Vehicle Routing Problem

Emma Krompaščíková (xkromp00)

vedoucí: Ing. Jan Klhůfek

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Zadanie

- S využitím genetického programování navrhněte způsob řešení vámi zvolené varianty logistického problému capacitated vehicle routing problem (CVRP) – můžete uvažovat variantu **s jedním depem**.
- Pro účely projektu si získejte nebo vhodně **vygenerujte** logistickou **síť** ,která má charakteristiku neorientovaného ohodnoceného grafu, kde hodnota hrany udává vzdálenost mezi dvojicí uzlů a hodnota uzlu objednávku, kterou je třeba do místa doručit.
- Graf nejprve rozšiřte do podoby **úplného grafu** (tak aby mezi všemi dvojicemi uzlů existovala ohodnocená hrana) a následně jej vhodně reprezentujte pro řešení daného problému s využitím flotily **N vozidel**.
- Uvažujte, že ne všechny uzly reprezentují místa, kam/odkud je potřeba něco dovézt/přivést - tuto skutečnost značí nenulová váha přiřazená k jednotlivým uzlům.
- Všechna vozidla **začínají** cestu výjezdem **z depa** a do depa se také musejí vrátit.
- Uvažujte pouze případy, kdy je instanci problému možné vyřešit s danou velikostí flotily a počtem objednávek.
- Hledejte řešení, které zajistí doručení všech objednávek a přitom minimalizuje celkovou ujetou vzdálenost.

