

## Евристичні алгоритми трасування.

Необхідно розробити алгоритм, що усуває недоліки хвильового алгоритму - збільшення швидкодії

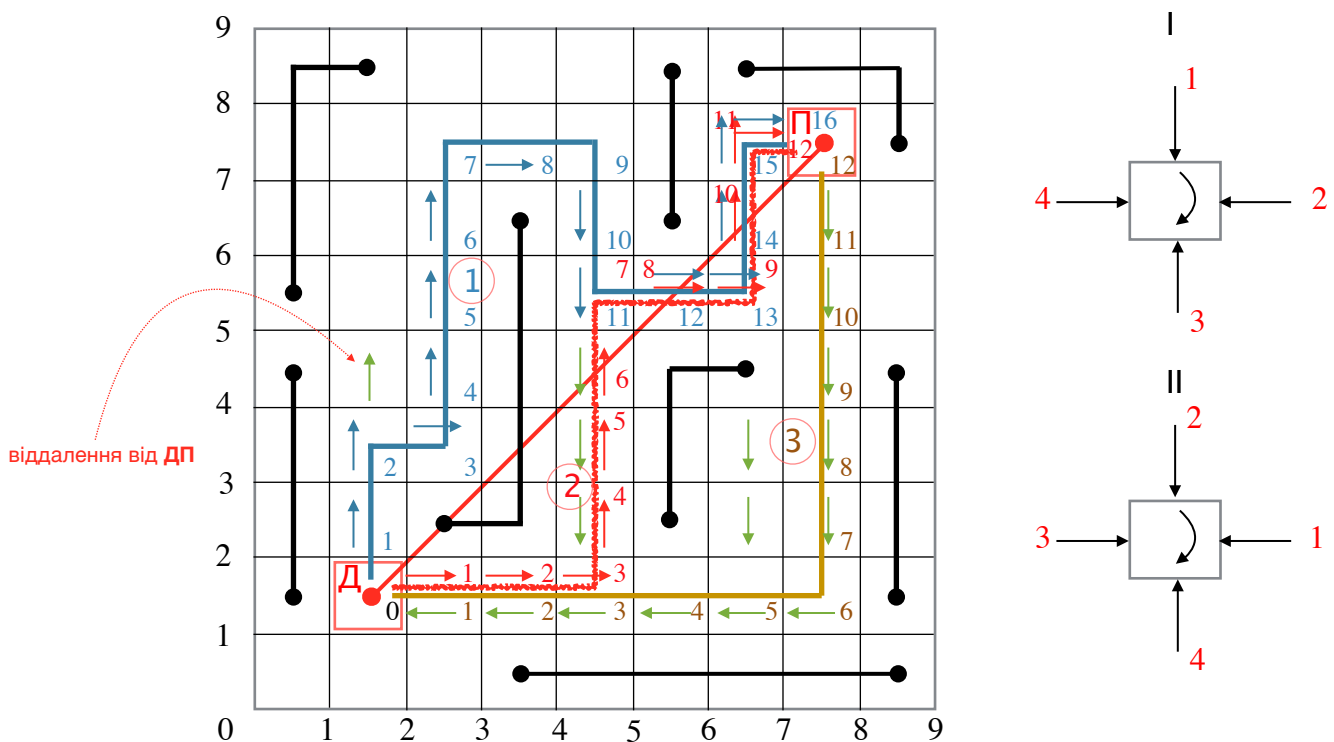
Алгоритми даної групи відносяться до найбільш швидкодійних та простих в програмуванні. Закладений в їх основу пріоритетний порядок побудови траси та обходу перешкод тягне за собою неоптимальність побудови траси. Даний алгоритм знаходить застосування, коли необхідно швидко виконати трасування ДП, ІС. Наприклад, в дослідному виробництві.

Розглянемо один з можливих варіантів алгоритму.

**“Д”** та **“П”** з'єднують прямою - мінімальна довжина між **“Д”** та **“П”**

На кожному кроці з числа вільних сусідніх сот вибирають ту, в якій відстань до **“П”** (по прямій) зменшується на максимально можливу величину або збільшується на мінімально допустиму величину. При рівних умовах вибираємо напрямку, що відповідає пріоритетному напрямку **ШК**. Якщо на шляху побудови траси зустрічаються перешкоди, то їх обходять по першому вільному напрямку, досліджуючи стан сусідніх сот в порядку вибраного пріоритету.

Розглянемо роботу алгоритму.



**ШК** визначає, як будемо обходити пряму **ДП**.

Задамо **ШК I, II**, тобто задамо пріоритетний напрямку - обходимо пряму **ДП** зліва (**I**) та зправа (**II**). Для першого випадку (**ШК**) вгору, вправо, вниз, вліво.

Приймаємо, що траса віддаляється від **П**, якщо на наступному кроці відстань до даної прямої **ДП** зростає.

Тоді мінімальна відстань до прямої відповідає верхній та правій сотам (див. **ШК**). Вибираємо верхню соту, тому що вона знаходиться на більш пріоритетному напрямку. Аналізуємо наступну соту та будемо провідник, який обійде перешкоду **зліва**.

