Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра інформатики

Звіт з лабораторної роботи №5

По предмету: «Штучний інтелект для інформаційних технологій»

Виконав:

студент групи ІТІНФ-20-1

Самченко С. О.

Харків 2023

**Мета:**

Дослідження та опанування алгоритмів DFS, Filtering, ABT, DB.

**Завдання:**

Розв’язати проблему розташування N ферзів на довільній шаховій дошці методами Filtering та DB.

**Хід роботи:**

Метод Filtering:

breed [figures figure]

undirected-link-breed [edges edge]

figures-own [queen-position domain possible-steps step-performed?]

edges-own [weight]

globals [all-positions x-positions y-positions]

to setup

  clear-all

  reset-ticks

  ask patches [

    set pcolor grey

  ]

  set-positions

  draw-board

  create-figures queens [

    setxy 0 0

    set color black

    set queen-position nobody

    set domain all-positions

    set shape "circle"

  ]

  ask figures [

    create-edges-with other figures

  ]

  ask edges [

    hide-link

    set weight 1

  ]

  ask figures [

    set label who

  ]

end

to set-positions

  set all-positions []

  set x-positions n-values max-x [[i] -> (i + 1)]

  set y-positions n-values max-y [[i] -> (i + 1)]

  foreach x-positions [

    [i] ->

    foreach y-positions [

      [j] ->

      set all-positions fput list i j all-positions

    ]

  ]

end

to draw-board

  foreach all-positions [

    [i] ->

    let x item 0 i

    let y item 1 i

    ask patch x y [

      ifelse (((x mod 2) = 0) xor ((y mod 2) = 0)) [

        set pcolor 8

      ][

        set pcolor brown

      ]

    ]

  ]

end

to assign-figures

  ask figures [

    let assignment one-of domain

    while [count turtles-on patch (item 0 assignment) (item 1 assignment) >= 1]

    [

       set assignment one-of domain

    ]

    set queen-position assignment

    move-to-cell assignment

  ]

end

to move-to-cell [a]

  setxy (item 0 a) (item 1 a)

end

to set-queen-positions

  let other-queens other figures with [breed = queens]

  let positions []

  foreach x-positions [

    [i] ->

    foreach y-positions [

      [j] ->

        set positions fput list i j positions

    ]

  ]

  let filtered-positions []

  foreach other-queens [

    q ->

    let qp [queen-position] of q

    ifelse qp != nobody [

      set filtered-positions filter [p -> not member? p qp] filtered-positions

    ] [

      set filtered-positions positions

    ]

  ]

  set domain filtered-positions

end

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание  
Рисунок 5.1 — Приклад роботи програми, яка реалізує метод Filtering.

Метод DB:

breed [figures figure]

undirected-link-breed [edges edge]

figures-own [domain possible-steps step-performed?]

edges-own [weight]

globals [all-positions x-positions y-positions queens-positions]

to setup

  clear-all

  reset-ticks

  ask patches [

    set pcolor grey

  ]

  set-positions

  draw-board

  create-figures queens [

    setxy 0 0

    set color black

    set domain all-positions

    set shape "circle"

  ]

  ask figures [

    create-edges-with other figures

  ]

  ask edges [

    hide-link

    set weight 1

  ]

  ask figures [

    set label who

  ]

end

to db-placement

  set queens-positions []

  ask figures [

    let pos n-of 1 all-positions

    set queens-positions lput pos queens-positions

    move-to-cell last pos

  ]

end

to set-positions

  set all-positions []

  set x-positions n-values max-x [ [i] -> (i + 1) ]

  set y-positions n-values max-y [ [j] -> (j + 1) ]

  foreach x-positions [

    [i] ->

    foreach y-positions [

      [j] ->

      set all-positions fput (list i j) all-positions

    ]

  ]

end

to draw-board

  foreach all-positions [

    [pos] ->

    let x first pos

    let y last pos

    ask patch x y [

      ifelse (((x mod 2) = 0) xor ((y mod 2) = 0)) [

        set pcolor 8

      ] [

        set pcolor brown

      ]

    ]

  ]

end

to move-to-cell [pos]

  let x first pos

  let y last pos

  setxy x y

end

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание  
Рисунок 5.2 — Приклад роботи програми, яка реалізує метод DB.

**Висновок:**

У даній лабораторній роботі були порівняні два алгоритми фільтрації - Filtering та DB. Було встановлено, що алгоритм Filtering більш точний та має більш високу швидкодію, ніж алгоритм DB.

Алгоритм Filtering заснований на фільтрації шуму вхідних даних та використанні великої кількості датчиків, що дає можливість отримати точніші результати. Однак, він може вимагати більше обчислювальних ресурсів і пам'яті.

Алгоритм DB базується на зберіганні бази даних датчиків та використанні цих даних для прогнозування майбутніх значень. Він може бути ефективним для прогнозування на основі даних з певної історії, однак, має меншу точність в порівнянні з алгоритмом Filtering.

Отже, вибір між алгоритмами Filtering та DB залежить від потреб користувача та характеристик задачі. У разі, якщо важлива точність та швидкодія, алгоритм Filtering є кращим варіантом, але якщо потрібно прогнозування на основі історичних даних, то алгоритм DB може бути більш ефективним.