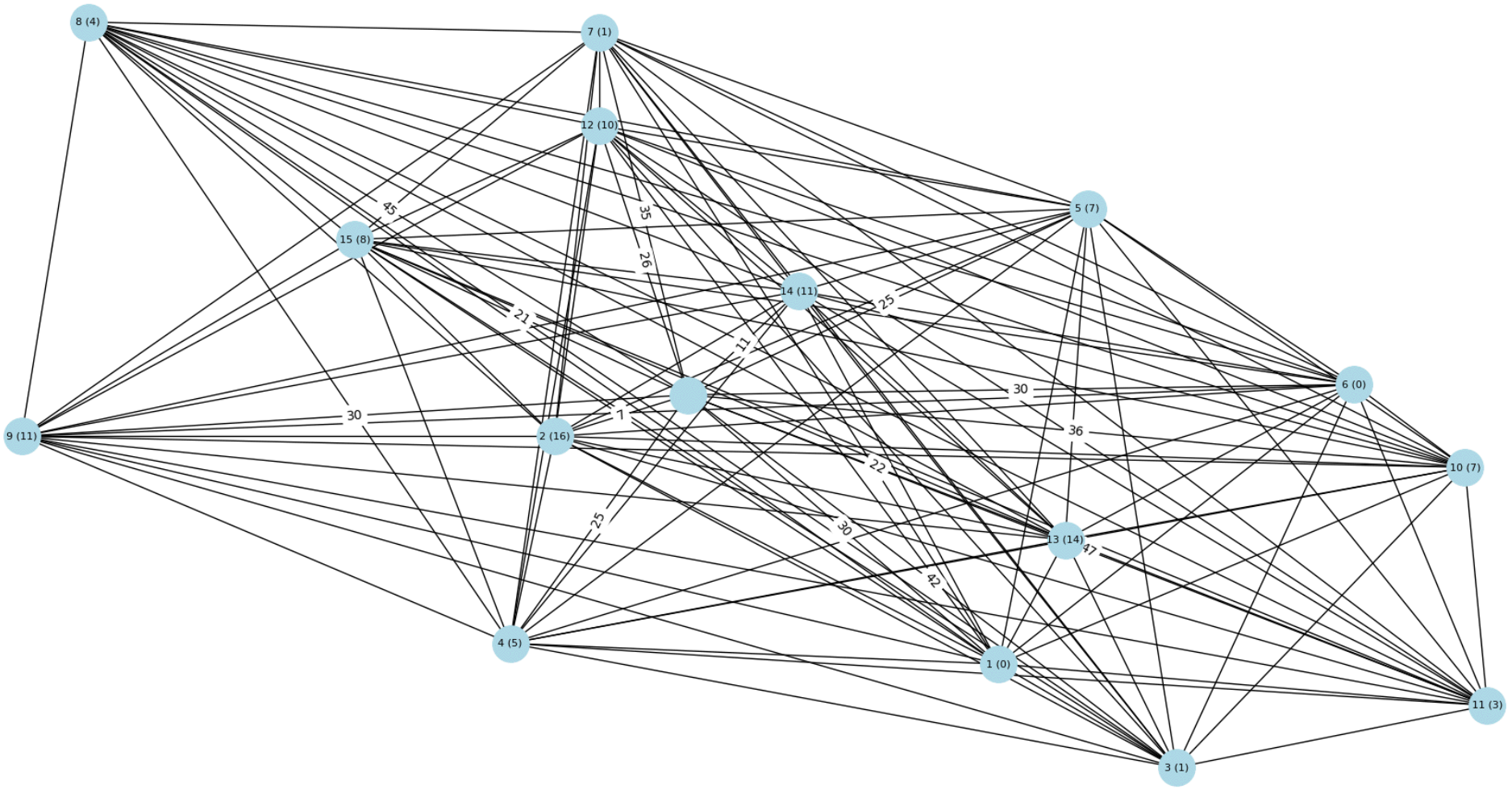
Graf a knižnice

- Python
  - **networkx**
  - **matplotlib.pyplot**
  - **numpy**

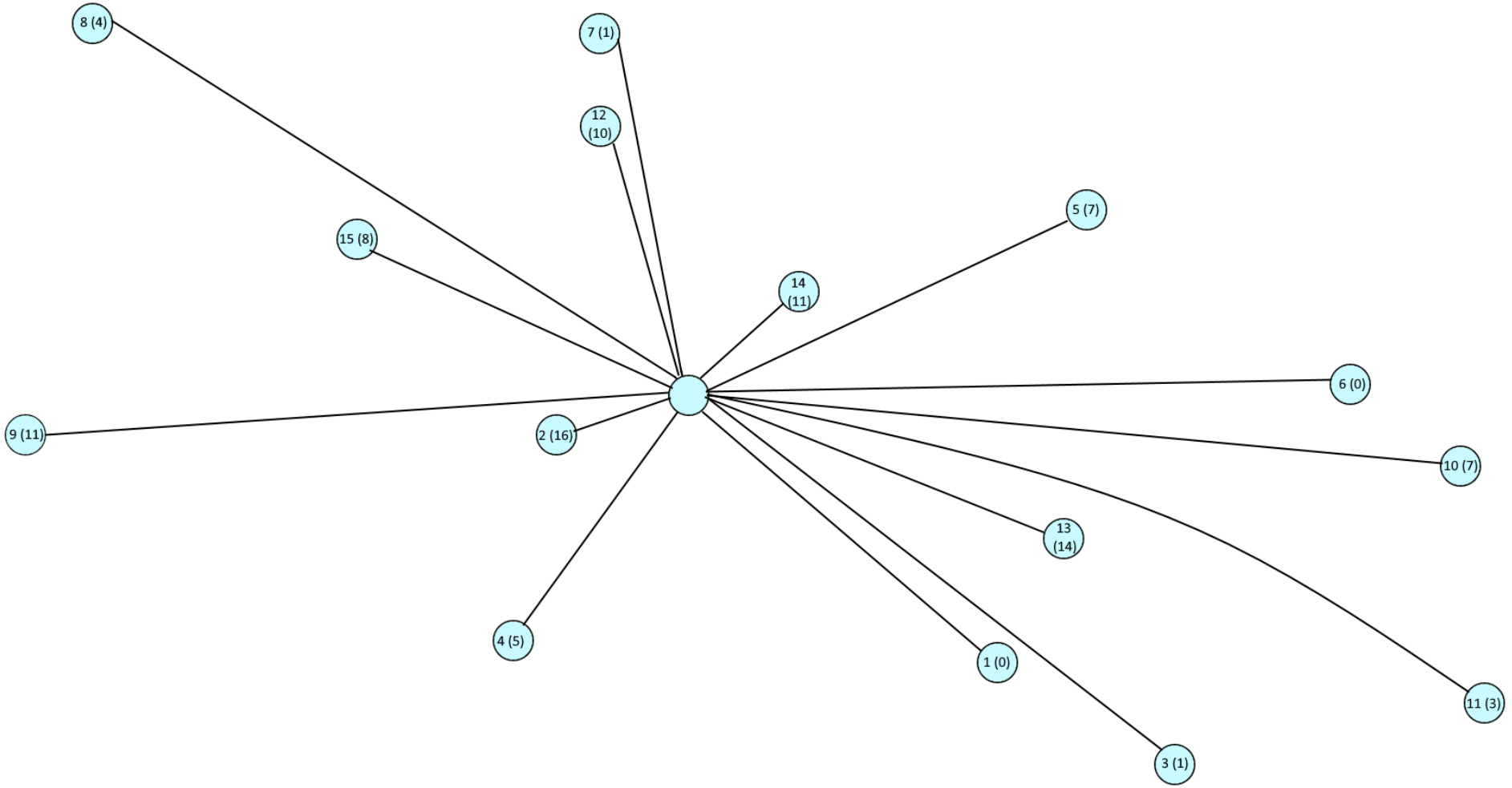
Prínos – GA, zakódovanie

- riešim CVRP
- používam mutovanie, kríženie nie
- 2 typy mutácii
  - zmena poradia uzlov v pláne trasy
    - [2, 3, 4, 6] -> [3, 2, 4, 6]
  - premiestnenie uzlu inému autu – kontrola kapacity!
    - [1, 3, 5, 7] , [2, 4, 6] -> [1, 3, 5] , [2, 4, 6, 7]
- jedinec reprezentuje jednu flotilu
- každé auto je jedným génom
- do novej generácie postupuje po 9tej generácii 25 najlepších jedincov a náhodná polovica zo zvyšku
- pre kapacity miest používam slovník

