратів) [3].

Конструкторський - розробка СЕП, МС, принципово-монтажних схем, схем з'єднань зовнішніх проводок, плану КМ, КЖ.

На МС вказують геометричне положення всіх компонентів вузла, а також відображають з'єднання складових частин установки і визначають дроти, джгути, кабелі, якими здійснюються ці з'єднання.

Схема з'єднань зовнішніх проводок в спрощеному вигляді відображає кабельні зв'язки між окремими шафами, панелями, апаратами, містить номенклатуру кабелів, їх маркування і точки підключення.

Схеми під'єднання рядів затискачів визначають безпосереднє під'єднання жил кабелів до рядів затискачів пристроїв, панелей, шаф.

План КМ відображає безпосередньо траси прокладки кабелю на плані об'єкта.

КЖ містить маркування, точки підключення (звідки-куди), марку, кількість жил, перетин, довжину кабелів. Всі ці дані зводяться в табличний вигляд.

Технологічний - проект організації будівництва в частині прокладки кабелю. Вказуються способи прокладки проводів і кабелів на лотках і в коробах (розсіпом, пучками, багатошарово і т.п.), технічні засоби і особливості монтажу.



Рис. 1.1. Аспекти ПЕКМ

З точки зору автоматизації конструкторський аспект проектування в регіоні практично не розвинений. Плани КМ виконуються фахівцями проектних інститутів вручну без використання спеціалізованих програмних засобів (за винятком графічних САПР).

Під електричною КМ ПС будемо розуміти сукупність кабелів і кабельних конструкцій, призначених для розподілу електроживлення власних потреб, освітлення, обігріву і т.д., а також збору і передачі сигналів приладів і пристроїв управління, електроавтоматики, блокування, вимірювання, захисту, сигналізації.

КМ може проходити як зовні по території ВРП, так і всередині приміщень ЗРП, ЗПУ і ін.

Для зовнішньої прокладки кабелю по ВРП використовуються кабельні короби і лотки.

Таким чином, створення плану КМ є завданням структурного синтезу з точки зору теорії автоматизованого проектування. Завдання структурного синтезу формулюється як задача прийняття рішень. Її суть полягає у визначенні мети, множини можливих рішень і обмежуючих умов [3].

Оформлення результатів проектування (проектної документації) -окремий випадок структурного синтезу.

### 1.3. Методика ПЕКМ

ПЕКМ є невід'ємною частиною процесу проектування всього енергооб'єкта, зокрема електричної підстанції. У загальному вигляді, процес проектування можна розбити на кілька етапів [4]:

1. Отримання завдання на проектування;
2. Вибір і узгодження схем електричних розподільних пристроїв ПС;
3. Вибір первинного електротехнічного обладнання;
4. Розробка плану ПС, на якому відображені всі будівлі, споруди, електричні апарати, зовнішні шафи;
5. Розробка планів будівель ЗПУ, закритого розподільчого пристрою (ЗРП) і т.д .;
6. Проектування заземлення, освітлення, власних потреб ПС;
7. Розробка функціональних і структурних схем для систем РЗА, АСУ ТП і ін;
8. Розробка МС для систем РЗА, АСУ ТП і ін. (рисунок 1.2);

9. Розробка МС панелей (рисунок 1.3), схем підключень клемних рядів (рисунок 1.4) і схем підключень зовнішніх проводок (рисунок 1.5) для окремих систем;

10. КЖ (рисунок 1.7) і план кабельної розкладки (рисунок 1.6).

Цей список не є вичерпним і може містити розробку додаткових комплектів документації. Послідовність етапів може бути змінена в залежності від конкретного об'єкта проектування і плану організації робіт у проектній організації, так само деякі пункти, які не залежать один від одного можливо виконувати паралельно [5].

На рисунку 1.2 в якості прикладу подано фрагмент принципово-монтажної схеми збору дискретних сигналів від шафи приводу бакового вимикача 110 кВ Siemens3AP1 DT в систему АСУ ТП через проміжну шафу затискачів ШЗ-3.

На даній схемі відображені номери клем підключення жил кабелю, ключі та допоміжні контакти реле в шафі приводу, маркування ланцюгів кабелю, а також клемники для підключення в проміжному шафі і шафі АСУ ТП.



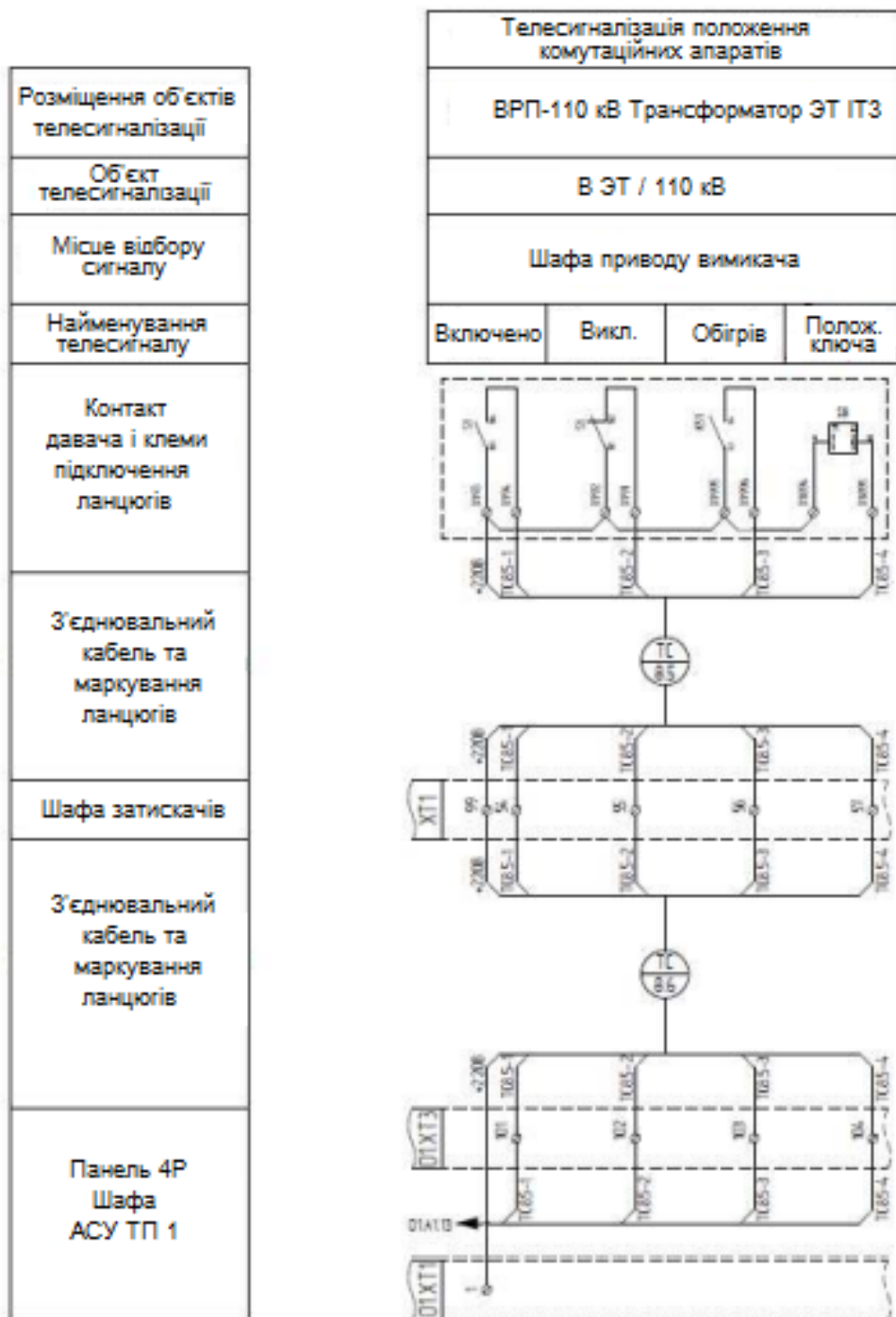


Рис. 1.2. Фрагмент принципово-монтажної схеми збору сигналів

Вихідною інформацією для такого виду схем є заводські схеми шаф, вимоги до номенклатури сигналів, які збираються.

Принципово-монтажна схема по формі аналогічна до СЕП, але додатково містить марки затискачів, провідників і кабелів.

МС розробляються для конкретних панелей і відображають з'єднання складових частин виробу (установки) і визначають дроти, кабелі, джгути, якими здійснюються ці з'єднання, а також місця або адреси їх приєднання до клемним затискачів або контактам пристроїв [7].

На рис.1.3 представлений фрагмент МС шафи пристрою зв'язку з об'єктом (ПЗО), в якому здійснюється збір дискретних сигналів і видача керуючих впливів за допомогою модулів введення-виведення А1, А2. На такій схемі відображається все основне обладнання, що входить до складу панелі і внутрішні зв'язки між елементами.

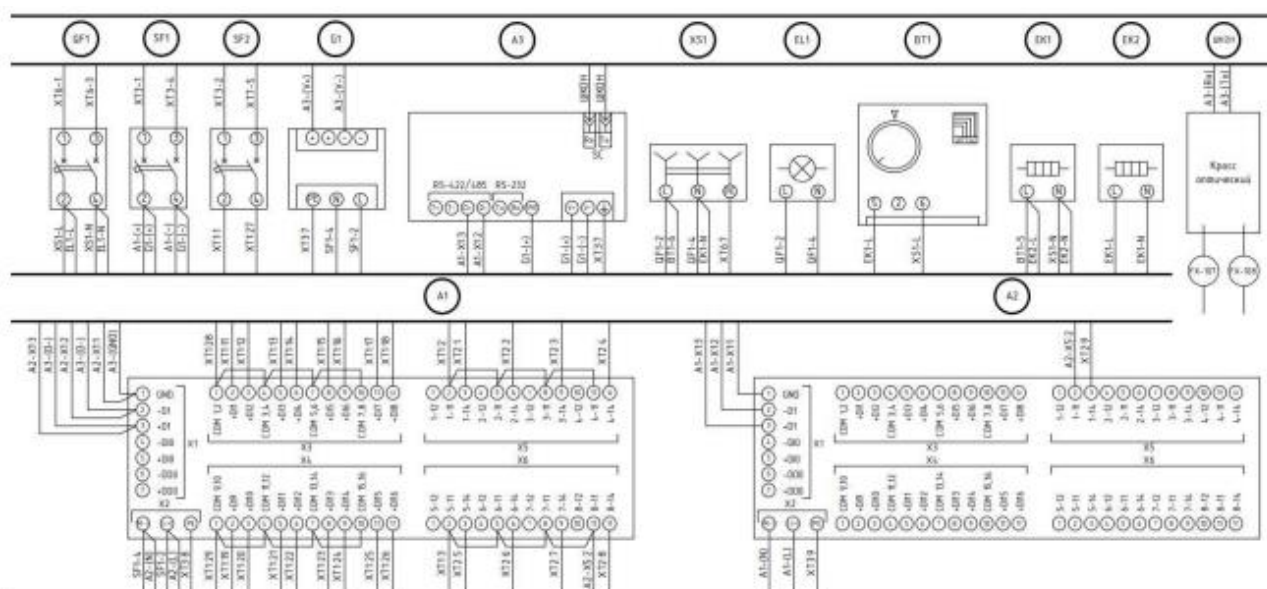


Рис. 1.3. Фрагмент МС шафи ПЗО

На МС шафи часто не відображаються зовнішні зв'язки, тому, для отримання повної і наочної картини для виробництва монтажу кабельної продукції необхідна розробка схем підключень клемних рядів, що відображають клемники шафи з кабелями, які під'єднуються до них.

На рис. 1.4 представлений фрагмент схеми підключень клемних рядів шафи проміжних затискачів. Тут вказується маркування кабелів, їх напрямки, адреси жил, котрі входять і виходять, а також маркування ланцюгів.

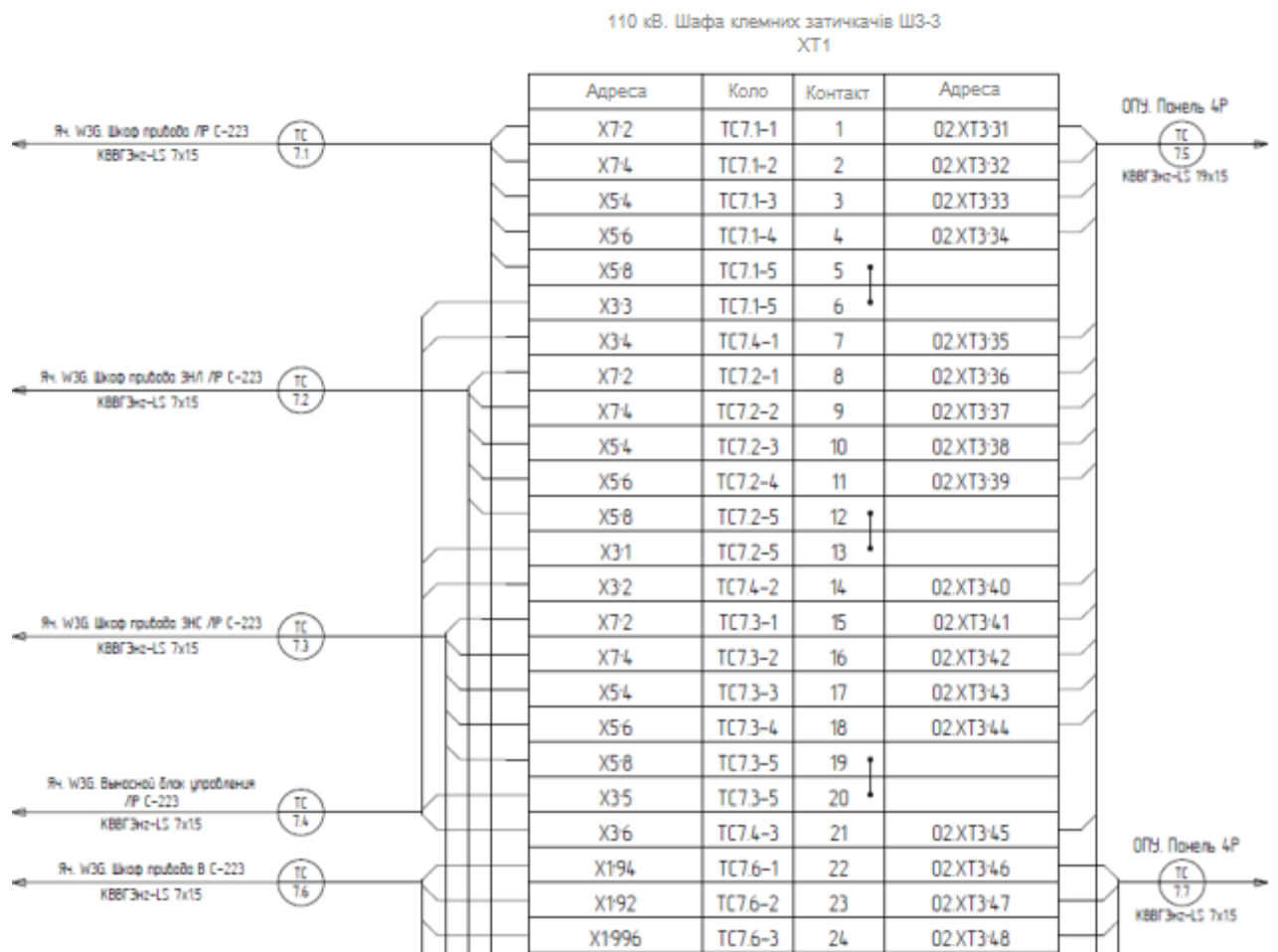


Рис. 1.4. Фрагмент схеми підключень клемних рядів

З представлених схем можна зробити висновок, що вони можуть містити інформацію, яка багаторазово повторюється і перетинається в різній формі, наприклад, принципово-монтажна схема і схема підключень клемних рядів, в кімнаті конференцій клем, марки ланцюгів і кабелю.

Багаторазова участь інформації в різних документах при ручному проектуванні призводить до великого обсягу рутинної роботи і є одним з основних джерел помилок. Так як процес проектування є ітераційним, то внесення змін до один документ вимагає переробки значної кількості взаємопов'язаних документів [8].

Автоматизація такого роду проектних робіт зводиться до автоматизації оформлення документації і ніяк не зачіпає процес синтезу конструкції [9].

Так само існують схеми з'єднань зовнішніх проводок (рисунок 1.5), на якій спрощено в структурному вигляді показані точки підключення і маркування кабелів без прив'язки до координат плану об'єкта.

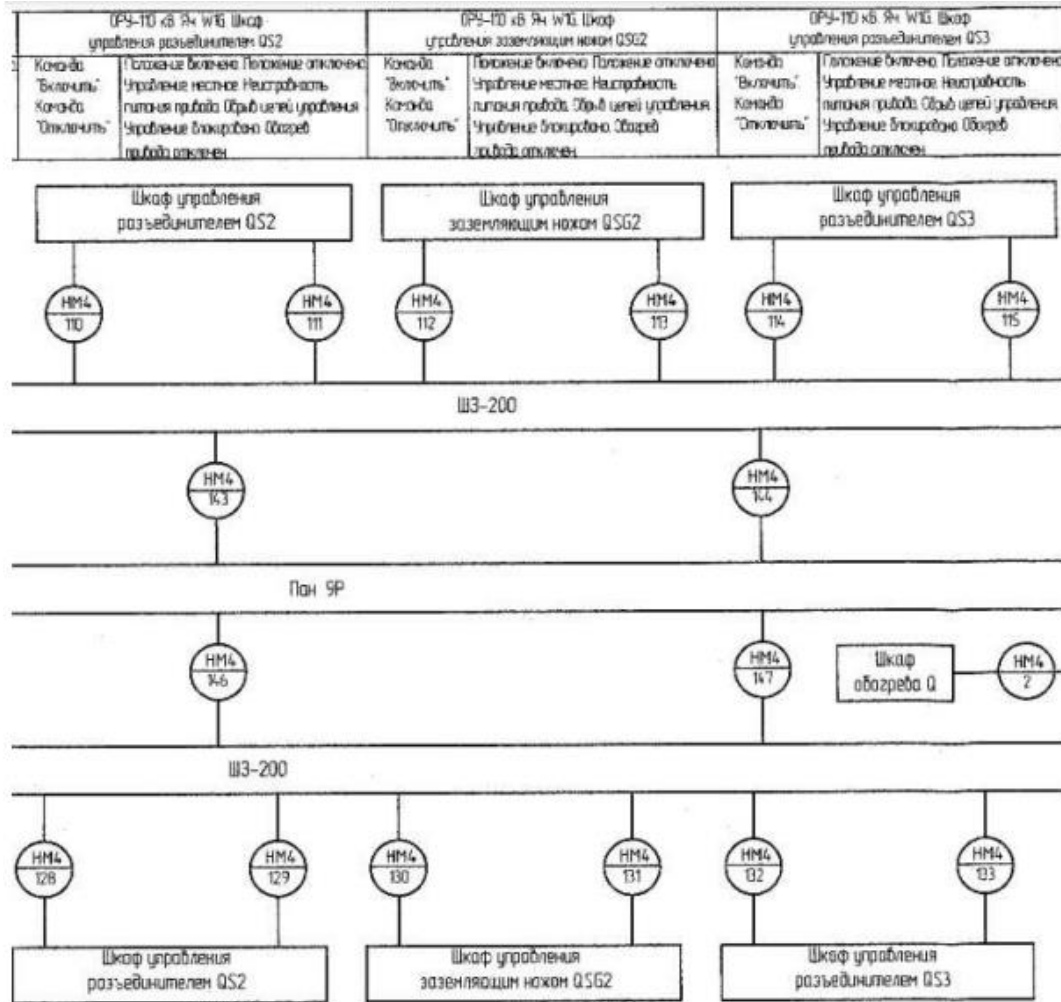


Рис. 1.5. Фрагмент схеми підключення зовнішніх проводок

План кабельної розкладки містить ту ж інформацію, що і схеми підключення зовнішніх проводок, але з прив'язкою до існуючого розташованого устаткування (рисунок 1.6).

