Трасування провідних з'єднань

1. Постановка задачі
2. Алгоритми побудови **НЗС** (найкоротших зв'язуючих сіток):
- алгоритм Краскала
- алгоритм Прима

Трасування провідних монтажних з'єднань є **завершуючим етапом** проектування. Для вирішення цюєї задачі необхідно у відповідності електричної принципової схеми (монтажної схеми) пристрою вибрати порядок з'єднувальних виводів (контактів) схеми та визначити конкретну геометрію трас. При цьому необхідно враховувати конкретні конструкторсько-технологічного обмеження.
На відміну від алгоритмів розбиття та розміщення алгоритми **трасування** є **спеціалізованими** , тобто пристосовані для рішення конкретної задачі (ортогональна траса або ні, можливий перехід з шару на шар в МДП, скільки шарів в МДП відводиться для траси і т.д.
Не дивлячись на швидкий ріст, широке застосування **друкованого монтажа,** питомого часу провідних з'єднань в складному ЕА застосування провідного монтажу залишаються *достатньо* **великим**. В серійному виробництві **провідний монтаж** використовується при електричному об'єднанні конструктивних одиниць *високих рівней* ієрархії - блоків, панелей, стійок та інше. При **мілкосерійному** виробництві монтаж використовують при розводці дискретних компонентів. В останній час успішно використовують ДПП з додатковим провідим монтажом.

Трасування провідних з'єднань виконують двома способами:
1. По прямим, що з'єднують виводи окремих компонентів - монтаж внавал
2. Монтаж за допомогою жгутів

Перший спосіб відрізняється простотою виконання та високою перешкодостійкостю, тому що дозволяє до мінімума скоротити довжину провідників та простяганням ділянок їх паралельного проходження.

Однак висока ймовірність помилок монтажу, а контроль якості монтажу складний. Мала ремонтнопридатність.

Другий спосіб припускає об'єднання окремих провідників в жгути, які укладаються в окремі канали, передбаченні в монтажному полі. Цей спосіб більш технологічний, спрощується процес контроля та усунення помилок, допущених при монтажі.

Таким чином з точки зору задач трасування провідних з'єднань безпосередньо вирішуваних конструктором, використання монтажу внавал припускає визначення пар безпосередньо з'єдуванних контактів. При монтажі в жгути необхідно не тільки виконати з'єднання пар контактів, але також виконати розкладання всіх провідників по каналам, тобто визначити послідовність з'єднання виводів та конфігурації провідників (*приклад привести*).

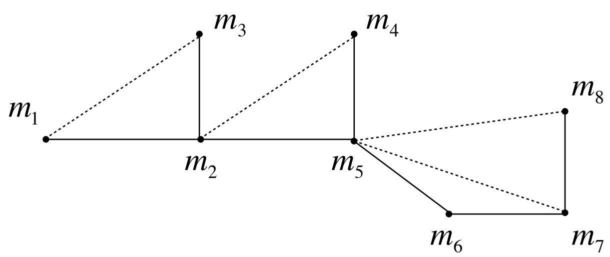
З урахуванням указаних особливостей виконання провідного монтажу **задача** *трасування провідних з'єднань* в загальному випадку формується наступним чином.

Задано місцезнаходження множини виводів (перед трасуванням вирішена задача розміщення) $M=\left\{m_1,m_2,\ldots,m_p\right\}$. У відповідності до схеми електричної принципової (СЕП) множину M розіб'ємо на пересічні підмножини $M^{(1)},M^{(2)},\ldots M^{(i)},\ldots M^{(k)}$, кожна з яких включає виводи, що підлягають електричному об'єднанню. Для кожної підмножини $M^{(i)}=\left\{m_1^{(i)},m_2^{(i)},\ldots,m_n^{(i)}\right\}\ i=\overline{1,k}$ потрібно визначити послідовність з'єднання виводів та конфігурацію провідників, що забезпечують при заданих обмеженнях **мінімальну сумарну довжину з'єднань** - критерій оптимізації.