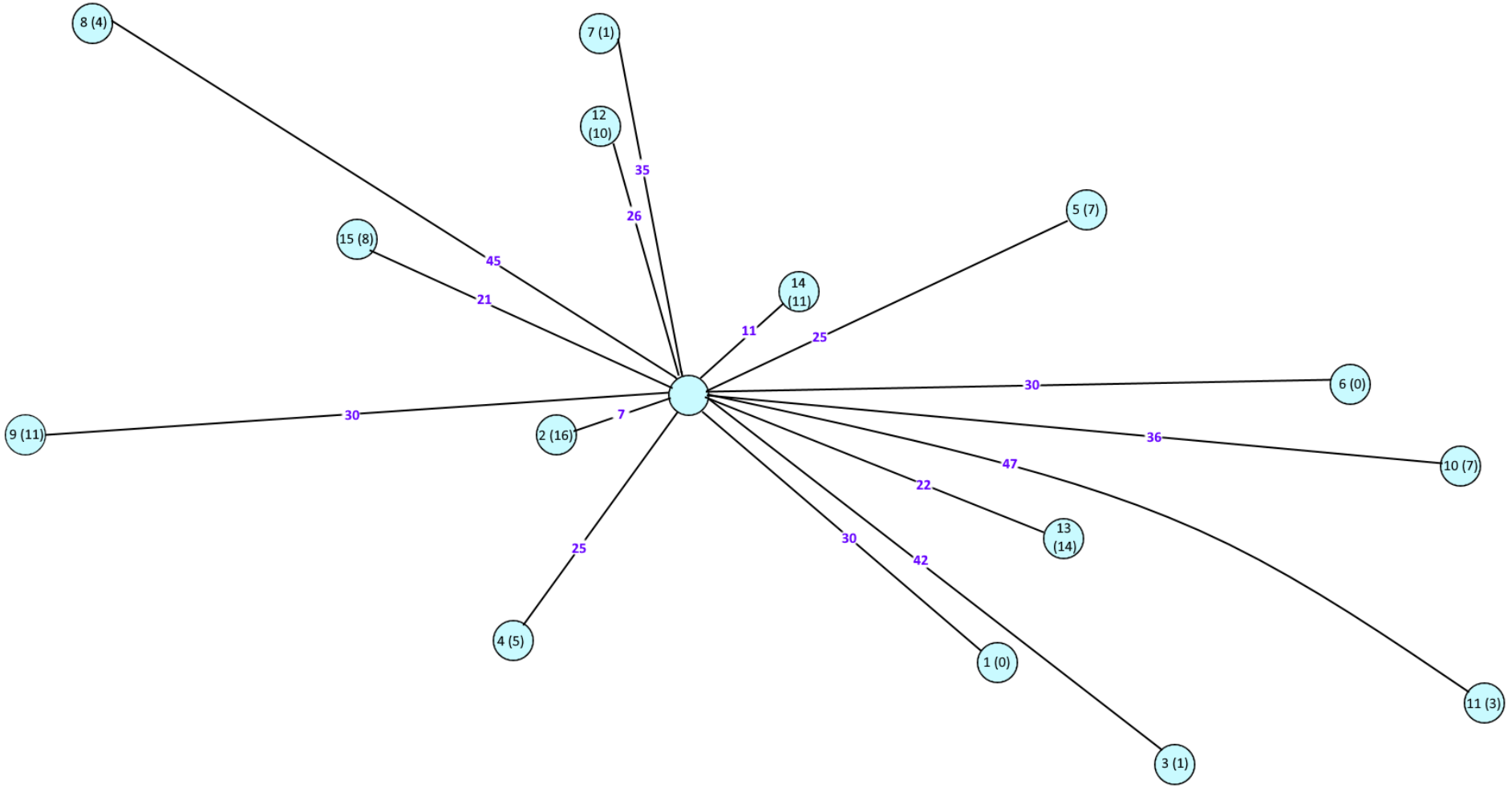
Ukážka grafu



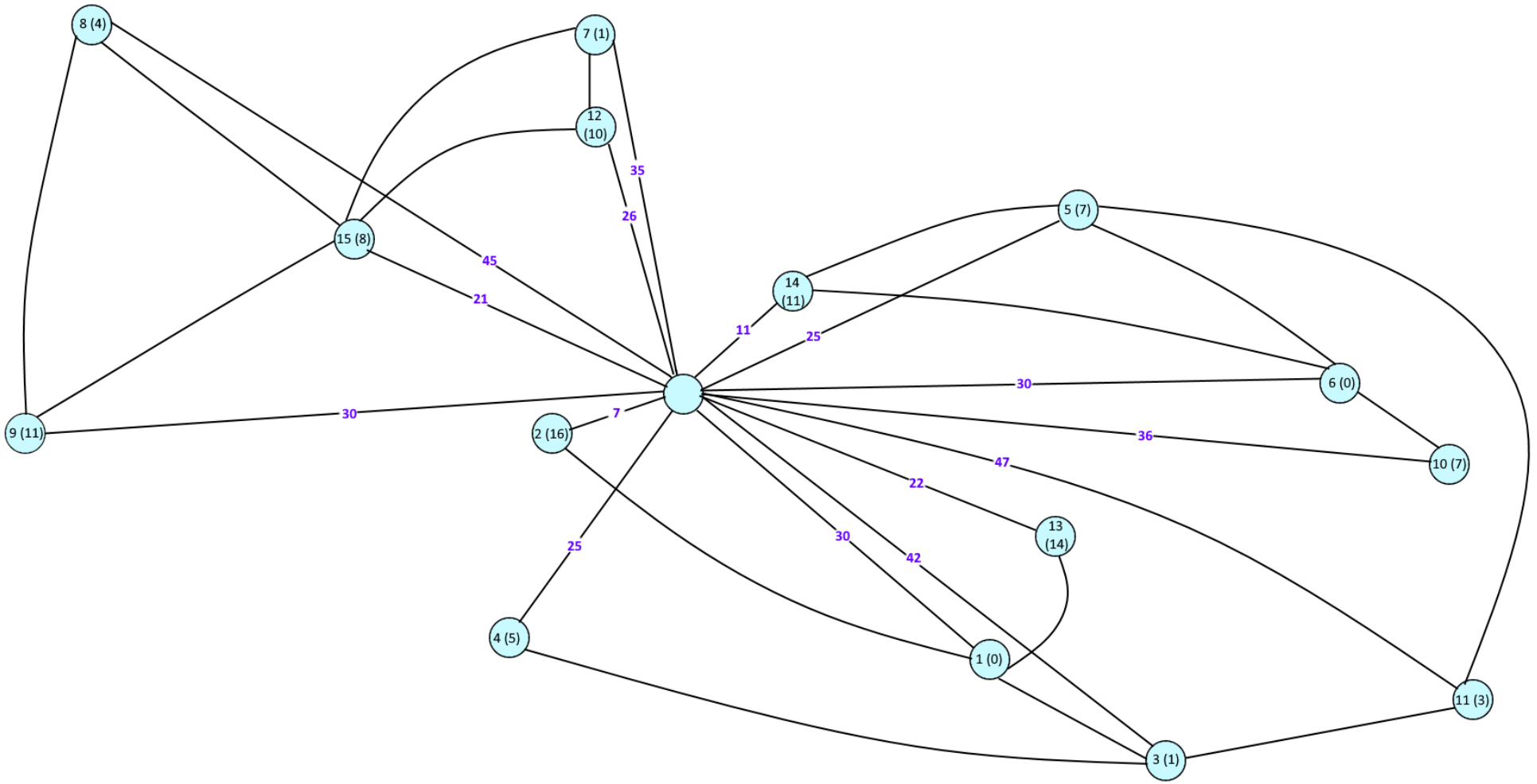
Ukážka grafu



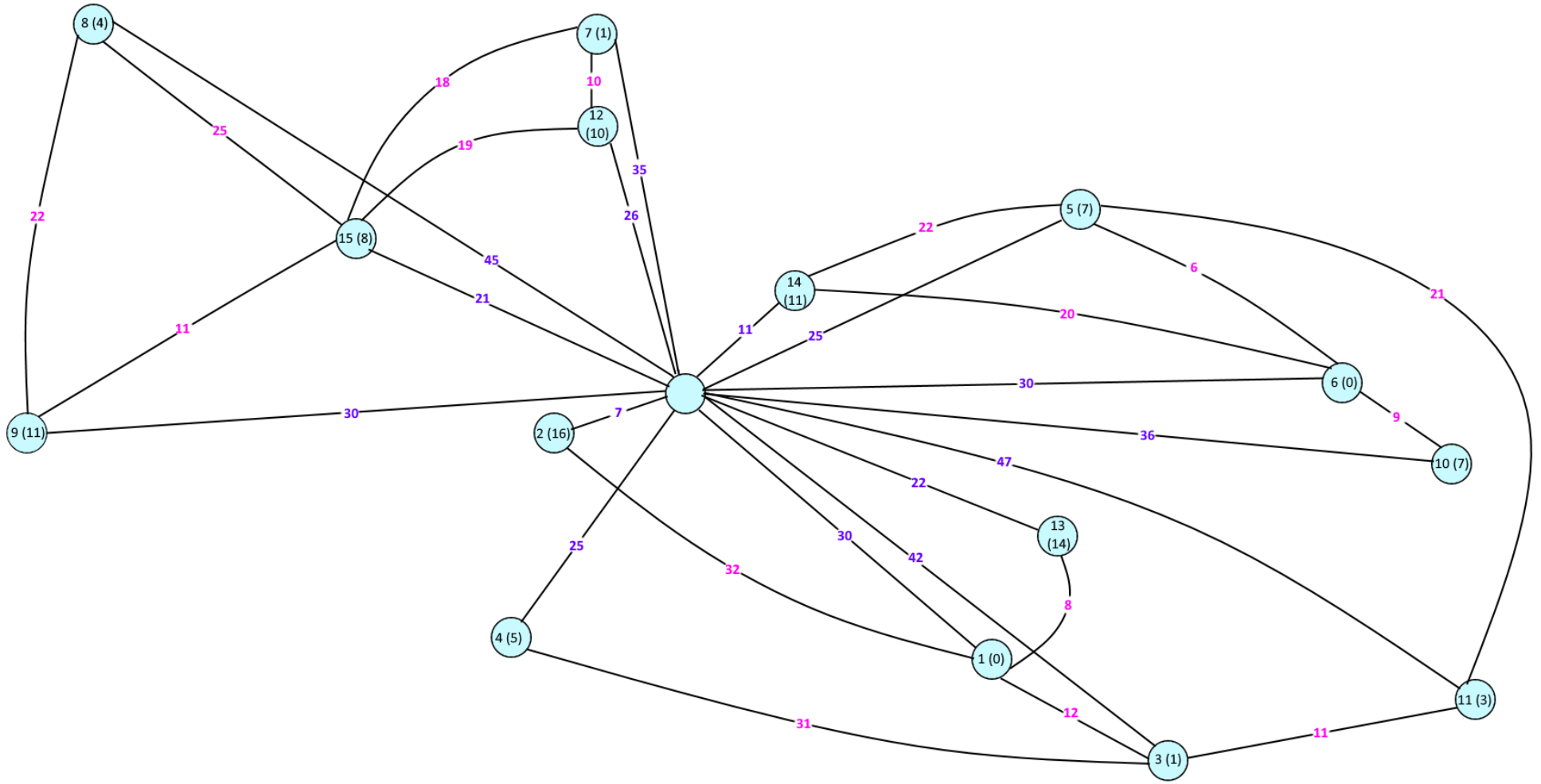
Ukážka grafu



Ukážka grafu



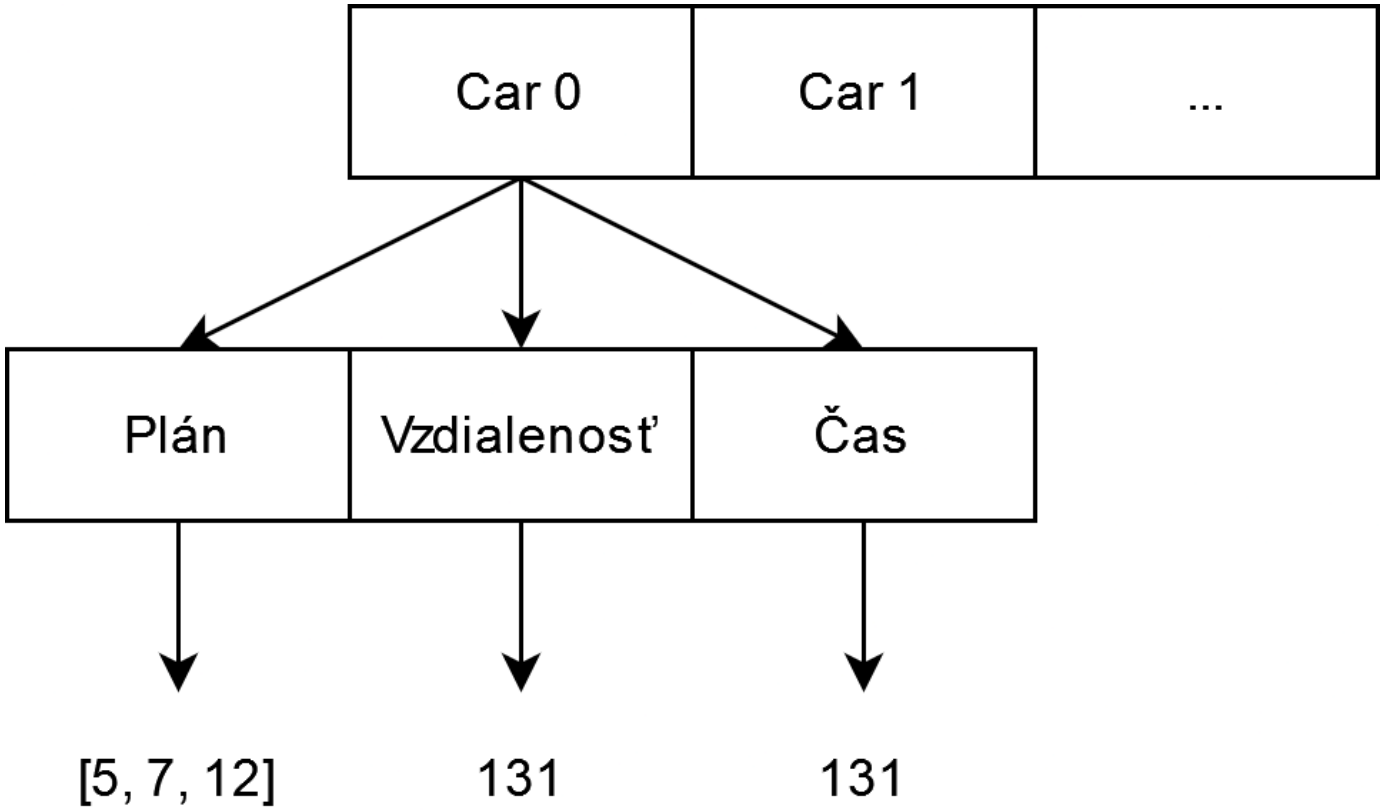
Ukážka grafu



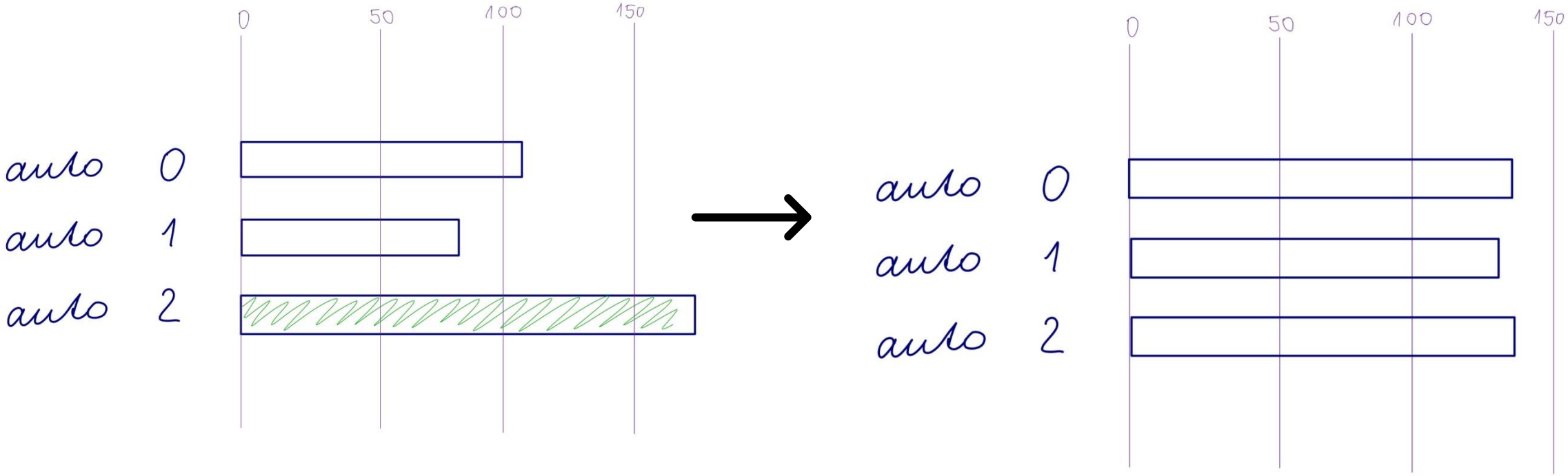
Ukážka matice súsednosti

```
[[ 0. 30.  7. 42. 25. 25. 30. 35. 45. 30. 36. 47. 26. 22. 11. 21.]
 [30.  0. 32. 12. 43. 44. 50. 65. 75. 60. 59. 23. 56.  8. 41. 51.]
 [ 7. 32.  0. 44. 33. 33. 37. 42. 52. 37. 43. 54. 34. 29. 18. 28.]
 [42. 12. 44.  0. 31. 32. 38. 77. 87. 72. 47. 11. 68. 20. 53. 63.]
 [25. 43. 33. 31.  0. 51. 55. 61. 70. 56. 61. 42. 52. 47. 36. 47.]
 [25. 44. 33. 32. 51.  0.  6. 61. 70. 56. 15. 21. 52. 47. 22. 47.]
