

# 1. Завдання на роботу

Завдання на роботу.

1. Написати програму реалізації алгоритму MMAS для завдання знаходження найкоротшого шляху на графові.
2. Для графів з файлів shp55.aco, shp95.aco і shp155.aco експериментально підібрати значення параметрів  $\alpha$ ,  $\beta$  і  $\rho$ , що дозволяють знайти найкоротший шлях від вершини з номером 0 до вершини з номером 54, 94 і 154 для графів shp55.aco, shp95.aco і shp155.aco відповідно.

## 2. Про реалізацію

Мурашиний алгоритм було реалізовано мовою C#.

Посилання на гітхаб репозиторій: [Bohdan628318ylypchenko/AntColonyOptimization](https://github.com/Bohdan628318ylypchenko/AntColonyOptimization)

Реалізація складається з 3 частин:

1. ant-core – бібліотека, що безпосередньо реалізує мурашиний алгоритм.
2. ant-test – юніт-тести бібліотеки ant-core
3. ant-demo – консольний застосунок, виконує мурашиний алгоритм на заданому графі із вказаними параметрами:

```
PS C:\Users\Bohdan\Programming\labs-5-1\ant2\ant-demo\bin\x64\Release\net8.0-windows> .\ant-demo.exe --help
ant-demo 1.0.0+6a31ab07d46573ded87bad9cf312f8de40e93a6a
Copyright (C) 2024 ant-demo

-r, --random-numbers-source-path    Required. Path to file to serve as random numbers source.
-p, --graph-path                    Required. Path to file containing graph description.
-f, --initial-ferment                Required. Initial ferment value.
-a, --cost-coefficient               Required. Probability cost coefficient.
-b, --ferment-coefficient             Required. Probability ferment coefficient.
-s, --vertex-index-start             Required. Index of start vertex.
-e, --vertex-index-end               Required. Index of end vertex.
-t, --ant-count                      Required. Ant count.
-i, --max-iteration-count             Required. Max iteration count.
-q, --ant-ferment-count              Required. Total ant ferment count. Used to calculate ferment delta.
-c, --ferment-expiration             Required. Ferment expiration coefficient. New ferment = c * old ferment + total delta.
--help                              Display this help screen.
--version                            Display version information.

PS C:\Users\Bohdan\Programming\labs-5-1\ant2\ant-demo\bin\x64\Release\net8.0-windows> |
```

Рис. 1 Параметри демонстраційного застосунку

Псевдокод реалізації:

1. place ants into start vertex
2. for (iteration count = 0; iteration count < max iteration count; iteration count++):
3.     run each ant though graph
4.     calculate ferment delta for each graph edge
5.     update ferments
6.     select best route from current run
7.     compare current best and global best

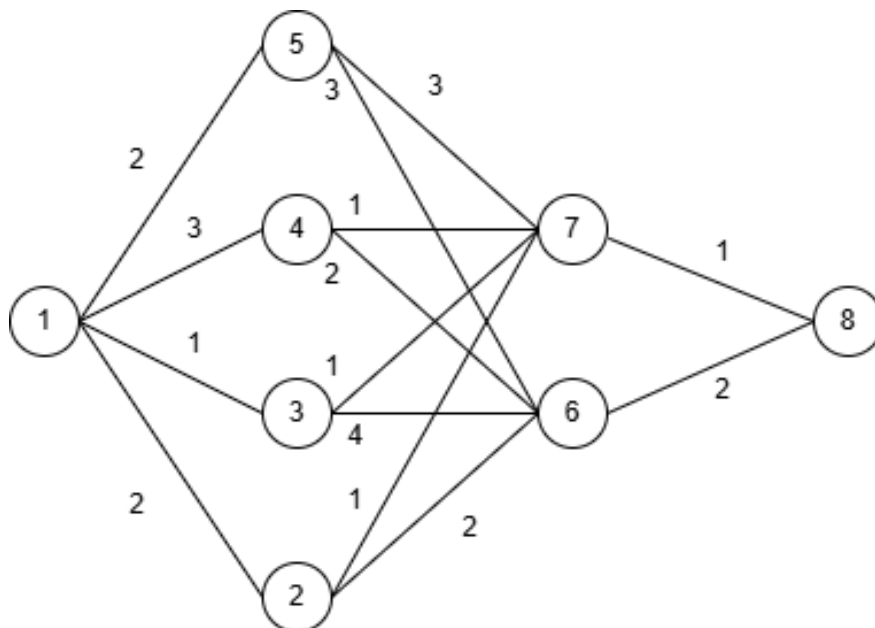
Кроки 1, 2, 4, 5, 6, 7 реалізовані класом AntColony. Крок 3 реалізується класом Ant. Граф, шлях мурахи реалізовані класами Graph, Path відповідно.

Варто відмітити, що під час завантаження графу із файлу всі значення вагів ребер, що рівні нулю, замінюються на велике число: кількість вершин графу \* найбільший ваговий коефіцієнт ребра \* 100. Таким чином, навіть якщо мураха зайде в «глухий кут» (тобто початковий граф є неповним), мураха зможе вийти з кута, пройшовши «неіснуючим» ребром. При цьому включення неіснуючих ребер до маршруту призводить до катастрофічного зростання вартості такого маршруту. Маршрут з неіснуючими ребрами завжди матиме більшу вартість у порівнянні з маршрутом, що включає лише справжні ребра.

### 3. Тестування

#### Власний приклад (8 вершин)

В якості найпростішого прикладу для тестування було створено граф:



Матриця вагових коефіцієнтів вершин (пропуски відповідають значенню 0, будуть замінені на  $7 * 8 * 100 = 5600$ ):

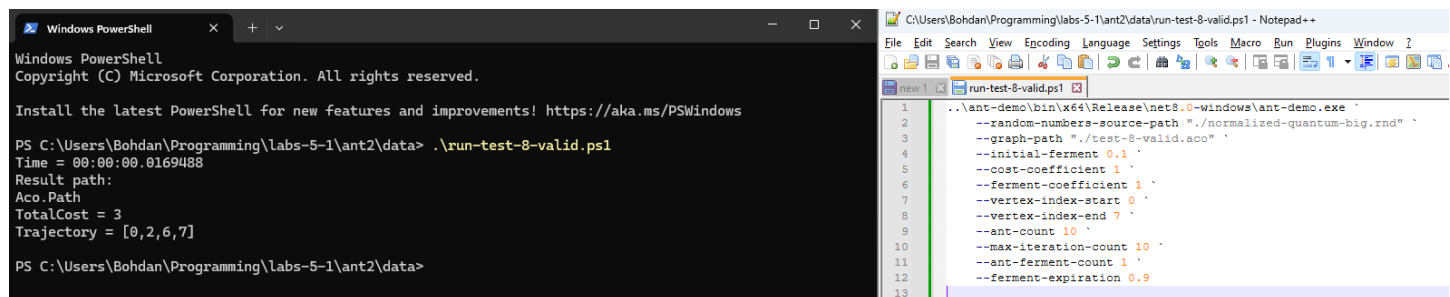
	1	2	3	4	5	6	7	8
1		2	1	3	6			
2	2					5	7	
3	1					4	1	
4	3					2	3	
5	6					3	3	
6		5	4	2	3			2
7		7	1	3	3			1
8						2	1	

Початкове положення – вершина 1. Кінцеве положення – вершина 8. Найкращий шлях із вершини 1 у вершину 8: [1, 3, 7, 8], ціна шляху дорівнює 3.