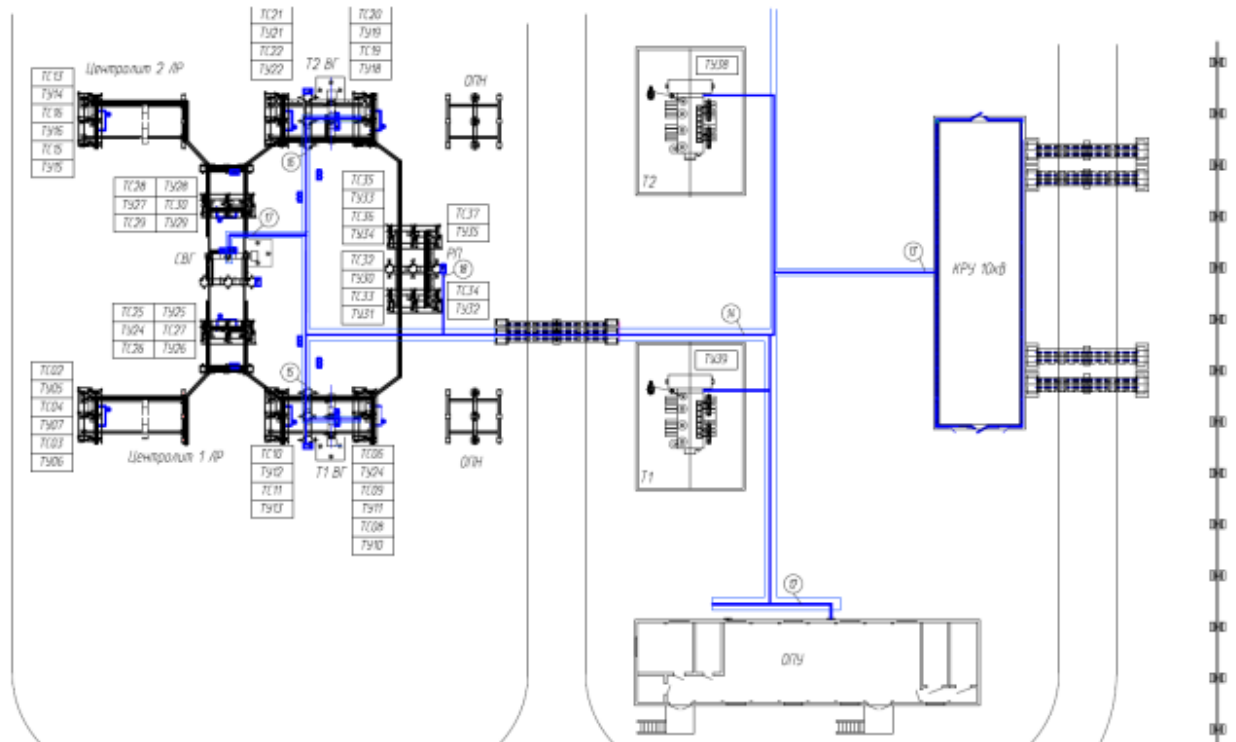


Рис. 1.6. Фрагмент плану кабельної розкладки

Підставою для складання плану кабельної розкладки і схем з'єднань зовнішніх провідок є КЖ, в загальному випадку містить точки початку і кінця кабелю, марку, кількість жил, перетин і довжину [10].

Приклад наведено на рис.1.7.

Маркування кабелів	Заводська марка кабелів		Кільк. залізаності жил	Напряга		Довжина кабелів, м		Примітка
	Тип	Число жил Сечення		Звідки	Куда	проект.	факт	
НП-01	КВВГЭнг(А)-ХЛ	10x15	7	ОПЗ, п.10Р Шкар ТМ №1	ОП4-110, ШЗ /Р 1/110 кВ	110		
НП-02	КВВГЭнг(А)-ХЛ	10x15	7	ОПЗ, п.10Р Шкар ТМ №1	ОП4-110, ШЗ СР 1/110 кВ	125		
НП-03	КВВГЭнг(А)-ХЛ	7x15	5	ОПЗ, п.10Р Шкар ТМ №1	ОП4-110, ШЗ ШР В 1Т/110 кВ	100		
НП-04	КВВГЭнг(А)-ХЛ	7x15	5	ОПЗ, п.10Р Шкар ТМ №1	ОП4-110, ШЗ1 В 1Т/110 кВ	95		
НП-05	КВВГЭнг(А)-ХЛ	10x15	7	ОПЗ, п.10Р Шкар ТМ №1	ОП4-110, ШЗ /Р 2/110 кВ	130		
НП-06	КВВГЭнг(А)-ХЛ	10x15	7	ОПЗ, п.10Р Шкар ТМ №1	ОП4-110, ШЗ СР 2/110 кВ	145		
НП-07	КВВГЭнг(А)-ХЛ	7x15	5	ОПЗ, п.10Р Шкар ТМ №1	ОП4-110, ШЗ ШР В 2Т/110 кВ	120		
НП-08	КВВГЭнг(А)-ХЛ	7x15	5	ОПЗ, п.10Р Шкар ТМ №1	ОП4-110, ШЗ1 В 2Т/110 кВ	115		

Рис 1.7. Фрагмент КЖ

Завдання побудови плану кабельної розкладки є типовим завданням синтезу конструкції, автоматизувати яку можна різними математичними методами.

#### 1.4. Висновки до розділу

У цьому розділі розглянуто проблематику процесу ПЕКМ, проаналізовано традиційна методика проектування підстанцій і вторинних ланцюгів зокрема з точки зору можливості автоматизації проектних робіт. Обґрунтовано рішення автоматизації проектних робіт по створенню плану кабельної розкладки, так як тут необхідно вирішити задачу оптимізації та синтезу конструкції.

## РОЗДІЛ 2

### ТЕОРЕТИЧНА ЧАСТИНА

#### 2.1. Алгоритмізація задачі ПЕКМ

Характерною особливістю комутаційних схем є присутність в них ланцюгів, що з'єднують кілька елементів. Сама схема в загальному випадку не задає конкретного способу реалізації таких багатокінцевих з'єднань - він повинен бути визначений в процесі вирішення основних завдань конструювання [11].

Серед різних варіантів опису комутаційних схем найбільшою спільністю і наочністю володіє опис схеми у вигляді графа. Таке представлення широко використовується при математичній постановці різних оптимізаційних задач конструювання і дозволяє в цілому ряді випадків знайти адекватні задачі в теорії графів і скористатися при вирішенні задач конструювання відомими математичними методами.

При виборі способу опису комутаційної схеми графом треба мати на увазі особливості схеми, а також допустиму ступінь ідеалізації моделі для використання в конкретному завданні.

Існує кілька способів опису схем графами: ГКС; ГЕК; ЗГС.

Моделі схеми у вигляді ГКС або ГЕК задають електрично пов'язані підмножини елементів, але залишають свободу у визначенні конкретних монтажних з'єднань групи виводів.

Варто нагадати, що в понятті графа враховується тільки факт присутності ребер між окремими вершинами, а не спосіб їх зображення. Так, одному і тому ж графу можуть відповідати різні креслення, що відрізняються розташуванням вершин і геометричною формою ребер.

Розрахунок оптимальних конфігурацій з'єднань становить основу для розробки схем проводового та друкованого монтажу. Графи з'єднань, як правило,

не містять циклів, оскільки видалення будь-якого ребра, що входить в цикл, що не порушує електричного зв'язку даної групи виводів.

Тому, можна вважати, що граф, який відповідає з'єднанню групи виводів - це зв'язний граф без циклів, тобто дерево з'єднань.

При дротовому монтажі можлива реалізація лише елементарних з'єднань двох точок типу «вивід-вивід».

В алгоритмах друкованого (плівкового) способу виконання з'єднань для здійснення необхідних електричних зв'язків крім виводів можна використовувати внутрішні точки провідників. Якщо провести аналогію для побудови плану кабельної розкладки, то алгоритми такого виду також можна використовувати для вирішення поставленого завдання побудови загальної траси КМ [11].

Врахування специфіки реалізації з'єднань дає змогу використовувати при рішенні завдання спрощені моделі опису схем, засновані на задачі «ступеня пов'язаності» елементів один з одним.

В графі  $G = (E, U)$  вершини відповідають елементам, тоді як ребра  $u_{ij}$  з приписаними до них вагами  $r_{ij} > 0$  - кількість ланцюгів між елементами (див. рис. 2.1).

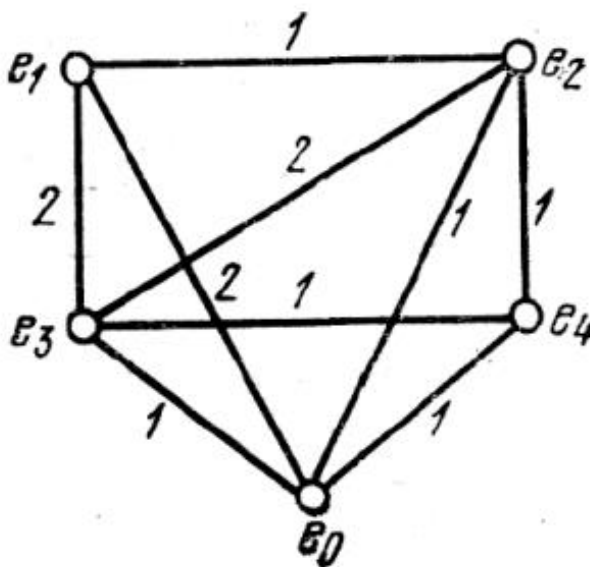


Рис. 2.1. ЗГС



Алгоритмічні методи трасування друкованих з'єднань істотно залежать від конструкції комутаційного поля і можуть бути розділені на дві основні групи [12] (рис. 2.2).



Рис. 2.2. Етапи трасування

До першої групи належать топографічні методи, в яких пріоритет надається метричному аспекту задачі. Друга група заснована на графо-теоретичному підході до вирішення задачі трасування.

## 2.2. Трасування дротових з'єднань

Побудова мінімальних дерев з'єднань є основним завданням, котре необхідно вирішити при трасуванні проводового монтажу. Завдання побудови мінімального дерева формулюється в такий спосіб.

Нехай  $P = \{p_1, p_2, \dots, p_n\}$  – множина точок площини, котрі відповідають виводам довільного ланцюга. Розглянемо повний граф  $G = (I, U)$ , вершини кого  $x \in X$  відповідають виводам ланцюга, а ребра  $u \in U$  з приписаною до них вагою  $\mu(u)$  характеризують з'єднання між парами виводів. Значення  $\mu(u)$  може дорівнювати відстані між відповідними точками множини. У загальному випадку  $\mu(u)$  може являти лінійну комбінацію декількох характеристик з'єднання:

$$\mu(u) = k_1 d_1(u) + k_2 d_2(u) + \dots + k_s d_s(u), \quad (2.1)$$

де  $k_1, k_2, \dots, k_s$  – коефіцієнти;  $d_s(u)$  – деяка характеристика з'єднання  $u$ .

Тепер вихідне завдання зводиться до визначення в графі  $G$  дерева, що містить всі вершини  $X$  та володіє мінімальною сумарною вагою ребер. Таке дерево називається мінімальним покриваючим деревом, або мінімальним зв'язуючим деревом.

Для комп'ютерної реалізації найбільш ефективним є алгоритм Прима [12], згідно якого наступні принципи повинні послідовно виконуватися:

- будь-яка ізольована вершина з'єднується з найближчою;
- всякий ізольований фрагмент (пов'язана група вершин) обов'язково з'єднується з найближчою вершиною при допомозі найкоротшого ребра.

Тут під відстанню між вершинами будемо розуміти значення  $\mu(u)$ , надане ребрах відповідного графа. Відстанню вершини від даного ізольованого фрагмента є мінімум його відстані до окремих вершин фрагмента.

На рис. 2.3 відстанню вершини  $x$  від фрагмента 1,2,3,4,5 є довжина ребра  $(5, x)$ .

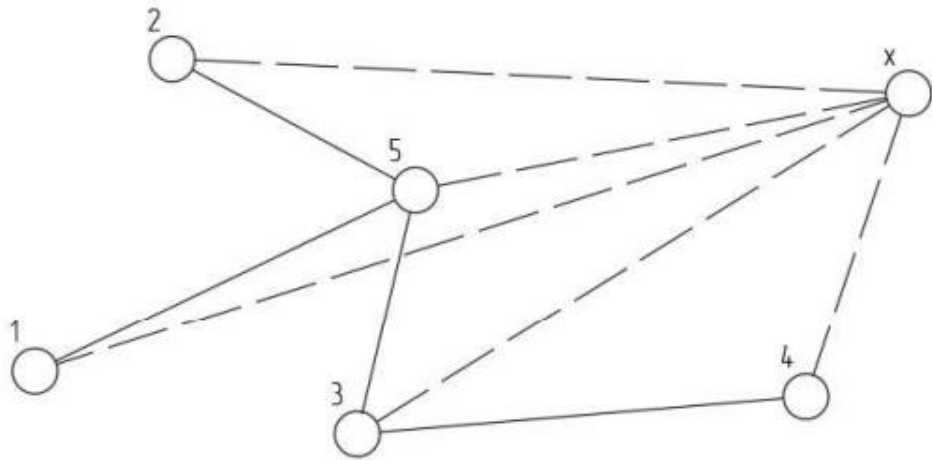


Рис. 2.3. Визначення відстані від вершини до фрагмента

Таким чином, алгоритм створення мінімального зв'язуючого дерева для ланцюга з  $n$  виводами може бути тепер описаний так:

- для довільного виводу ланцюга отримати найближчий і виконати з'єднання;
- за кожним наступним кроком  $i=2,3,\dots,n-1$  з множини непід'єднаних виводів вибрати той, який знаходиться ближче за інші (в зазначеному вище сенсі) до групи вже пов'язаних виводів, і під'єднати його до цієї групи по найкоротшому шляху.

Дерево, котре буде побудоване таким чином, володітиме мінімальною сумарною довжиною з'єднань.

Іноді при побудові зв'язує дерева в якості значення  $\mu(u)$  приймають сумарну оцінку, що включає як довжину ребра  $d(u)$ , так і кількість перетинів  $h(u)$  цього ребра з ребрами вже побудованих дерев:

$$\mu(u) = k_1 d(u) + k_2 h(u), \quad (2.2)$$

Зокрема, така оцінка використовується при побудові зв'язуючих дерев для схем друкованого монтажу. В цьому випадку процедура Прима залишається без змін, а відстань між выводами ланцюга розраховується по (2.2).

Мінімальне дерево, котре створене з використанням алгоритму Прима, має властивість, за якою ступінь будь-якої вершини (число інцидентних ребер) не більше 6. Як правило, при розробці МС вводиться обмеження на максимальне число з'єднань  $\lambda$ , що підходять до одного виводу. Якщо  $\lambda < 6$ , необхідно використовувати спеціальні алгоритми побудови мінімальних зв'язуючих дерев.

Побудова мінімального дерева з обмеженням на ступеня вершин може бути здійснено при використанні процедур, заснованих на методі гілок і меж. Однак для практичних цілей перевагу слід віддати евристичним алгоритмам.

Зокрема, можна використовувати модифіковані принципи Прима: будь-яка ізольована вершина обов'язково з'єднується з найближчою, котра ще не з'єднана з  $\lambda$  іншими вершинами; всякий ізольований фрагмент з'єднується найбільш коротким ребром з найближчою вершиною, що не з'єднана з  $\lambda$  іншими вершинами.

Дослідження показують, що алгоритм, побудований на підставі цих принципів, призводить до отримання дерев з довжиною, що перевищує мінімальну не більше ніж на 5% при числі виводів  $n \leq 15$ .

Модифіковані принципи Прима використовуються при паралельному нарощуванні декількох фрагментів дерева. На підставі проведеної серії експериментів ( $\lambda = 2,3$ ) з'ясовується, що такий спосіб дає дерева з меншою довжиною з'єднань в порівнянні зі способом послідовного нарощування одного ізольованого фрагмента [11].

У деяких випадках, крім обмеження на ступеня вершин зв'язуючого дерева, задається початкова і кінцева точка ланцюга. Наприклад, це має місце при розробці МС для високочастотних ланцюгів, коли необхідно пов'язати в певній послідовності джерело сигналу і кілька навантажень. Тоді задача зводиться до побудови найкоротшого шляху між двома заданими выводами, що проходить

через всі інші виводи ланцюга. Дане завдання споріднене до задачі про маршрут комівояжера, але відрізняється від останньої тим, що шлях обходу повинен бути розімкненим і з'єднувати дві задані точки. Дотримуючись термінології теорії графів, виникає задача побудови найкоротшого гамільтонового ланцюга між заданими початковою і кінцевою вершинами. Розглянемо алгоритм, що дає наближене рішення цієї задачі. Основу алгоритму становить  $(n-1)$  - кроковий процес вибору найкоротших ребер в повному графі, перевірки кожного ребра на виконання обмежень завдання і складання з вибраних ребер шляху, що з'єднує задані точки.