При зміні порядку **ШК**: вправо, вверх, вліво, вниз - форма провідника зміниться- провідник обійде перешкоду **зправа** (варіант 2)

З аналізу провідників:

**перший провідник** -  $L_1 = 16$  кроків,  $N_1 = 7$  поворотів

**другий провідник** -  $L_2 = 12$  кроків,  $N_2 = 4$  поворотів

*Провідники будуються швидко, тому що аналізуємо малу кількість сот.*

**Слідую** - що вони **складної** конфігурації та не завжди оптимальні по довжині.

Для порівняння **третій провідник**, що побудований хвильовим алгоритмом.

**третій провідник** -  $L_3 = 12$  кроків,  $N_3 = 1$  поворотів. *Для побудови **третього провідника** необхідно проаналізувати велику кількість сот - хвильовий алгоритм.*

Час побудови **ПР1** та **ПР2** зменшився, тому що потрібно аналізувати тільки соти, по яких прокладається траса.

При хвильовому алгоритмі кількість аналізованих сот збільшується - як результат збільшується й час на побудову траси.

Аналогія з теорії авторегулювання.

Користуючись термінологією теорій авторегулювання можна сказати, що цей алгоритм використовує **стеження** траси, що прокладається, за відстанню між прямою та трасою. На кожному кроці цю відстань прагнуть робити мінімальною за винятком обходу перешкоди.

### Променевий алгоритм.

Таким чином ідея зменшення часу на побудову траси нам зрозуміла - це зменшена кількість сот, по яким проходить хвиля при її розповсюдженні. Ця ідея в алгоритмах реалізується по-різному.

В алгоритмі зустрічної хвилі два джерела, але все одно аналізується великі площі, що займають перша та друга хвилі.

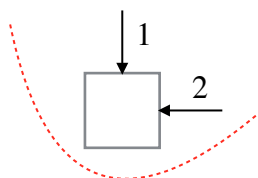
В евристичному алгоритмі, скажемо так не аналізується не одна лишня сота, тобто аналізується стільки сот, скільки вийшло до провідника. Правда - помітний виграш.

В евристичному алгоритмі час можна зменшити, якщо примініти метод зустрічної хвилі.

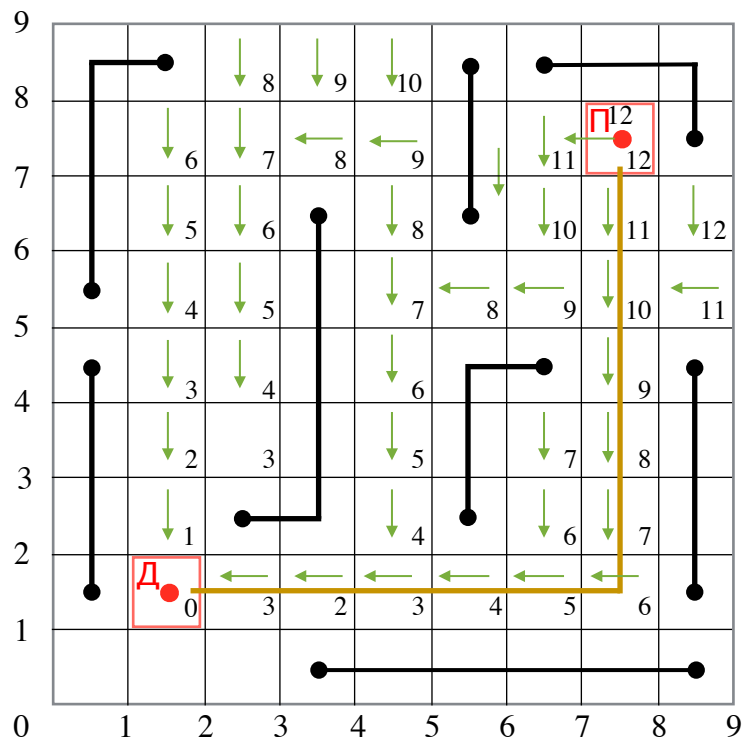
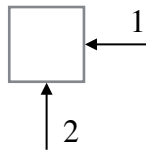
З урахуванням вказаної ідеї можна запропонувати ай такий метод - **променевий алгоритм**.

Давайте згадаємо як розповсюджується хвиля: пам'ятаєте, що хтось говорив, що хвиля розповсюджується **у всіх напрямках** по вільним сусіднім сотам. Тобто в правилах ми все заклали.

Давайте поміняємо правила. Нехай хвиля розповсюджується в I квадратні координат **ДП**. Така зміна правил заснована на тому, що "**Д**" та "**П**" знаходяться в кутах прямокутника. Тому нехай хвиля розповсюджується від "**Д**" до "**П**" в просторі прямокутника, що обмежений горизонтальними та вертикальними сторонами прямокутника, тобто



Якщо хочемо зменшити час побудови траси - тоді використовуємо алгоритм зустрічної хвилі, тоді хвиля



В заключення можна сказати наступне:

В променевому алгоритмі вибір соти для побудови траси між “Д” та “П” виконується по завчасно вибраним напрямкам, що називаються **променями**. Це дозволяє зменшити число сот, що аналізуються на етапі розповсюдження хвилі, а відповідно скоротити час побудови траси.

**Недолік:** обмеження провідника в напрямках зменшує ймовірність побудови траси складної форми, висока ймовірність блокування траси.