Оберемо наступні значення параметрів алгоритму:

```
initial-ferment 0.1
cost-coefficient 1
ferment-coefficient 1
vertex-index-start 0
vertex-index-end 7
ant-count 10
max-iteration-count 10
ant-ferment-count 1
ferment-expiration 0.9
```

Результат виконання алгоритму:



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

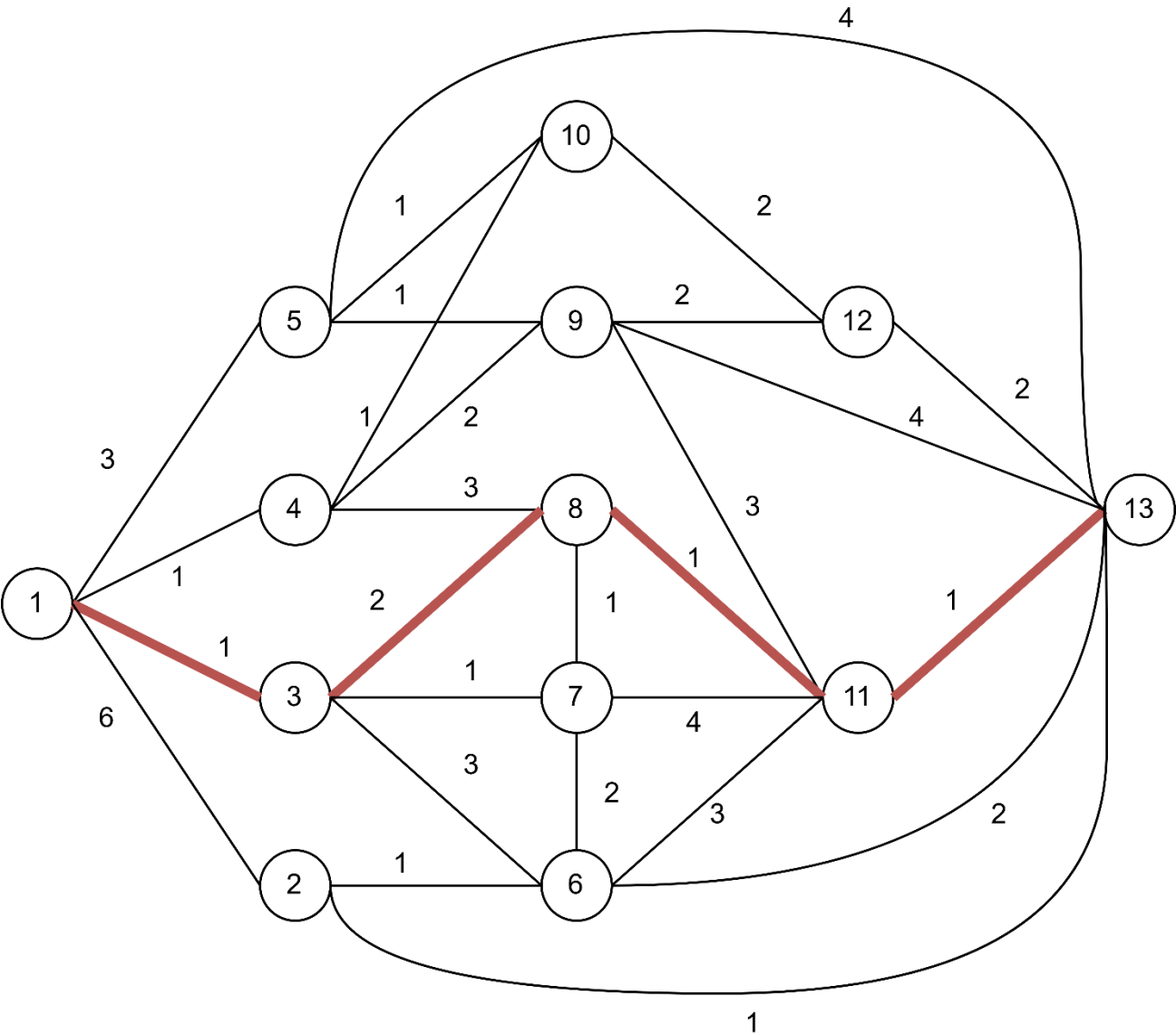
PS C:\Users\Bohdan\Programming\labs-5-1\ant2\data> .\run-test-8-valid.ps1
Time = 00:00:00.0169488
Result path:
Aco.Path
TotalCost = 3
Trajectory = [0,2,6,7]