Нехай задано розташування точок на рис. 2.4 (а).  $1^*$  та  $2^*$  - відповідно початкова і кінцева точки шляху. Складемо впорядковану по зростанню довжин послідовність ребер повного графа  $G : (1^*-3), (1^*-2^*), (2^*-3), (4-5), (3-4), (3-5), (1^*-4), (2^*-5), (2^*-4), (1^*-5)$ .

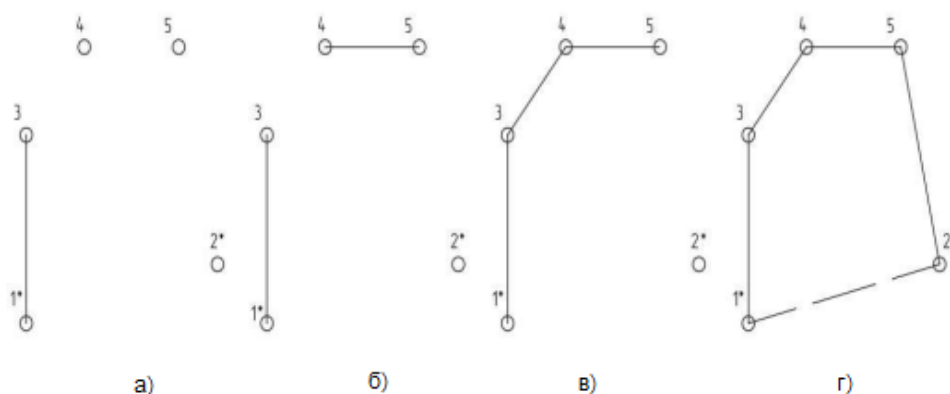


Рис. 2.4. Побудова шляху

Чергове ребро  $i=1,2,\dots,n-1$  вибирається один по одному з цією послідовності при виконанні наступних умов:

1. Ребро не з'єднує задані кінцеву і початкові точки  $1^*$  та  $2^*$ .
2. При включенні ребра в шлях ступінь вершин, що з'єднуються цим ребром, не перевищує допустимої ( $\lambda=1$  для початкової та кінцевої точок і  $\lambda=2$  для інших точок).

3. Ребро не утворює циклу з ребрами, вже включеними в шлях.

4. При включенні в шлях будь-якого ребра, крім  $(n-1)$ -го, початкова і кінцева точки залишаються непов'язаними.

Умови 1-3 безпосередньо впливають з обмежень задачі. Умова 4 перешкоджає утворенню тупикових ситуацій, тобто такого положення, при якому подальше формування шляху стає неможливим - все під'єднані точки, крім початкової і кінцевої, мають ступінь  $\lambda=2$ . Покроковий процес формування шляху зображений на рис.1.11 а-г).

Крок 1. Вибираємо ребро  $(1^*-3)$ , так як воно задовольняє всім умовами.

Крок 2. Ребро  $(1^*-2^*)$  відкидається, тому що не задовольняється умова 1, а ребро  $(2^*-3)$  - так як не задовольняється умова 4. Вибирається ребро  $(4-5)$ .

Крок 3. Вибирається ребро  $(3-4)$ .

Крок 4. Ребра  $(3-5)$  і  $(1^*-4)$  відкидаються через невиконання умови 3. Вибирається ребро  $(2^*-5)$ . Результуючий шлях  $1^* - 3 - 4 - 5 - 2^*$ .

Зауважимо, що якщо зняти обмеження про крайніх точках шляху, то даний алгоритм призводить до коротшого шляху  $2 - 1 - 3 - 4 - 5$ .

Використання описаних вище процедур для побудови зв'язуючих дерев з обмеженим ступенем вершин забезпечує цілком прийнятні результати. Середнє подовження монтажних з'єднань в порівнянні з оптимальним рішенням порядку 5%.

### 2.3. Побудова дерев Штейнера

При проектуванні друкованих з'єднань особливий інтерес представляє побудова зв'язуючих дерев спеціального виду - дерев Штейнера [13, 14].

Відома проблема Штейнера полягає в наступному. Дана множина точок на площині:  $P = \{p_1, p_2, \dots, p_n\}$ . Потрібно знайти дерево  $T = (X, U)$  з множиною вершин  $X$  і множиною ребер  $U$ , для якого  $P \subset X$  і сумарна довжина ребер  $U$

мінімальна. Таким чином, на відміну від мінімального зв'язуючого дерева, дерево Штейнера містить деяку множину вершин  $X \setminus P$ , відмінних від вихідної множини виводів. Використання дерев такого типу дозволяє істотно поліпшити характеристики друкованих з'єднань, зокрема зменшити їх довжину, спростити форму і т. д.

Алгоритми побудови мінімальних зв'язуючих дерев не залежать від способу оцінки відстаней між виводами. У той же самий час методи рішення задачі Штейнера істотно залежать від метрики. Проведені дослідження виявили у дерев Штейнера деякі характерні властивості, які послужили основою для розробки точних і наближених алгоритмів розв'язання задачі.

Оскільки при машинному трасуванні друкованих з'єднань, як правило, використовується ортогональна опорна сітка, зупинимося на завданні побудови дерев Штейнера в ортогональній метриці.

В роботі Ханана запропонований алгоритм для побудови мінімального дерева Штейнера при  $n \leq 5$  і описані властивості таких дерев для довільного  $n$ .

Нехай  $P = \{p_i\}, i = 1, 2, \dots, n$  - задана множина точок, а  $Q = \{q_j\}, j = 1, 2, \dots, k$  - додаткові точки при вирішенні задачі. Має місце наступна теорема: якщо  $\{x_p\}$  і  $\{y_p\}$  - множина координат заданих точок, а  $\{x_{q_j}, y_{q_j}\}$  - координати будь-якої вершини  $q_j \in Q$ , то існує таке рішення задачі, що  $x_{q_j} \in \{x_p\}$  і  $y_{q_j} \in \{y_p\}$  для всіх  $j = 1, 2, \dots, k \leq n-2$ .

Таким чином, теорема стверджує, що при знаходженні додаткових точок можна обмежитися вузлами ортогональної сітки, побудованої на заданих точках (див. рис. 2.5). При цьому число цих точок  $|Q| \leq n-2$ . Крім того, для ступенів вершин дерева Штейнера виконуються співвідношення:  $1 \leq w(p_i) \leq 4$  і  $w(q_j) = 3$  або 4.

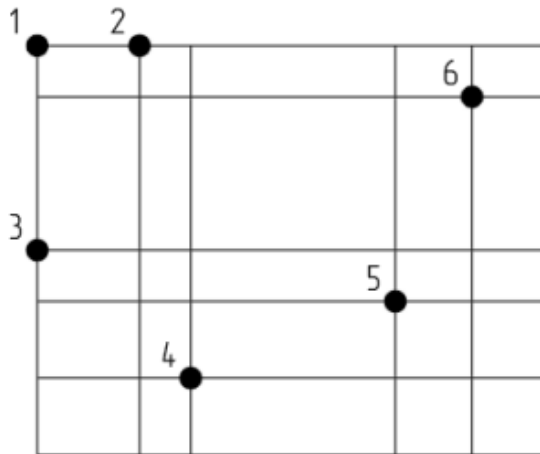


Рис. 2.5. Ортогональна сітка

Для відшукування положення додаткових точок в принципі можна використовувати метод скороченого перебору або алгоритм, заснований на методі гілок і меж.

Точний метод вирішення задачі побудови мінімального дерева Штейнера також запропонований Шейном. Метод використовує апарат булевої алгебри.

Співставимо ортогональній сітці граф  $G = (V, E)$ , вершини якого відповідають вузлам сітки, а ребра - відрізкам, що з'єднують відповідні вузли. Припишемо ребрам ваги, рівні відстані між вузлами сітки. Тоді вихідна задача стає еквівалентною до задачі відшукування в графі  $G = (V, E)$ , дерева  $G' = (V', E')$ , що має мінімальну сумарну вагу ребер і включає заданий набір вершин  $P$  графа  $G: P \subset V' \subset V$ .

Співставимо будь-якому ребру  $e$  булеву змінну  $x_e$ , тоді будь-якому шляху  $s = e_1, e_2, \dots, e_k$  відповідає кон'юнкція булевих змінних  $X_s = \bigwedge_{i=1}^k x_{e_i}$ , а деякій множині шляхів  $M_{l,t}$  між двома вершинами  $v_l$  і  $v_t$  - діз'юнктивна форма  $u_{l,t} = \bigvee_{s \in M_{l,t}} X_s$ . Позначимо через  $u_{i,i+1}$  булеву функцію, відповідну множині шляхів, що з'єднують вершини  $p_i$  і  $p_{i+1}$  даного кола. З огляду на введені позначення, для



заданої множини вершин :  $P = \{p_1, p_2, \dots, p_n\}$ , можливо написати булеву функцію, що дозволяє виділити дерева мінімальної довжини:  $u(1, 2, \dots, n) = \bigwedge_{i=1}^{n-1} u_{i,i+1}$ .

Після приведення цієї функції до диз'юнктивного вигляду всі мінімальні дерева будуть визначатися елементарними кон'юнкціями, котрі мають мінімальний сумарну вагу. Під вагою кон'юнкції мається на увазі сума ваг булевих змінних, які входять до неї.

Відзначимо, що процедура перебору спрощується, якщо в ортогональній сітці всі ребра мають однакову довжину. У цьому випадку вага будь-якої булевої змінної дорівнює 1 і вибір мінімального дерева зводиться до знаходження кон'юнкції, що має мінімальне число букв.

Аналогічний метод запропонований М. Е. Штейном для побудови найпростіших зв'язуючих дерев. Найпростішим автор називає дерево, розташоване на мінімальній кількості горизонтальних і вертикальних магістралей ортогональної сітки. Такий критерій пов'язаний зі спробою врахувати при побудові окремих ланцюгів специфіку проектування двосторонніх друкованих плат з ортогональними з'єднаннями. Дійсно, в цьому випадку утворюються зв'язуючі дерева, котрі мають мале число поворотів з'єднань, внаслідок чого зменшується число міжшарових переходів.

Очевидно, що пряме застосування методу булевих функцій для побудови дерев Штейнера практично неможливо вже при відносно невеликих значеннях, оскільки число елементарних кон'юнкцій різко зростає. Тому доцільно зменшити розмірність задачі перед застосуванням методу. Це можливо завдяки деяким додатковим властивостям зв'язуючих дерев в ортогональній метриці.

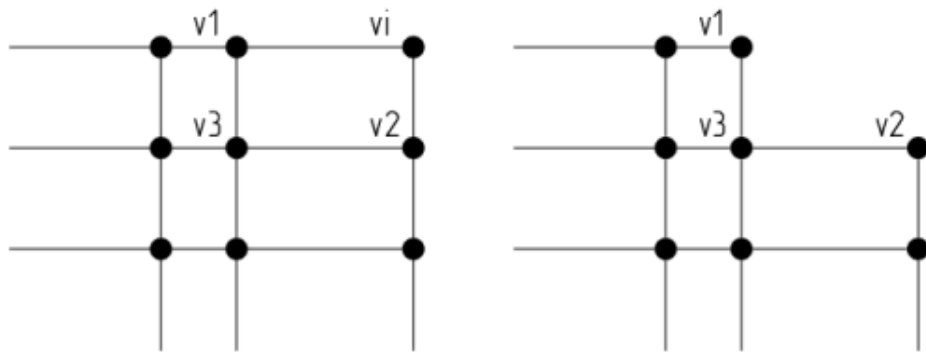


Рис. 2.6. Видалення ребер

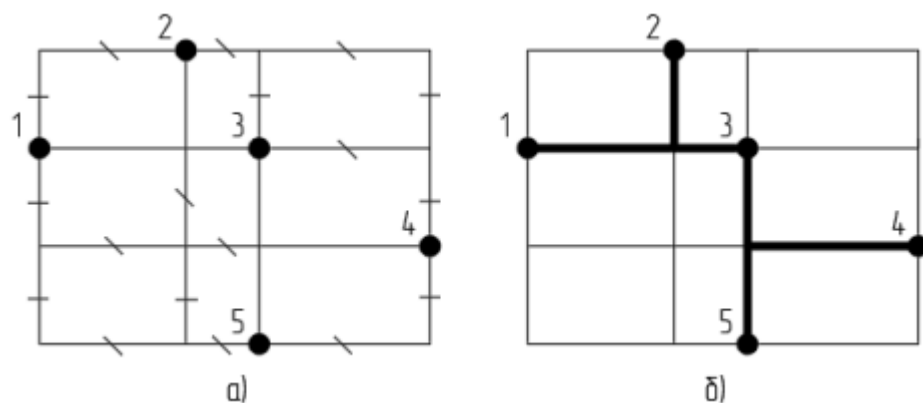


Рис. 2.7. Побудова мінімального дерева

а - послідовне видалення ребер, б - мінімальне дерево

Нехай  $v_i \in P$  деякий вузол сітки, що має ступінь  $w(v_i) = 2$ . Нехай далі  $v_1$  і  $v_2$  - вузли, сусідні до  $v_i$ , і існує вузол  $v_3$ , сусідній до  $v_1$  і  $v_2$ . Тоді дерево мінімальної довжини може бути знайдено і в сітці, отриманій видаленням з вихідної сітки ребер  $(v_1, v_i)$  і  $(v_2, v_i)$ . Доведення випливає з того факту, що якщо ребра  $(v_1, v_i)$  і  $(v_2, v_i)$  входять до складу мінімального дерева, то дерево, яке має ту ж сумарну довжину ребер, може бути отримано при заміні ребер  $(v_1, v_i)$  і  $(v_2, v_i)$  на ребра  $(v_1, v_3)$  і  $(v_2, v_3)$ . Послідовне застосування цього правила в ряді випадків призводить до отримання рішення задачі.

Відзначимо, що якщо в процесі видалення ребер деякий вузол  $p_0 \in P$  стає висячим ( $w(p_0)=1$ ) і  $(p_0, v_1, v_2, \dots, v_k)$  - деякий шлях в сітці, утвореній після видалення ребер, такий, що  $w(v_i) = 2, i = 1, 2, \dots, k - 1$ , то цей шлях обов'язково повинен увійти в рішення, оскільки в іншому випадку вершину  $p_0$  не можна було б поєднати з іншими вершинами з  $P$ . Цей шлях можна видалити з сітки, замінивши вершину  $p_0$  вершиною  $v_k$ .

Застосуємо процедуру видалення ребер наприклад, показаному на рис. 2.6. Результуюча сітка з виділеними ребрами показана на рис. 2.8. Оскільки шлях  $(7, v, 7')$  обов'язково увійде в рішення, завдання зводиться до з'єднання точок  $1, \dots, 7'$  в зменшеній сітці. Однак і після такого спрощення задача відшукування мінімального рішення залишається громіздкою.

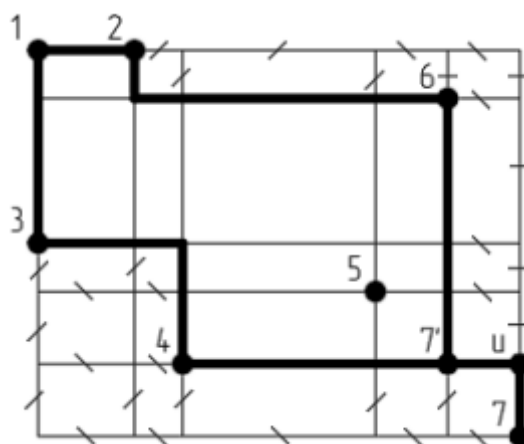


Рис. 2.8. Зменшення розміру сітки

Розглянемо два наближених методи побудови дерева Штейнера в ортогональній метриці.

Нехай  $P = \{p_i\}, i = 1, 2, \dots, n$  - множина виводів одного ланцюга. Побудуємо базисну ортогональну сітку магістралей, що проходять через ці точки (див. рис. 2.9).

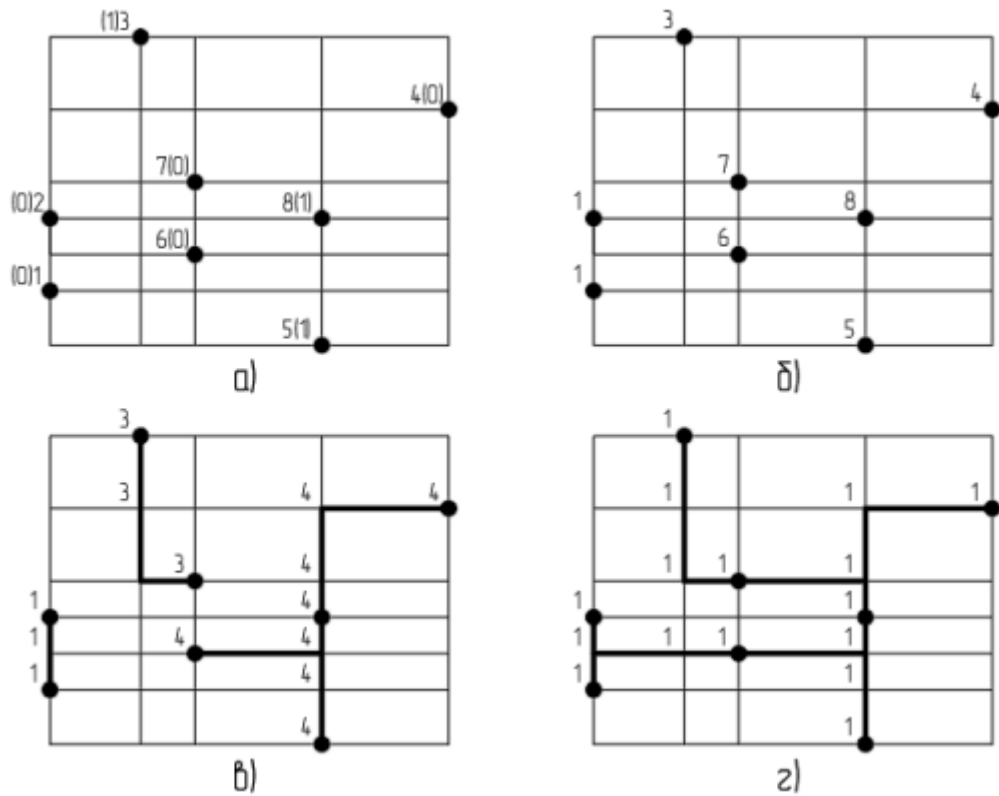


Рис. 2.9. Розмітка вузлів решітки

Алгоритм побудови дерева включає наступні операції.

