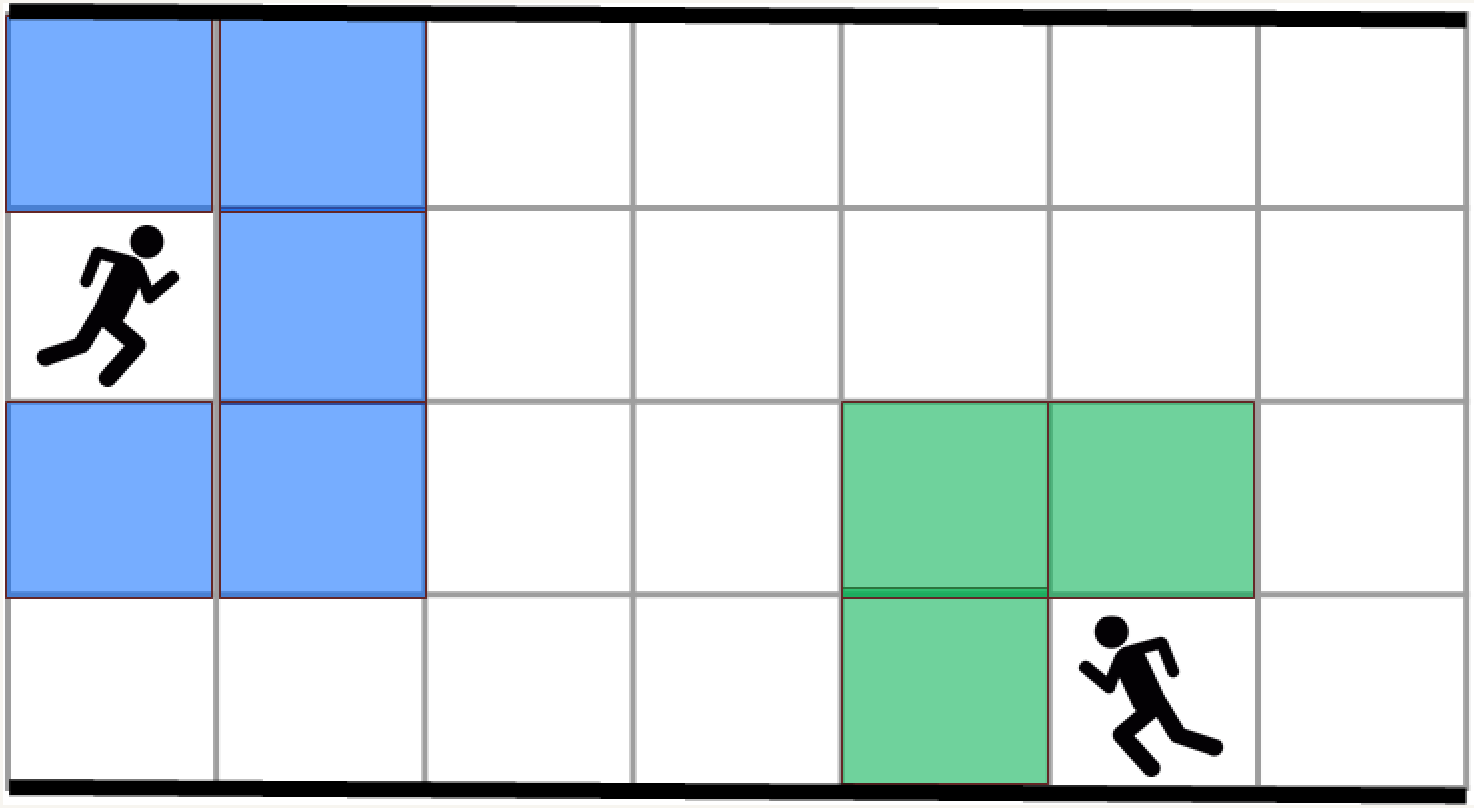
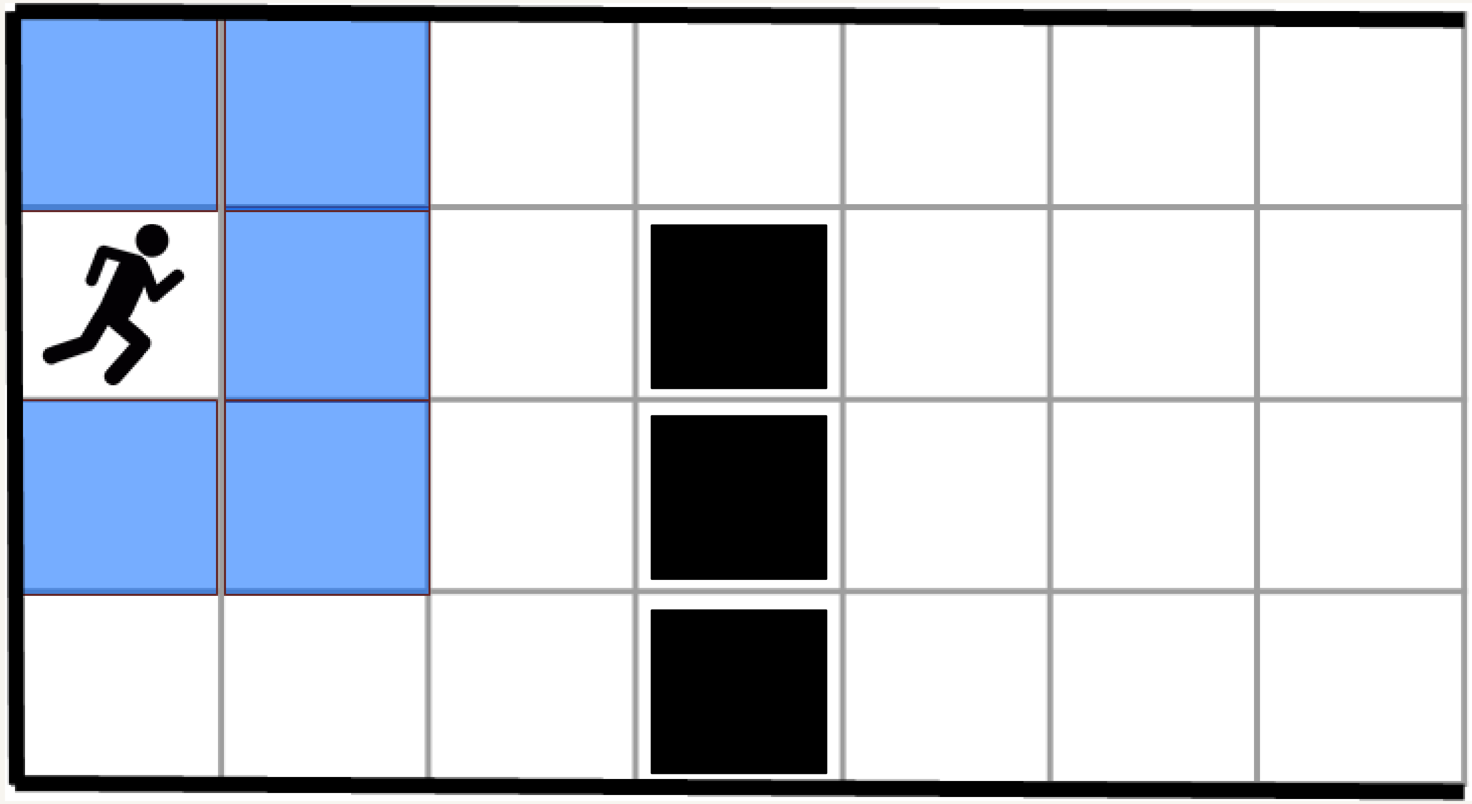
Modelowanie ruchu pieszych podczas ewakuacji

Rozszerzenie modelu

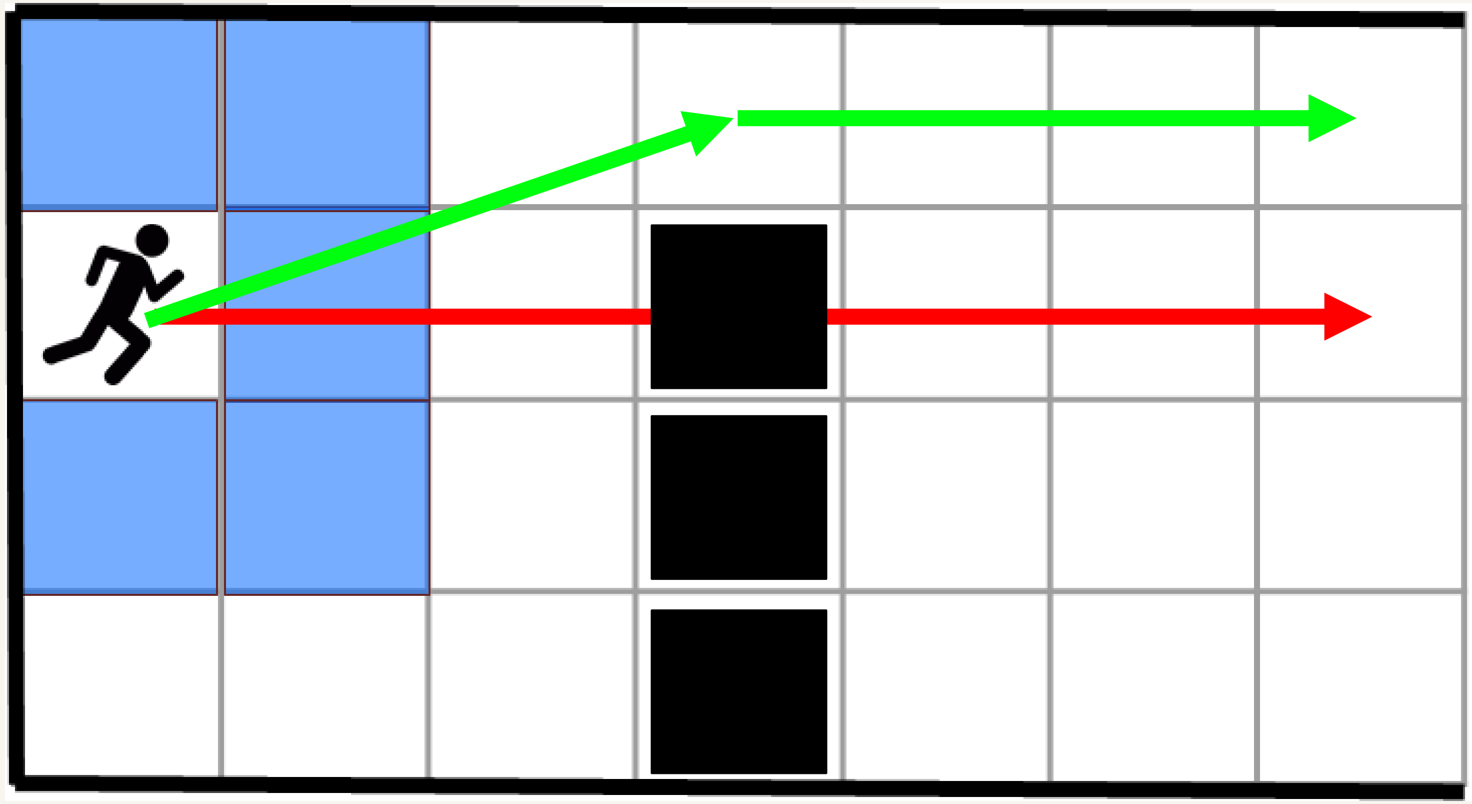
Przypomnienie i rozszerzenie modelu



Przypomnienie i rozszerzenie modelu



Jak przyznawać punkty użyteczności za ruch?