PS C:\Users\Bohdan\Programming\labs-5-1\ant2\data>

C:\Users\Bohdan\Programming\labs-5-1\ant2\data\run-test-8-valid.ps1 - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
new 1 run-test-8-valid.ps1
1 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe `
2 --random-numbers-source-path ".\normalized-quantum-big.rnd" `
3 --graph-path ".\test-8-valid.aco" `
4 --initial-ferment 0.1 `
5 --cost-coefficient 1 `
6 --ferment-coefficient 1 `
7 --vertex-index-start 0 `
8 --vertex-index-end 7 `
9 --ant-count 10 `
10 --max-iteration-count 10 `
11 --ant-ferment-count 1 `
12 --ferment-expiration 0.9
13
```

Реалізація знайшла найдешевший шлях із вершини 1 до вершини 8 (програма нумерує вершини з нуля).

Власний приклад (13 вершин, 1)



	1	2	3	4	5	6	7	8	9	10	11	12	13
1		6	1	1	3								
2	6					1							1
3	1					3	1	2					
4	1							3	2	1			
5	3								1	1			4
6		1	3				2				3		2
7			1			2		1			4		
8			2	3			1				1		
9				2	1						3	2	4
10				1	1							2	
11						3	4	1	3				1
12									2	2			2
13		1			4	2			4		1	2	

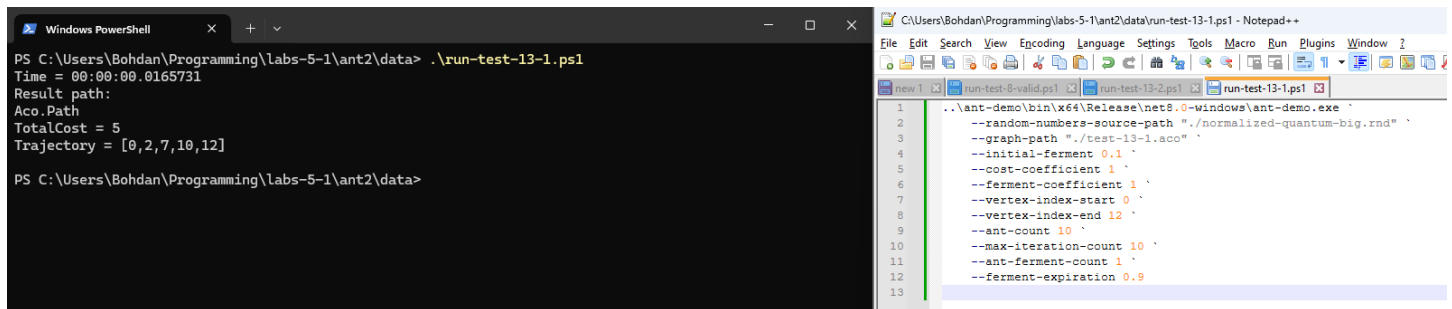
Початкове положення – вершина 1. Кінцеве положення – вершина 13.

Найкращий шлях із вершини 1 у вершину 13: [1, 3, 8, 11, 13]. Ціна шляху дорівнює 5.

Оберемо наступні значення параметрів алгоритму:

```
initial-ferment 0.1
cost-coefficient 1
ferment-coefficient 1
vertex-index-start 0
vertex-index-end 12
ant-count 10
max-iteration-count 10
ant-ferment-count 1
ferment-expiration 0.9
```

Результат виконання алгоритму:



The screenshot shows two windows. On the left is a Windows PowerShell terminal window with the following output:

```
PS C:\Users\Bohdan\Programming\labs-5-1\ant2\data> .\run-test-13-1.ps1
Time = 00:00:00.0165731
Result path:
Aco.Path
TotalCost = 5
Trajectory = [0,2,7,10,12]
PS C:\Users\Bohdan\Programming\labs-5-1\ant2\data>
```

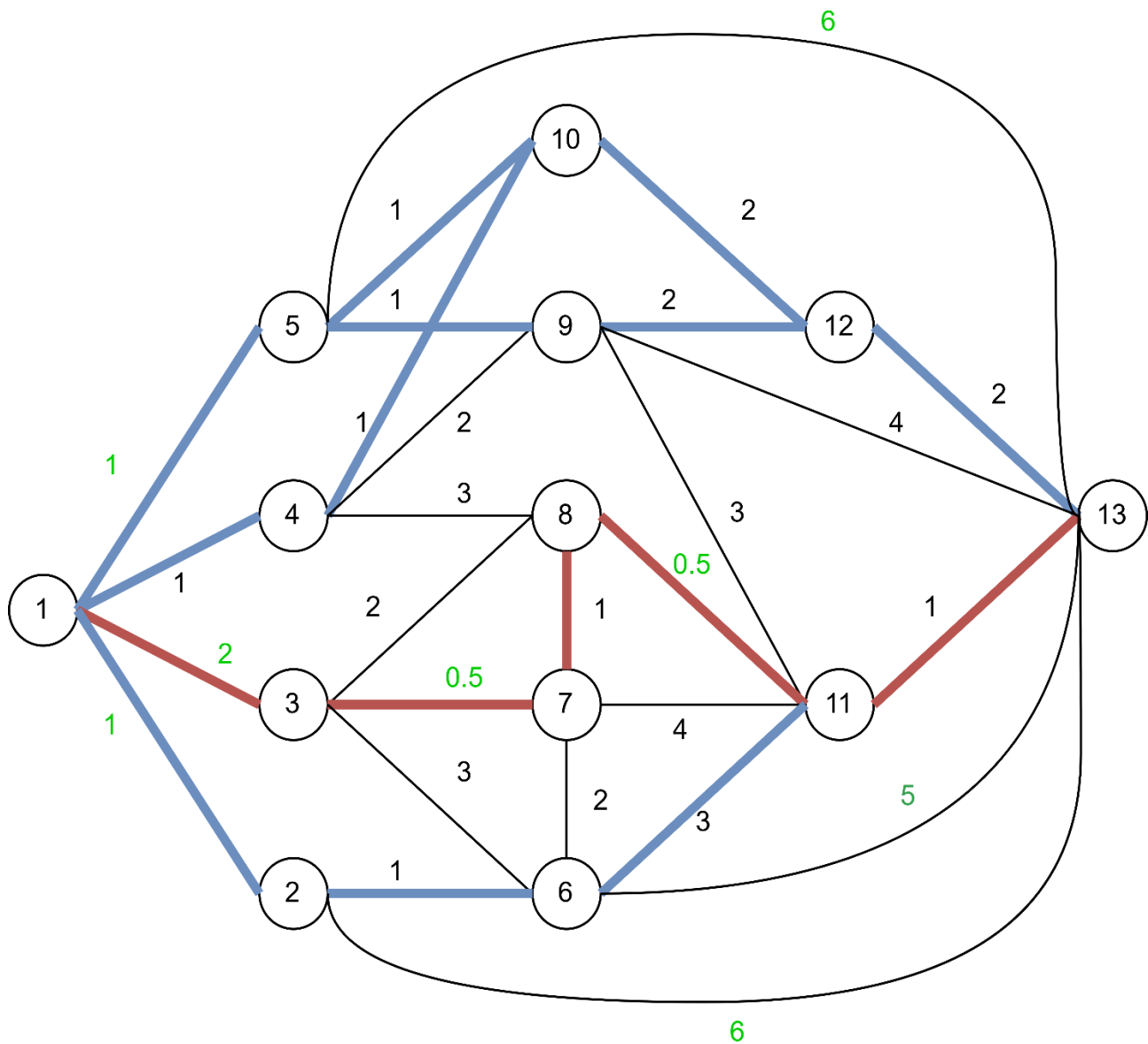
On the right is a Notepad++ window showing the content of the run-test-13-1.ps1 script:

```
1 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe `
2 --random-numbers-source-path ".\normalized-quantum-big.rnd" `
3 --graph-path ".\test-13-1.aco" `
4 --initial-ferment 0.1 `
5 --cost-coefficient 1 `
6 --ferment-coefficient 1 `
7 --vertex-index-start 0 `
8 --vertex-index-end 12 `
9 --ant-count 10 `
10 --max-iteration-count 10 `
11 --ant-ferment-count 1 `
12 --ferment-expiration 0.9
13
```

Реалізація знайшла найкращий шлях.

## Власний приклад (13 вершин, 2)

Змінимо попередній граф, «замаскувавши» найдешевший шлях. Тепер граф має багато шляхів, ціна яких близька до 5 (6 або 5.5). Відволікаючі розв’язки позначено синім кольором, змінені вагові коефіцієнти – зеленим. Тепер найкращим шляхом є [1, 3, 7, 8, 11, 13].

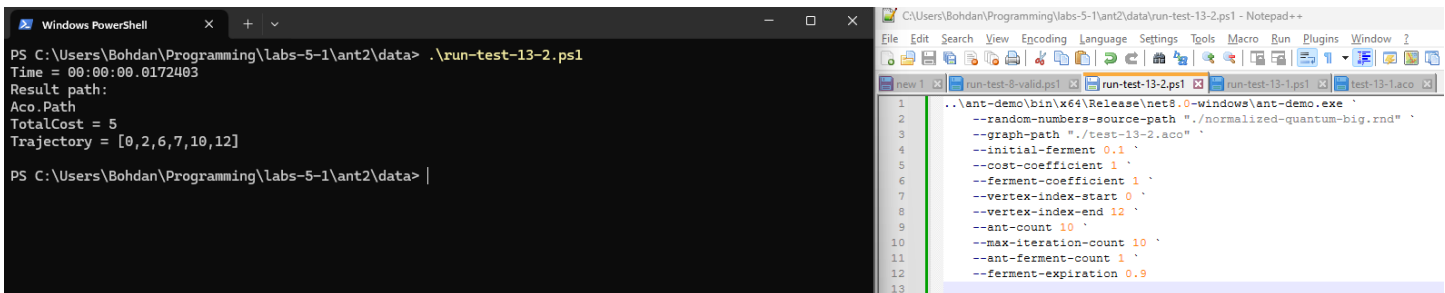


	1	2	3	4	5	6	7	8	9	10	11	12	13
1		1	2	1	3								
2	1					1							6
3	2					3	0.5	2					
4	1							3	2	1			
5	3								1	1			4
6		1	3				2				3		5
7			0.5			2		1			4		
8			2	3			1				0.5		
9				2	1						3	2	4
10				1	1							2	
11						3	4	0.5	3				1
12									2	2			2
13		6			4	5			4		1	2	

Параметри алгоритму візьмемо ті самі, що і у попередньому прикладі:

```
initial-ferment 0.1
cost-coefficient 1
ferment-coefficient 1
vertex-index-start 0
vertex-index-end 12
ant-count 10
max-iteration-count 10
ant-ferment-count 1
ferment-expiration 0.9
```

Результат виконання алгоритму:



The screenshot shows two windows. On the left, a Windows PowerShell window displays the output of the command `.\run-test-13-2.ps1`. The output indicates the time taken (00:00:00.0172403), the result path, the total cost (5), and the trajectory `[0,2,6,7,10,12]`. On the right, a Notepad++ window shows the contents of the `run-test-13-2.ps1` script, which lists various parameters for the `ant-demo.exe` application, including `--initial-ferment 0.1`, `--cost-coefficient 1`, `--ferment-coefficient 1`, `--vertex-index-start 0`, `--vertex-index-end 12`, `--ant-count 10`, `--max-iteration-count 10`, `--ant-ferment-count 1`, and `--ferment-expiration 0.9`.

Реалізація знайшла найкращий шлях.

**SHP 55,95,155**

У попередніх прикладах мурашиний алгоритм не потребував ретельного підбору параметрів для знаходження найкращого шляху. Тому під час дослідження графів shp 55, 95, 155 хотілось би провести декілька запусків, змінюючи параметри таким чином, що буде наявне поступове покращення результату.

Набір параметрів для графу shp55:

# Greedy 1

```
..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "./normalized-quantum-big.rnd" -p "./SHP55.aco" -f 0.1 -a 1 -b 0 -s 0 -e 54 -t 10 -i 100 -q 100 -c 0.9
```

# Greedy 2

```
..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "./normalized-quantum-big.rnd" -p "./SHP55.aco" -f 0.1 -a 1 -b 0 -s 0 -e 54 -t 10 -i 200 -q 100 -c 0.9
```

# Greedy 3

```
..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "./normalized-quantum-big.rnd" -p "./SHP55.aco" -f 0.1 -a 1 -b 0 -s 0 -e 54 -t 10 -i 300 -q 100 -c 0.9
```

# Greedy 4

```
..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "./normalized-quantum-big.rnd" -p "./SHP55.aco" -f 0.1 -a 1 -b 0 -s 0 -e 54 -t 10 -i 400 -q 100 -c 0.9
```

```
# Greedy 5
```

```
..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "./normalized-quantum-big.rnd" -p "./SHP55.aco" -f 0.1 -a 1 -b 0 -s 0 -e 54 -t 50 -i 400 -q 100 -c 0.9
```

```
# Greedy 6
```

```
..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "./normalized-quantum-big.rnd" -p "./SHP55.aco" -f 0.1 -a 1 -b 0 -s 0 -e 54 -t 50 -i 500 -q 100 -c 0.9
```

```
# Greedy 7
```

```
..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "./normalized-quantum-big.rnd" -p "./SHP55.aco" -f 0.1 -a 1 -b 0 -s 0 -e 54 -t 50 -i 1000 -q 100 -c 0.9
```