1. Пронумерувати за годинниковою стрілкою, по спіралі точки множини  $P$ , починаючи з точки  $p_{i_*}$ , для якої  $x_{i_*} = \min_i x_i, y_{i_*} = \min_i y_i$ , де  $I_*$  - множина точок, для яких  $x_i = x_{i_*}$ .

2. Присвоїти кожній точці  $p_i$  одну з двох ознак, що визначають напрямок переважного проведення траси від цієї точки: 0 – вздовж осі  $X$ , 1 - вздовж осі  $Y$ .

Ознаки 0 та 1 присвоюються по черзі, починаючи з точки  $p_1 = p_{i_*}$ , котра позначена ознакою 0. При цьому точки однієї ділянки з послідовності  $p_1, p_2, \dots, p_n$ , що знаходяться на одній горизонталі (вертикалі), отримують одну і ту ж ознаку.

3. Починаючи з точки  $p_1$ , знайти точку  $p_i$ , для якої  $d_{i1} = |x_i - x_1| + |y_i - y_1| = \min_j d_{j1}$ . Побудувати фрагмент дерева, що з'єднає  $p_1$  з  $p_i$ , вибираючи з двох можливих варіантів той, який відповідає ознаці точки  $p_1$ .

4. Всім вершинам сітки, через які пройшов фрагмент, присвоїти найменший з номерів кінцевих точці фрагмента.

5. Операції п.3 і п.4 виконувати для інших точок в порядку зростання їх номерів до тих пір, поки не будуть побудовані фрагменти для всіх точок  $p_i (i = 1, 2, \dots, n)$ .

6. Повторювати п.3 - п.5 до тих пір, поки всі  $p_i (i = 1, 2, \dots, n)$  не отримають номер 1.

Розмітка вузлів решітки за вище наведеним алгоритмом показана на рис. 2.9.

Блок схема даного алгоритму показана на рис. 2.10.

Також існує евристичний алгоритм вирішення цієї задачі, заснований на безпосередньому використанні алгоритму Прима (див. рис. 2.11). На першому етапі за допомогою алгоритму Прима будується мінімальне зв'язуюче дерево  $T = (P, U)$ , де  $P$  - множина виводів ланцюга,  $u \in U$  - ребра, реалізовані найкоротшими шляхами  $u'$  і  $u''$ . В результаті отримуємо список з'єднань для дерева  $T$ : (1,2), (2,5), (5,6), (5,4), (4,3), (6,7), (6,8).

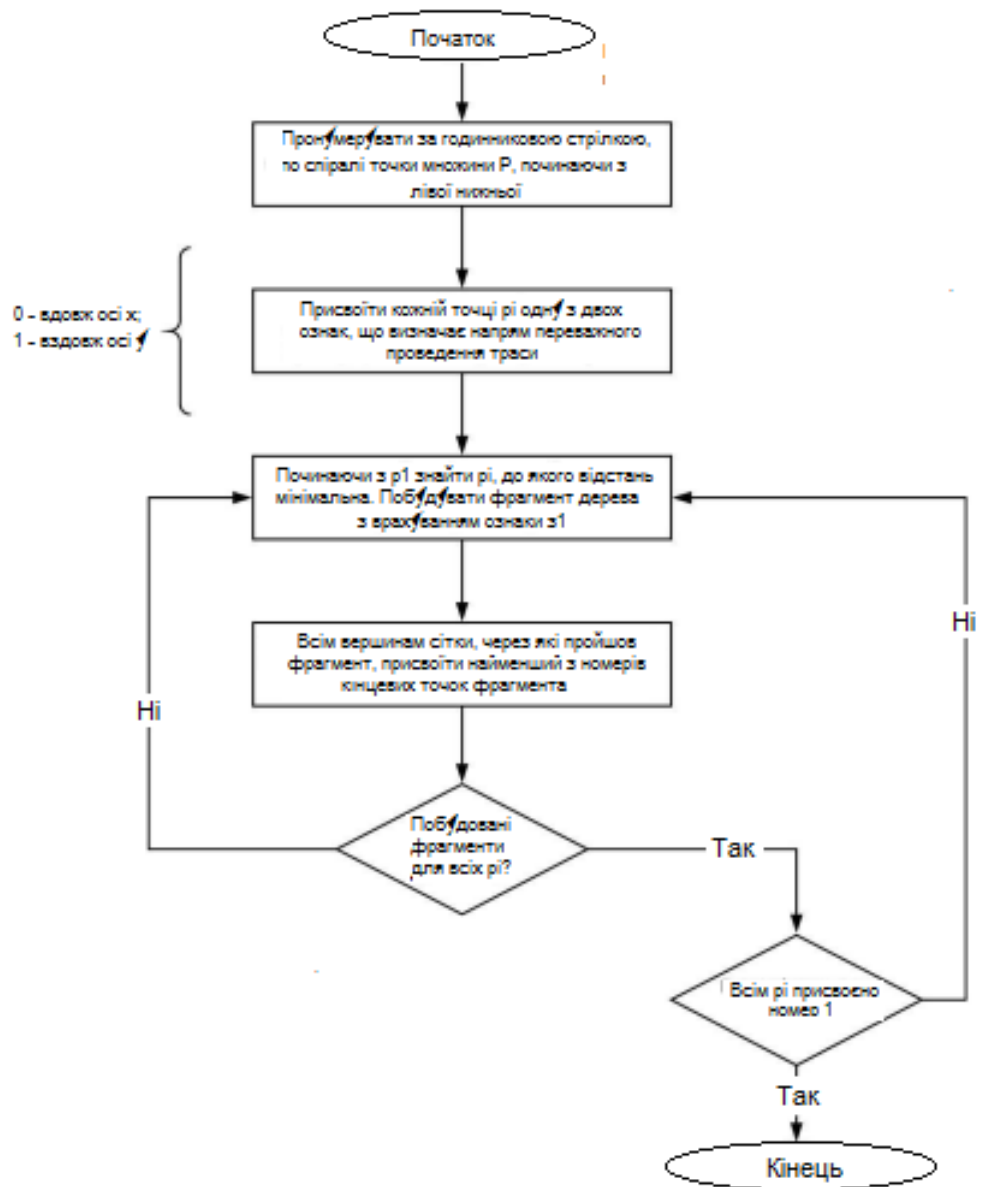


Рис. 2.10. Блок-схема алгоритму побудови дерева Штейнера

Другий етап полягає в суміщенні шляхів, котрі відповідають ребрам  $u$ :

1. Вибирається чергова під'єднана вершина  $p_i$  зі списку  $S$ .
2. Аналізуються всі ребра, інцидентні вершині  $p_i$  і вибираються їх реалізації, що дають максимальне суміщення шляхів з урахуванням раніше побудованих з'єднань.
3. Операції п.1, п.2 виконуються до тих пір, поки для всіх ребер дерева  $T$  не будуть отримані однозначні реалізації.

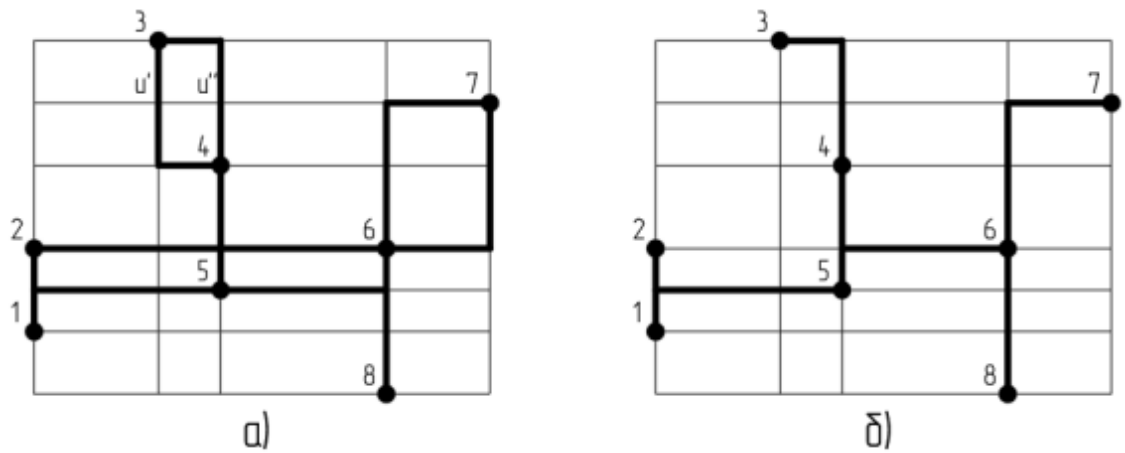


Рис. 2.11. Ілюстрація суміщення шляхів:

а - мінімальне зв'язуюче дерево, б - результат суміщення

Розглянуті алгоритми побудови штейнерових дерев припускають наявність базової опорної сітки для виводів ланцюга і можуть бути безпосередньо використані для трасування друкованих та дровових з'єднань лише в найпростіших конструкціях. У деяких програмах трасування спочатку для всіх ланцюгів будуються дерева Штейнера, а потім здійснюється розподіл пересічних відрізків з'єднань по різних шарах (для друкованих з'єднань). Якщо алгоритмічна методика трасування передбачає послідовну реалізацію ланцюгів, в цьому випадку дерева друкованих з'єднань можуть бути створені з використанням хвильового алгоритму.

#### 2.4. Хвильовий алгоритм

Застосування хвильового алгоритму дає змогу отримувати дерева з'єднань, близькі по довжині до дерев Штейнера [13]. Крім того, при побудові дерев для окремих ланцюгів вдається врахувати різні обмеження на їх взаємне розташування, зокрема неможливість їх перетинів, рівномірність розташування з'єднань і ін.

Етапи хвильового алгоритму:

1. На масштабній сітці комутаційного поля відзначаються зайняті і вільні дискрети (траса може проходити тільки через вільні дискрети; всі раніше прокладені провідники через дискрети і контактні майданчики - це зайняті дискрети).

2. Вводиться вагова функція  $F(f_1, f_2, \dots, f_g)$  як критерій якості шляху (наприклад,  $f_1$  - допустима довжина,  $f_2$  - кількість перетинів,  $f_3$  - число вигинів, ...,  $f_n$  - число перехідних отворів).

3. Поширення хвилі для кожної ланцюга від початку А (джерела) у всі сторони до кінця В (приймача) полягає в присвоєнні дискретам, сусіднім з раніше розглянутими, певного значення вагової функції F. Процес поширення хвилі триває до тих пір, поки фронт хвилі, що розширюється, не досягає дискрети В (кінця ланцюга). Це означає, що шлях побудувати можна. Якщо на черговому етапі поширення хвилі не виявиться жодної вільної дискрети, то шлях побудувати не можна.

4. Будується шлях траси від дискрети В по пройденим дискретам для усунення невизначеності проведення траси в разі, якщо декілька сусідніх дискрет мають однакову вагу, використовуються вагові координати (пріоритет поворотів: вгору, направо, вниз, ліворуч), які вказують перевагу напрямку траси.

Для підвищення швидкодії хвильових алгоритмів застосовується одночасне поширення хвилі від джерела і приймача до їх зустрічі, поширення хвилі тільки в області прямокутника, що створює штучні кордони по дискретах початку і кінця ланцюга, поширення хвилі не по всій множині кліток, а по потоковій мережі, уповільнення хвилі, стиснення і розтягнення мережі.

Слід зазначити, що на поширення хвилі витрачається близько 90% часу обчислень, а на проведення ланцюга - 10%. Значними недоліками цього алгоритму є низька швидкодія та необхідний великий об'єм пам'яті для зберігання даних щодо поточного стану всіх комірок комутаційного поля, а також побудова лише з'єднань типу «вхід-вихід». Крім того, з огляду на специфіку предметної



області, де для побудови плану кабельної розкладки на комутаційному полі доцільно використовувати нерегулярну сітку, використання розглянутого хвильового алгоритму не є пріоритетним.

## 2.5. Алгоритм Дейкстри

Є алгоритмом на графах [14], який представив у 1959 р. нідерландський вчений Е. Дейкстра.

Етапи алгоритму.

1. Всім вершинам, окрім першої, присвоюється вага рівна нескінченності, а першій вершині - 0.

2. Усі вершини не виділено.

3. Перша вершина стає поточною.

4. Вага всіх вершин, котрі невиділені, рахується за формулою: вага невиділеної вершини є мінімальне число з старої ваги даної вершини, суми ваги поточної вершини і ваги ребра, що з'єднує поточну вершину з невиділеною.

5. Серед невиділених вершин шукається вершина з мінімальною вагою. Якщо така не знайдена, тобто вага всіх вершин дорівнює нескінченності, то маршрут не існує. Отже, вихід. Інакше, поточною стає знайдена вершина. Вона ж виділяється.

6. Якщо поточною вершиною виявляється кінцева, то шлях знайдений, і його вага є вагою кінцевої вершини.

7. Перехід на крок 4.

Отже, алгоритм Дейкстри знаходить найкоротші шляхи від однієї вершини графа до всіх інших, що з точки зору поставленого завдання не є доцільним. Крім того, алгоритм не передбачає використання ортогональної нерівномірної сітки.

## 2.6. Висновки до розділу

Розглянуто класичні алгоритми трасування дротових і друкованих з'єднань. Проведений аналіз особливостей електричних КМ дозволив визначити ряд факторів, що ускладнюють використання одних алгоритмів (потрібно ортогональна нерівномірна сітка) на користь алгоритму побудови мінімальних дерев Штейнера.

## РОЗДІЛ 3

### ПРАКТИЧНА ЧАСТИНА

#### 3.1. Інструментальні засоби проектування КМ

##### 3.1.1. Model Studio CS Кабельне господарство

Програмний комплекс ModelStudioCS Кабельне господарство [15] використовується для тривимірного компонування кабельних конструкцій будь-якого рівня складності і автоматичної тривимірної розкладки кабелів.

Розробники реалізували спеціальні алгоритми, що не мають аналогів, котрі дозволяють здійснити автоматичну розкладку кабелів у відповідності до вимог нормативних документів.

Основні можливості ModelStudioCS Кабельне господарство:

- розміщення в тривимірному просторі основного і допоміжного обладнання;
- конструювання та компонування в тривимірному просторі кабельних конструкцій будь-якого рівня складності та елементів будівельних конструкцій;
- одержання оформлених планів (див. рис. 3.1), перерізів на кабельні конструкції;
- формування і випуск набору табличної документації по кабельній розкладці - з рамками, штампами, емблемами і т. п., в тому числі і КЖ.

В процесі роботи досить вибрати з БД кабель, вказати шафу і обладнання, яке підключається до шафи. Програма визначає найкоротший шлях по трасах від одного об'єкта до іншого. На пошук шляху діють різного роду обмеження, такі як наявність на даній трасі раніше прокладених кабелів, тип, група і марка прокладених кабелів, наявне місце на полиці або трасі для укладання кабелю, ознака резерву кабелю.

Враховуються всі вимоги, що пред'являються до розкладці в ПУЕ.

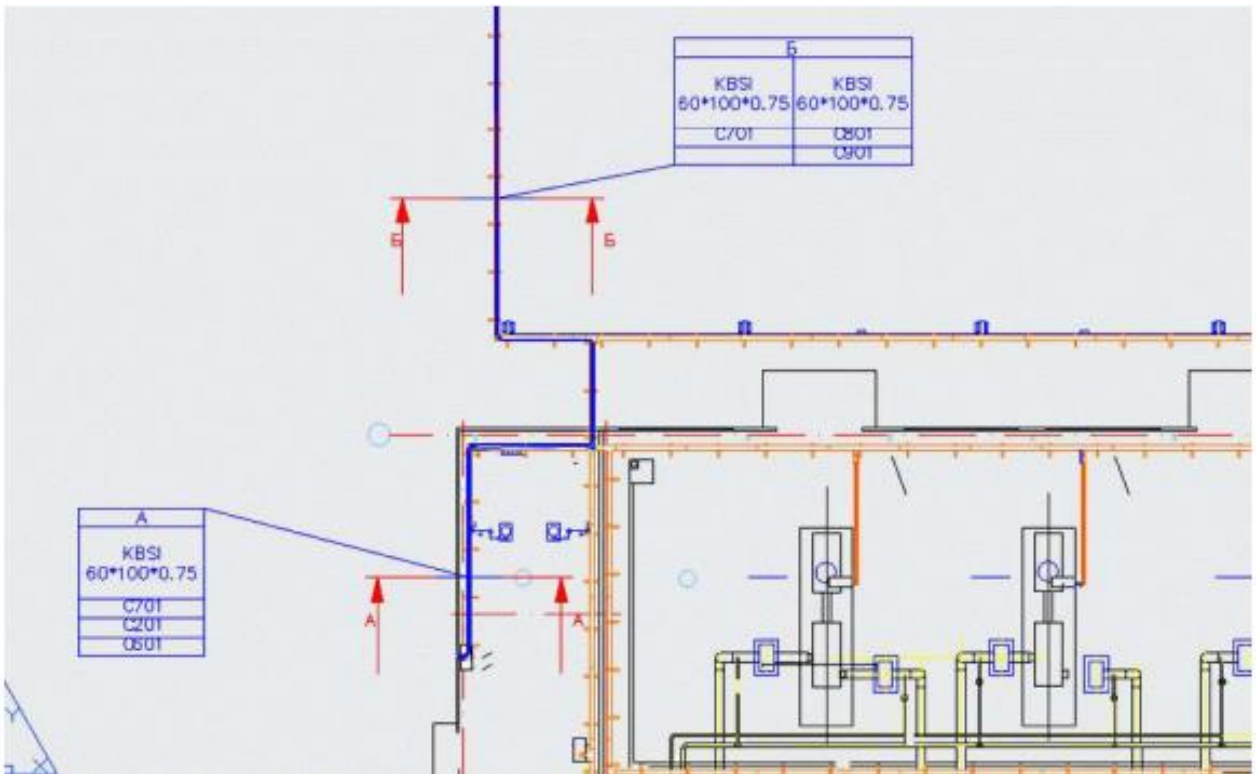


Рис. 3.1. Кабельна розкладка в ModelStudioCS

Можливе ручне налаштування, наприклад, заборона проходження кабелю по певній трасі.