Algorytm Dijkstry

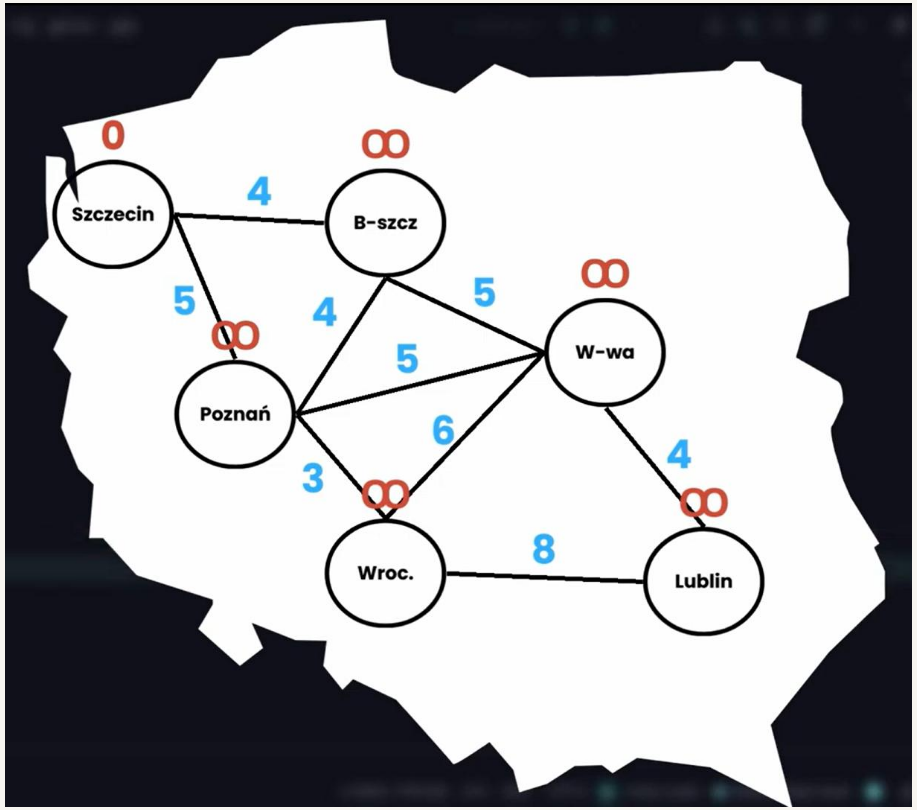
Służy do znajdowania najkrótszych ścieżek z wybranego wierzchołka do pozostałych wierzchołków w grafie o nieujemnych wagach krawędzi.

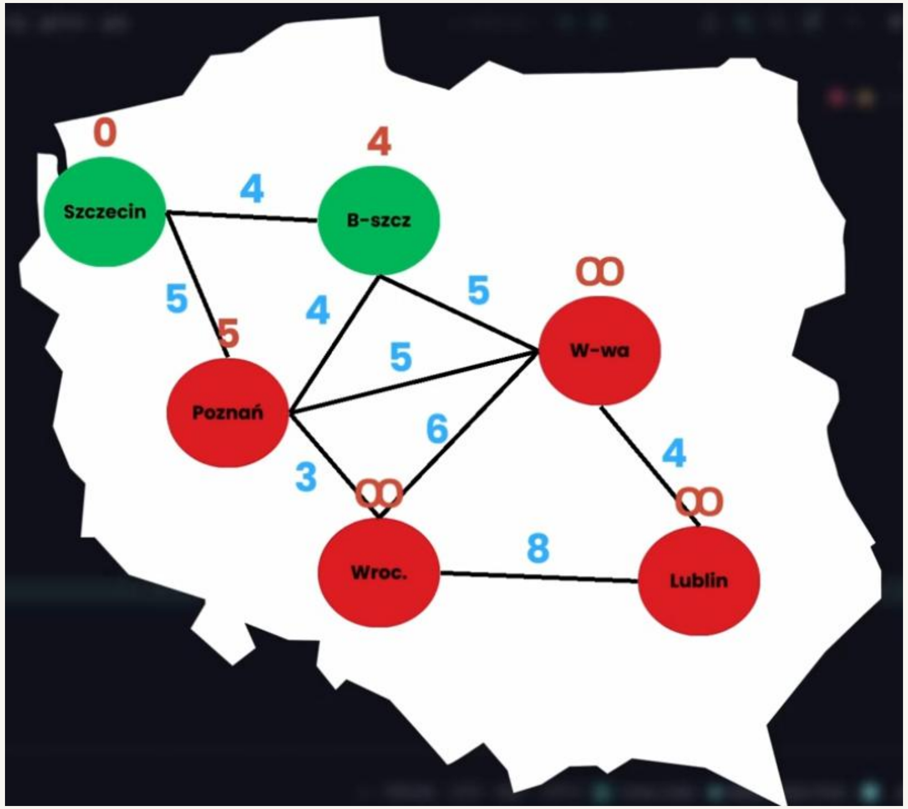
Algorytm:

- Ustalamy $d[s]=0$, $d[v]=\infty$ dla wszystkich pozostałych wierzchołków
- Tworzymy kolejkę priorytetową dla wszystkich wierzchołków grafu (aktualna odległość od wierzchołka źródłowego)
- Dla każdego sąsiada wierzchołka v dokonujemy relaksacji poprzez u: $d[u] + w(u,v) < d[v]$. Jeśli jest krótsza niż dotychczasowa ścieżka to $d[v] := d[u]+w(u,v)$

Algorytm Dijkstry

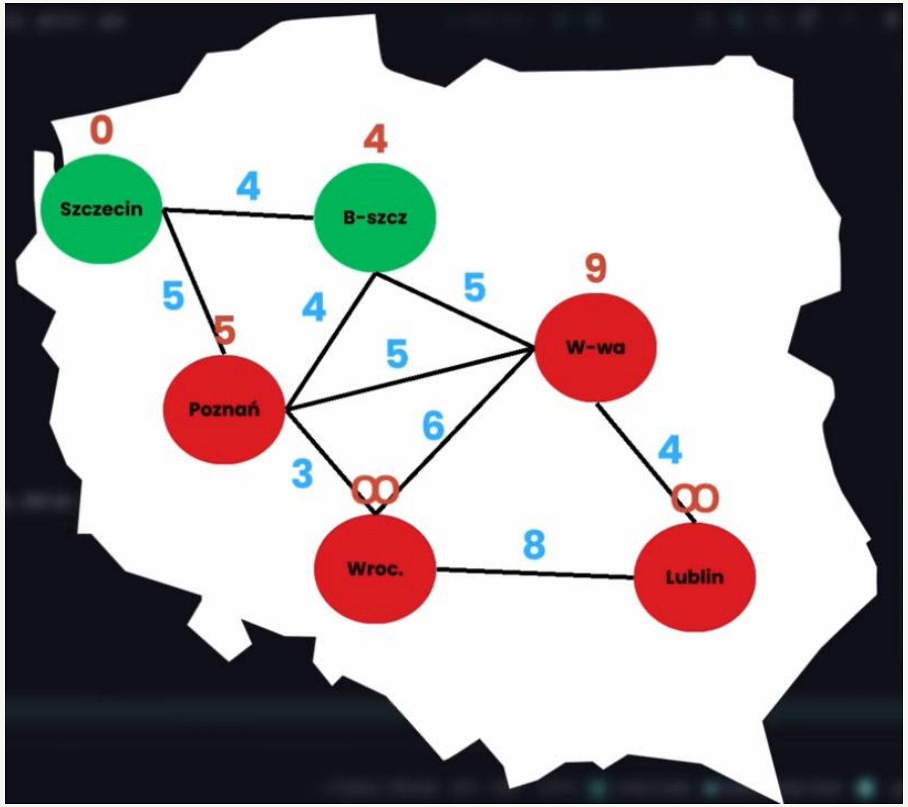
Wybieramy nieodwiedzone miasto
z najmniejszą liczbą godzin