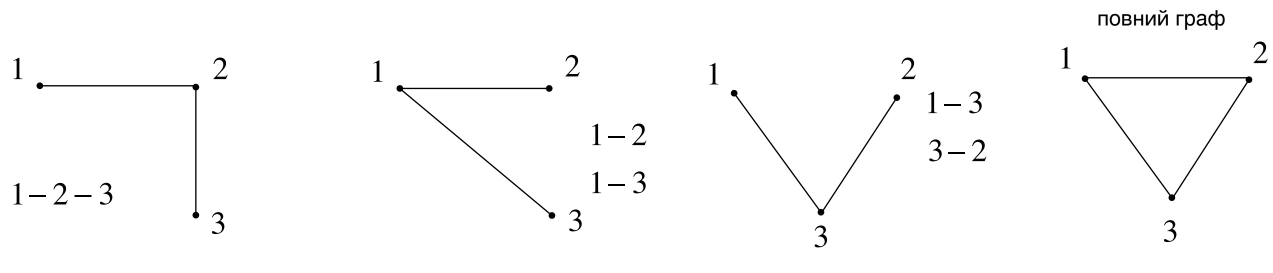
При джгутовому монтажі після визначення пар з'єднуваних контактів, вирішують задачу формування жгутів (конфігурацію провідників), при якій визначається таке розподілення провідників, які забезпечують мінімальну сумарну довжину з'єднань при максимальному числі трас простої конфігурації.
При формуванні жгутів, необхідно враховувати параметри сигналу в провідниках. Так, при одночасному трасуванні потужних та слабосигнальних (чуттєвих до наведень) ланцюгів очевидно в першу чергу прокладуються слабосигнальні ланцюги з ціллю мінімізації їх довжини. Потім визначається конфігурація потужних ланцюгів. Після цього виконується трасування всіх інших ланцюгів.

Таким чином вимоги до перешкодостійкості приводять до зміни порядку трасування окремих ланцюгів.
Задача побудови ланцюгів з мінімальною сумарною довжиною ребер зводиться до задачі побудови **НЗС**.

Ця **задача проста**, коли до ланцюгу входить **мала** *кількість виводів* та виявляється достатньо складною при великий потужності підмножини $M^{(i)}$, тому що число зв'язуючих дерев електричних ланцюгів визначає $t_n = n^{n-2}$, де n - число вершин, що входить до множини $M^{(i)}$. Очевидно, що серед різних варіантів з'єднань (див. рис.) вершин (виводів) підмножини $M^{(i)}$ доцільно розглядати лише деревоподібні структури.

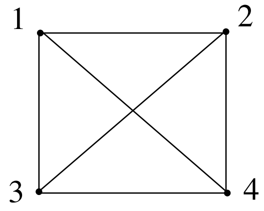


Практично **задача** *зводиться* до відшуку **оптимального дерева**, а для цього необхідно переглянути $t_n = n^{n-2}$ варіантів.
$n = 3 \quad t_n = 3^{3-2} = 3$



Число ребер в графі $N = \dfrac{n(n-1)}{2} = \dfrac{3\times 2}{2} = 3$

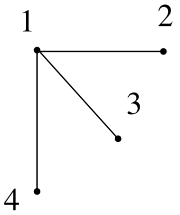
Число дерев $t = n^{n-2} = 3$, які включають всі вершини



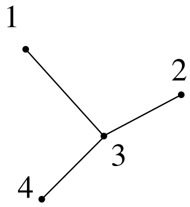
1)

При $n = 4 \quad t_n = 4^{4-2} = 16$ варіантів дерев

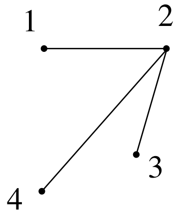
$N = \dfrac{4(4-1)}{2} = 6$



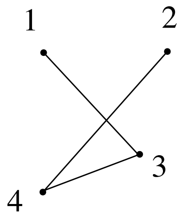
$1-2 \quad 1-3 \quad 1-4$



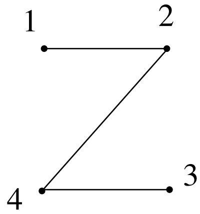
$1-3 \quad 3-4 \quad 3-2$



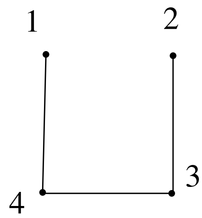
$1-2 \quad 2-3 \quad 2-4$



$1-3 \quad 3-4 \quad 4-2$



$1-2 \quad 2-4 \quad 2-3$



$2-3 \quad 3-4 \quad 4-1$

Кількість еквіпотенціальних ланцюгів, що з'єднують виводи, визначають як
$t_n = n^{n-2}$
n - кількість виводів в еквіпотенційному ланцюгу.

при $n = 3 \quad t_n = 3$ варіантів ланцюгів
$n = 4 \quad t_n = 16$ варіантів ланцюгів
$n = 5 \quad t_n = 125$ варіантів ланцюгів

тому ми сказали, що при малому n виявляється достатньо складною при великій потужності множини M^i.
Ця задача вирішується просто алгоритмами побудови **НЗС**: алгоритм Краскала та алгоритм Прима.