З використанням ModelStudioCS Кабельне господарство рішення задачі кабельної розкладки зводиться до вирішення інженерної задачі розташування кабельних конструкцій і кабелю з урахуванням вимог ПУЕ. При цьому підрахунок довжини, кількості конструкцій і матеріалів програма виконує автоматично.

### 3.1.2. САПР ЦВК

ЦВК є електротехнічною САПР, яка виконує автоматизоване формування за СЕП МС [16].

Всі документи виконуються у відповідності з ЄСКД і галузевими нормативами. САПР ЦВК є проблемно-орієнтованою надбудовою над графічними системами AutoCAD і ZWCAD.

Основою є:

- текстові та графічні БД, що містять інформацію про технічні характеристики апаратури, умовні графічні позначення елементів схем, що дозволяють легко формувати з готових фрагментів різні документи;
- засоби управління проектом, що забезпечують систематизоване зберігання інформації, простий доступ до великої кількості документів, реалізацію проектних процедур.

Даний програмний продукт не передбачає побудову планів кабельної розкладки. Орієнтований тільки на СЕП, МС та схеми кабельних з'єднань для ланцюгів вторинної комутації (рис. 3.2).

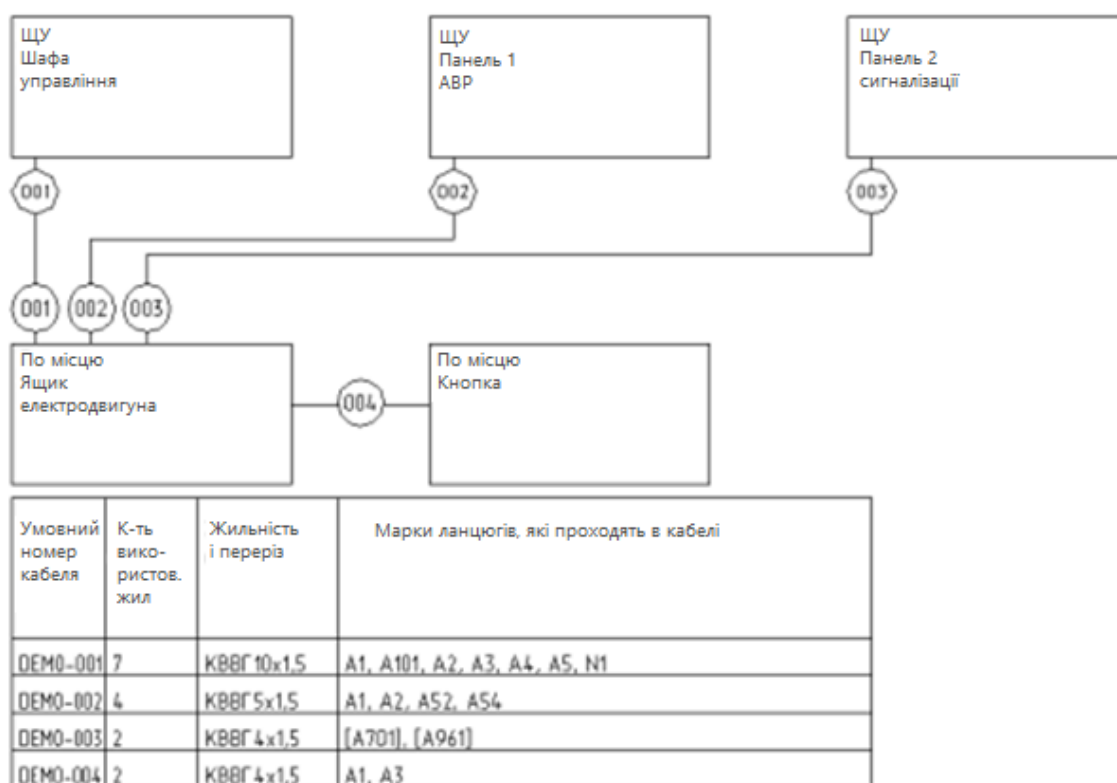


Рис. 3.2. Схема кабельних з'єднань в САПР ЦВК

### 3.1.3. Eplan FieldSys

Дозволяє проводити трасування кабелів за планом території, з прив'язкою до реального масштабу [16]. Це комплексне рішення включає кабельний маршрутизатор, пов'язаний з БД інструмент проектування кабельних трас, а також автоматизований аналіз даних.

У комплексі це надає можливість зробити процес проектування значно швидшим, спростити складання і обслуговування, забезпечити поліпшену якість розроблюваної документації.

FieldSys представляє собою модуль до EPLAN.

Вихідними даними для проектування в FieldSys є виконаний в масштабі 2D-план розміщення обладнання в форматах \* .dxf або \* .dwg. На імпортованій схемою, використовуючи функціонал EPLAN, розмічається система кабельних трас з точками з'єднання пристроїв і переходами, за якими надалі буде виконуватися трасування кабелів. На наступному етапі на схемі розміщується обладнання, яке буде з'єднано по розмічених трасах. Грунтуючись на з'єднаннях між пристроями, визначеними в електромонтажних схемах, EPLAN прокладає всі кабельні з'єднання за певними трасами (рис. 3.3).

Можливість завдання індивідуальних правил допомагає спланувати оптимальні маршрути прокладки кабелів. Також розраховується їх довжина - на підставі довжин сегментів кабельних трас, переходів, початкових і кінцевих резервів для підключення, а також заданих допусків. Після трасування можлива генерація звітних документів.

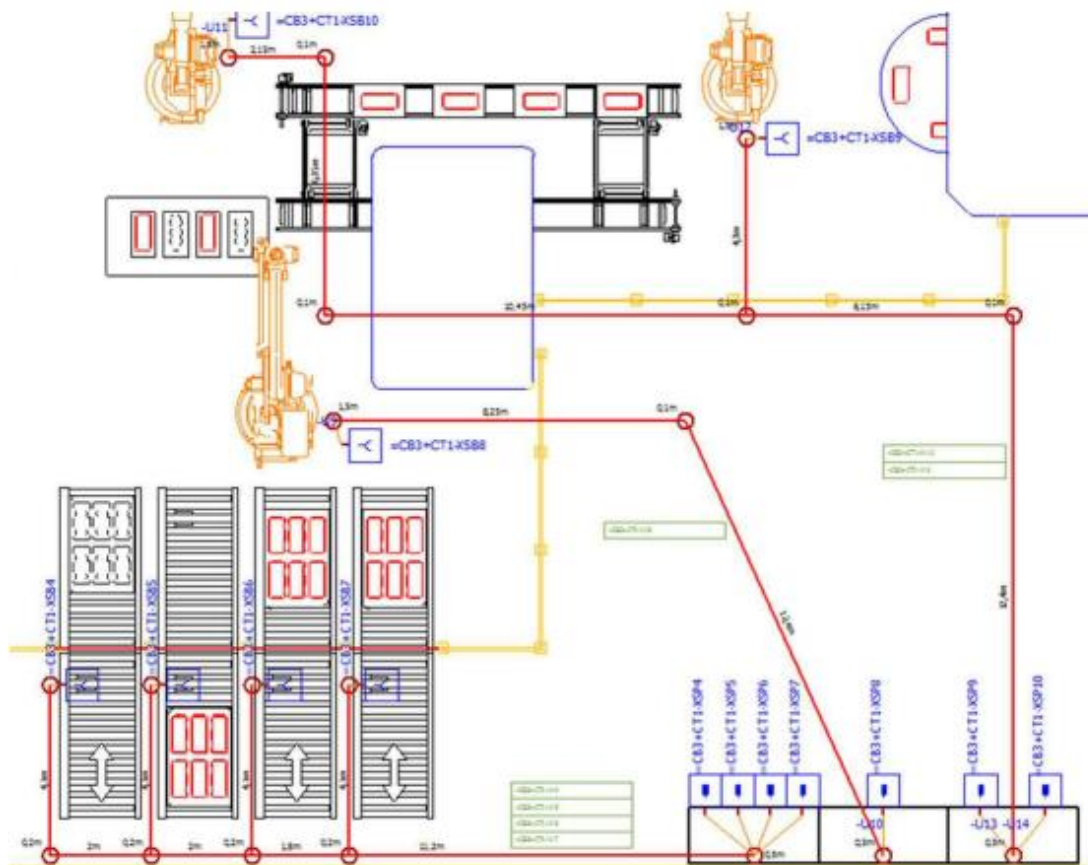


Рис. 3.3. Кабельна розкладка в EplanFieldSys

Наявність електронної документації за кабельними трасами, маршрутами кабелів і різних звітів надає додаткові можливості роботи з проектом - не тільки на стадії монтажу, але і в рамках експлуатації (наприклад, при усуненні неполадок).

#### 3.1.4. КОМПАС-Графік

КОМПАС-Графік + є прикладною бібліотекою проектування: призначена для спрощення підбору світильників, трасування систем електропостачання та електроосвітлення на планах будівель і випуску проектної документації [15]. Дозволяє виконати трасування кабелем електроспоживачів та електроустановочних виробів за групами підключення, вибір перетину кабелю і кількості жил.

Створена для розробки проектів в частині силового електрообладнання, внутрішнього електроосвітлення та електропостачання.

Процес проектування силової електрики включає:

- розміщення електрообладнання на плані;
- підбір обладнання з каталогу;
- трасування кабелю;
- вибір типу кабелю та розрахунок його довжини;
- створення та наповнення КЖ;
- побудова СЕП розподільної і живильної мережі;
- створення специфікацій;
- складання силових однолінійних схем;
- випуск плану розташування електрообладнання і електропроводок.

Оформлення плану розташування обладнання і прокладання кабелю здійснюється на архітектурній підоснові, яка може бути завантажена або побудована за допомогою Бібліотеки проектування будівель і споруд (див. рис. 3.4).

На наступних етапах формується КЖ, йде підрахунок необхідного обладнання, елементів, матеріалів і відомості в специфікації, складається однолінійна схема і оформляються плани розташування.



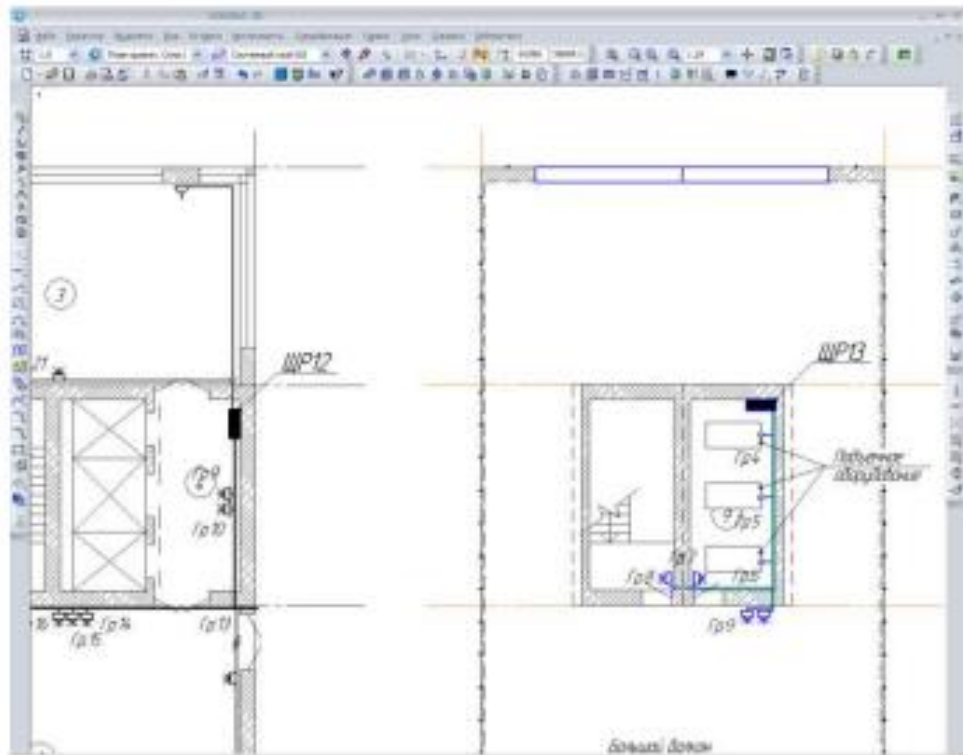


Рис. 3.4. Кабельна розкладка в КОМПАС-Графік

Можливий контроль кабелів і трас в тривимірному просторі з метою перевірки отриманих конструктивних рішень (рис. 3.5).

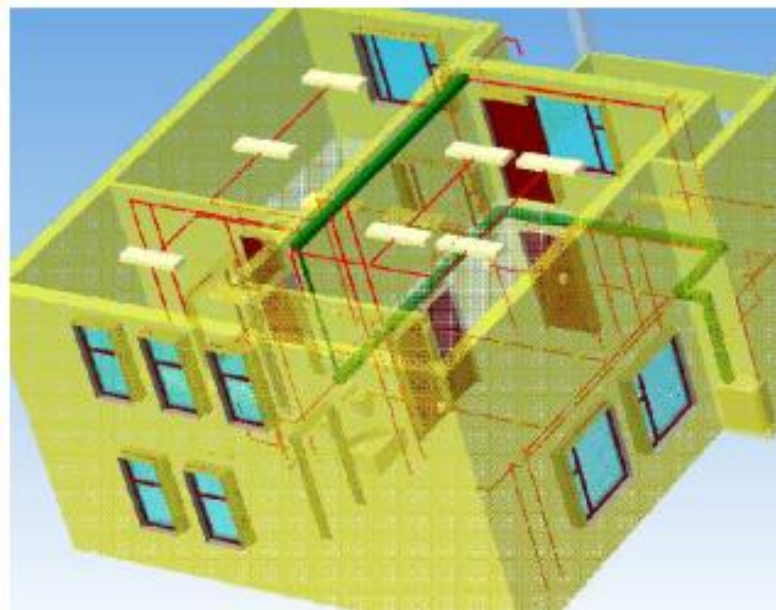


Рис. 3.5. Тривимірне представлення плану кабельних трас

### 3.1.5. E3.Series

Програмний продукт виробництва Zuken, що може вирішувати завершений комплекс проектних завдань у сфері проектування систем електротехніки, АСУ ТП, КВП - починаючи від створення функціональних схем, закінчуючи кресленнями компоновки шаф і панелей, а також випуском конструкторської документації.

З точки зору завдання проектування КМ в E3.Series існує додатковий модуль E3.Instrumentation, котрий встановлюється «Поверх» базового E3.Cable. Даний модуль дозволяє здійснювати розкладку і підрахунок довжин прокладених кабелів з використанням масштабу креслення або окремої області (див. рис. 3.6). Вихідна інформація про точки з'єднання береться зі схем з'єднань зовнішніх провідок, однолінійних СЕП залежно від призначення кабелів.

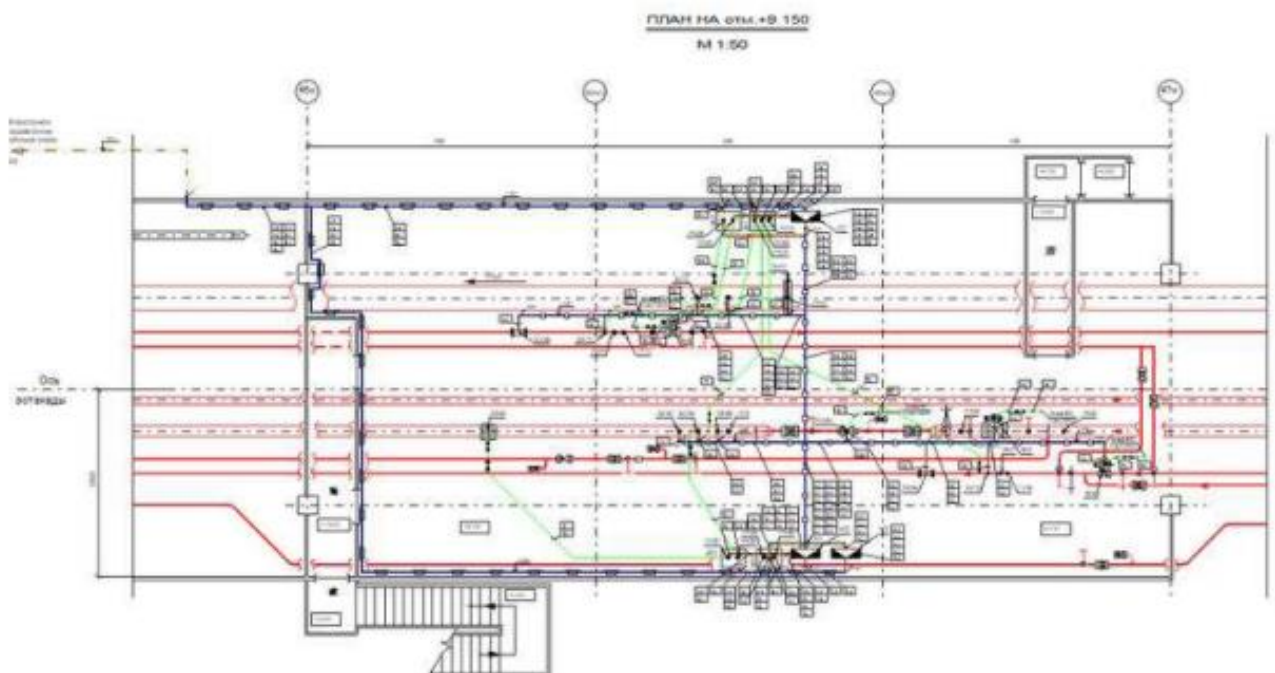


Рис. 3.6. Кабельна розкладка в E3.Series

Модульний підхід передбачає використання E3.Series в повному циклі проектування і не дозволяє отримувати вихідні дані для кабельної розкладки з

сторонніх САПР. Іншими словами, завдання можна вирішити тільки при комплексному використанні даної САПР.

### 3.1.6. AutoCAD Electrical

Звичний AutoCAD, що має додаткові функції для створення проектів області електротехніки, автоматики, АСУ ТП, КВП і т. д [17].

Додані інструменти автоматизують процес створення СЕП, схем з'єднань, креслень компоновок. Існує велика вбудована бібліотека графічних образів для СЕП.

Даний продукт орієнтований в основному на рівень ПС і СЕП, конфігурації ПЛК. Для автоматизованої кабельної розкладки можливостей даного програмного товару не представлено.

### 3.1.7. ADVJToolkit

Програмний комплекс ADVJToolkit є надбудовою над AutoCAD [17].