 [30. 50. 37. 38. 55.  6.  0. 65. 75. 60.  9. 27. 56. 52. 20. 51.]
 [35. 65. 42. 77. 61. 61. 65.  0. 43. 29. 71. 82. 10. 57. 46. 18.]
 [45. 75. 52. 87. 70. 70. 75. 43.  0. 22. 81. 91. 44. 67. 56. 25.]
 [30. 60. 37. 72. 56. 56. 60. 29. 22.  0. 66. 77. 30. 52. 41. 11.]
 [36. 59. 43. 47. 61. 15.  9. 71. 81. 66.  0. 36. 62. 58. 29. 57.]
 [47. 23. 54. 11. 42. 21. 27. 82. 91. 77. 36.  0. 73. 31. 43. 68.]
 [26. 56. 34. 68. 52. 52. 56. 10. 44. 30. 62. 73.  0. 48. 37. 19.]
 [22.  8. 29. 20. 47. 47. 52. 57. 67. 52. 58. 31. 48.  0. 33. 43.]
 [11. 41. 18. 53. 36. 22. 20. 46. 56. 41. 29. 43. 37. 33.  0. 32.]
 [21. 51. 28. 63. 47. 47. 51. 18. 25. 11. 57. 68. 19. 43. 32.  0.]]
```