Алгоритм побудови НЗС

Алгоритм побудови **НЗС** заснований на послідовному виборі з множини зв'язків самих коротких, та які не утворюють циклів з раніш відображеними зв'язками. Утворена **НЗС** повинна з'єднувати всі виводи, але не закорочувати їх (приклад вище 1)). Приєднання окремих зв'язків продовжується до тих пір, поки не буде отримано мінімальне дерево, що включає всі вершини (виводи).

Для побудови **НЗС** використовують два алгоритми:
- алгоритм Краскала
- алгоритм Прима

M - виводи окремих КЕ, що входять до ланцюга

Алгоритм Краскала

Згідно задачі трасування провідних з'єднань

- задано місце знаходження множини точок (підмножини <u>виводів</u> (виводів - не X) КЕ), що входять до ланцюга

$$M^i = \{ m_1^{(i)}, m_2^{(i)}, \ldots, m_n^{(i)} \} \; i = \overline{1,k}$$

Необхідно визначити послідовність вибору виводів - зв'язків

- на множині M^i будуємо повний граф $G = \{ M^i, V^i \}$ та визначаємо число всіх ребер (зв'язків) повного графу

$N = \dfrac{n(n-1)}{2}$ $\quad n$ - число вершин в графі (підмножина контактних майданчиків (КМ), що потрібно з'єднати)

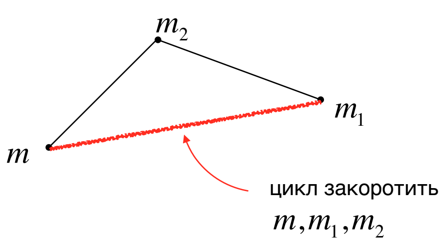
- обчислюємо довжину всіх ребер графу (будь-яким способом) - в евклідовій або ортогональній метриці
- впорядковуємо список ребер по довжині так, щоб виконалася умова

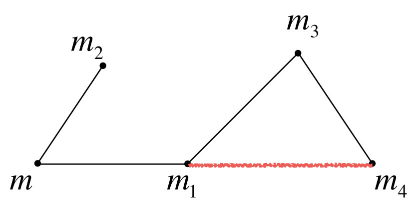
$$\forall v_i \in V \left[d(v_i) \le d(v_{i+1}) \right] \quad 1 \le i = N$$

$d(v_i)$ - довжина ребра i

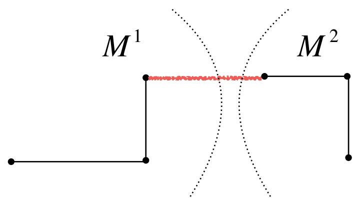
Тобто в результаті отримаємо список довжин (упорядкованих).

- послідовно вибираємо з списку довжин ребра з мінімальною довжиною та починаємо будувати дерево. При кожному виборі ребра перевіряємо умову утворення циклів. Утворення циклів визначаємо по належності їх ребер дереву, що будується. Ребра, що створюють цикл, не приєднують до дерева, що будується.





m_4 - на данному кроці в КСС включати не можна

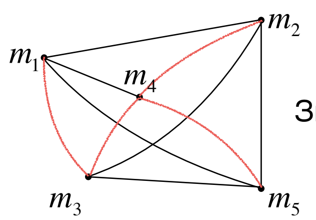


Переваги:

Алгоритм дозволяє паралельну побудову декількох піддерев

Недоліки:

- кожен раз потребується перевірка на утворення циклів
- я б ще відмітив й такий недолік - нема матриці, до якої ми звикли, але її можна використовувати



<u>Дано</u> $M - \{ m_1, m_2, m_3, m_4, m_5 \}$

<u>Побудувати</u> НЗС, включаючи 5 вершин.

Звертаю увагу: дано множину виводів (а не КЕ), на яких необхідно побудувати трасу, тобто зв'язати їх провідником.

Робимо так:

- будуємо повний граф та визначаємо кількість ребер

- визначаємо $N = \frac{n(n-1)}{2} = \frac{5\times 4}{2} = 10 \quad t_n = n^{n-2} = 5^3 = 125$

- визначаємо довжину кожного зв'язку. Нехай довжини будуть такими:

Зразу упорядковуємо довжини по зростанню. Нехай довжини будуть такими:

$\vee\ d_{45} = 1.8$

$\vee\ d_{24} = 2.8$ -три вывода - перевіримо створення циклу

$\vee\ d_{34} = 3$ - 4 вивода - перевіряємо створення циклу

$d_{35} = 3.2$ - цикл

$d_{25} = 4$ - цикл

$\vee\ d_{13} = 4$ - дерево побудоване

$\vee\ d_{14} = 5$

$d_{12} = 5.2$

$d_{23} = 6$

$d_{15} = 8$

Слідує припустити, що затребувані будуть дані малих значень d_{ij}, а декілька послідуючих значень не будуть в деяких випадках зовсім затребувані (використовувані)

$d = d_{45} + d_{24} + d_{34} + d_{13}$ - умовно коротка сумарна довжина, тому що $d_{35} = 3.2 < d_{13} = 4$

Алгоритм закінчує роботу як буде побудоване дерево (будуть з'єднані виводи в M^i .