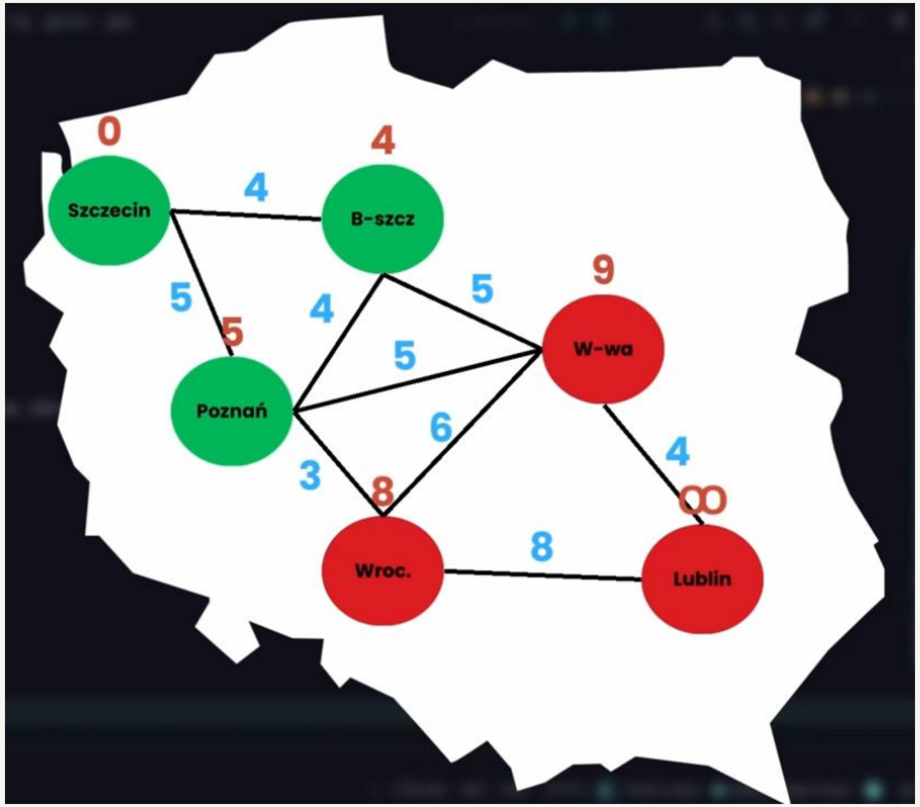




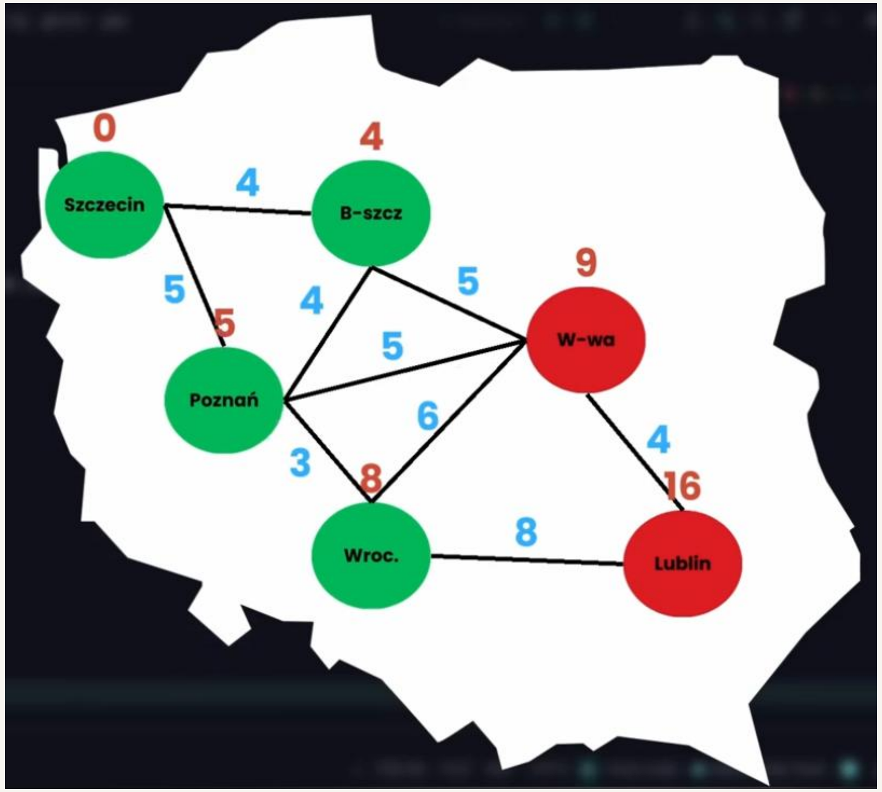
Algorytm Dijkstry

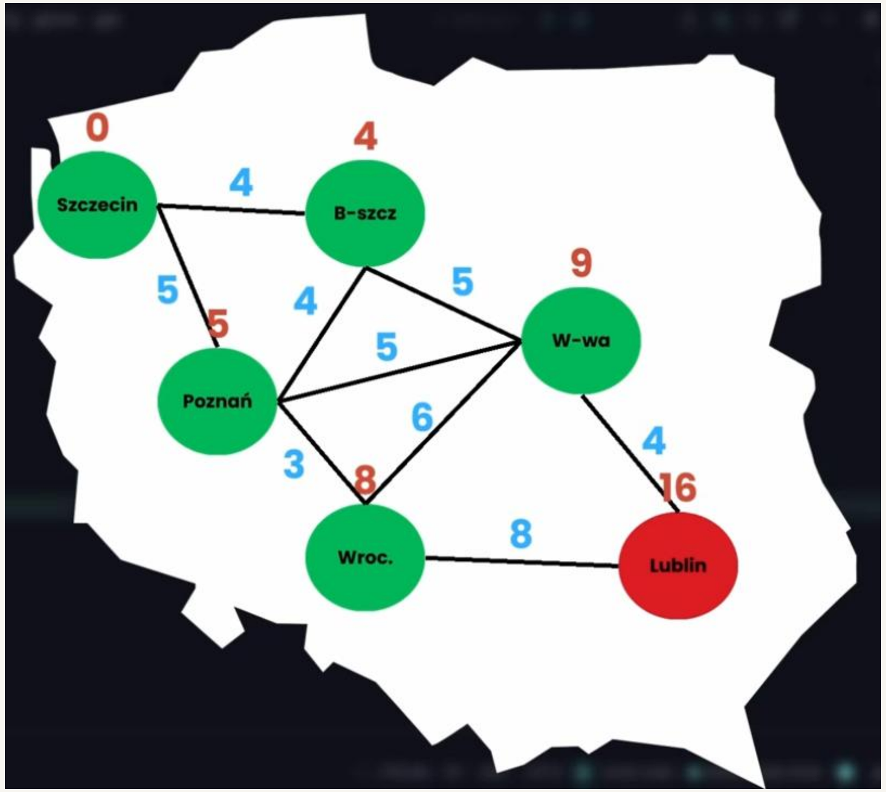
Celem jest by każde miasto miało najmniejszą możliwą liczbę godzin potrzebną by do niego dotrzeć



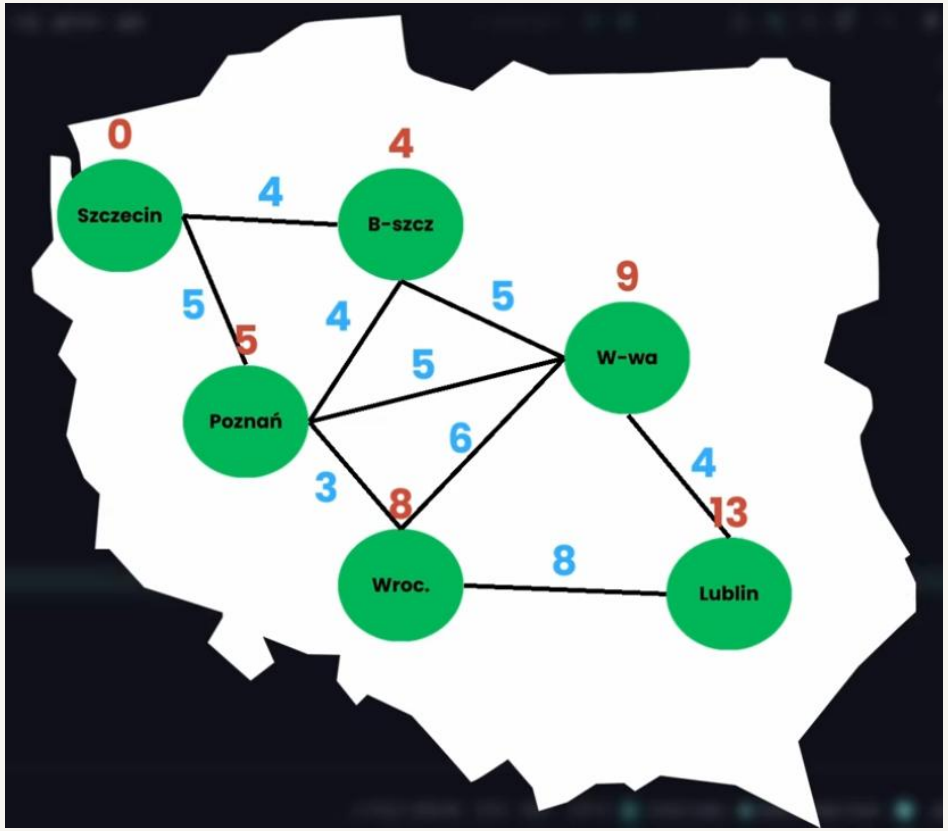


Algorytm Dijkstry





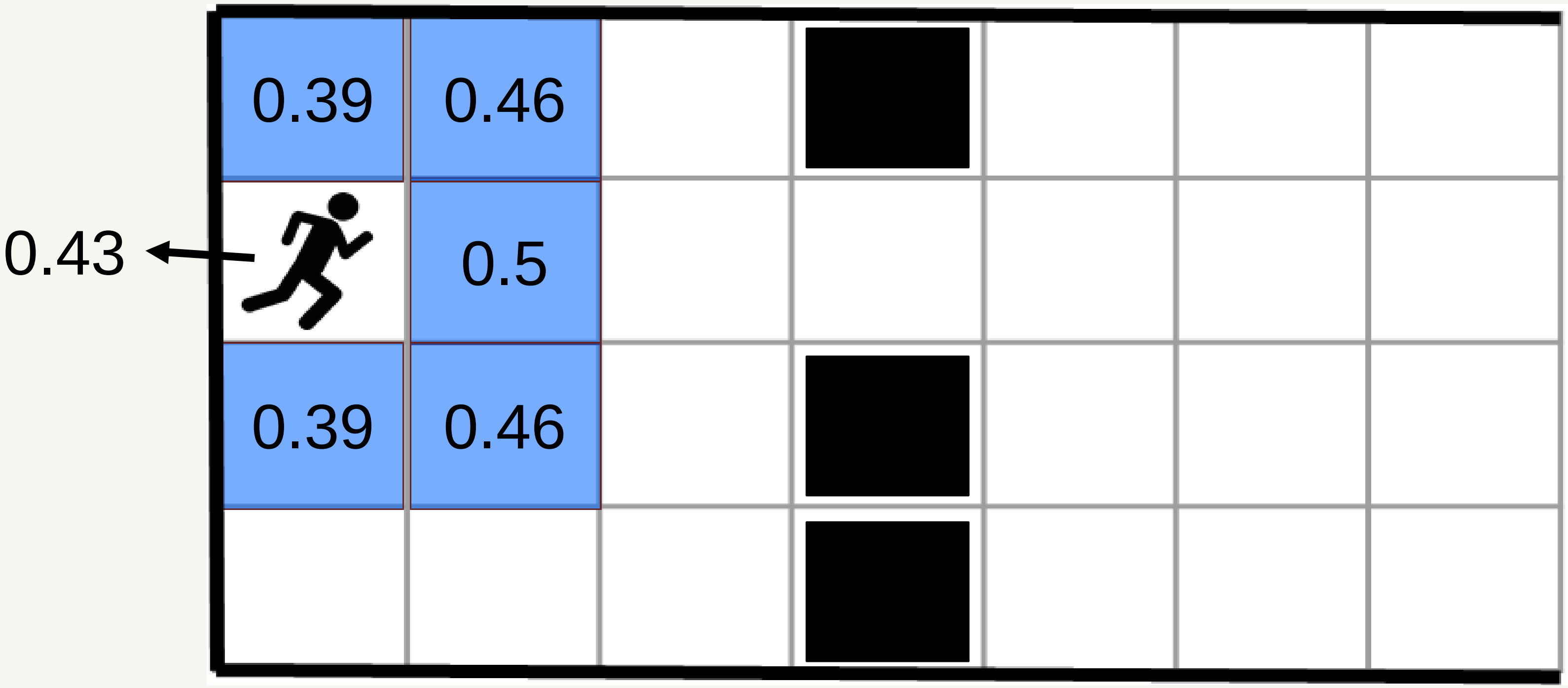
Algorytm Dijkstry



Jak wykorzystać algorytm Dijkstry?