Flotila aut



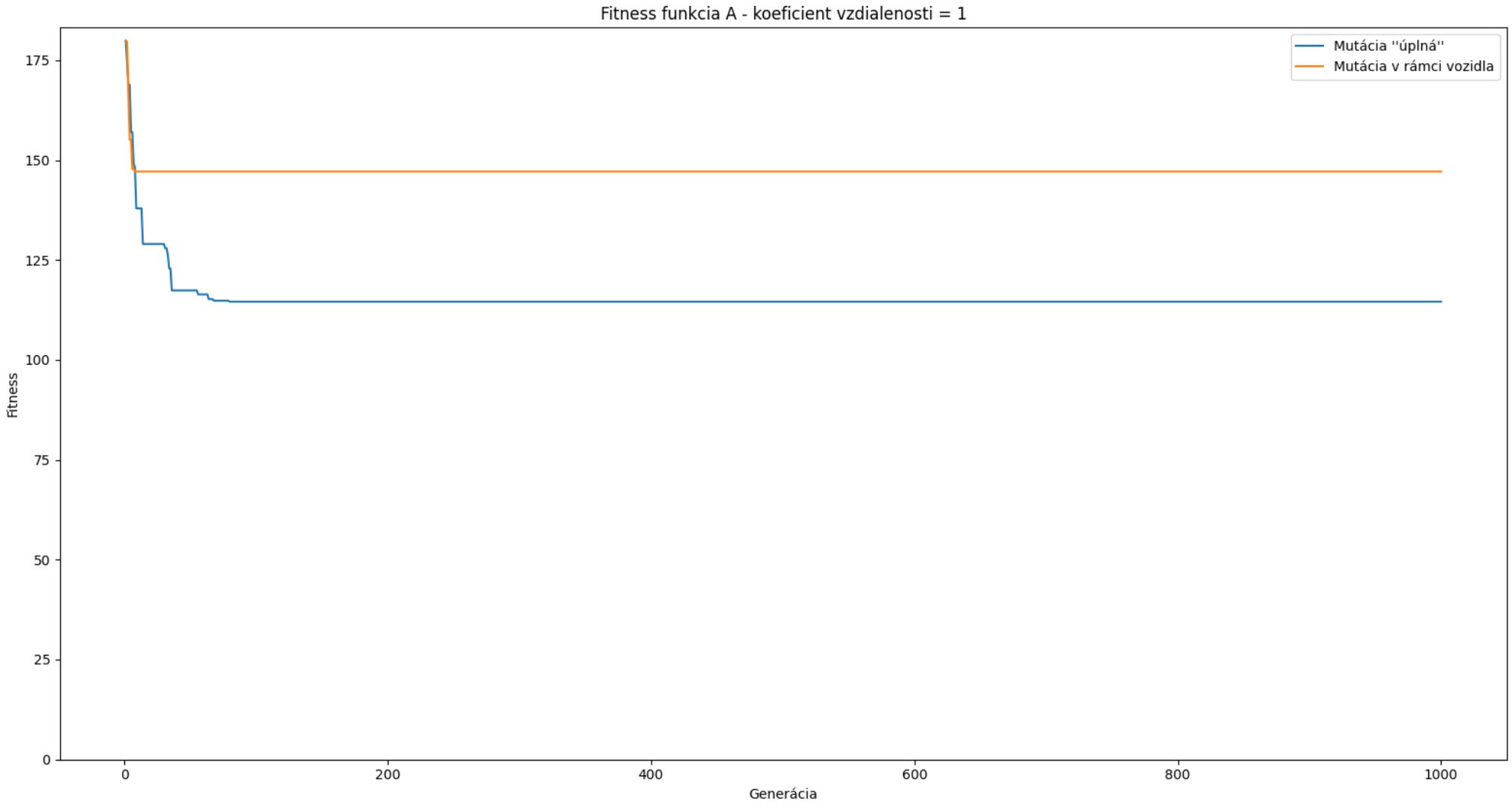
Časová a vzdialenostná optimalizácia



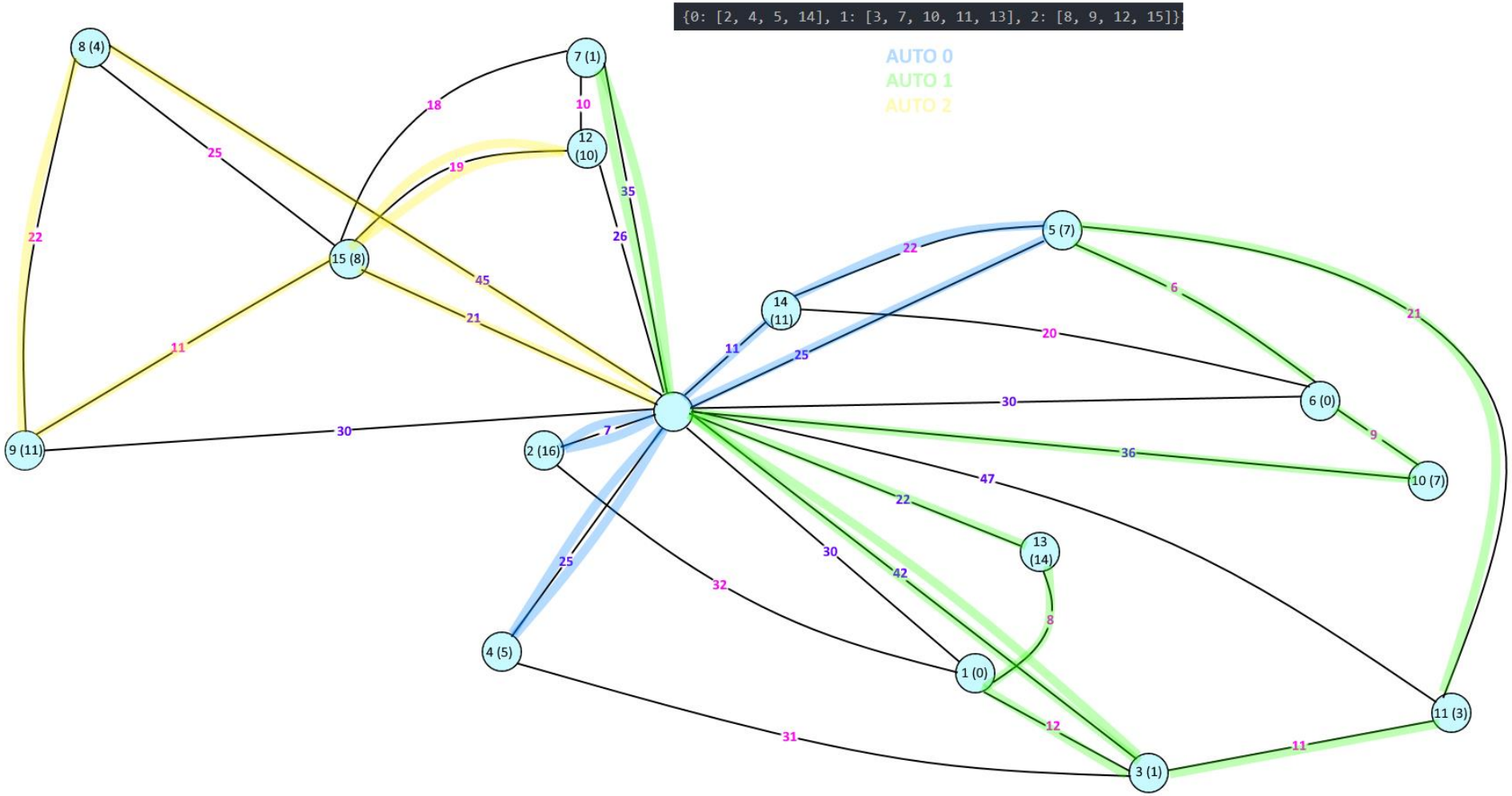
Fitness funkcia

```
fitness = ((COEFICIENT_DISTANCE*distance/fleet)+COEFICIENT_TIME*time)
```