Дана версія програми призначена для:

- скорочення кількості рутинних операцій вимірювання довжини трас кабелів;
- автоматичне визначення кабелів, що проходять через зазначену користувачем точку траси і вставка переліку кабелів;
- розрахунку реальних довжин з усіма поправками;
- забезпечує автоматичний перерахунок довжин всіх кабелів при зміні поправочного домножуючого коефіцієнта і масштабу креслення;
- вставка підрахованих значень на схемі з'єднань зовнішніх проводок;
- підрахунок підсумкового кількості кабелю, труби і металорукавів за типами і вставка у вигляді таблиці переліку елементів;

– пропонує додатковий інструментарій для редагування різних об'єктів, в тому числі і для роботи з блоками в AutoCAD.

В якості вихідних даних використовується схема з'єднань зовнішніх проводок з проставленими на ній блоками кабелів. Дана схема обов'язково повинна знаходитися в тому ж файлі, в якому розробляється план кабельної розкладки. Кабельні траси створюються вручну, розкладка окремих кабелів здійснюється за раніше створеними трасами.

Передбачено алгоритм автоматичної розкладки кабелю за вказаними початковою і кінцевою точкою і заздалегідь підготовленими трасами.

Приклад кабельної розкладки в ADV\_Toolkit показано на рис. 3.7

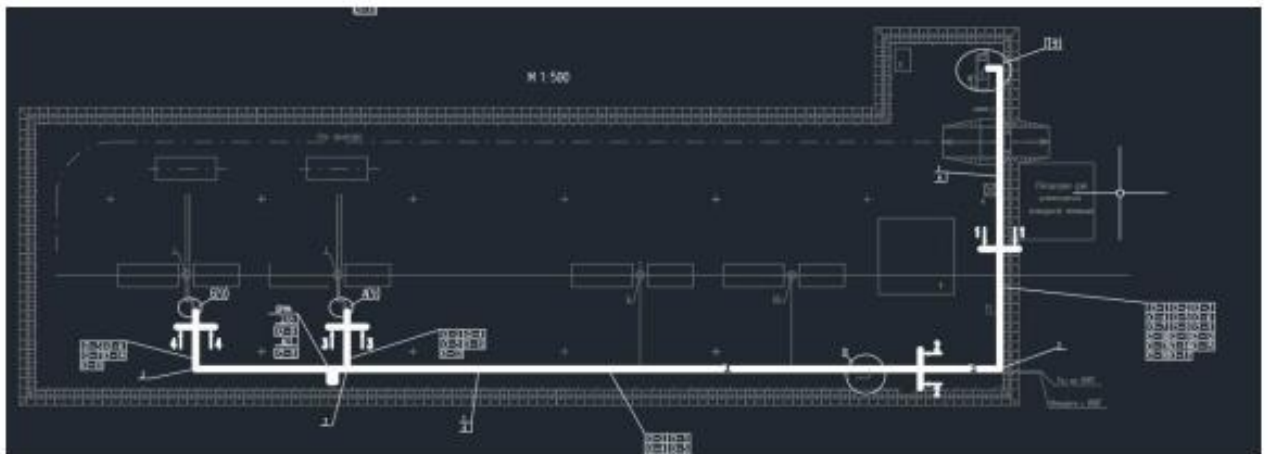


Рис. 3.7. Кабельна розкладка в ADV\_Toolkit

### 3.2. Порівняльний аналіз САПР

Зведемо оглядову інформацію в таблицю 3.1 для наочності порівняння.

Основний критерій - орієнтованість на проектування КМ і закладений функціонал для трасування і побудови кабельної розкладки.

Деякі розглянуті продукти не задовольняють вимогам завдання, в них відсутній необхідний функціонал.

## Порівняння можливостей досліджених САПР

№ з/п	Найменування	Переваги	Недоліки
1.	Model Studio CS Кабельне господарство	- формування документації у відповідності з державними стандартами; - реалізовані алгоритми автоматичної розкладки кабеля; - можливість отримання вихідних даних зі сторонніх пакетів	- висока вартість ліцензії; - складність освоєння;
2.	САПР ЦВК	- інтеграція з AutoCAD або ZWCAD; - генерація КЖ	- немає можливості проектування плану КМ
3.	Eplan FieldSys	- автоматизована побудова кабельних трас; - можливість імпорту вихідних даних в форматах * .dxf або * .dwg; - поповнювана база даних кабельних конструкцій; - генерація звітних документів	- ускладнена техпідтримка; - висока вартість ліцензії
4.	КОМПАС-Графік + бібліотека ЄС	- трасування кабелю по споживачах; - формування документації у відповідності до вітчизняних стандартів	- вузька область застосування (для електроосвітлення); - є додатковою бібліотекою
5.	E3.Series	- автоматизована побудова кабельних трас	- висока вартість ліцензії; - обов'язкове використання цього ж продукту для підготовки вихідних даних
6.	AutoCAD Electrical	- генерація КЖ	- немає автоматичного трасування кабельних мереж; - орієнтованість на рівень МС і СЕП
7.	ADV_Toolkit	- приватне рішення, яке вільно поширюється; - автоматизована побудова кабельних трас	- складність в освоєнні і підготовці вихідних даних

У представлених систем виявлено основний недолік – кабель прокладається по підготовленим вручну трасах, закладені алгоритми не передбачають оптимізацію довжин кабельних конструкцій.

### 3.3. Програмна реалізація

На початковому етапі розробки програмного продукту необхідно провести вивчення, аналіз методології розробки, котра буде адаптованою до кожного конкретного проекту.

Програмна реалізація обраного алгоритму виконувалася на мові C ++ [21, 22], для експорту даних з модуля трасування в універсальний графічний редактор AutoCAD використовувалася вбудована мова AutoLISP [23]. Схема взаємодії та обміну даними між модулем трасування і AutoCAD показана на рис. 3.8.

В якості вхідних даних модуля трасування використовується файл списку з'єднань в форматі \* .asc , що містить найменування точок підключення і ланцюгів, які їх зв'язують. Прототипом цього файлу є КЖ.



Рис. 3.8. Схема взаємодії

Далі запускається алгоритм побудови мінімальних дерев Штейнера в модулі трасування з виводом на попередню візуалізацію і генерацією вихідного файлу формату \* .scr , що містить координати оптимально розташованих ланцюгів у вигляді поліліній.

Даний файл завантажується в AutoCAD і траси накладаються на план розташування обладнання, тим самим отримуємо оптимальний план кабельної розкладки з точки зору мінімізації довжини кабельних з'єднань.

На початковому етапі розробки системи були побудовані діаграми у відповідності до нотації UML [25]. Для опису функціональності і поведінки всього розроблюваного об'єкта була побудована Use Case diagram, яка наведена на рис. 3.9.

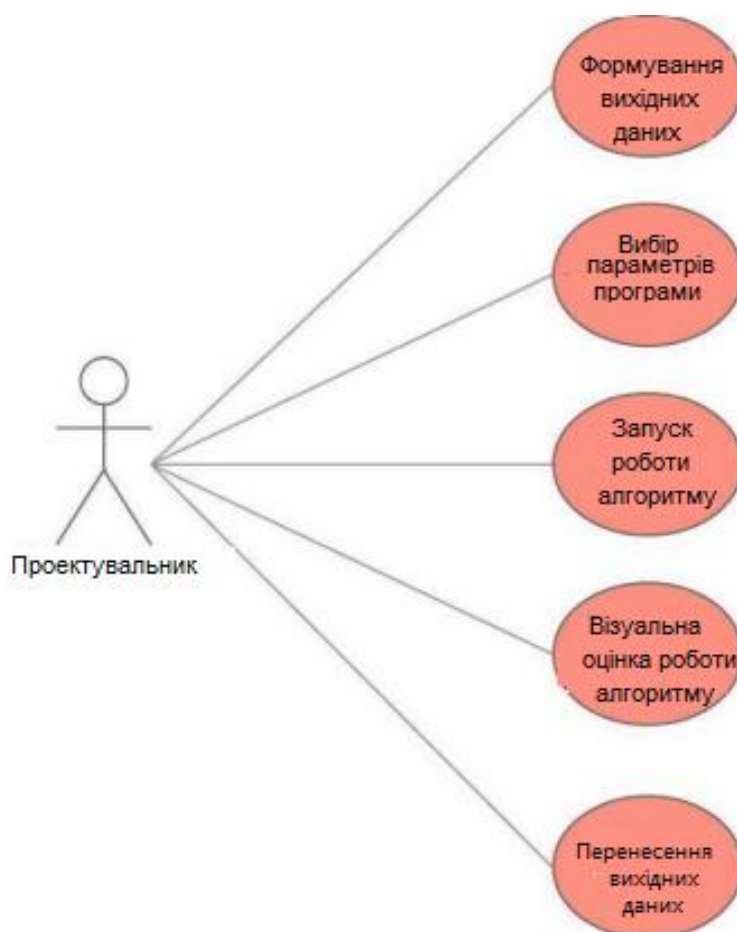


Рис. 3.9. Діаграма варіантів використання

### 3.3.1. Модуль трасування

З причини складності реалізації алгоритму побудови мінімальних дерев на стандартній мові програмування AutoLISP, вбудованій в AutoCAD, реалізація здійснювалася як окремий програмний модуль на мові C++ [21].

У додатку Б наведено програмний код реалізації алгоритму побудови мінімальних дерев Штейнера.

На рис. 3.10 представлена діаграма послідовності (Sequence diagram).

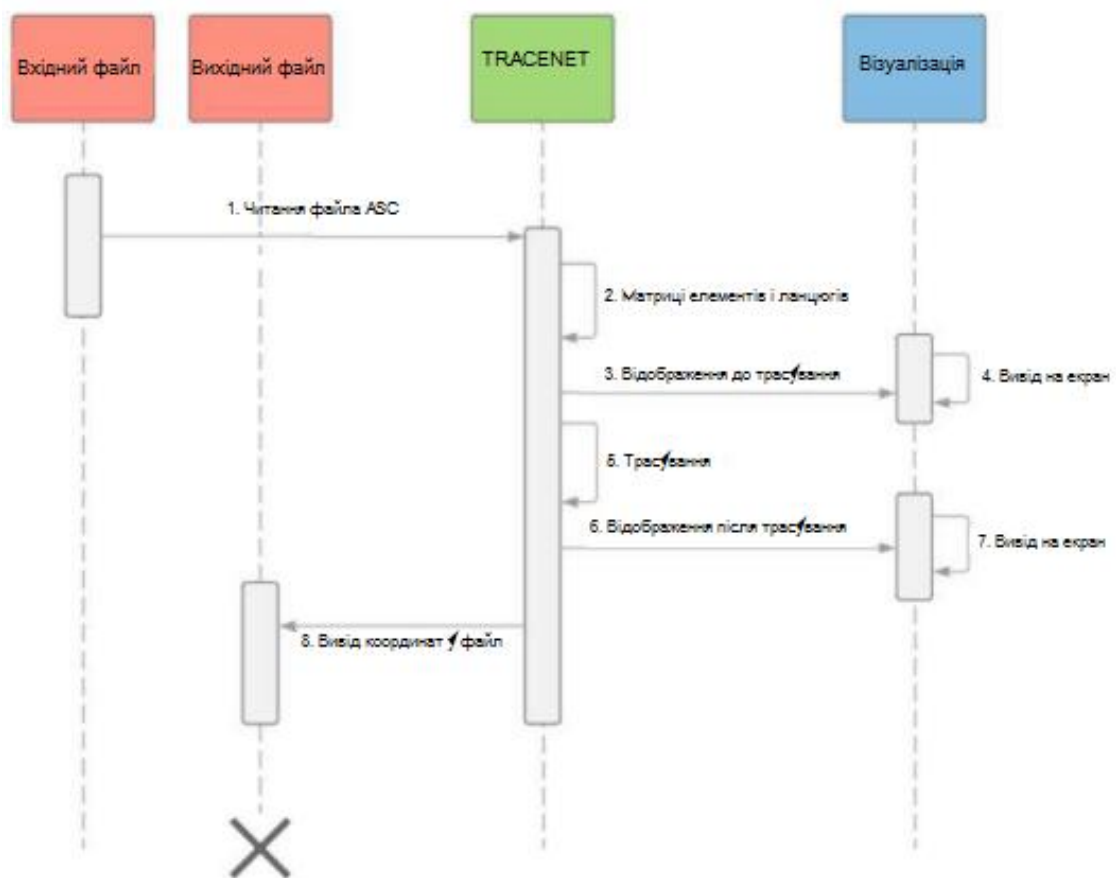


Рис. 3.10. Діаграма послідовності для програми

Як приклад вхідного файлу для модуля трасування представимо наступні рисунки 3.11 та 3.12.



```

*NET "1"
*NET "1" OPU."5" SZ."5"

*NET "2"
*NET "2" SZ."1" QS1."1" QS1."1" OPU."1" QS2."1"

*NET "3"
*NET "3" DT."1" OPU."1"

*NET "4"
*NET "4" OPU."1" ZRU."1" |

```

Рис. 3.11. Формат вхідного файлу

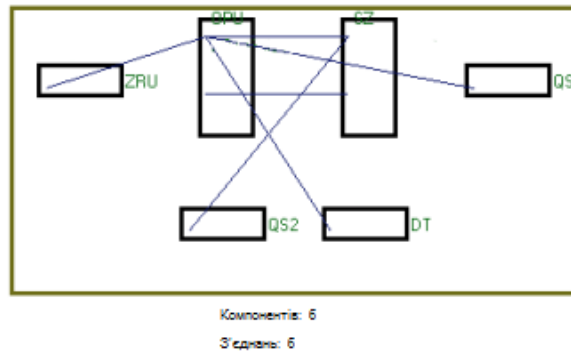
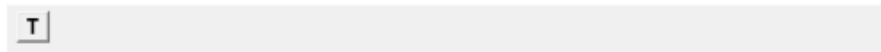


Рис. 3.12. Інтерфейс модуля трасування

В результаті роботи програмного модуля, одержуємо наступний попередній малюнок трасування , наведений на рис. 3.13.

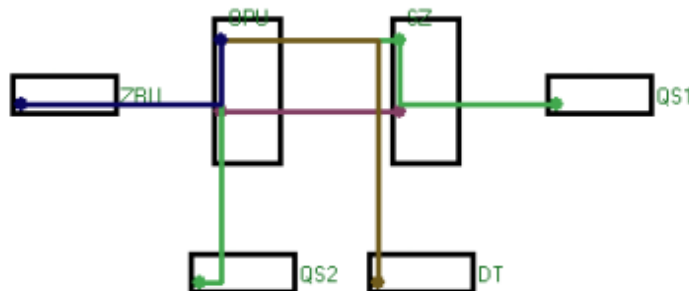


Рис. 3.13. Попередня візуалізація

Після роботи основного модуля трасування і формування вихідних даних необхідно завантажити отримані дані в універсальне графічне середовище проектування.

### 3.3.2. Інтеграція з графічною САПР

Проведено аналіз і реалізована можливість інтеграції отриманих трас з програмного модуля трасування в САПР AutoCAD з метою побудови плану КМ на існуючому плані об'єкта з прив'язкою до координат. Траси експортуються у вигляді списку координат, далі проводиться автоматичне відмальовування за допомогою скрипта на мові AutoLISP.

Як видно з самої назви, його основне призначення - це опрацювання списків, котрі складаються з «атомів» (якихось абстрактних елементів, котрі є формально необмеженими по довжині ланцюжками символів). Атоми можуть вважатися як рядками, так і числами, або бути якимись логічними структурами з вкладеннями на безкінечну глибину підсписків як ієрархічне дерево. Для обробки списків застосовується функціональна модель, котра заснована на теорії lambda-обчислень.

У складі системи AutoCAD є інтерпретатор мови AutoLISP [23]. Якщо при генерації AutoCADa інтерпретатор AutoLISPa був підключений, то він завантажується в оперативку після запуску графічного редактора ACAD і доступний протягом всього сеансу роботи з ACAD.

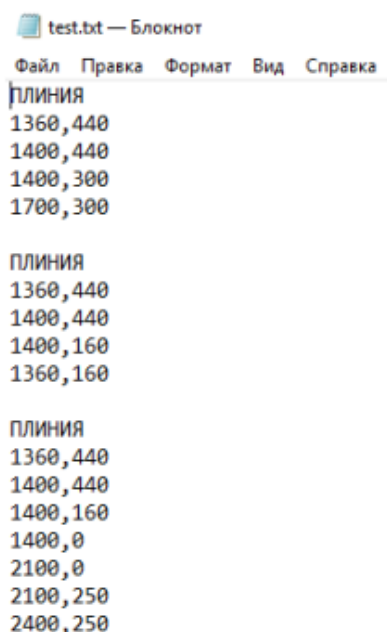
Таким чином, графічний редактор ACAD і інтерпретатор мови AutoLISP є єдиною системою: будь-яка функція AutoLISPa може бути викликана з графічного редактора, і будь-яка команда редактора може бути використана в програмі на AutoLISPe. Можливості застосування AutoLISPa вельми широкі і різноманітні [24].

Значною перевагою застосування AutoCAD є можливість автоматизації часто повторюваних операцій або дій. Один з таких механізмів автоматизації - скрипти або сценарії.

З точки зору комп'ютерної термінології, сценарій - це програма, яка виконується без втручання користувача. Для AutoCAD файл скрипта – це звичайний .txt файл в кодуванні ASCII, котрий має перелік інструкцій для командного рядка. Як правило, файли сценарію створюються поза програмою за допомогою текстового редактора (в даному випадку вихідний файл модуля трасування) в форматі ASCII і з розширенням \* .scr .

Кожен рядок сценарію містить команду, і кожен пробіл в сценарії є значущим, оскільки він виконує роль, аналогічну натискання клавіші ENTER чи ПРОБІЛ. Необхідно знати послідовність запитів для команди, щоб правильно скласти послідовність відповідей в сценарії.

Приклад такого сценарію для відтворення поліліній відповідно до заданих координат представлений на рис. 3.14.



```
test.txt — Блокнот
Файл  Правка  Формат  Вид  Справка
ПЛИНИЯ
1360,440
1400,440
1400,300
1700,300

ПЛИНИЯ
1360,440
1400,440
1400,160
1360,160

ПЛИНИЯ
1360,440
1400,440
1400,160
1400,0
2100,0
2100,250
2400,250
```

Рис 3.14. Структура сценарію

### 3.3. Результати роботи

Після роботи з вихідними даними модуля, шляхом завантаження сценарію в AutoCAD, отримуємо готовий план кабельної розкладки, розташований поверх ТОПОПЛАН об'єкта для прив'язки до існуючих координат. На рис. 3.15 представлений фрагмент плану кабельної розкладки для тестового прикладу, що розробляється для апробації отриманих рішень.