$\sum L = 1.8 + 2.8 + 3 + 4$ од. довжини

Можно стверджувати, що

1) перши два ребра завжди будуть включені в **НЗС**

2) **НЗС** побудоване за менше число кроків буде оптимальніше

Підмножина M^i, буде стільки, скільки ланцюгів. Задача побудови **НЗС** вирішується для ланцюгів, що містять число КП рівне та більше трьох

$n = 2 \quad t = 2^{2-2} = 1$

$n = 3 \quad t = 3^{3-2} = 3$

$n = 4 \quad t = 4^{4-2} = 16$

В процесі вирішення можна отримати декілька окремих дерев, які в кінці кінців створюють однозв'язне дерево. Алгоритм дає оптимальне рішення.

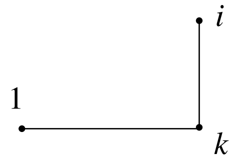
Алгоритм Прима

Суть алгоритму складається в послідовному приєднанні вершин до дерева, що будується. Перші два кроки алгоритму такі ж як і в алгоритмі Краскала.

- будуємо повний граф
- визначаємо відстань між усіма парами вершин
- формуємо матрицю довжин (таким чином в алгоритмі Краскала теж може бути $\|D\|$)

$$D = \left|d_{ij}\right|_{\underbrace{n\times n}_{\text{скільки виводів входить до КСС}}}$$

$$
D = \begin{array}{c|cccccccc}
 & 1 & 2 & \cdots & i & \cdots & k & \cdots & n \\
1 & 0 & d_{12} & \cdots & d_{1i} & \cdots & d_{1k} & \cdots & d_{1n} \\
2 & d_{21} & 0 & \cdots & d_{2i} & \cdots & d_{2k} & \cdots & d_{2n} \\
\vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
i & d_{i1} & d_{i2} & \cdots & 0 & \cdots & d_{ik} & \cdots & d_{in} \\
\vdots & \cdots & \cdots & \cdots & \cdots & 0 & \cdots & \cdots & \cdots \\
k & d_{k1} & d_{k2} & \cdots & d_{ki} & \cdots & 0 & \cdots & d_{kn} \\
\vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
n & d_{n1} & d_{n2} & \cdots & d_{ni} & \cdots & d_{nk} & \cdots & 0
\end{array}
$$



$d_{1k} = \min$

$d_{ki} = \min$

- аналізуємо перший рядок матриці D (перщий рядок вибираємо без всякого аналізу) та в ньому вибираємо елемент з мінімальним значенням - нехай $d_{1k} = \min$.
- в матриці D викреслюємо 1-й та k-й стовпчики. З'єднуємо 1й та k-й виводи.
- в новій матриці аналізуємо 1й та kй рядки, і в них вибирається елемент з мінімальним значенням , нехай $d_{ki} = \min$, з'єднуємо k та і виводи.
- викреслюється стовпчик і та в матриці аналізуються 1,k та і рядки та знову вибирається в них мінімальний елемент та так до тих пір поки не будуть проаналізовані всі рядки (поки n-1 стовпчиків будуть викреслені).

Іншими словами - з матриці послідовно вибираємо самі короткі зв'язки.

Приклад

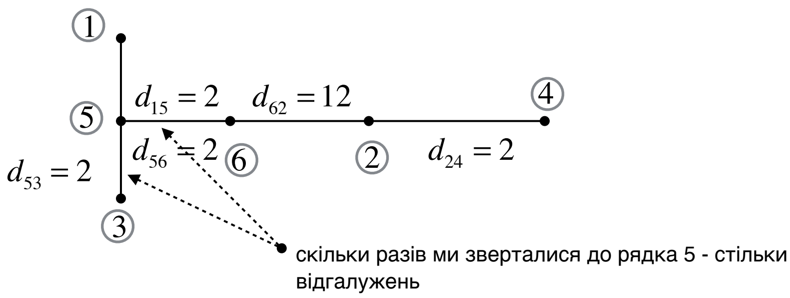
Нехай після визначення довжин між парами вершин повного графу матриця довжин має наступний вид

$$
D = \begin{array}{c|cccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 \\
1 & 0 & 4 & 5 & 10 & 2^{1\,\min} & 7 \\
2 & 4 & 0 & 12 & 2^{5\,\min} & 5 & 1 \\
3 & 5 & 12 & 0 & 4 & 2 & 7 \\
4 & 10 & 2 & 4 & 0 & 9 & 15 \\
5 & 2 & 5 & 2^{2\,\min} & 9 & 0 & 2^{3\,\min} \\
6 & 7 & 1^{4\,\min} & 7 & 15 & 2 & 0
\end{array}
$$