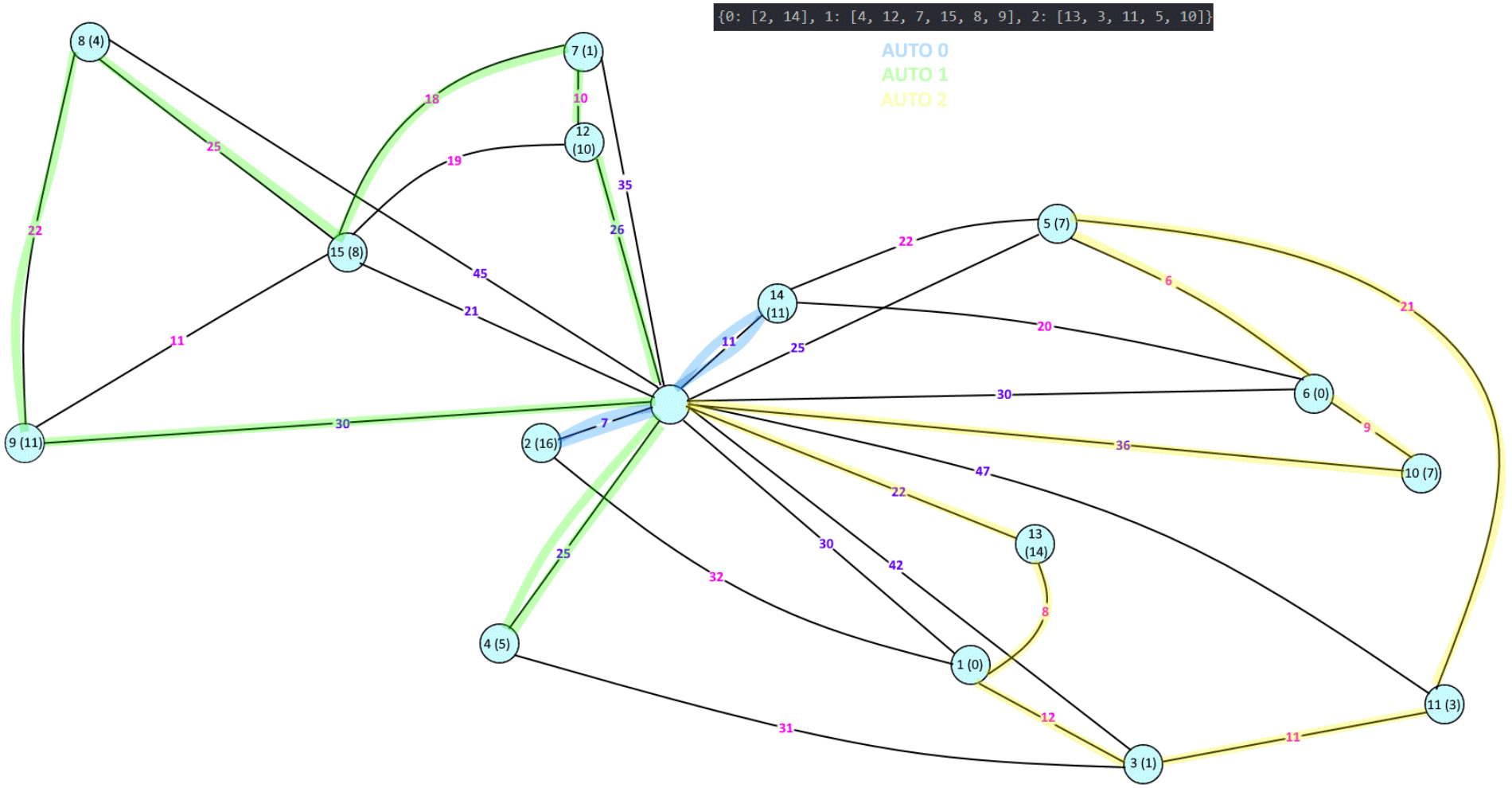
Fitness optimalizovaná pre vzdialenosť



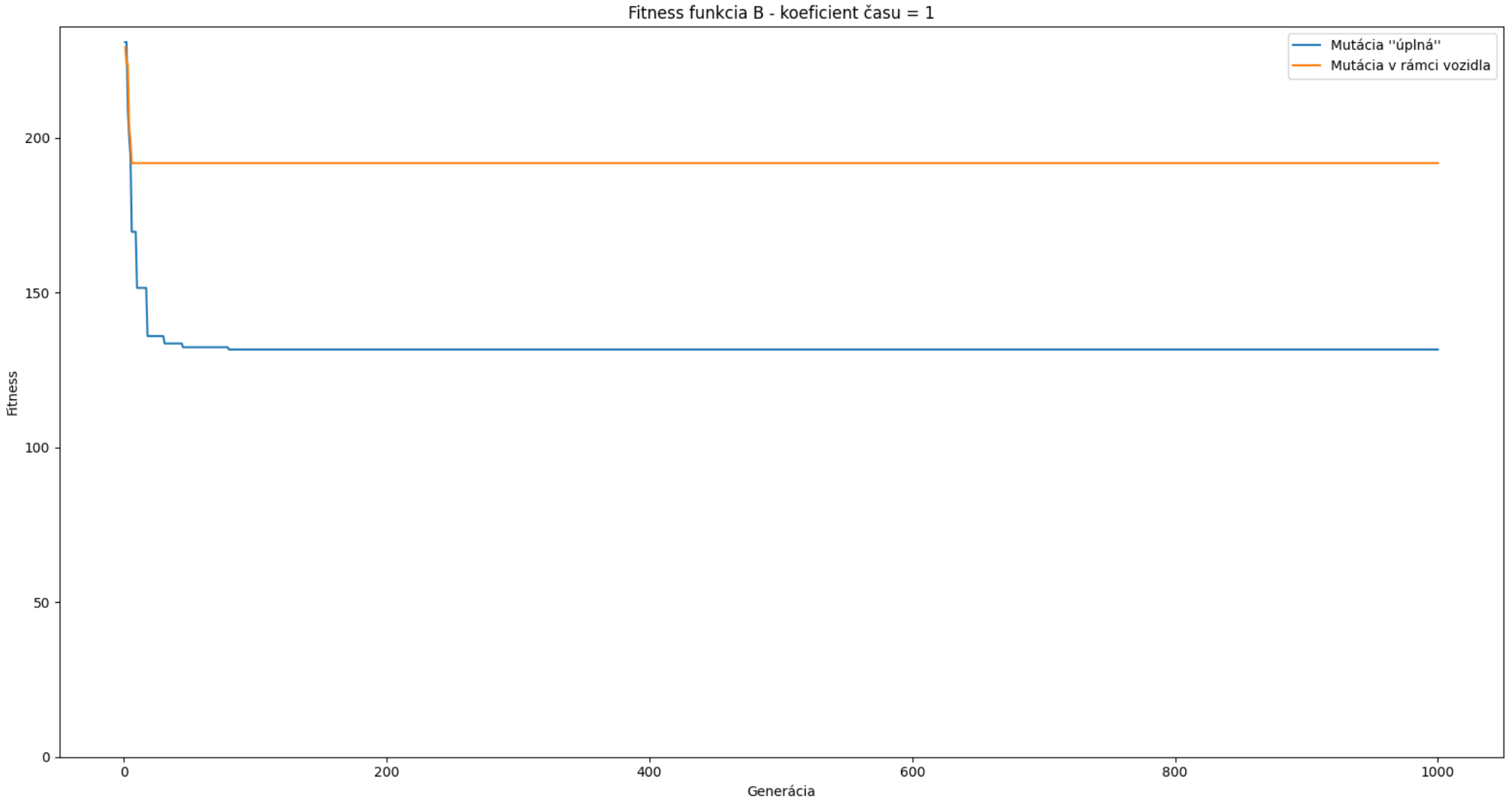
Fitness optimalizovaná pre vzdialenosť – najlepšia trasa prvej generácie



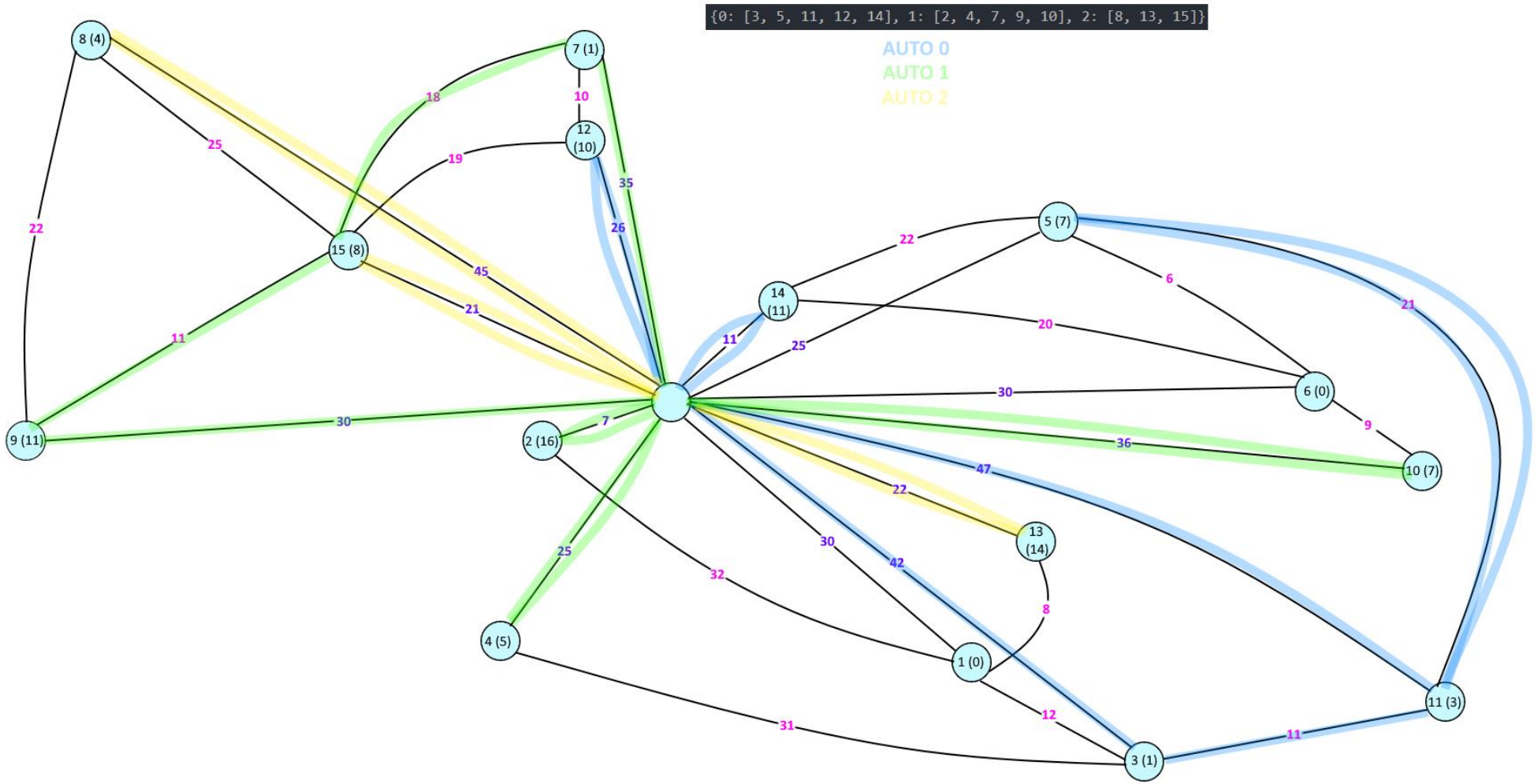
Fitness optimalizovaná pre vzdialenosť – najlepšia trasa tisícej generácie