-1	-1	-1	-1	-1	-1	-1	-1	0.37	0.35
-1	-1	-1	-1	-1	-1	-1	-1	0.42	0.37
0.68	0.69	0.68	0.66	0.61	0.57	0.52	0.48	0.43	0.39
0.72	0.74	0.72	0.68	0.63	0.59	0.54	0.5	0.45	0.41
0.77	0.78	-1	-1	-1	-1	-1	-1	-1	-1
0.81	0.83	-1	-1	-1	-1	-1	-1	-1	-1
0.83	0.87	-1	-1	-1	-1	-1	-1	-1	-1
0.85	0.89	0.94	0.96	0.96	0.96	0.96	0.94	0.89	0.85
0.87	0.91	0.96	1	1	1	1	0.96	0.91	0.87

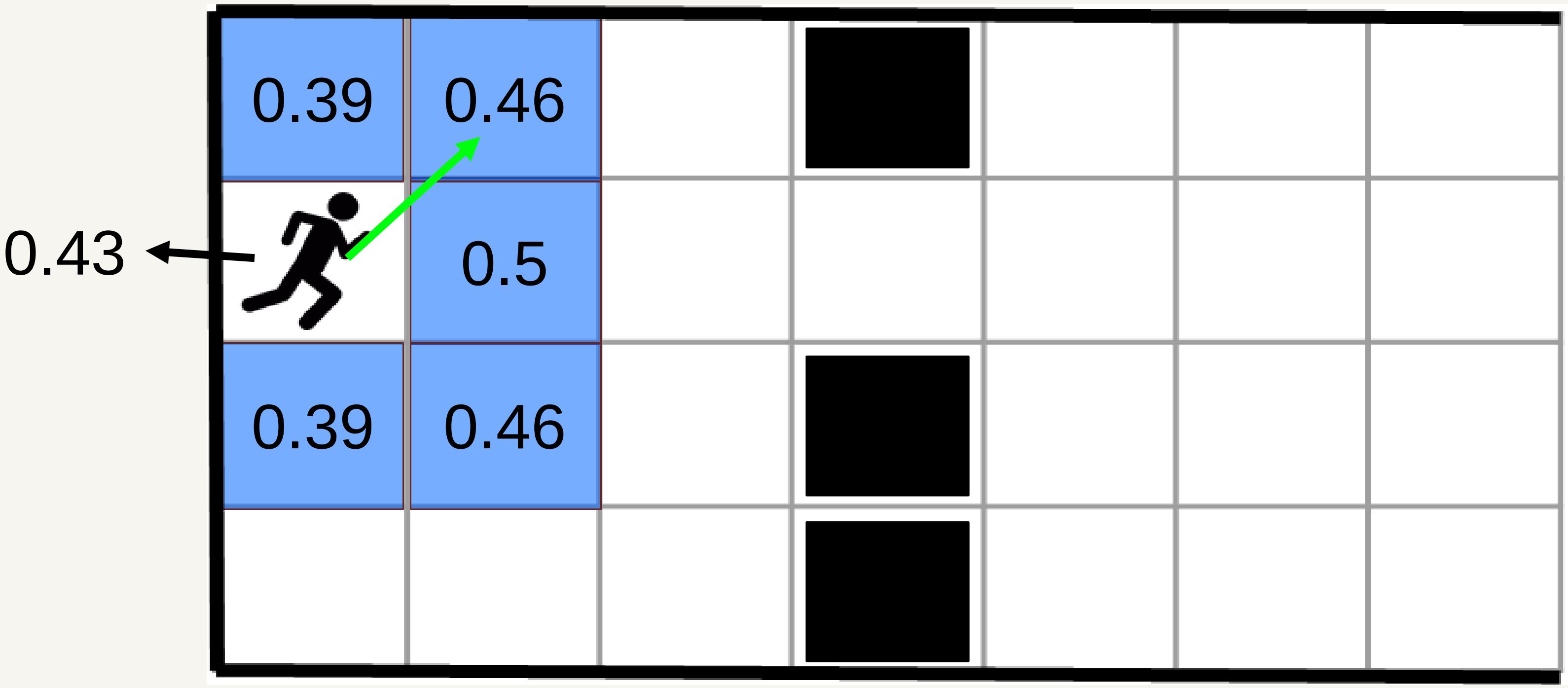
Jak wykorzystać algorytm Dijkstry?



$$U_{move} = \frac{P_{cell} - P_{agent}}{D_{max}}$$

Umove - punkty użyteczności za ruch, Pcell - liczba w komórce do której ruch rozpatrujemy, Pagent - punkty w komórce w której znajduje się agent, Dmax - największa możliwa wartość Pcell - Pagent w danym ruchu.

Jak wykorzystać algorytm Dijkstry?