```
# Ferments 1
```

```
..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "./normalized-quantum-big.rnd" -p "./SHP55.aco" -f 0.1 -a 1 -b 1 -s 0 -e 54 -t 50 -i 400 -q 100 -c 0.9
```

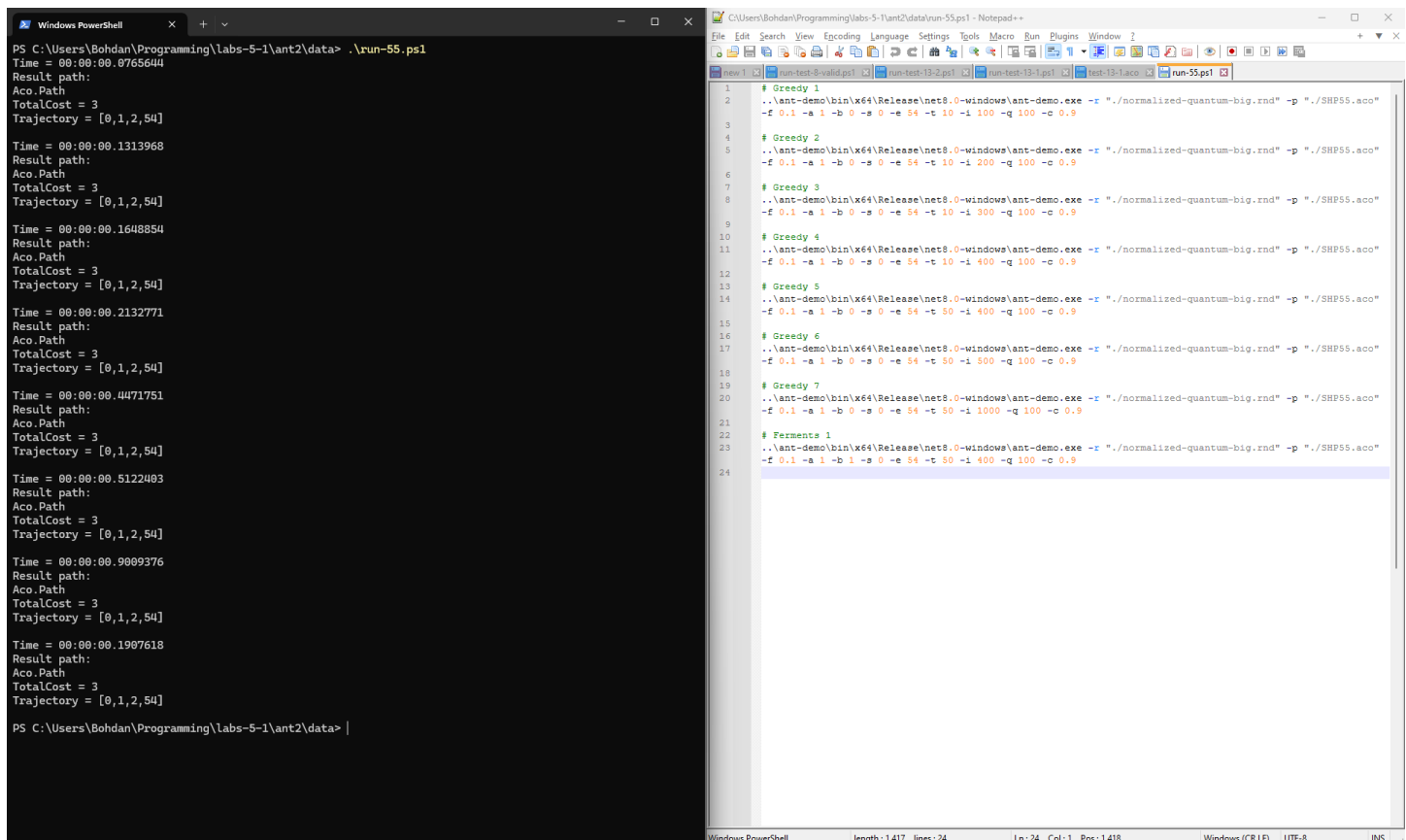
Перші 7 досліджень встановлюють параметр  $b$  (ferment-coefficient) рівним 0. Тобто для перших 7 досліджень вплив ферменту на поведінку алгоритму відсутній. Тоді поведінка алгоритму має деяку схожість із жадібним алгоритмом: імовірність вибору вершини  $A$  залежить виключно від ціни шляху між поточною вершиною та вершиною  $A$ .

Останнє дослідження встановлює коефіцієнт ціни та коефіцієнт ферменту рівними 1.

Дослідження також змінюють кількість ітерацій (від 100 до 1000), кількість мурах (10, 50).

Для графів  $shp$  95, 155 параметри варіювались за аналогічним принципом.

Результати досліджень графу  $shp55$ :



```
PS C:\Users\Bohdan\Programming\labs-5-1\ant2\data> .run-55.ps1
Time = 00:00:00.0765644
Result path:
Aco.Path
TotalCost = 3
Trajectory = [0,1,2,54]

Time = 00:00:00.1313968
Result path:
Aco.Path
TotalCost = 3
Trajectory = [0,1,2,54]

Time = 00:00:00.1648854
Result path:
Aco.Path
TotalCost = 3
Trajectory = [0,1,2,54]

Time = 00:00:00.2132771
Result path:
Aco.Path
TotalCost = 3
Trajectory = [0,1,2,54]

Time = 00:00:00.4471751
Result path:
Aco.Path
TotalCost = 3
Trajectory = [0,1,2,54]

Time = 00:00:00.5122403
Result path:
Aco.Path
TotalCost = 3
Trajectory = [0,1,2,54]

Time = 00:00:00.9009376
Result path:
Aco.Path
TotalCost = 3
Trajectory = [0,1,2,54]

Time = 00:00:00.1907618
Result path:
Aco.Path
TotalCost = 3
Trajectory = [0,1,2,54]

PS C:\Users\Bohdan\Programming\labs-5-1\ant2\data>

1 # Greedy 1
2 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "./normalized-quantum-big.rnd" -p "./SHP55.aco"
3 -f 0.1 -a 1 -b 0 -s 0 -e 54 -t 10 -i 100 -q 100 -c 0.9
4
5 # Greedy 2
6 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "./normalized-quantum-big.rnd" -p "./SHP55.aco"
7 -f 0.1 -a 1 -b 0 -s 0 -e 54 -t 10 -i 200 -q 100 -c 0.9
8
9 # Greedy 3
10 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "./normalized-quantum-big.rnd" -p "./SHP55.aco"
11 -f 0.1 -a 1 -b 0 -s 0 -e 54 -t 10 -i 400 -q 100 -c 0.9
12
13 # Greedy 4
14 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "./normalized-quantum-big.rnd" -p "./SHP55.aco"
15 -f 0.1 -a 1 -b 0 -s 0 -e 54 -t 50 -i 400 -q 100 -c 0.9
16
17 # Greedy 5
18 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "./normalized-quantum-big.rnd" -p "./SHP55.aco"
19 -f 0.1 -a 1 -b 0 -s 0 -e 54 -t 50 -i 500 -q 100 -c 0.9
20
21 # Greedy 6
22 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "./normalized-quantum-big.rnd" -p "./SHP55.aco"
23 -f 0.1 -a 1 -b 0 -s 0 -e 54 -t 50 -i 1000 -q 100 -c 0.9
24
25 # Ferments 1
26 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "./normalized-quantum-big.rnd" -p "./SHP55.aco"
27 -f 0.1 -a 1 -b 1 -s 0 -e 54 -t 50 -i 400 -q 100 -c 0.9
```



## Результати досліджень графу shp95:

```
Windows PowerShell
PS C:\Users\Bohdan\Programming\labs-5-1\ant2\data> .\run-95.ps1
Time = 00:00:00.1337829
Result path:
Aco.Path
TotalCost = 12
Trajectory = [0,1,3,5,94]

Time = 00:00:00.2098035
Result path:
Aco.Path
TotalCost = 12
Trajectory = [0,1,3,5,94]