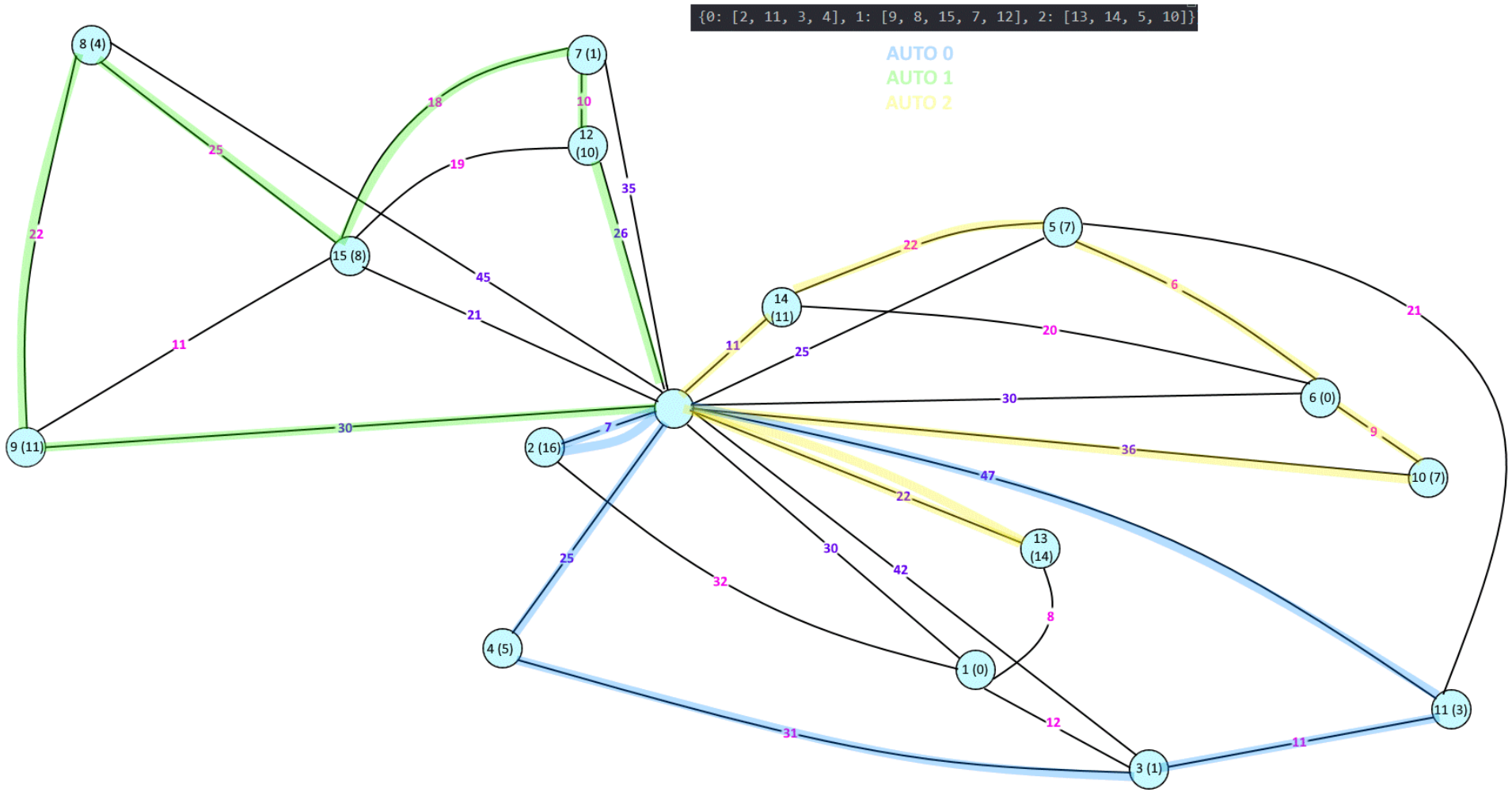
Fitness optimalizovaná pre čas



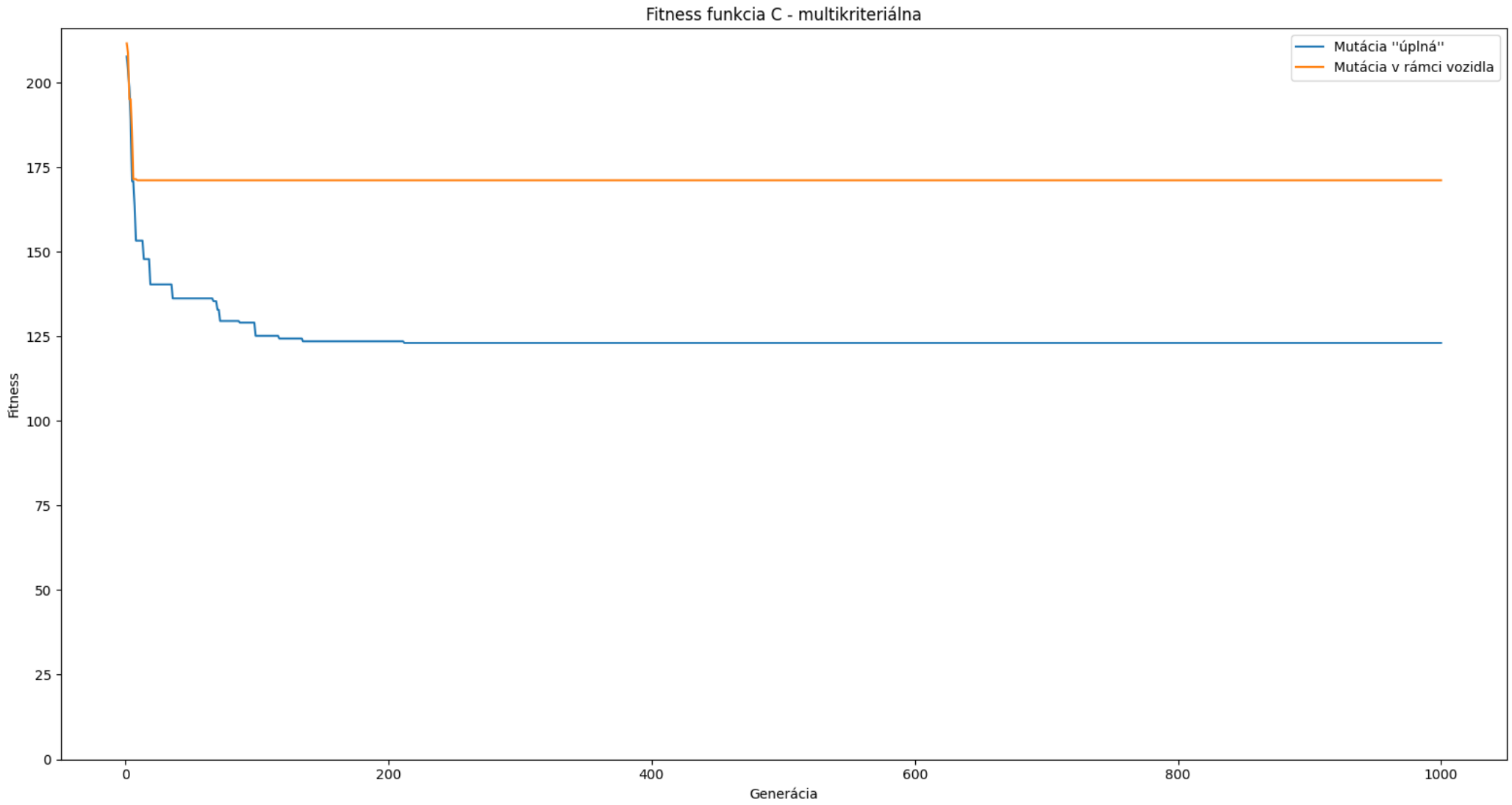
Fitness optimalizovaná pre čas – najlepšia trasa prvej generácie