$$Umove = \frac{Pcell - Pagent}{Dmax}$$

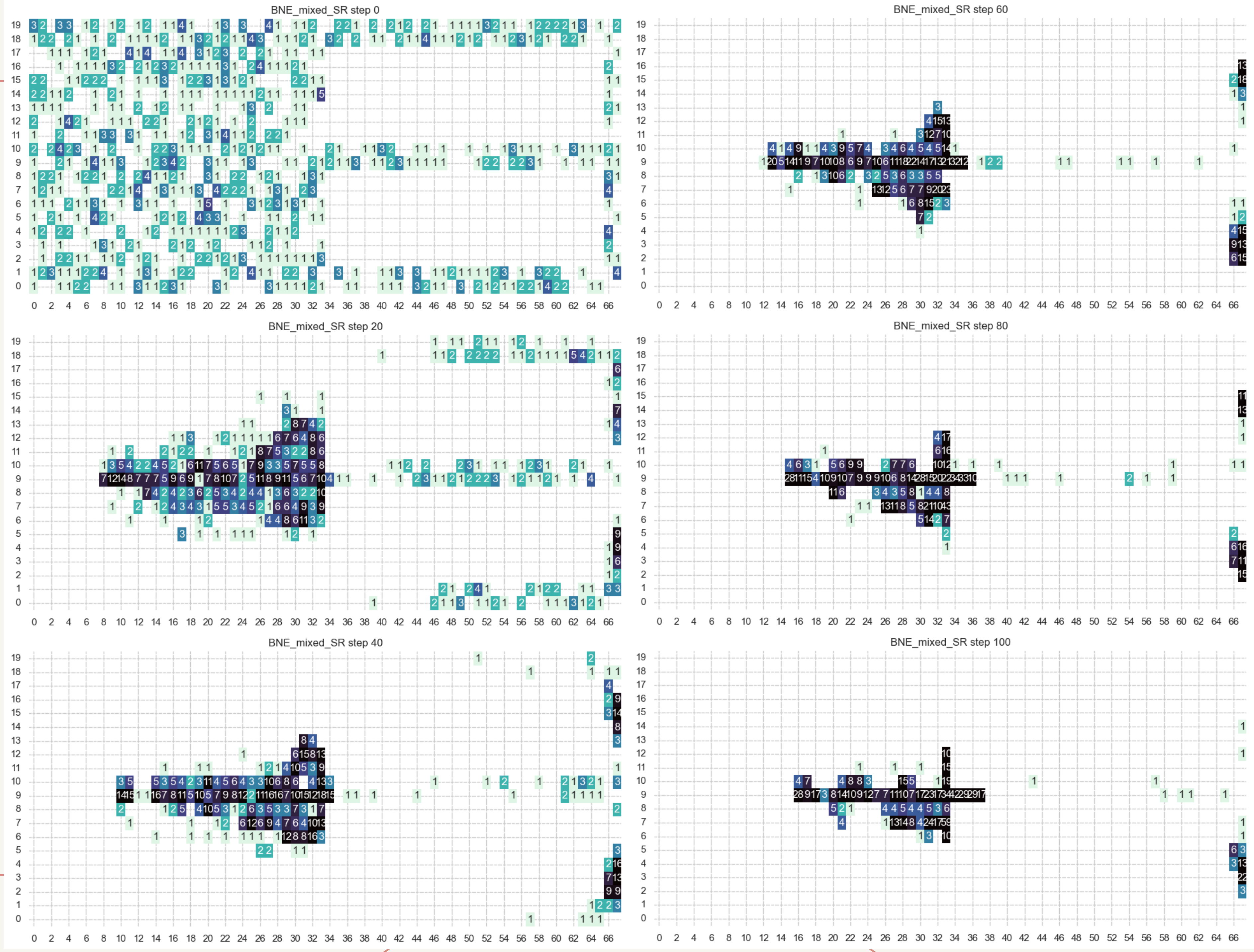
Dla ruchu oznaczonego zieloną strzałką

Umove =
=(0.46-0.43)/(0.5-0.43)=
=0.03/0.07≈0.43

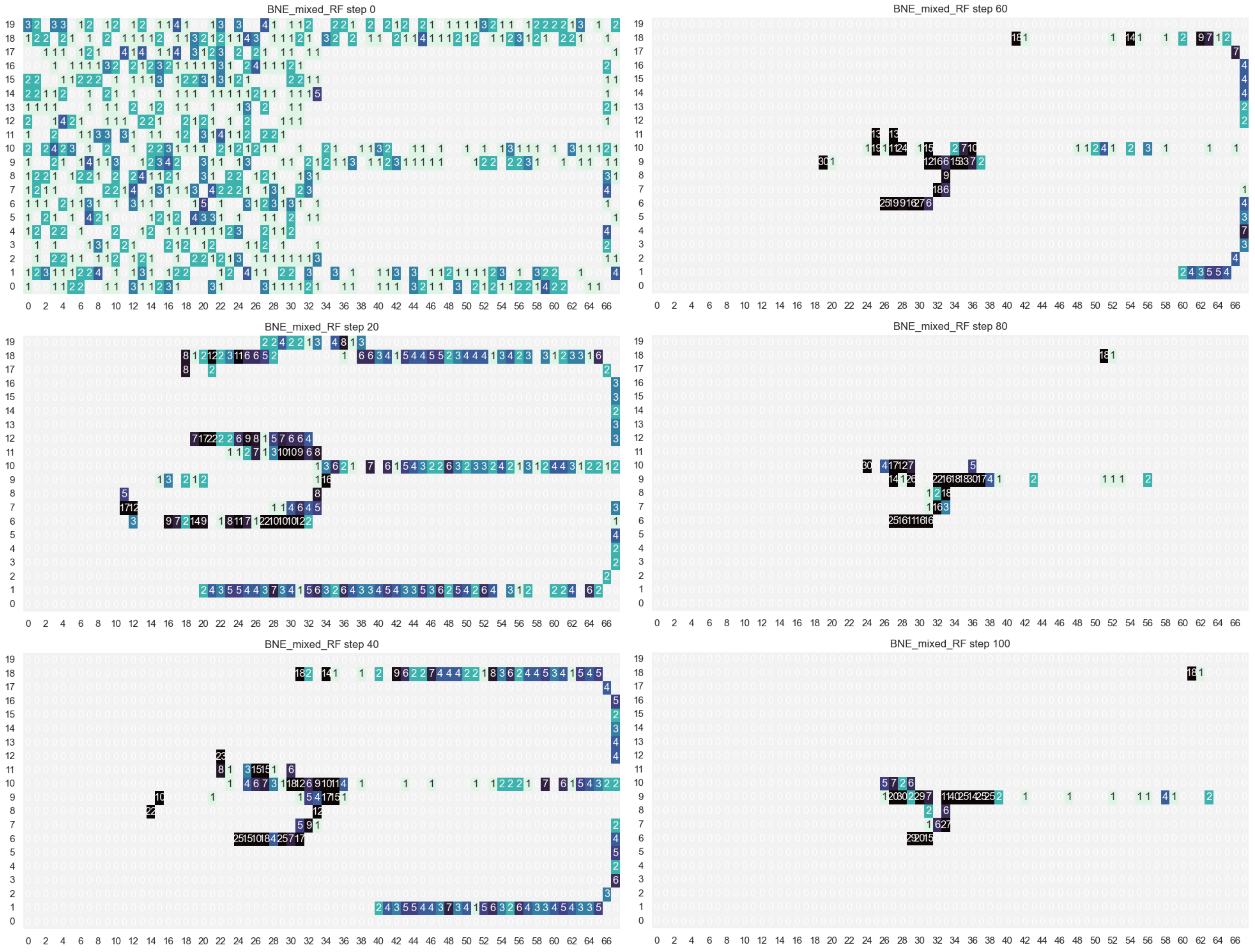
Czas na symulacje!