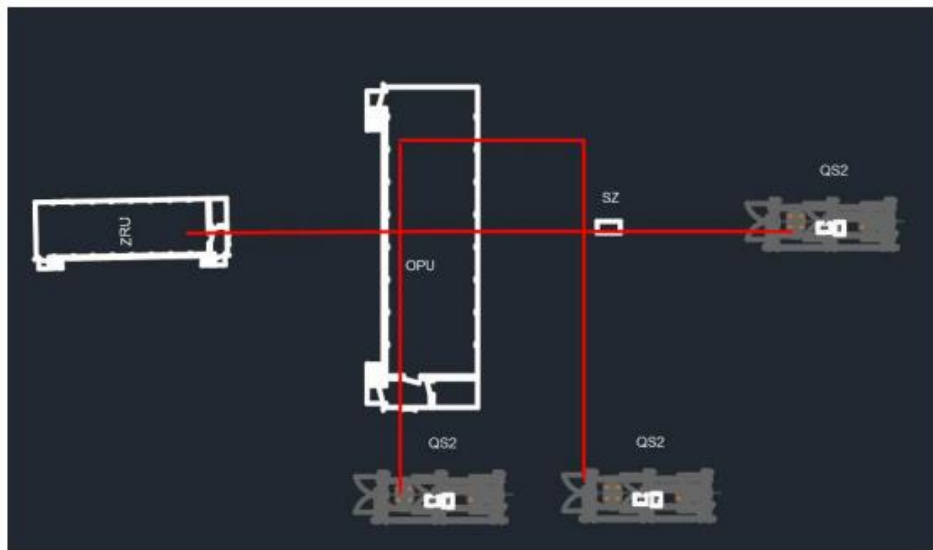


Рис. 3.15. Фрагмент плану кабельної розкладки

На рис. 3.16 наведено приклад плану кабельної розкладки для проєктованого конкретного об'єкта.

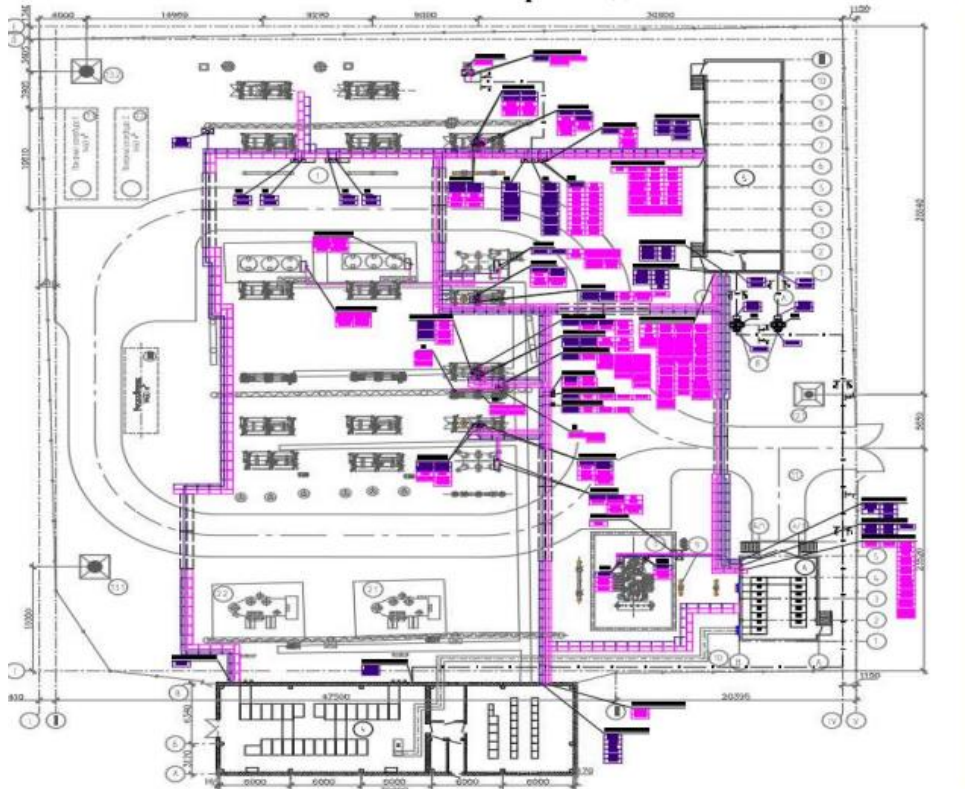


Рис. 3.16. План кабельної розкладки

### 3.4. Висновки до розділу

Розглянуто основні особливості електротехнічних САПР як розповсюджуваних на комерційній основі, так і розроблених окремими проєктувальниками на основі приватних рішень. Проаналізовано можливість використання продуктів для вирішення поставленого завдання, а саме побудови плану КМ. Деякі розглянуті продукти не задовольняють вимогам завдання, в них відсутній необхідний функціонал.

Програмна реалізація обраного алгоритму виконувалася на мові C ++, для експорту даних з модуля трасування в універсальний графічний редактор AutoCAD використовувалася вбудована мова AutoLISP. Побудовані діаграми використання та послідовностей. Розроблено програмний код реалізації алгоритму побудови мінімальних дерев Штейнера. Забезпечена інтеграція з графічною САПР.

## РОЗДІЛ 4

### ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

#### 4.1. Охорона праці

Метою кваліфікаційної роботи магістра є розробка методичного підходу для забезпечення інформаційного супроводу процесу ПЕКМ. Оскільки, проведення робіт з розробки та використання системи передбачає використання комп'ютерної техніки, зокрема ПК та периферійних пристроїв, то обов'язковим є дотримання вимог з охорони праці і техніки безпеки.

Для ефективної і безпечної роботи колективу працівників з розробки ПЗ комп'ютерних систем, в тому числі і фахівців із забезпечення інформаційного супроводу процесу ПЕКМ, необхідно організувати безпечні умови праці. При цьому керівник організації несе безпосередню відповідальність за порушення нормативно-правових актів з охорони праці [26]. Окрім цього, на робочих місцях працівників необхідно забезпечити дотримання вимог, затверджених Наказом Мінсоцполітики від 14.02.2018 за № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». Згідно Вимог приміщення, де розміщені робочі місця операторів, крім приміщень, у яких розміщені робочі місця операторів великих ЕОМ загального призначення (сервер), мають бути оснащені системою автоматичної пожежної сигналізації відповідно до цих вимог;

– переліку однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації, затвердженого наказом Міністерства України з питань надзвичайних ситуацій та у справах захисту населення від наслідків Чорнобильської катастрофи від 22.08.2005 N 161, зареєстрованого в Міністерстві юстиції України 05.09.2005 за N 990/11270 (НАПБ Б.06.004-2005);

– Державних будівельних норм "Інженерне обладнання будинків і споруд. Пожежна автоматика будинків і споруд", затверджених наказом Держбуду України від 28.10.98 N 247 (далі - ДБН В.2.5-56:2014, з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками.

В інших приміщеннях допускається встановлювати теплові пожежні сповіщувачі. Приміщення, де розміщені робочі місця операторів, мають бути оснащені вогнегасниками, кількість яких визначається згідно з вимогами ДСТУ 4297:2004 «Пожежна техніка. Технічне обслуговування вогнегасників». Загальні технічні вимоги і з урахуванням граничнодопустимих концентрацій вогнегасної рідини відповідно до вимог НАПБ А.01.001-2014. Приміщення, в яких розміщуються робочі місця операторів сервера загального призначення, обладнуються системою автоматичної пожежної сигналізації та засобами пожежогасіння відповідно до вимог ДБН В.2.5-56:2014, ДБН В.2.5-56:2010, НАПБ А.01.001-2014 і вимог нормативно-технічної та експлуатаційної документації виробника. Проходи до засобів пожежогасіння мають бути вільними.

Лінія електромережі для живлення комп'ютера та периферійних пристроїв повинні бути виконаними як окрема групова трипровідна мережа шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Не допускається використовувати нульовий робочий провідник як нульовий захисний провідник. Нульовий захисний провідник прокладається від стійки групового розподільного щита, розподільного пункту до розеток електроживлення. Не допускається підключати на щиті до одного контактного затискача нульовий робочий та нульовий захисний провідники.

Площа перерізу нульового робочого та нульового захисного провідника в груповій трипровідній мережі має бути не менше площі перерізу фазового провідника. Усі провідники мають відповідати номінальним параметрам мережі та навантаження, умовам навколишнього середовища, умовам розподілу

провідників, температурному режиму та типам апаратури захисту, вимогам НПАОП 40.1-1.01-97.

У приміщенні, де одночасно експлуатуються понад п'ять комп'ютерів, на помітному, доступному місці встановлюється аварійний резервний вимикач, який може повністю вимкнути електричне живлення приміщення, крім освітлення. Комп'ютери повинні підключатися до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення.

У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їхня конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним. Не допускається підключати комп'ютери до звичайної двопровідної електромережі, в тому числі – з використанням перехідних пристроїв. Електромережі штепсельних з'єднань та електророзеток для живлення комп'ютерної техніки повинні бути виконаними за магістральною схемою, по 3-6 з'єднань або електророзеток в одному колі. Штепсельні з'єднання та електророзетки для напруги 12 В та 42 В за своєю конструкцією мають відрізнятися від штепсельних з'єднань для напруги 127 В та 220 В. Штепсельні з'єднання та електророзетки, розраховані на напругу 12 В та 42 В, мають візуально (за кольором) відрізнятися від кольору штепсельних з'єднань, розрахованих на напругу 127 В та 220 В.

При розробці методичного підходу для забезпечення інформаційного супроводу процесу ПЕКМ, важливим, з точки зору охорони праці, є забезпечення достатньої величини природного та штучного освітлення, які визначені у НПАОП 0.00-7.15-18. Організація робочого місця фахівця із дослідження методів та програмно-апаратних засобів оптимізаційних процесів на основі ГА повинна забезпечувати відповідність усіх елементів робочого місця та їх розташування ергономічним вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце для



виконання робіт у положенні сидячи. Загальні ергономічні вимоги». Відстань від екрана до ока фахівців, які працюють за комп'ютером визначається згідно з вимогами ДСанПіН 3.3.2.007-98.

Розміщення принтера або іншого пристрою введення-виведення інформації на робочому місці має забезпечувати добру видимість екрана комп'ютера, зручність ручного керування пристроєм введення-виведення інформації в зоні досяжності моторного поля згідно з вимогами ДСанПіН 3.3.2.007-98.

Таким чином, у результаті аналізу вимог щодо охорони праці користувачів комп'ютерів, визначено особливості організації робочих місць, вимог з електробезпеки, природного та штучного освітлення для ефективної і безпечної роботи фахівців з дослідження методів та програмно-апаратних засобів оптимізаційних процесів на основі ГА.

#### 4.2. Комп'ютерне забезпечення процесу оцінки радіаційної та хімічної обстановки

Екологічне співтовариство розробило сімейство інструментів комплексної екологічної оцінки. Програмне забезпечення і послуги (ESS), комерційна група ПАСА, включаючи AirWare (для повітряних проблеми якості), WaterWare (для якості води), CityWare (якість повітря і води в контексті великих міст) і EIAxpert (для надання допомоги із загальним впливом на навколишнє середовище). Функціональність в цілому схожа на RAISON, хоча з великим акцентом на моделювання і меншим акцентом на керування даними. Знову ж таки, інструменти ESS розроблені як модульні набори інструментів (доступні спеціальні системи для вирішення конкретних завдань). Компоненти включають стандартні імітаційні моделі, включаючи моделі ISC і PBM Агентства з охорони навколишнього середовища США, управління даними, в тому числі ГІС, аналіз даних (наприклад, аналіз часових рядів даних спостережень), візуалізація, а також оптимізація [23].

Іноді немає готових моделей, придатних для конкретного застосування, але тягар розробки нової програми на Фортрані або С / С ++ є надмірним. Розробка моделі оточення може відносно легко реалізувати власні моделі комп'ютерів і не турбуватися про включення процедур для вирішення рівнянь, візуалізації і т. д. Як правило, за допомогою цих інструментів користувач просто повинен вказати свою модель, використовуючи або математичні рівняння, або спеціальні графічні символи або значки, які безпосередньо представляють поведінку системи.

На даний момент є розроблені моделі комп'ютерного забезпечення процесу для оцінки радіаційної та хімічної обстановки.

GEMS – це система на основі моделей, яка підтримує оцінки схильності і ризику, надаючи доступ до одиночних і мультимедійних моделям експозиції, фізико-хімічні властивості методи оцінки, статистичний аналіз, графічні та картографічні програми з відповідними даними на навколишнє середовище, джерела, рецептори і популяції. У розробці з 1981 року, GEMS надає аналітикам 84 84 інтерактивний, легко досліджуваний інтерфейс для різних моделей, програм і даних, які необхідні для оцінки хімічного впливу і ризику [25].

HSPF – це комплексний пакет для моделювання кількості і якості стоків з багатоцільових водозборів і процесів радіації, що відбуваються в потоках або повністю змішаних озерах. Це дозволяє інтегроване моделювання землі і ґрунту, процесів забруднення при гідравлічній і осадово-хімічній взаємодії. Результатом моделювання є тимчасові дані витрати стоку, концентрація поживних речовин і пестицидів, а також дані кількості і якості води в будь-якій точці водозбору. Алгоритми якості води включають динаміку BOD / DO, вуглець, азот і фосфор. Процеси трансформації, які включені в модель це: гідроліз, фотоліз, окислення, випаровування, сорбція і біодеградація. Вторинні або «дочірні» хімічні речовини також моделюються.

Вимоги до даних для моделі можуть бути досить широкими в залежності від конкретного застосування.

Модель MMSOILS – це методологія оцінки впливу на людину і ризику для здоров'я, пов'язаних з викидами забруднень з небезпечних відходів. Мультимедійна модель, що стосується перенесення хімічної речовини в ґрунтові води, поверхневі води, атмосферу і накопичення в їжі. Шляхи впливу на людину, які розглянуті в методології включають: потрапляння в ґрунт, вдихання летких речовин в повітря і тверді частинки, шкірний контакт, прийом питної води і т.д. Ризик, пов'язаний із загальною дозою опромінення, розраховується на основі хімічної токсичності [24].

#### 4.3. Висновки до розділу

В цьому розділі проаналізовано важливі питання охорони праці та безпеки в надзвичайних ситуаціях, висвітлено питання комп'ютерного забезпечення процесу оцінки радіаційної та хімічної обстановки.

## ВИСНОВКИ

В результаті виконання кваліфікаційної роботи був проведений аналіз процесу ПЕКМ, здійснена програмна реалізація алгоритму трасування і інтеграція отриманих рішень в графічну САПР для побудови плану кабельної розкладки. Тим самим розроблено методичний підхід для забезпечення інформаційного супроводу процесу ПЕКМ.

Запропоновано принципово новий підхід, який передбачає мінімізацію довжини з'єднань при побудові плану кабельної розкладки на основі алгоритму побудови мінімальних дерев Штейнера.

Програмна реалізація обраного алгоритму виконувалася на мові C ++, для експорту даних з модуля трасування в універсальний графічний редактор AutoCAD використовувалася вбудована мова AutoLISP. Побудовані діаграми використання та послідовностей. Розроблено програмний код реалізації алгоритму побудови мінімальних дерев Штейнера. Забезпечена інтеграція з графічною САПР

Розробка і впровадження методики інформаційного супроводу процесу ПЕКМ дає змогу:

- оптимізувати протяжність кабелю і кабельних конструкцій, тим самим знизивши матеріальні витрати на будівельно-монтажні роботи;
- збільшити продуктивність праці для проектувальника;
- суттєво зменшити кількість технічних помилок;
- покращити якість документації для продукції, яка випускається.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Библия электрика: ПУЭ. Новосибирск: Норматика, 2018. 672 с.
2. Наумчук О. М. Основы систем автоматизованого проектування: Інтерактивний комплекс навчально-методичного забезпечення. Рівне: НУВГП, 2008. 136 с.
3. Тимченко А. А. Основы системного проектування та системного аналізу складних об'єктів: Основы САПР та системного проектування складних об'єктів: Підручник. К.: Либідь, 2003. 272 с.
4. Андреев В. А. Релейная защита, автоматика и телемеханика в системах электроснабжения: учебник для вузов. Москва: Высшая школа, 1985. 391 с.
5. Панченко С. В., Блиндюк В. С., Баженов В. М. Релейний захист і автоматика: Навч. посібник. Харків: УкрДУЗТ, 2021. 276 с.
6. Иванов А. О., Тужилин А.А. Теория экстремальных сетей. Москва-Ижевск: Институт компьютерных исследований, 2003. 424 с.
7. Верити Б. Кабельные системы: проектирование, монтаж и обслуживания. М.: КУДИЦ-ОБРАЗ, 2004. 400 с.
8. Норенков И. П. Основы автоматизированного проектирования: учеб. Для ВУЗов. М. : Изд-во МГТУ им. Н. Э. Баумана, 2009. 430с.
9. Чумаков О. А., Стасевич Н. А. Основы систем автоматизированного проектирования: учеб. – метод. пособие. Минск: БГУИР, 2012. 95 с.
10. Анастасиев П.И. Проектирование кабельных сетей и проводок. М.: Энергия, 1980. 379 с.
11. Денисенко В. В. Компьютерное управление технологическим процессом, экспериментом, оборудованием. М.: Горячая линия-Телеком, 2009. 608 с.
12. Энциклопедия АСУ ТП. URL: [https://www.bookasutp.ru/Chapter9\\_2\\_4.aspx](https://www.bookasutp.ru/Chapter9_2_4.aspx) (дата звернення 22.10.2021).