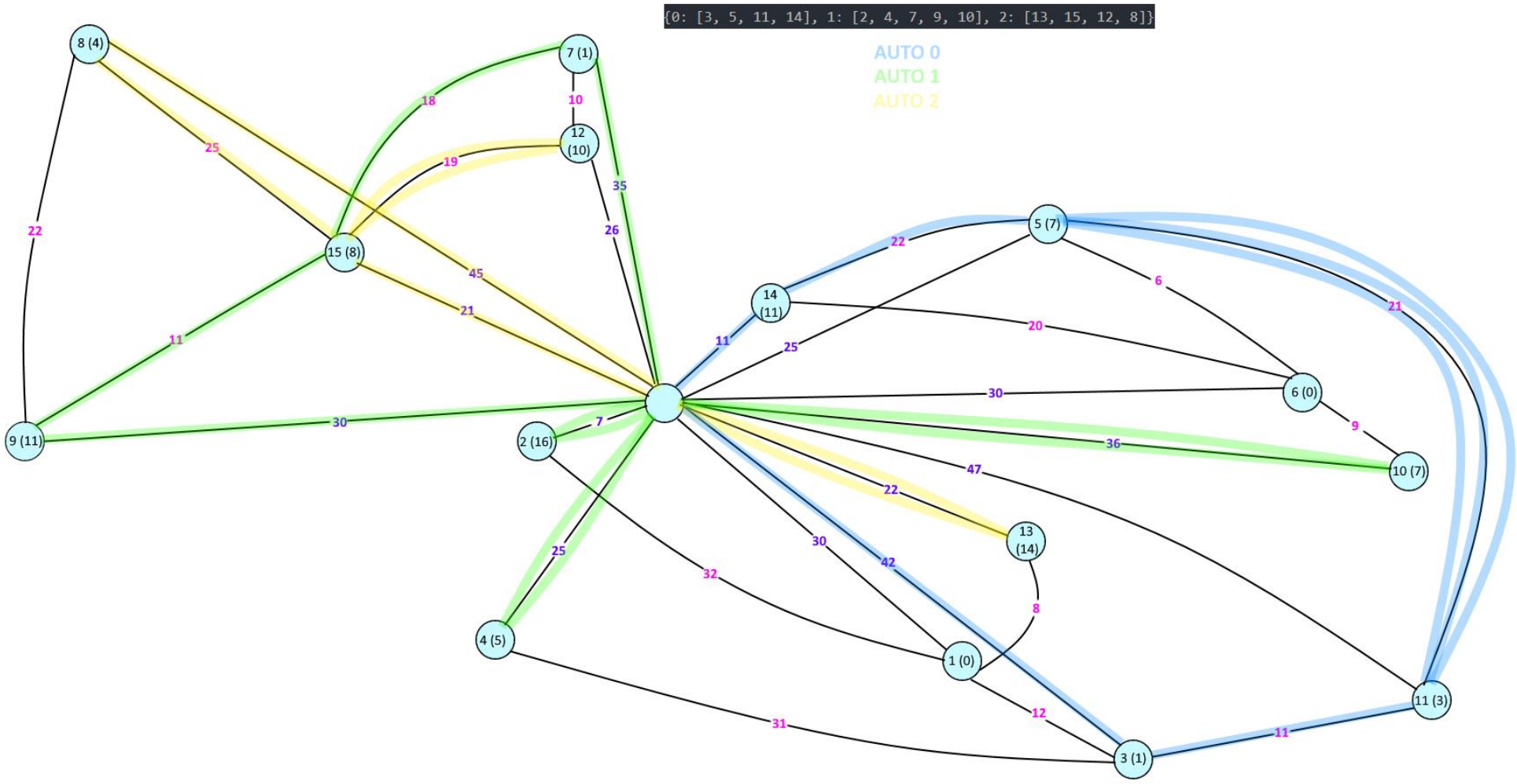
Fitness optimalizovaná pre čas – najlepšia trasa tisícej generácie



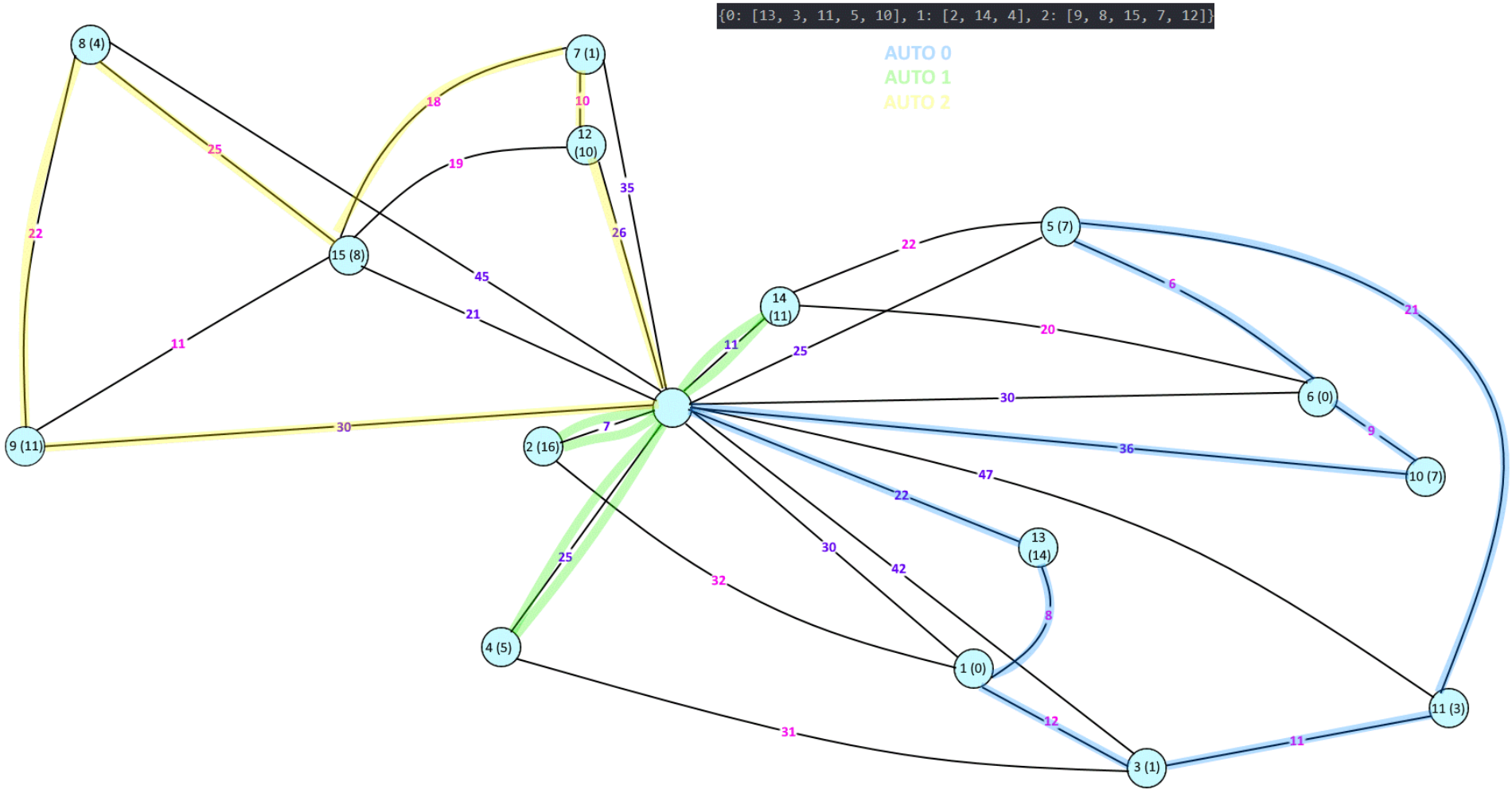
Multikriteriálna fitness



Multikriteriálna fitness – najlepšia trasa prvej generácie



Multikriteriálna fitness – najlepšia trasa tisícej generácie



Děkuji za pozornost!