Klatki z symulacji:
Map_type: two blocks

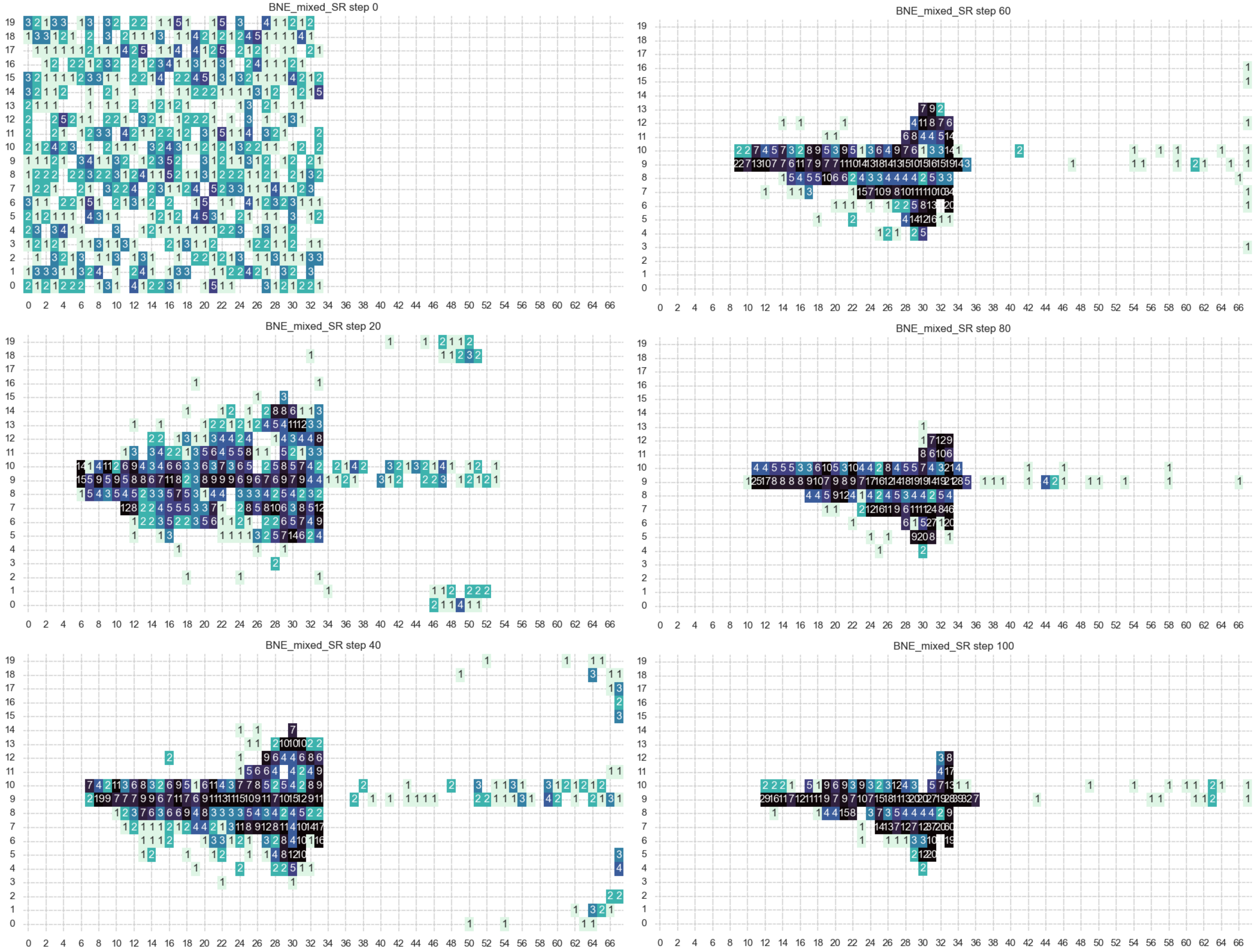
BNE, two_blocks



RF, two_blocks

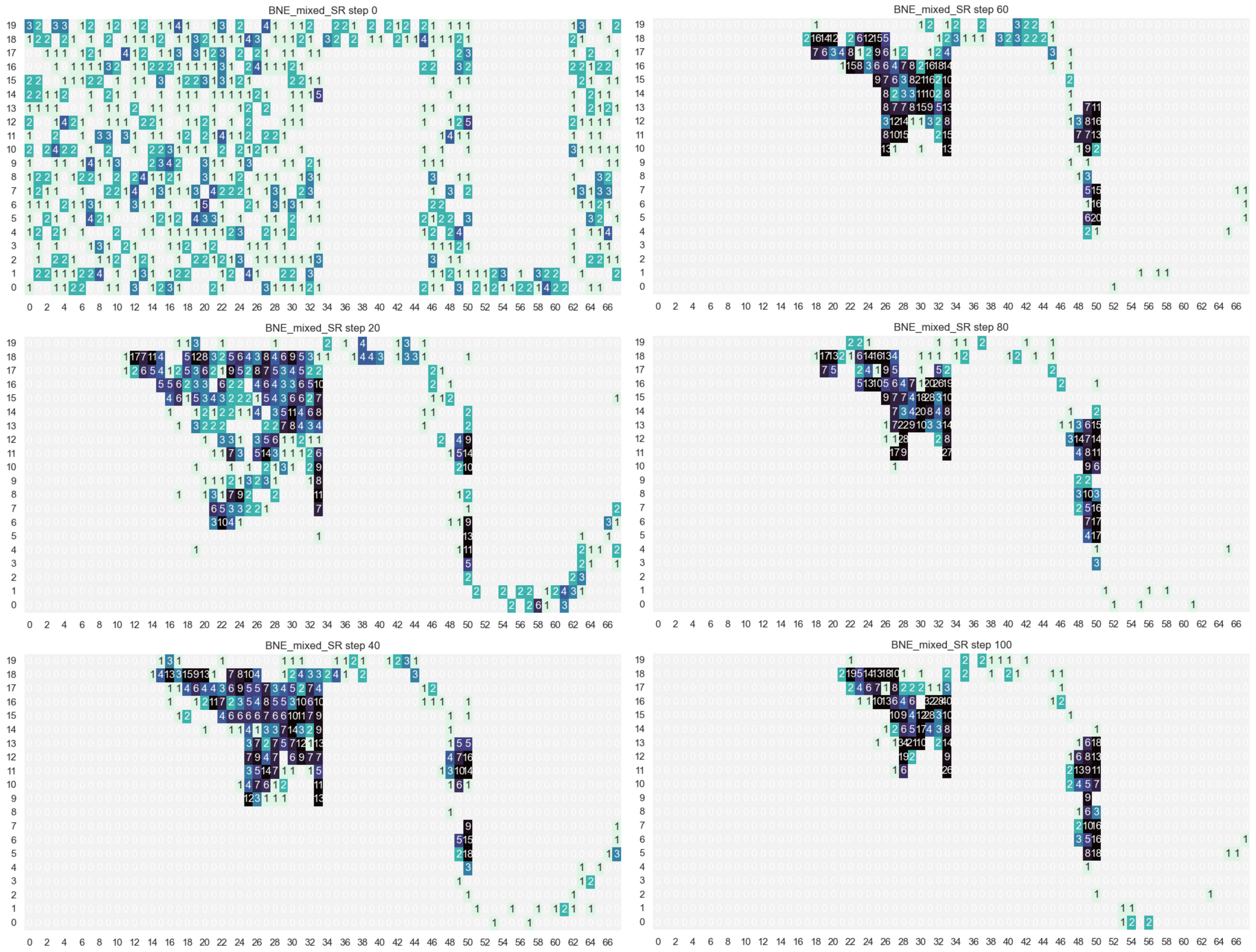


BNE, two_blocks, left_half

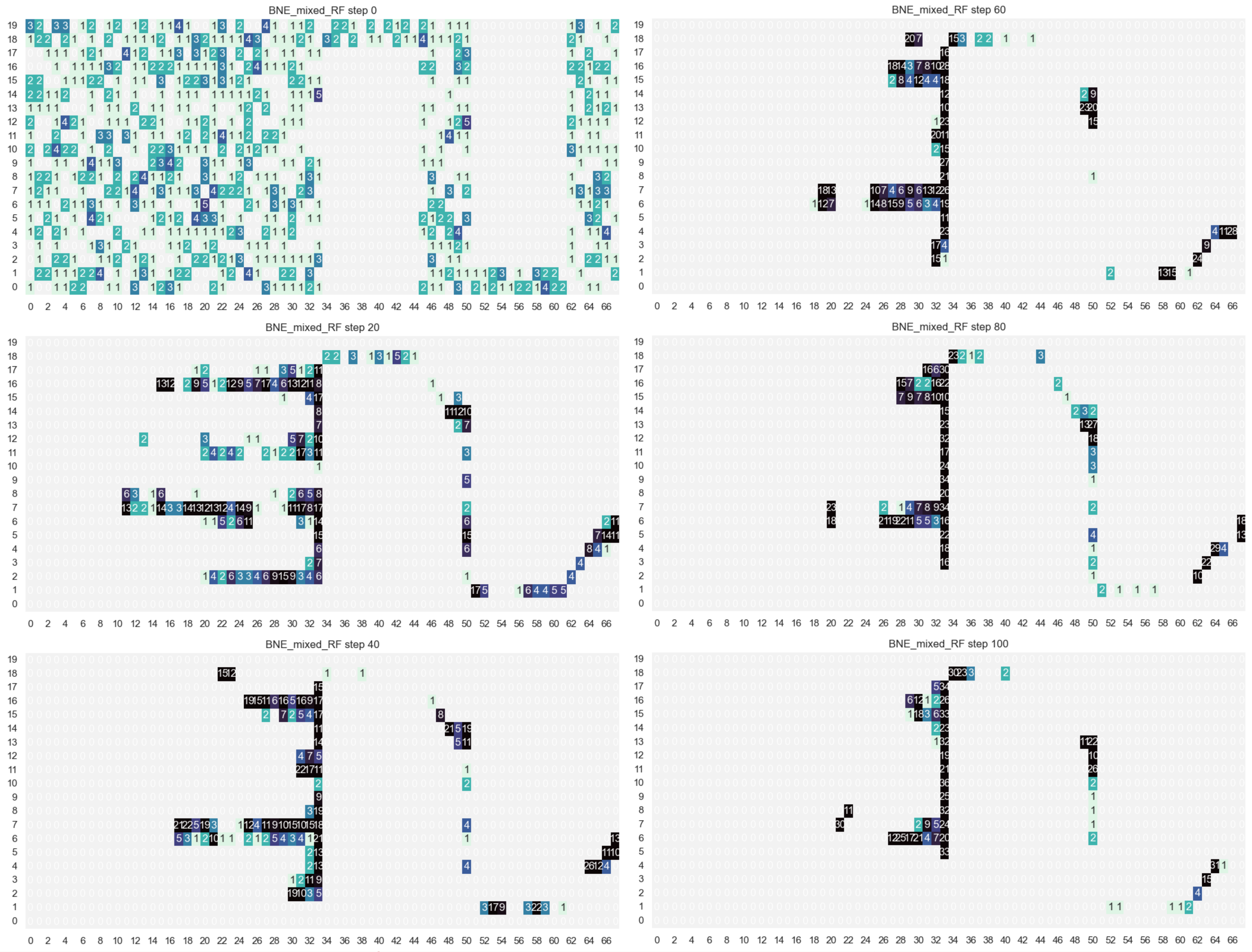


Klatki z symulacji:
Map_type: snake

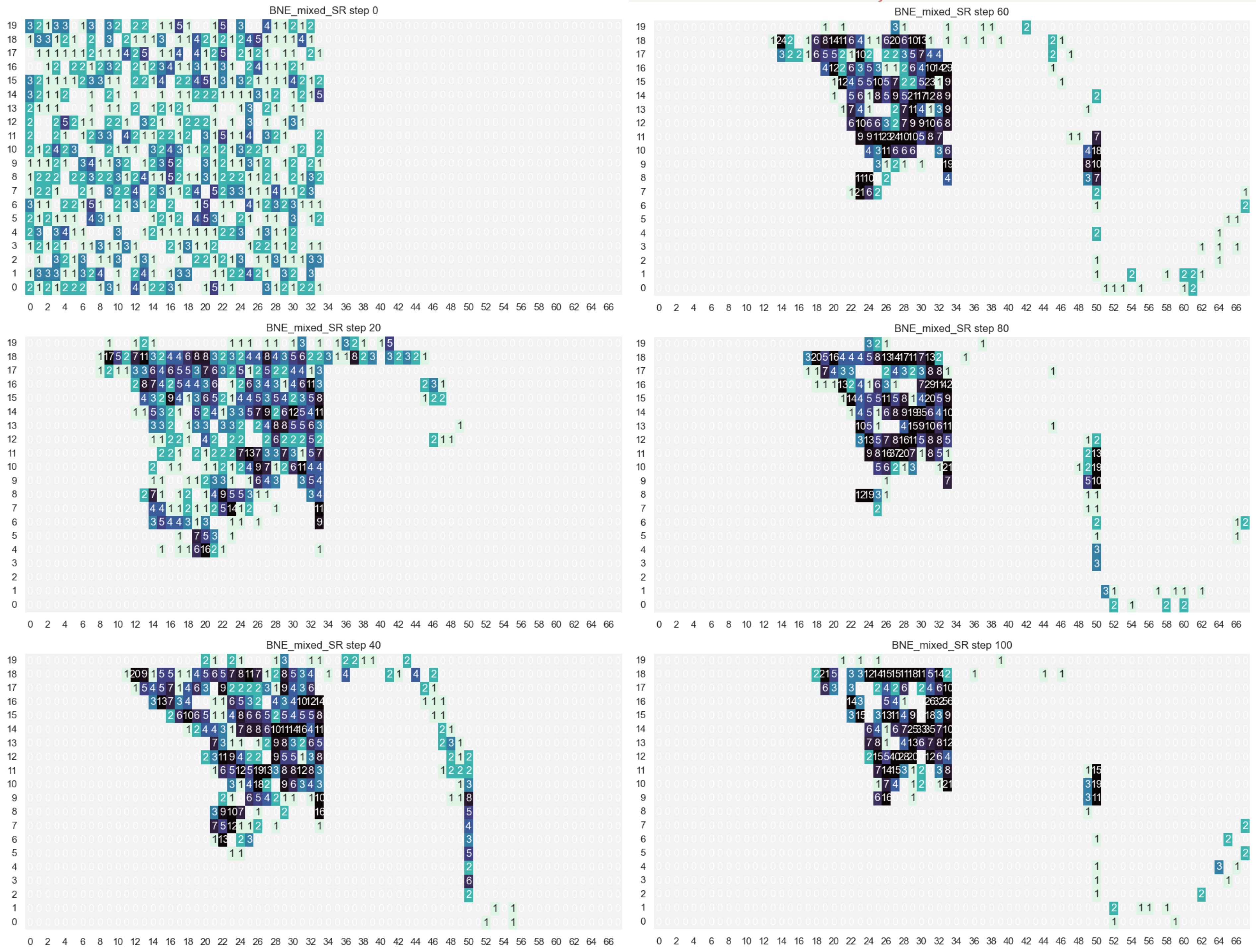
BNE, snake


BNE_mixed_SR step 0
BNE_mixed_SR step 20
BNE_mixed_SR step 40
BNE_mixed_SR step 60
BNE_mixed_SR step 80
BNE_mixed_SR step 100