Time = 00:00:00.2605828
Result path:
Aco.Path
TotalCost = 12
Trajectory = [0,1,3,5,94]

Time = 00:00:00.3368834
Result path:
Aco.Path
TotalCost = 12
Trajectory = [0,1,3,5,94]

Time = 00:00:00.8904207
Result path:
Aco.Path
TotalCost = 12
Trajectory = [0,1,3,5,94]

Time = 00:00:01.0520661
Result path:
Aco.Path
TotalCost = 12
Trajectory = [0,1,3,5,94]

Time = 00:00:01.8562005
Result path:
Aco.Path
TotalCost = 12
Trajectory = [0,1,3,5,94]

Time = 00:00:00.3374172
Result path:
Aco.Path
TotalCost = 12
Trajectory = [0,1,3,5,94]

PS C:\Users\Bohdan\Programming\labs-5-1\ant2\data> |

C:\Users\Bohdan\Programming\labs-5-1\ant2\data\run-95.ps1 - Notepad++
1 # Greedy 1
2 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP95.aco"
3 -f 0.1 -a 1 -b 0 -s 0 -e 94 -t 10 -i 100 -q 100 -c 0.9
4
5 # Greedy 2
6 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP95.aco"
7 -f 0.1 -a 1 -b 0 -s 0 -e 94 -t 10 -i 200 -q 100 -c 0.9
8
9 # Greedy 3
10 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP95.aco"
11 -f 0.1 -a 1 -b 0 -s 0 -e 94 -t 10 -i 300 -q 100 -c 0.9
12
13 # Greedy 4
14 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP95.aco"
15 -f 0.1 -a 1 -b 0 -s 0 -e 94 -t 10 -i 400 -q 100 -c 0.9
16
17 # Greedy 5
18 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP95.aco"
19 -f 0.1 -a 1 -b 0 -s 0 -e 94 -t 10 -i 500 -q 100 -c 0.9
20
21 # Greedy 6
22 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP95.aco"
23 -f 0.1 -a 1 -b 0 -s 0 -e 94 -t 10 -i 1000 -q 100 -c 0.9
24
25 # Greedy 7
26 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP95.aco"
27 -f 0.1 -a 1 -b 0 -s 0 -e 94 -t 10 -i 1000 -q 100 -c 0.9
28
29 # Ferments 1
30 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP95.aco"
31 -f 0.1 -a 1 -b 1 -s 0 -e 94 -t 10 -i 400 -q 100 -c 0.9
```

## Результати досліджень графу shp155:

```
Windows PowerShell
PS C:\Users\Bohdan\Programming\labs-5-1\ant2\data> .\run-155.ps1
Time = 00:00:00.2254884
Result path:
Aco.Path
TotalCost = 5
Trajectory = [0,1,3,5,7,154]

Time = 00:00:00.3481404
Result path:
Aco.Path
TotalCost = 5
Trajectory = [0,1,3,5,7,154]

Time = 00:00:00.4653022
Result path:
Aco.Path
TotalCost = 5
Trajectory = [0,1,3,5,7,154]

Time = 00:00:00.5678192
Result path:
Aco.Path
TotalCost = 5
Trajectory = [0,1,3,5,7,154]

Time = 00:00:01.8752247
Result path:
Aco.Path
TotalCost = 5
Trajectory = [0,1,3,5,7,154]

Time = 00:00:02.3676429
Result path:
Aco.Path
TotalCost = 5
Trajectory = [0,1,3,5,7,154]

Time = 00:00:04.3488183
Result path:
Aco.Path
TotalCost = 5
Trajectory = [0,1,3,5,7,154]

Time = 00:00:00.4598178
Result path:
Aco.Path
TotalCost = 5
Trajectory = [0,1,3,5,7,154]

PS C:\Users\Bohdan\Programming\labs-5-1\ant2\data> |

C:\Users\Bohdan\Programming\labs-5-1\ant2\data\run-155.ps1 - Notepad++
1 # Greedy 1
2 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP155.aco"
3 -f 0.1 -a 1 -b 0 -s 0 -e 154 -t 10 -i 100 -q 100 -c 0.9
4
5 # Greedy 2
6 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP155.aco"
7 -f 0.1 -a 1 -b 0 -s 0 -e 154 -t 10 -i 200 -q 100 -c 0.9
8
9 # Greedy 3
10 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP155.aco"
11 -f 0.1 -a 1 -b 0 -s 0 -e 154 -t 10 -i 300 -q 100 -c 0.9
12
13 # Greedy 4
14 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP155.aco"
15 -f 0.1 -a 1 -b 0 -s 0 -e 154 -t 10 -i 400 -q 100 -c 0.9
16
17 # Greedy 5
18 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP155.aco"
19 -f 0.1 -a 1 -b 0 -s 0 -e 154 -t 10 -i 500 -q 100 -c 0.9
20
21 # Greedy 6
22 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP155.aco"
23 -f 0.1 -a 1 -b 0 -s 0 -e 154 -t 10 -i 1000 -q 100 -c 0.9
24
25 # Greedy 7
26 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP155.aco"
27 -f 0.1 -a 1 -b 0 -s 0 -e 154 -t 10 -i 1000 -q 100 -c 0.9
28
29 # Ferments 1
30 ..\ant-demo\bin\x64\Release\net8.0-windows\ant-demo.exe -r "/normalized-quantum-big.rnd" -p "/SHP155.aco"
31 -f 0.1 -a 1 -b 1 -s 0 -e 154 -t 10 -i 400 -q 100 -c 0.9
```

Для кожного графу всі дослідження знаходять однаковий шлях, вплив ферментів не має значення:

Граф shp55: [0, 1, 2, 54], ціна = 3

Граф shp95: [0, 1, 3, 5, 94], ціна = 12

Граф shp155: [0, 1, 3, 5, 7, 154], ціна = 5

Імовірно, ці шляхи є найкращими.

Здатність алгоритму знаходити найкращий шлях для графів shp55,95,155 без впливу ферментів може бути пов’язана з великим розкидом вагів графу:

```
1 c FILE: yuzSHP55.aoo
2 c
3 c SOURCE: Yu.Zorin (yuzorin@gmail.com)
4 c
5 c DESCRIPTION: Random graph instance generated for semester project
6 c for AI course
7 c first line after comment (p) gives problem's size
8 c
9 p 55
10 i 0 1 0 210 188 0 144 0 163 0 0 0 0 46 113 271 266 0 261 0 234 42 256 0 0 124 148 169 148 232 121 197 164 207 262 126 215 0 265 26 29 0 156 189 205 142 241 229 146 142 218 41 243 62 262
11 i 1 0 1 89 0 45 114 61 3 11 33 36 0 107 35 110 30 0 0 0 177 29 66 28 0 263 225 201 0 29 64 0 90 38 17 105 273 170 174 52 187 257 170 169 27 3 208 34 69 41 67 254 0 265
12 i 0 1 0 212 239 56 0 107 49 19 0 42 267 257 0 243 0 0 0 177 240 79 69 206 179 170 0 175 233 167 260 0 0 38 259 123 0 0 0 268 201 8 0 0 163 23 0 47 0 0 136 62 143 30 107 1
13 i 210 89 212 0 136 154 0 200 73 0 98 0 116 49 0 0 0 246 183 0 65 0 162 155 0 22 199 95 171 49 264 270 40 0 107 10 95 0 212 227 266 0 40 41 11 2 212 147 0 87 18 18 170 59 266
14 i 188 0 239 136 0 0 130 70 126 193 209 222 199 157 35 0 58 240 131 190 0 267 0 55 228 217 0 231 235 194 42 56 91 69 115 218 52 111 0 53 47 107 200 235 161 86 93 24 145 0 168 267 224 21 13
15 i 0 45 56 154 0 0 53 217 101 207 0 0 146 0 48 58 213 4 0 121 136 52 63 252 0 201 0 136 0 253 54 0 201 142 0 46 157 135 254 256 251 0 0 151 188 0 0 0 256 22 48 40 140 14
16 i 144 114 0 0 130 53 0 14 22 171 184 0 275 0 193 75 129 0 274 0 93 149 3 0 0 66 135 79 37 0 0 189 244 68 103 256 178 0 244 272 204 33 259 181 275 220 0 179 63 188 77 93 120 169 69
17 i 0 61 107 200 70 217 14 0 0 0 63 83 57 197 31 107 242 137 197 29 123 108 0 7 0 98 175 28 93 76 54 0 188 0 36 0 90 247 74 228 196 0 26 0 41 218 159 0 0 39 10 0 0 82 147
18 i 163 3 49 73 126 101 22 0 0 83 0 113 0 0 0 114 0 217 0 93 194 96 122 125 0 0 42 103 253 0 0 20 188 207 0 77 260 0 129 180 175 108 179 27 0 54 0 9 127 259 0 188 120 32 123 254
19 i 0 11 19 0 193 207 171 0 83 0 146 0 163 133 251 252 0 0 260 260 195 217 187 231 4 226 57 117 181 162 124 0 0 80 174 0 97 120 9 253 0 67 104 0 14 175 103 0 143 14 61 2 102 189 0
20 i 0 33 0 98 209 0 184 63 0 146 0 0 117 1 257 175 130 165 117 265 200 149 56 171 267 0 87 131 88 209 0 31 0 164 0 0 230 171 0 0 242 142 39 224 0 61 0 249 0 0 168 126 162 132 0
21 i 0 36 42 0 222 0 0 83 113 0 0 0 255 0 0 219 0 0 0 61 147 30 114 21 0 0 168 37 77 3 138 117 0 0 223 145 183 97 69 212 89 203 210 249 163 0 179 255 69 179 244 168 53 0
22 i 0 0 267 116 199 146 275 57 0 163 117 255 0 0 176 30 239 133 165 0 193 185 252 27 189 23 0 115 152 0 266 207 83 188 51 0 103 194 272 144 137 0 222 0 20 85 44 186 250 75 0 60 186 0 57
23 i 46 107 257 49 157 0 0 197 0 113 1 0 0 0 45 100 143 122 0 14 168 85 107 0 63 0 0 247 65 81 165 114 251 4 154 0 194 0 27 127 0 161 88 202 0 112 208 19 78 25 44 80 194 0 215
24 i 113 35 0 0 35 48 193 31 114 251 257 0 176 45 0 45 4 186 247 9 105 47 275 0 143 261 230 165 40 70 0 239 0 244 130 18 239 56 120 128 236 0 0 0 126 97 0 193 112 32 19 0 0 73 126
25 i 271 110 243 0 0 48 75 107 0 252 175 219 30 100 45 0 164 0 0 204 0 234 202 0 50 89 0 0 186 15 141 0 56 171 0 162 85 102 0 0 114 249 0 0 0 31 225 214 225 0 243 0 264 266 250
26 i 266 30 0 0 58 58 129 242 217 0 130 0 239 143 4 164 0 15 179 0 262 92 121 47 81 220 118 224 71 77 258 6 40 0 34 0 53 195 237 26 147 203 0 256 24 261 49 83 249 0 186 111 182 111 152
27 i 0 0 90 246 240 213 0 137 0 0 165 0 133 122 186 0 15 0 81 81 0 3 0 50 226 6 176 251 27 263 105 216 49 230 266 147 265 179 0 0 0 179 58 0 0 85 199 0 98 65 167 226 207 202 231
28 i 261 0 177 183 131 4 274 197 93 260 117 0 165 0 247 0 179 81 0 9 113 125 251 0 229 110 224 0 196 0 0 0 77 16 177 174 0 98 0 251 29 0 253 0 125 0 0 128 214 234 252 0 258 22 0
29 i 0 0 240 0 190 0 0 29 194 260 265 0 0 14 9 204 0 81 9 0 242 120 0 178 20 67 75 196 59 186 35 6 272 160 0 86 200 253 207 40 111 146 91 0 0 0 231 181 157 0 0 54 31 130 192
30 i 234 0 79 65 0 121 93 123 96 195 200 61 193 168 105 0 262 0 113 242 0 103 34 36 30 250 109 214 127 5 184 24 0 155 21 0 237 248 70 0 197 85 138 47 132 96 266 91 220 0 227 163 222 0 155
31 i 42 177 69 0 267 136 149 108 122 217 149 147 185 85 47 234 92 3 125 120 103 0 0 195 149 0 47 106 262 0 141 100 263 0 136 0 0 0 15 75 0 77 228 201 172 157 113 162 164 183 114 148 112 1 134
32 i 256 29 206 162 0 52 3 0 125 187 56 30 252 107 275 202 121 0 251 0 34 0 0 229 271 214 0 0 174 25 3 0 216 245 3 0 243 0 34 133 257 246 0 202 0 222 0 56 0 274 213 263 0 211
33 i 0 66 179 155 55 63 0 7 0 231 171 114 27 0 0 0 47 50 0 178 36 195 0 0 0 22 85 37 268 172 0 244 0 137 15 0 3 33 3 66 241 150 79 70 140 94 180 81 0 265 0 207 247 0