в п'ятому рядку два рівних мінімальних елемента - алгоритм рекомендує вибрати 1й

6 виводів , що входять в провідник

6 виводів необхідно з'єднати між собою. Необхідно визначити послідовність з'єднання виводів.



Аналізуємо 1-й рядок. Переглядаємо перший рядок матриці: $d_{15} = 2 = \min$. Побудову дерева починаємо з ребра між вершинами **1-5**.

*Аналізуємо рядки **1-5***. Викреслюємо **1-5** стовпчики (тобто першого разу викреслюємо два стовпчики) та в рядках **1-5** вибираємо мінімальний елемент - $d_{53} = \min = 2$. В пятому рядку два рівних мінімальних елемента $d_{53} = d_{56} = 2$. Алгоритм рекомендує вибрати 1-й елемент тобто в рядку ij (елемент з меншим j) - $d_{53} = 2$. Будуємо гілку **5-3**.

*Аналізуємо рядки **1-5-3***. Викреслюємо третій стовпчик. Аналізуємо **1**, **3**, **4** рядки, вибираємо $d_{56} = 2 = \min$. Будуємо гілку **5-6**.

*Аналізуємо рядки **1-5-3-6***. Викреслюємо шостий стовпчик та в рядках **1**, **3**, **5**, **6** визначаємо $d_{62} = 1 = \min$. Будуємо гілку **6-2**.

*Аналізуємо рядки **1-5-3-6-2***. Викреслюємо 2й стовпчик та в рядках **1**, **3**, **5**, **6**, **2** вибираємо $d_{24} = \min = 2$, будуємо гілку **2-4**.

НЗС $= d_{15} - d_{53} - d_{56} - d_{62} - d_{24}$

$L_{\text{НЗС}} = d_{15} + d_{53} + d_{56} + d_{62} + d_{24} = 2 + 2 + 2 + 1 + 2 = 9$ од. довжини

Недолік:
Нема обмежень на число ребер, інцидентних одній вершині, скільки разів ми звернулися до рядку 5 - стільки відгалужень. На практиці виходячи з вимог технологічності, таке обмеження завжди присутнє (число інцидентних вершин не більше 3).

В алгоритмі Прима виникнення окремих дерев в процесі вирішення задачі неможливо. Суть алгоритму складається в послідовності приєднання вершин до будуємого дерева.

В заключення.

Найбільш ефективним з точки реалізації на ЕОМ являється алгоритмом Прима, в якому виконується послідовне приєднання ізольованих вершин з найближчою. Всякий ізолюємий фрагмент (зв'язана група вершин) з'єднується з найближчою вершиною найкоротшим ребром.

Алгоритм побудови мінімального зв'язуючого дерева для ланцюга n виводами може бути описане так:

1. Для довільного виводу ланцюга знайти найближчий та виконати з'єднання (тобто побудову дерева починаємо не з першої вершини)
2. На кожних послідуючих кроках $i = 2, 3, \ldots n-1$ з множини непід'єднаних виводів вибрати той, що який знаходиться ближче інших до групи вже зв'язаних, та приєднати його до цюєї групи по найкоротшому шляху (тобто знову аналізуємо рядки)

Побудоване таким чином дерево буде мати мінімальну сумарну довжину з'єднань.

Тепер вихідна задача зводиться до визначення в графі G дерева, що включає всі вершини X та має мінімальну сумарну вагу ребер.

Таке дерево з різній літературі називають мінімальним покриваючим деревом мінімального зв'язуючим деревом **НЗС**, деревом Прима.

Після розгляду прикладу задачу трасування провідних з’єднань можна сформулювати більш просто наступним чином:

- задано множина виводів, які необхідно з’єднати електричним ланцюгом.
- оскільки координати всіх виводів відомі, то задача зводиться до побудови на фіксованих вершинах графу-дерева- G з мінімальною сумарною довжиною ребер.

Таке дерево в різній літературі називають мінімальним покриваючим деревом, **НЗС**, деревом Прима.