RF, snake

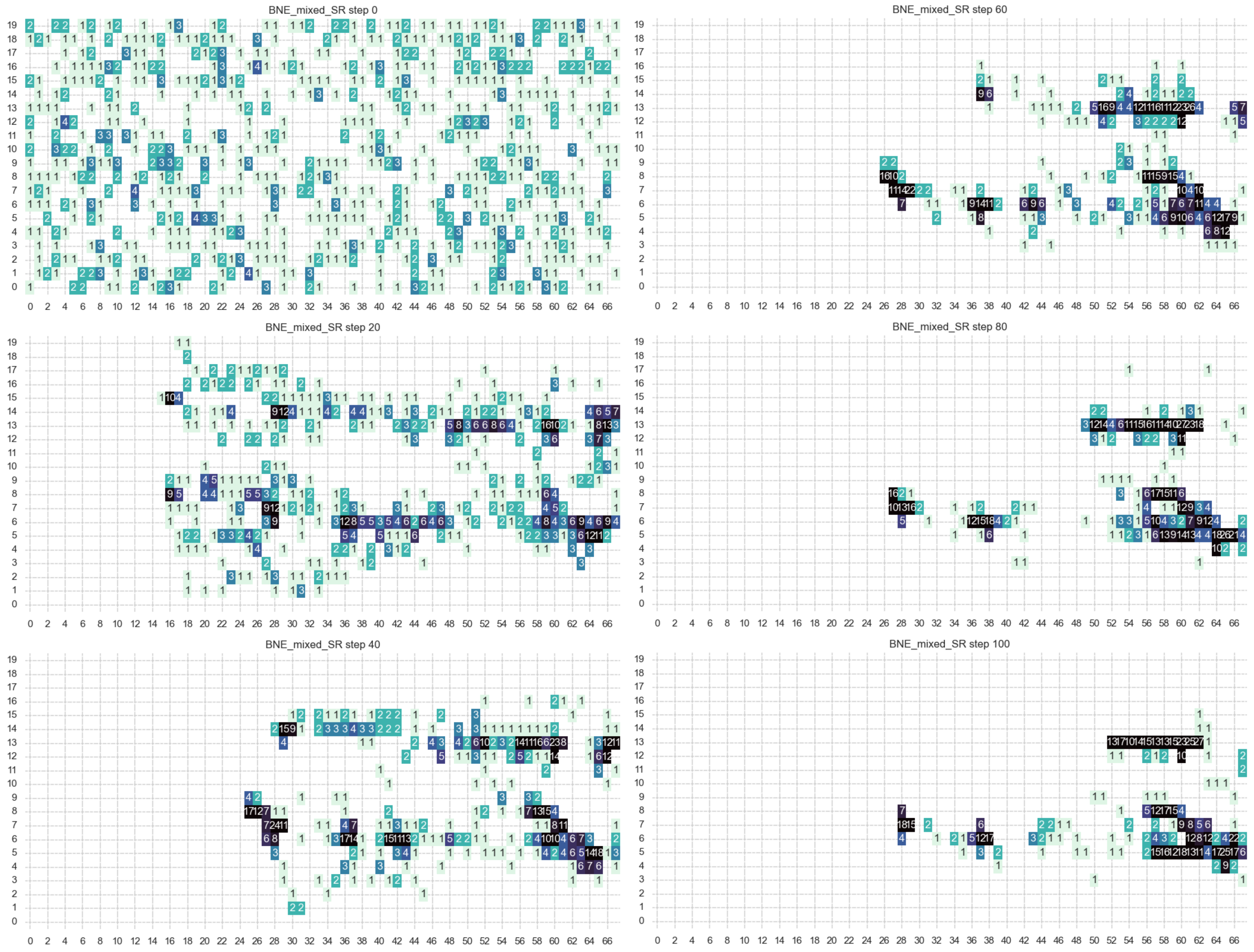

BNE_mixed_RF step 0
BNE_mixed_RF step 60
BNE_mixed_RF step 20
BNE_mixed_RF step 80
BNE_mixed_RF step 40
BNE_mixed_RF step 100