34 i 0 28 170 0 228 252 0 0 0 4 267 21 189 63 143 50 81 226 229 20 30 149 229 0 0 0 202 99 173 222 0 240 0 42 120 0 158 173 187 73 127 269 122 33 0 248 235 122 261 37 57 41 0 189 0
35 i 124 0 0 22 217 0 66 98 42 226 0 0 23 0 261 89 220 6 110 67 250 0 271 0 0 0 57 128 0 0 113 120 275 176 36 186 119 0 115 146 190 254 140 256 0 177 78 0 83 0 94 27 93 164 0
36 i 148 263 175 199 0 201 135 175 103 57 87 0 0 0 230 0 118 176 224 75 109 47 214 22 202 57 0 120 82 0 55 213 37 212 0 0 0 250 159 7 0 104 44 236 59 168 151 251 29 144 246 0 0 77 0
37 i 169 225 233 95 231 0 79 28 253 117 131 168 115 247 165 0 224 251 0 196 214 106 0 85 99 128 120 0 0 0 91 229 21 100 0 0 0 74 57 113 32 8 39 142 0 96 145 219 165 222 0 14 104 71
38 i 148 201 167 171 235 136 37 93 0 181 88 37 152 65 40 186 71 27 196 59 127 262 0 37 173 0 82 0 0 73 0 218 85 160 106 0 91 5 267 248 24 37 158 267 246 0 49 219 0 59 173 98 100 116
39 i 232 0 260 49 194 0 0 76 0 162 209 77 0 81 87 10 15 77 263 0 186 5 0 174 268 222 0 0 0 73 0 163 273 0 49 185 200 186 200 111 248 160 193 0 184 0 0 199 235 154 177 180 170 272 246 21
40 i 121 29 29 42 253 0 54 20 124 0 3 266 165 0 141 258 105 0 35 184 141 25 172 0 113 55 91 0 163 0 261 157 0 65 0 79 0 28 13 164 91 40 158 107 163 145 126 0 205 0 169 111 253 0
41 i 197 64 0 270 56 54 189 0 188 0 31 138 207 114 239 0 6 216 0 6 24 100 3 0 240 120 213 229 0 273 261 0 224 162 241 150 237 142 242 111 10 136 128 58 70 135 0 254 237 240 272 0 26 50 137
42 i 164 0 38 40 91 0 244 188 207 0 0 117 83 251 0 56 40 49 77 272 0 263 0 244 0 275 37 21 218 0 157 224 0 230 20 0 141 108 104 3 111 192 0 31 75 57 244 185 202 2 42 254 0 0 247
43 i 207 90 259 0 69 201 68 0 0 80 164 0 188 4 244 171 0 230 16 160 155 0 216 0 42 176 212 100 85 49 0 162 230 0 0 134 0 111 38 259 0 214 32 78 0 198 11 0 0 104 0 0 36 0 246
44 i 262 38 123 107 115 142 103 36 77 174 0 0 51 154 130 0 34 266 177 0 21 136 245 137 120 36 0 0 160 185 65 241 20 0 0 207 162 128 207 197 70 0 252 0 76 0 50 0 0 21 186 266 229 0 0
45 i 126 17 0 10 218 0 256 0 260 0 0 223 0 0 18 162 0 147 174 86 0 0 3 15 0 186 0 0 106 200 0 150 0 134 207 0 127 0 0 0 189 150 97 22 0 35 238 161 0 0 246 59 90 67
46 i 215 105 0 95 52 46 178 90 0 97 230 145 103 194 239 85 53 265 0 200 237 0 0 0 158 119 0 0 0 186 79 237 141 0 162 127 0 257 0 64 199 220 224 194 0 48 156 0 179 116 143 193 25 0 0
47 i 0 273 0 0 111 157 0 247 129 120 171 183 194 0 56 102 195 179 98 253 248 0 243 3 173 0 250 0 91 200 0 142 108 111 128 0 257 0 194 111 0 0 46 163 40 252 233 170 157 0 196 23 188 271
48 i 265 170 268 212 0 135 244 74 180 9 0 97 272 27 120 0 237 0 0 207 70 15 0 33 187 115 159 74 5 111 28 242 104 38 207 0 0 194 0 83 0 15 0 77 153 71 130 103 134 108 26 231 134 131 0
49 i 26 174 201 227 53 254 272 228 175 253 0 69 144 127 128 0 26 0 251 40 0 75 34 3 73 146 7 57 267 248 13 111 3 259 197 0 64 111 83 0 57 113 104 272 151 167 3 259 0 8 192 200 243 80 126
50 i 29 52 8 266 47 256 204 196 108 0 242 212 137 0 236 114 147 0 29 111 197 0 133 66 127 190 0 113 248 160 164 10 111 0 70 0 199 0 0 57 0 147 198 176 0 189 141 0 143 88 85 121 95 0 110
51 i 0 187 0 0 107 251 33 0 179 67 142 89 0 161 0 249 203 179 0 146 85 77 257 241 269 254 104 32 24 193 91 136 192 214 0 189 220 0 15 113 147 0 233 0 230 49 177 0 253 221 0 127 49 120 227
52 i 156 257 0 40 200 0 259 26 27 104 39 203 222 88 0 0 0 58 253 91 138 228 246 150 122 140 44 8 37 0 40 128 0 32 252 150 224 0 0 104 198 233 0 173 87 99 220 261 0 153 152 227 67 232 161
53 i 189 170 163 41 235 0 181 0 0 224 210 0 202 0 0 256 0 0 0 47 201 0 79 33 256 236 39 158 184 158 58 31 78 0 97 194 46 77 272 176 0 173 0 86 108 0 0 0 73 0 64 0 149 0
54 i 205 169 23 11 161 151 275 41 54 14 0 249 20 0 126 0 4 125 0 132 172 202 70 0 0 59 142 267 0 107 70 75 0 76 22 0 163 153 151 0 230 87 86 0 129 0 0 38 0 187 98 0 83 129
55 i 142 27 0 2 86 188 220 218 0 175 61 163 85 112 97 31 261 85 0 0 96 157 0 140 248 177 168 0 246 0 163 135 57 198 0 0 48 40 71 167 189 49 99 108 129 0 0 0 246 225 53 87 30 243 0
56 i 241 3 47 212 95 0 0 159 9 103 0 0 44 208 0 225 49 199 0 231 266 113 222 94 235 78 151 96 0 199 145 0 244 11 50 35 156 252 130 3 141 177 220 0 0 0 0 161 99 236 139 20 236 0 0
```

Приклад значень вагів графу shp55

Наявні малі значення (1, 4, 3, 46, 40, 35 тощо) і великі значення (210, 265, 189 тощо). Такий розмах значно полегшує пошук шляху, адже найкращий шлях додатково «підсвітується» ребрами з великою вагою.

Подальша оптимізація параметрів не має сенсу: алгоритм знаходить оптимальні шляхи за задовільний час (від 100/200 мілісекунд до 4 секунд для графу shp155 з 1000 ітераціями).