13. Алямовский А. А. Компьютерное моделирование в инженерной практике. СПб.: БХВ-Петербург, 2005. 800 с.
14. Саєнко С. Ю., Нечипоренко І. В. Основи САПР. Х. : ХДУХТ, 2017. 120 с.
15. Проектування кабельного господарства за допомогою ПЗ ModelStudioCS Кабельне господарство. URL: [www.msca.ru/news/actions-model-studio-cs/news\\_20140415\\_1.html](http://www.msca.ru/news/actions-model-studio-cs/news_20140415_1.html) (дата звертання: 06.11.2021)
16. Ли К. Основы САПР (CAD / CAM / CAE). С.-Пб.: Питер, 2004. 560 с/
17. Зуев С. А., Полещук Н. Н. САПР на базе AutoCAD – как это делается. С.-Пб.: БХВ-Петербург, 2004. 1168 с.
18. Товт В.А. Основи наукових досліджень та методика підготовки дипломних робіт: навч. Посіб. Ужгород: «ТОВ "РіК-У"», 2019. 139 с.
19. Єдина система конструкторської документації (ЄСКД). URL: <http://csm.kiev.ua/nd/nd.php?b=2&l=20> (дата звертання: 19.11.2021).
20. Митчик І.М. До проблеми трасування провідного монтажу. Інформаційні моделі, системи та технології: *Праці ІХ наук.-техн. конф.* (Тернопіль, 08-09 грудня 2021 р.), Тернопіль, 2021. – С. 138.
21. Дейтел Х., Дейтел П. Как программировать на С. М.: ООО "Бином-Пресс", 2008. 1456 с.
22. Шилдт Г. С++: базовий курс. М.: Изд.дом "Вильямс", 2010. 624 с.
23. Кудрявцев Е.М. AutoLISP. Программирование в AutoCAD. М.: «ДМК», 1999. 368 с.
24. Список функций Visual Lisp. URL: <http://www.cad.dp.ua/stats/vlisp.php> (дата звертання: 21.10.2021).
25. Фаулер М. UML. Основы. Краткое руководство по стандартному языку объектного моделирования. СПб.: Символ-Плюс, 2011. 192 с.
26. Толок А.О. Крюковська О.А. Безпека життєдіяльності: Навч. посібник. 2011. 215 с.

27. Зеркалов Д.В. Охорона праці в галузі: Загальні вимоги. Навчальний посібник. К.: Основа. 2011. 551 с.

28. USNRC. 1979. Расчет выбросов радиоактивных материалов в газообразных и жидких сточных водах из реакторов с кипящей водой (код BWRGALE), NUREG-0016/

ДОДАТОК А

Тези конференції

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ІВАНА ПУЛЮЯ

МАТЕРІАЛИ

ІХ НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,  
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



8–9 грудня 2021 року

ТЕРНОПІЛЬ  
2021



<b>А.М. Луцків, Г.А. Абоах, Р.К. Рунімбо, В.М. Соболю</b> РОЗВ'ЯЗАННЯ ЗАДАЧ МАШИННОГО НАВЧАННЯ У СЕРЕДОВИЩАХ ІЗ РОЗПОДІЛЕНОЮ ПАМ'ЯТТЮ	
<b>A.M. Lutskiv, H.A. Aboah, R.K. Rawimbo, V.M. Sobol</b> RESOLVING MACHINE LEARNING TASKS IN DISTRIBUTED MEMORY ENVIRONMENT	130
<b>С.В. Солєнко, Р.О. Жароцький</b> канд. техн. наук МОЖЛИВОСТІ СЕРЕДОВИЩА PLUTUS PLAYGROUND ДЛЯ НАПИСАННЯ ТА ТЕСТУВАННЯ СМАРТ-КОНТРАКТІВ	
<b>S. Solenko, R. Zharovskyi Ph.D.</b> PLUTUS PLAYGROUND ENVIRONMENT CAPABILITIES FOR WRITING AND TESTING SMART CONTRACTS	131
<b>Д.В. Купищєць, Ю.З. Лєщєщєв</b> ЗАСТОСУНОК ДЛЯ МОНИТОРИНГУ ДАНИХ РОЗУМНОГО БУДИНКУ	
<b>D. Kupyshets, Ph.D. Yu. Z. Leshchyshev</b> SMART HOME DATA MONITORING APPLICATION	133
<b>В. Хвостівський, Г. Осухівська, Л. Хвостівська</b> ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ОПРАЦЮВАННЯ МЕРЕЖЕВОГО ТРАФІКУ	
<b>V. Khvostivskyy, H. Osukhivska, L. Khvostivska</b> NETWORK TRAFFIC PROCESSING SYSTEM SOFTWARE	134
<b>Н. Шаблій, А. Шаблій</b> АНАЛІЗ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ МЕТОДІВ БІОМЕТРИЧНОЇ АУТЕНТИФІКАЦІЇ ЗА КЛАВІАТУРНИМ ПОЧЕРКОМ	
<b>N. Shablyi, A. Shablyi</b> ANALYSIS OF INFORMATION TECHNOLOGIES OF BIOMETRIC AUTHENTICATION METHODS KEYSTROKE DYNAMICS	135
<b>Н. Шаблій, А. Шаблій</b> АРХІТЕКТУРА ІНФОРМАЦІЙНОЇ СИСТЕМИ БІОМЕТРИЧНОЇ АУТЕНТИФІКАЦІЇ ЗА КЛАВІАТУРНИМ ПОЧЕРКОМ	
<b>N. Shablyi, A. Shablyi</b> ARCHITECTURE OF BIOMETRIC AUTHENTICATION INFORMATION SYSTEM KEYSTROKE DYNAMICS	136
<b>В.В. Яцишин, Ю.Р. Мєльник, А.В. Волощєв</b> КОМПОНЕНТИ КОМП'ЮТЕРНОЇ СИСТЕМИ ПРОГНОЗУВАННЯ ПОВЕДІНКИ РУХУ АВТОМОБІЛЯ	
<b>V.V. Yatsyshyn, Yu.R. Melnyk, A.V. Voloshchuk</b> COMPONENTS OF COMPUTER SYSTEM FOR CAR MOVEMENT BEHAVIOR PREDICTION	137
<b>І.М. Митчєв</b> ДО ПРОБЛЕМИ ТРАСУВАННЯ ПРОВОДОВОГО МОНТАЖУ	
<b>I.M. Mytchuk</b> O A PROBLEM ON TRACING OF WIRE ASSEMBLING	138

**ДО ПРОБЛЕМИ ТРАСУВАННЯ ПРОВОДОВОГО МОНТАЖУ**

UDC 004.021

**I.M. Mytchuk****TO A PROBLEM ON TRACING OF WIRE ASSEMBLING**

Трасування кабельних мереж є однією з найбільш важливих і трудомістких завдань при проектуванні об'єктів електроенергетики. Як правило, ці роботи виконуються вручну фахівцем сектора первинних з'єднань проектною організацією. Підрахунок довжин кабелів і траси розкладки вибираються з великим наближенням, що часто призводить до необґрунтованих переїнтерат і здорожчення кабельної продукції на етапах закупівлі і будівництва об'єкта.

В алгоритмічному плані задача трасування – це побудова для всіх ланцюгів схеми реальних оптимальних монтажних з'єднань. Розрізняють метричний і топологічний аспекти завдання трасування. Метричний аспект передбачає врахування конструктивних розмірів елементів, з'єднань і комутаційного поля. Топологічний аспект пов'язаний з вибором допустимого просторового розташування окремих монтажних з'єднань на комутаційному полі при обмеженнях на кількість перетинів з'єднань, число шарів комутаційної схеми тощо. Алгоритмічні методи трасування друкованих з'єднань істотно залежать від конструкції комутаційного поля і можуть бути розділені на дві основні групи. До першої групи належать топографічні методи, в яких пріоритет надається метричному аспекту задачі. Друга група заснована на графо-теоретичному підході до вирішення задачі трасування.

Побудова мінімальних дерев з'єднань є основним завданням, котре необхідно вирішити при трасуванні проводового монтажу. Для комп'ютерної реалізації найбільш ефективним є алгоритм Прима, згідно якого наступні принципи повинні послідовно виконуватися: будь-яка іпольована вершина з'єднується з найближчою; всякий іпольований фрагмент (пов'язана група вершин) обов'язково з'єднується з найближчою вершиною при допомозі найкоротшого ребра. Під відстанню між вершинами варто розуміти значення ваги, надане ребрам відповідного графа. Відстанню вершини від даного іпольованого фрагмента є мінімум його відстані до окремих вершин фрагмента. Таким чином, алгоритм створення мінімального зв'язуючого дерева для ланцюга з  $n$  виводами може бути тепер описаний так: для довільного виводу ланцюга отримати найближчий і виконати з'єднання; за кожним наступним кроком  $i=2,3,\dots,n-1$  з множини непід'єднаних виводів вибрати той, який знаходиться ближче за інші (в означеному вище сенсі) до групи вже пов'язаних виводів, і під'єднати його до цієї групи по найкоротшому шляху. Дерево, котре буде побудоване таким чином, володітиме мінімальною сумарною довжиною з'єднань.

При проектуванні друкованих з'єднань, особливий інтерес представляє побудова зв'язуючих дерев спеціального виду – дерев Штейнера [1]. Алгоритми побудови мінімальних зв'язуючих дерев не залежать від способу оцінки відстаней між виводами. У той же самий час методи рішення задачі Штейнера істотно залежать від метрики. Проведені дослідження виявили у дерев Штейнера деякі характерні властивості, які послужили основою для розробки точних і наближених алгоритмів розв'язання задачі.

**Література.**

1. Иванов А. О., Тужилин А.А. Теория экстремальных сетей. Москва- Ижевск: Институт компьютерных исследований, 2003. 424 с.

## ДОДАТОК Б

### Програмний код реалізації алгоритму побудови мінімальних дерев

#### Штейнера

```
// Крок 1) За годинниковою стрілкою по спіралі нумеруємо точки
// множини P (тільки що поставлені в масиві монтажного простору
// точки)
//
TRect Field; // Прямокутник, що обмежує точки множини P
Field.Left = MAXINT; // Початкова ініціалізація координат його
// сторін
Field.Right = 0;
Field.Top = MAXINT;
Field.Bottom = 0;
num = 1;
// Цикл по всьому масиву монтажного простору для пошуку
// прямокутника, що обмежує точки множини P
for (int y = 0; y <MAPY * PARTSIZE; y++)
for (int x = 0; x <MAPX * PARTSIZE; x++)
// Якщо точка не порожня і вона позначена як тільки що поставлена
// (не нумерована)
if (mas [x] [y] && GetPriznak (mas [x] [y]) == NONUM) {
// Розширення сторін прямокутника так, щоб він включав дану точку
Field.Left = min (x, Field.Left);
Field.Right = max (x, Field.Right);
Field.Top = min (y, Field.Top);
Field.Bottom = max (y, Field.Bottom);
}
int cx = Field.Right - Field.Left, // Розміри прямокутника
cy = Field.Bottom - Field.Top;
// Цикл руху по спіралі (квадратній) в масиві монтажного простору і
// нумерація точок множини P
```

```

// px, py - координати лівого нижнього кута поточного витка спіралі
for (int py = 0, px = 0; py < cy / 2 + 1 && px < cx / 2 + 1; py ++, px
++) {
int x2 = cx - px, // координати правого верхнього кута поточного
// витка спіралі
y2 = cy - py;
// Цикл руху вгору по лівій стороні квадратної спіралі
for (int y = py; y <= y2; y ++ )
NumPoint (Field.Left + px, Field.Bottom - y);
// нумерація точок, які зустрічаються
// Цикл руху вправо по верхній стороні квадратної спіралі
for (int x = px; x <= x2; x ++ )
NumPoint (Field.Left + x, Field.Bottom - y2);
// нумерація точок, які зустрічаються
// Цикл руху вниз по правій стороні квадратної спіралі
for (int y = y2; y >= py; y -- )
NumPoint (Field.Left + x2, Field.Bottom - y);
// нумерація точок, які зустрічаються
// Цикл руху вліво по нижньому боці квадратної спіралі
for (int x = x2; x >= px; x -- )
NumPoint (Field.Left + x, Field.Bottom - py);
// нумерація точок, які зустрічаються
}
//
// Крок 2) Присвоїти кожній точці множини P ознаки переважного
// проведення траси (0 - уздовж осі x, 1 - уздовж осі y)
// Ознака присвоюється поперемінно, починаючи з P1 = ознака 0;
//Точки на одній вертикалі/горизонталі отримують одну і ту ж ознаку
//
int priznak = 0; // ознака поточної точки
// Цикл по всіх пронумерованих точках множини P для призначення
// ознаки
for (int n = 1; n < num; n ++ )
// Цикл по всьому масиву монтажного простору для пошуку координат

```

```

// поточної точки множини P
for (int y = 0; y <MAPY * PARTSIZE; y ++)
for (int x = 0; x <MAPX * PARTSIZE; x ++)
// Якщо знайдена необхідна точка множини P і їй ще не призначено
// ознака
if (GetNum (mas [x] [y]) == n && GetPriznak (mas [x] [y]) == NOPR) {
// Призначаємо ознаку точці
mas [x] [y] = Pixel (mas [x] [y], priznak);
// Цикл пошуку в масиві монтажного простору точок на одній вертикалі
// з даною
for (int ly = 0; ly <MAPY * PARTSIZE; ly ++)
// Якщо знайдена непорожня точка і їй ще не призначено ознаку
if (mas [x] [ly] && GetPriznak (mas [x] [ly]) == NOPR)
// Призначаємо точці таку саму ознаку
mas [x] [ly] = Pixel (mas [x] [ly], priznak);
// Цикл пошуку в масиві монтажного простору точок на одній
// горизонталі з даною
for (int lx = 0; lx <MAPX * PARTSIZE; lx ++)
// Якщо знайдена непорожня точка і їй ще не призначено ознака
if (mas [lx] [y] && GetPriznak (mas [lx] [y]) == NOPR)
// Призначаємо точці таку саму ознаку
mas [lx] [y] = Pixel (mas [lx] [y], priznak);
priznak = 1 - priznak; // міняємо ознаку на протилежну (з 0 на 1 і з
// 1 на 0)
}
//
// Крок 3) Починаючи з P1 знайти таку Pi, відстань до якої
// мінімальна
// Побудувати фрагмент дерева.
// Прапорець - чи залишаються ще точки для трасування
bool working;
// Цикл трасування всіх точок
do {
// Цикл по всіх номерах точках множини P

```

```

for (int nm = 1; nm < num; nm ++ ) {
TPoint Best, Best2; // координати знайдених найкращих точок,
int bestJlmg = MAXINT, // відстань між ними,
best_num = -1, // найменший з їх номерів
priznak; // ознака
// Цикл перебору в масиві монтажного простору точок з поточним
// номером
// for (int y = 0; y < MAPY * PARTSIZE; y ++ )
// for (int x = 0; x < MAPX * PARTSIZE; x ++ ) {
for (int y = Field.Top; y <= Field.Bottom; y ++ )
for (int x = Field.Left; x <= Field.Right; x ++ ) {
// Якщо точка порожня або її номер не той, який потрібен
if (GetNum (mas [x] [y]) != nm)
continue; // перехід до наступної ітерації (наступної точки)
// Цикл пошуку в масиві монтажного простору точки найбільш
// близької по відстані до поточної
// for (int py = 0; py < MAPY * PARTSIZE; py ++ )
// for (int px = 0; px < MAPX * PARTSIZE; px ++ ) {
for (int py = Field.Top; py <= Field.Bottom; py ++ )
for (int px = Field.Left; px <= Field.Right; px ++ ) {
if (! mas [px] [py])
continue; // перехід до наступної ітерації (наступної точки)
int num = GetNum (mas [px] [py]); // отримуємо номер точки
// Якщо точка порожня або це точка тієї ж ланцюга, що і поточна
if (num == nm)
continue; // перехід до наступної ітерації (наступної точки)
// Обчислюємо відстань між точками по метриці
int lng = abs (x - px) + abs (y - py);
// Якщо відстань менша за найкращу
if (lng < best_lng) {
bestJlmg = lng; // запам'ятаємо цю відстань як найкращу
// запам'ятаємо найменший з їх номерів точок
best_num = min (GetNum (mas [x] [y]), GetNum (mas [px] [py]));
// запам'ятаємо ознаку точки, яка має найменший номер

```

```

if (best_num == GetNum (mas [x] [y]))
priznak = GetPriznak (mas [x] [y]);
else
priznak = GetPriznak (mas [px] [py]);
// Запам'ятаємо координати обох точок як найкращих кандидатів на
// прокладку траси
Best = Point (x, y);
Best2 = Point (px, py);
}
}
}
// Якщо були знайдені 2 вершини для прокладки траси
if (best_num > 0) {
// Побудуємо фрагмент дерева
Connect (Best.x, Best.y, GetPriznak (mas [Best.x] [Best.y]),
Best2.x, Best2.y,
best_num);
//
// Крок 4) Всім точкам монтажного простору, через які пройшов
// фрагмент привласнити найменший з номерів кінцевих точок фрагмента
//
ReNum (bestJnum, priznak); // Виконання кроку 4) алгоритму
}
}
working = false; // Скидаємо прапор
// Цикл по всьому масиву монтажного простору для перевірки, чи
// залишаються ще точки для трасування
for (int y = 0; y < MAPY * PARTSIZE; y++)
for (int x = 0; x < MAPX * PARTSIZE; x++) {
// Отримуємо номер поточної точки
int num = GetNum (mas [x] [y]);
// Якщо точка не порожня і її номер не дорівнює 1
if (num && num != 1)
working = true; // встановлюємо прапорець

```

```

}
} While (working); // Поки залишаються точки для трасування
// Цикл по всьому масиву монтажного простору для підрахунку SDS
for (int y = 0; y <MAPY * PARTSIZE; y ++)
for (int x = 0; x <MAPX * PARTSIZE; x ++)
// Якщо точка не порожня
if (mas [x] [y]) {
// Якщо від неї прокладена траса вниз
if (mas [x + 1] [y])
boardnets [nnet] .SDS ++; // збільшуємо на 1 піксель
// Якщо від неї прокладена траса вправо
if (mas [x] [y + 1])
boardnets [nnet] .SDS ++; // збільшуємо на 1 піксель
}
// Цикл по всьому масиву монтажного простору для відображення траси
// на канві
for (int y = 0; y <MAPY * PARTSIZE; y ++)
for (int x = 0; x <MAPX * PARTSIZE; x ++)
// Якщо точка не порожня
if (mas [x] [y])
// На канві відображаємо точку траси у вигляді маленького кола
boardnets [nnet] .tracednet-> Canvas-> Ellipse (x-TRACKSZ + 1,
y-TRACKSZ + 1,
x + TRACKSZ, y + TRACKSZ);
// Відображення монтажного простору зі схемою на екрані
// Form1-> DrawSchematic ();

```