BNE, snake, left_half

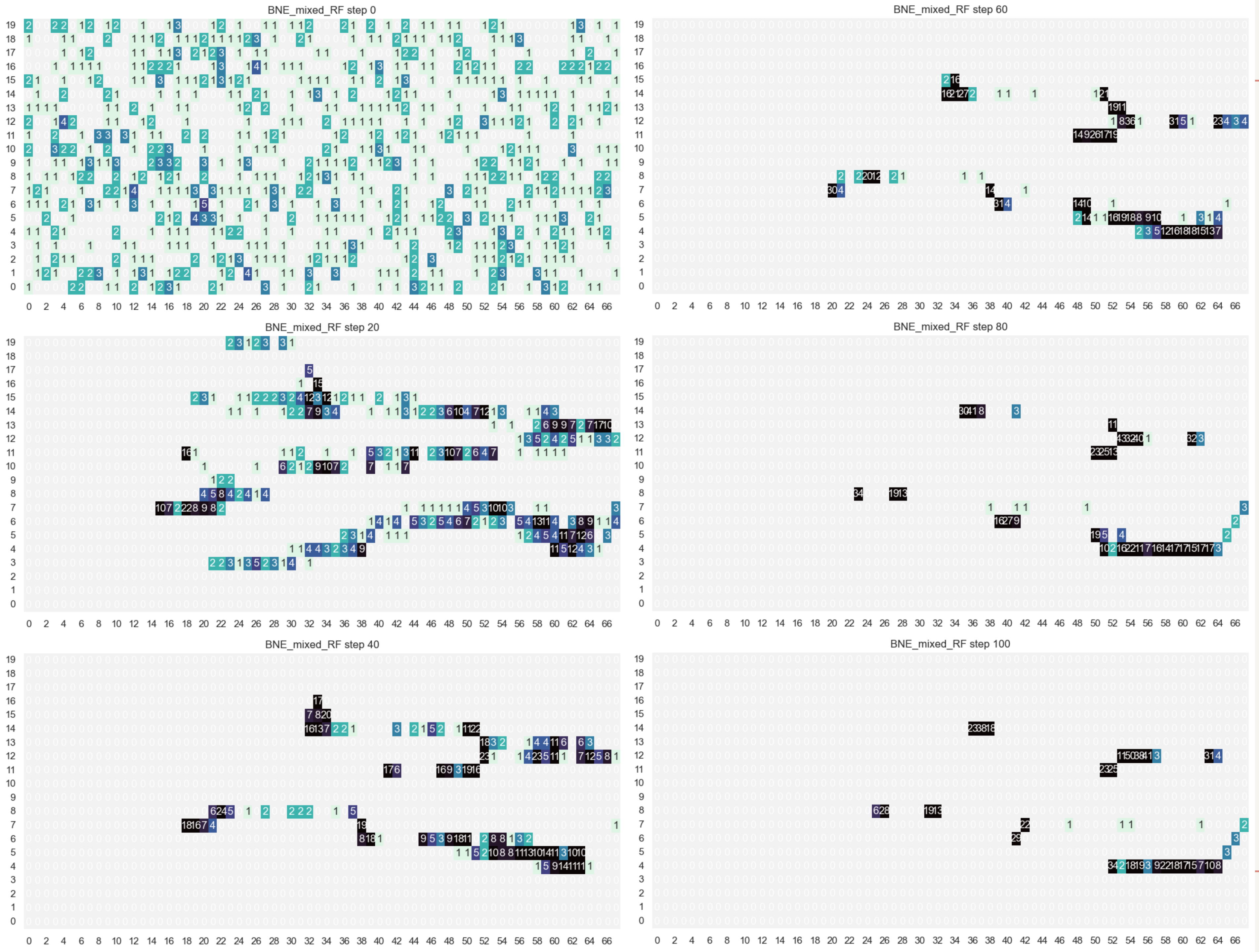


Klatki z symulacji:
Map_type: random_squares

BNE, random_squares

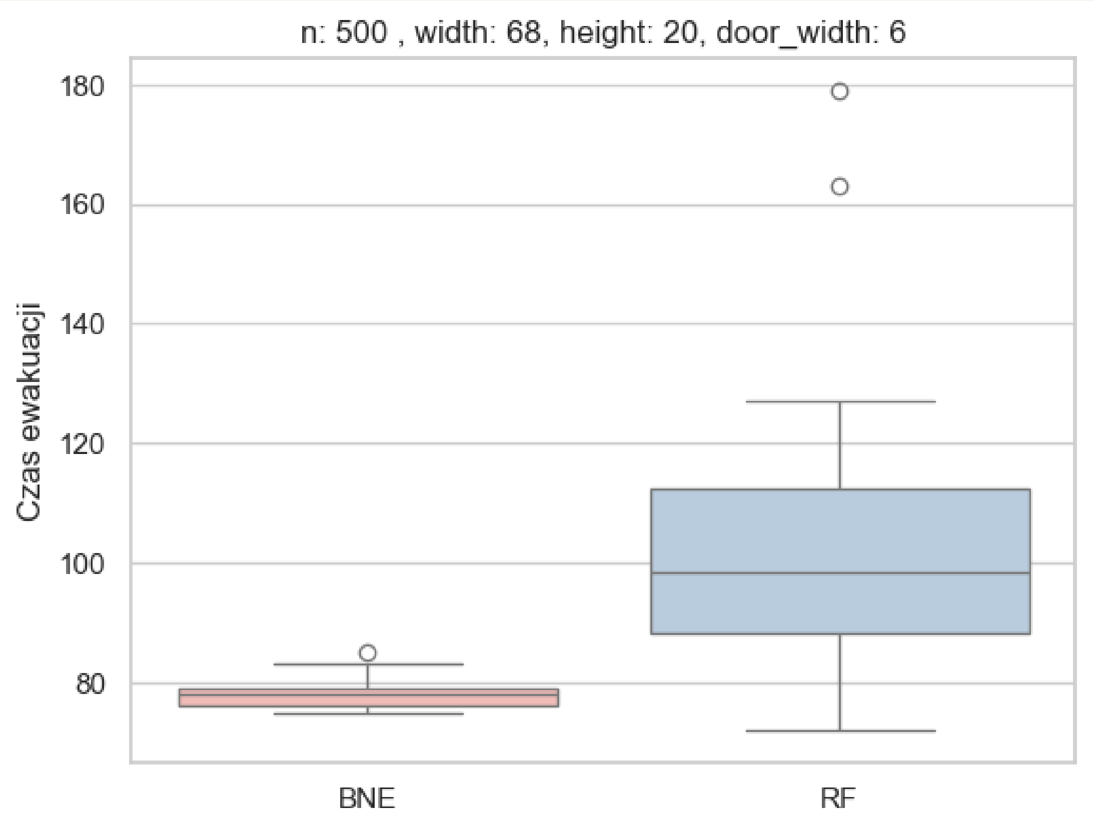


RF, random_squares

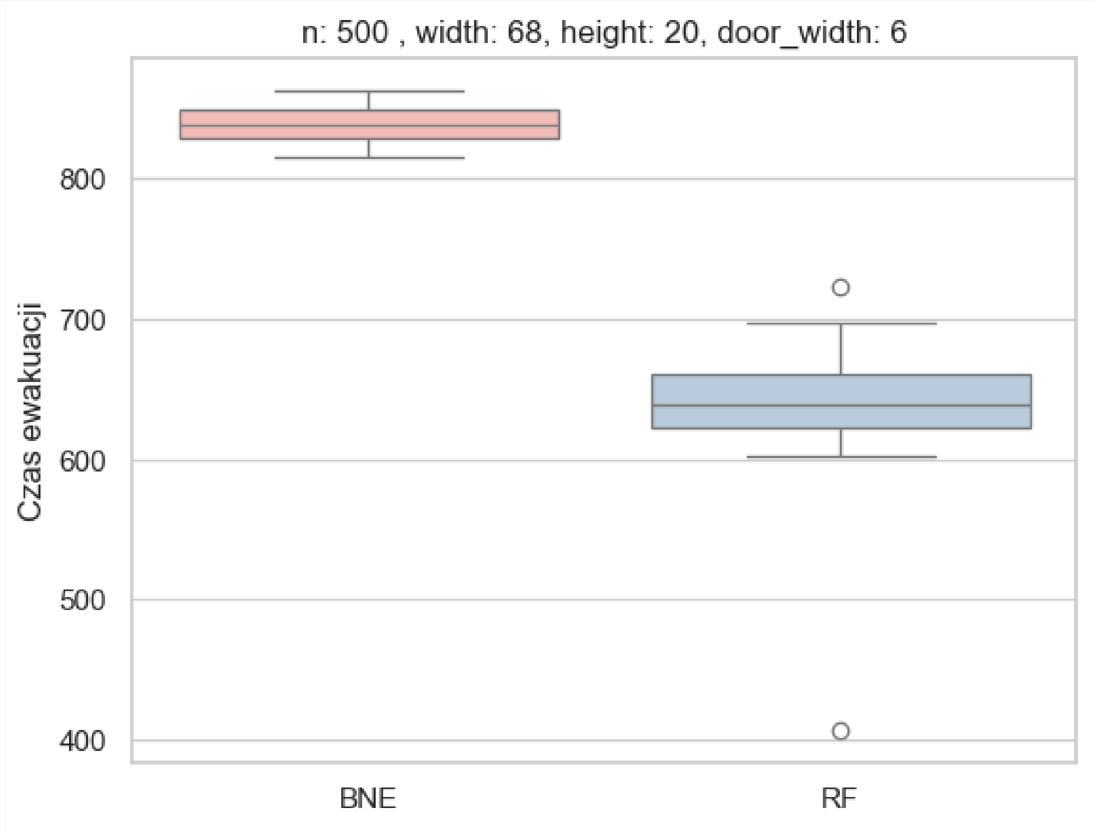


Wyniki oraz wnioski

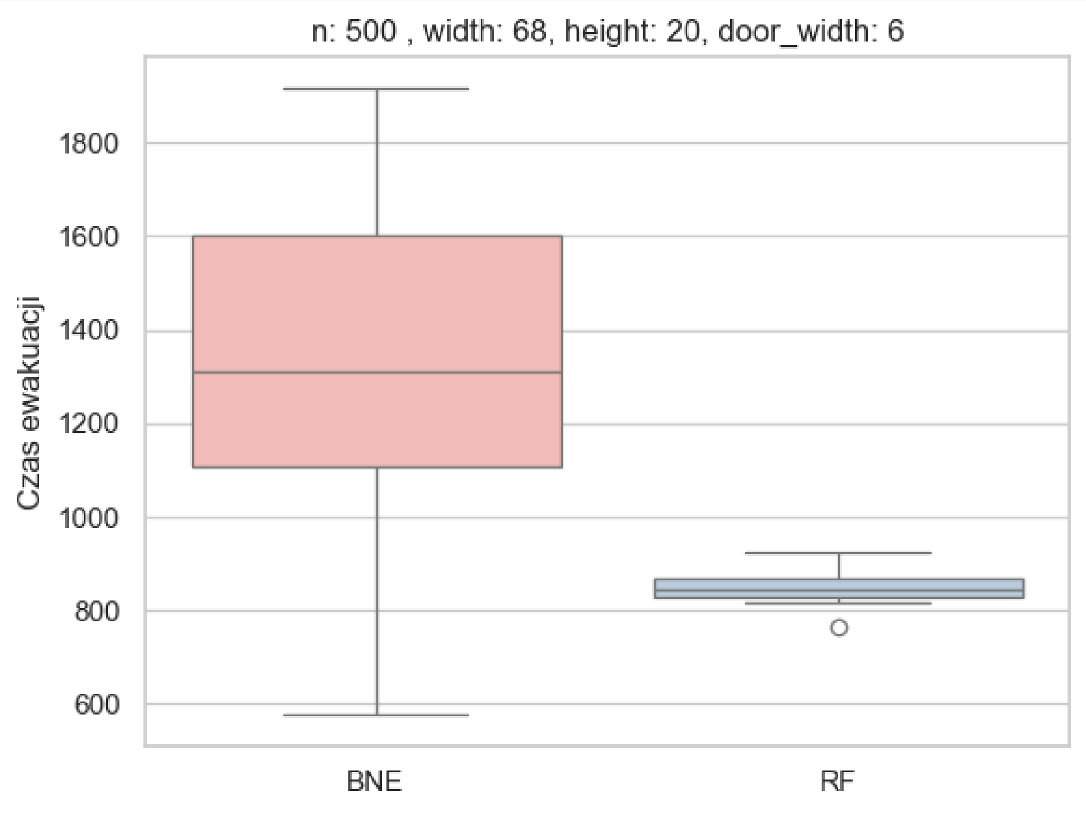
Map_type: empty



Map_type: two_blocks



Map_type: snake

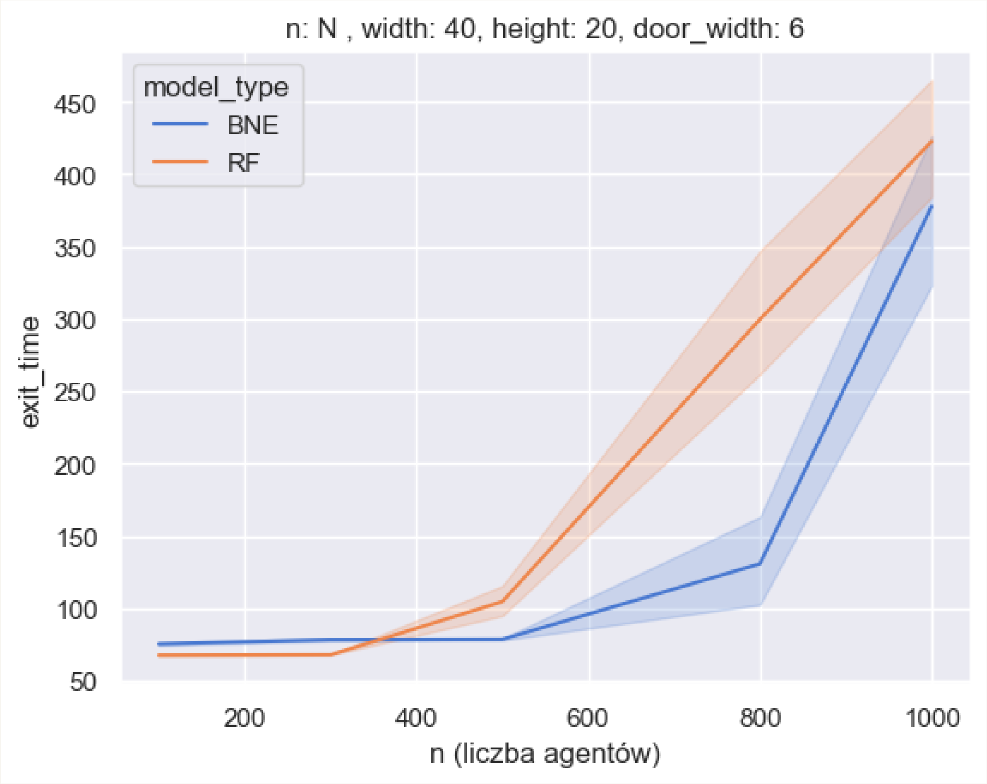


Wnioski:

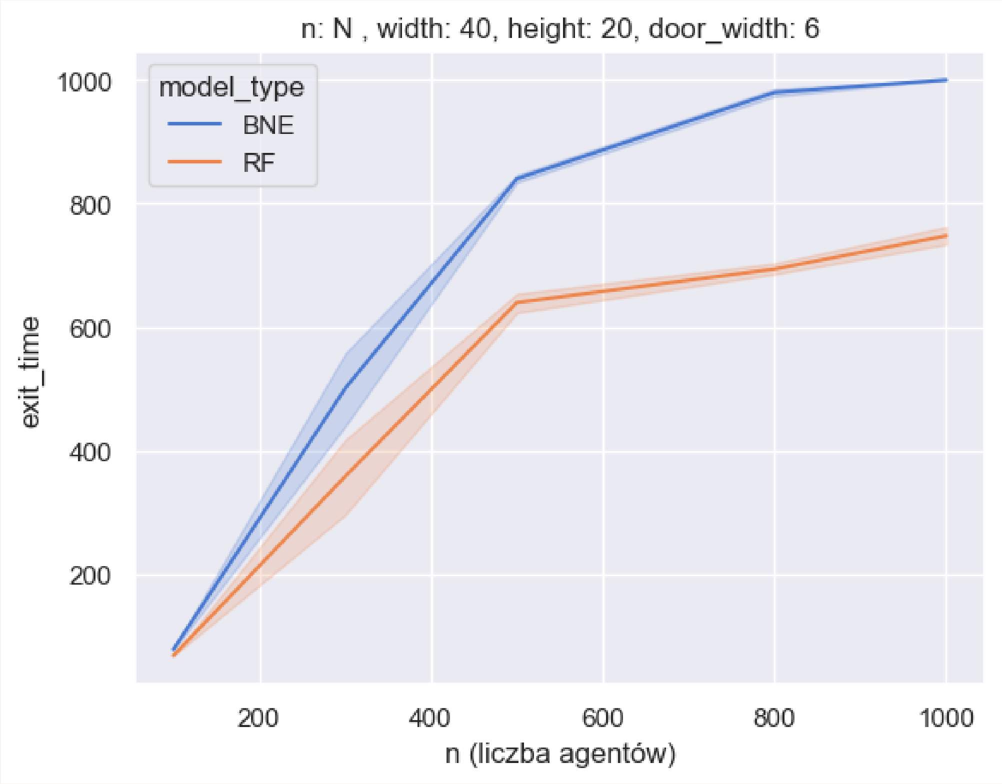
- Na trasie z przeszkodami lepszy jest RF
- RF jest bardziej stabilny

Porównanie czasów ewakuacji

Map_type: empty



Map_type: two_blocks



Dziękujemy za uwagę!

Źródła

- Liao, Guo, Zhu, Shang - *Enhancing emergency pedestrian safety through flow rate design: Bayesian-Nash Equilibrium in multi-agent system*, Computers & Industrial Engineering, Volume 137, 2019
- Wang, Y., Ge, J., & Comber, A. (2023). An Agent-Based Simulation Model of Pedestrian Evacuation based on Bayesian Nash Equilibrium. Journal of Artificial Societies and Social Simulation, 26(3), 6.
- Wang, Y., Ge, J. and Comber, A. (Accepted: 2025) Modelling emergent pedestrian evacuation behaviors from intelligent, game-playing agents. Journal of Computational Social Science. ISSN 2432-2717