# Розділ 4. Дослідження алгоритму

Для проведення експерименту був розроблений програмний продукт. Кожен алгоритм був по 10 разів запущений на випадкових даних розмірності від 3 до 10, тобто було здійснено 70 прогонів алгоритму. На кожній розмірності було визначено середній час роботи алгоритму. За цими даними були побудовані та проаналізовані графіки.

Час роботи генетичного алгоритму в залежності від розмірності представлений у наступній таблиці:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | avg |
| 3 | 0,00079 | 0,00075 | 0,00188 | 0,00086 | 0,00128 | 0,0007 | 0,00073 | 0,00364 | 0,001 | 0,00199 | 0,001362 |
| 4 | 0,00207 | 0,00081 | 0,00185 | 0,00079 | 0,00082 | 0,00194 | 0,00194 | 0,0022 | 0,00089 | 0,00216 | 0,001547 |
| 5 | 0,00092 | 0,00088 | 0,00067 | 0,00082 | 0,00127 | 0,00089 | 0,00112 | 0,00197 | 0,00089 | 0,00082 | 0,001025 |
| 6 | 0,00083 | 0,00097 | 0,00105 | 0,00109 | 0,00084 | 0,00096 | 0,00091 | 0,00086 | 0,00109 | 0,00101 | 0,000961 |
| 7 | 0,00318 | 0,00096 | 0,00252 | 0,00097 | 0,00088 | 0,00364 | 0,00109 | 0,00132 | 0,00227 | 0,00174 | 0,001857 |
| 8 | 0,00214 | 0,00327 | 0,00128 | 0,00163 | 0,0052 | 0,00094 | 0,00169 | 0,0019 | 0,00099 | 0,00508 | 0,002412 |
| 9 | 0,002 | 0,00124 | 0,00632 | 0,00208 | 0,0027 | 0,00149 | 0,00354 | 0,00162 | 0,00207 | 0,00176 | 0,002482 |
| 10 | 0,00289 | 0,00489 | 0,00195 | 0,00473 | 0,00259 | 0,00302 | 0,00442 | 0,00244 | 0,01102 | 0,00641 | 0,004436 |

Таблиця 1. Час та середній час роботи генетичного алгоритму на 10 запусках в залежності від розміру вхідних даних

Рисунок 1. Залежність швидкості від розміру вхідних даних

Графік залежності показує стрімкий зріст часу роботи зі збільшенням розмірності матриці.

Нижче представлені час таблиця з часом роботи та графік залежності часу від розмірності мурашиного алгоритму.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | avg |
| 3 | 0,002 | 0,00082 | 0,00085 | 0,0018 | 0,00214 | 0,00225 | 0,00086 | 0,0008 | 0,00122 | 0,00079 | 0,001353 |
| 4 | 0,0018 | 0,00247 | 0,0037 | 0,00093 | 0,0015 | 0,00362 | 0,00174 | 0,00363 | 0,0027 | 0,0035 | 0,002559 |
| 5 | 0,00111 | 0,00286 | 0,00243 | 0,00113 | 0,00149 | 0,00163 | 0,00396 | 0,00211 | 0,00629 | 0,0022 | 0,002521 |
| 6 | 0,00234 | 0,0043 | 0,00457 | 0,00289 | 0,00309 | 0,00179 | 0,00337 | 0,00304 | 0,00492 | 0,00345 | 0,003376 |
| 7 | 0,00241 | 0,0028 | 0,00181 | 0,00465 | 0,00298 | 0,00578 | 0,00271 | 0,00234 | 0,00252 | 0,00632 | 0,003432 |
| 8 | 0,00475 | 0,00482 | 0,00309 | 0,00302 | 0,00204 | 0,00227 | 0,00509 | 0,00601 | 0,00306 | 0,00507 | 0,003922 |
| 9 | 0,00402 | 0,00296 | 0,00255 | 0,00379 | 0,00252 | 0,00662 | 0,00247 | 0,00606 | 0,00308 | 0,00318 | 0,003725 |
| 10 | 0,00324 | 0,00406 | 0,00458 | 0,00797 | 0,0049 | 0,00936 | 0,01239 | 0,00388 | 0,00608 | 0,00585 | 0,006231 |

Таблиця 2. Час та середній час роботи алгоритму мурашиних колоній на 10 запусках в залежності від розміру вхідних даних

Рисунок 2. Залежність швидкості від розміру вхідних даних

Як видно з графіку, зростання є більш плавним, окрім розмірності 10, де час роботи зростає майже в двічі.

Порівняємо на графіку обидва алгоритми.

Рисунок 3. Порівняльна характеристика швидкості роботи алгоритмів

Де помаранчевим представлений генетичний алгоритм, а синім – мурашиних колоній.

Як видно з двох графіків, швидкість роботи генетичного алгоритму значно вища за швидкість мурашиного, проте це не говорить про перевагу генетичного алгоритму. Перевіримо точність обох алгоритмів, аби прийти до висновку про те, який алгоритм все ж таки краще.

На відомих даних розмірності від 3 до 10, для яких відомі оптимальні розв’язки (відомий оптимальний результат для кожного вхідного набору даних). Кожен алгоритм був запущений 1000 разів та взятий відсоток правильного знаходження оптимальних результатів. Вхідні дані додаються у додатках.

Приведемо графіки точності для генетичного алгоритму та алгоритму мурашиних колоній та їх порівняльний графік.

Рисунок 4. Точність генетичного алгоритму в залежності від розміру вхідних даних.

Рисунок 5. Точність алгоритму мурашиних колоній в залежності від розміру вхідних даних.

Рисунок 6. Порівняльна характеристика точності алгоритмів в залежності від розміру вхідних даних

Де генетичний алгоритм представлений помаранчевим, а мурашиний – синім.

Як видно з дослідження, хоча генетичний алгоритм працює значно швидше, він має надзвичайно низьку точність на вхідних даних розмірності більше 5, в той час, як алгоритм мурашиних колоній має досить непогану точність на усіх вхідних даних. Низька точність в окремих випадках може бути викликана специфічними вхідними даними.

Додатково розглянемо роботу алгоритму на великих наборах даних (n є [30, 35]), програма не здатна обробляти матриці розмірності більше 35 через специфіку GET-запиту. Розглянемо діаграми, що демонструють швидкості роботи алгоритмів на великих наборах даних та їх порівняння з малим розміром даних.

Рисунок 16. Демонстрація роботи мурашиного алгоритму на великих наборах даних.

Подібний аналіз для генетичного алгоритму не доцільним, так як уже при розмірності 20 алгоритм працює близько 15 секунд.

Як висновок можна сказати, що хоч генетичний алгоритм працює швидше, швидкість зростання часу роботи в залежності від вхідних даних значно нижча за мурашиний, він має надзвичайно низьку точність, яка викликана повністю випадковою побудовою розв’язку.