**Списъкът на съседите, ако разглеждаме Марковската верига като граф, е:**

(през цялото време за групите ще използвам стойности от 0 до 11, за да е консистентно с индексацията на масива, т.е. I = {0, …., 11} )

0 -> 0, 2, 3, 4, 9
1 -> 0, 3, 4, 5, 10
2 -> 1, 4, 5, 6, 11
3 -> 2, 5, 6, 7, 11
4 -> 3, 6, 7, 8, 11
5 -> 4, 7, 8, 9, 11
6 -> 5,  8, 9, 10, 11
7 -> 6,  9, 10, 11
8 -> 7, 10, 11
9 -> 8, 10, 11
10 -> 9, 11
11 -> 10

В кода се конструира преходната матрица transitionMatrix и след извеждането ѝ имаме:

Transition matrix:
0
 :
[0.918, 0.0, 0.04, 0.03, 0.01, 0.0, 0.0, 0.0, 0.0, 0.002, 0.0, 0.0]
1
 :
[0.918, 0.0, 0.0, 0.04, 0.03, 0.01, 0.0, 0.0, 0.0, 0.0, 0.002, 0.0]
2
 :
[0.0, 0.918, 0.0, 0.0, 0.04, 0.03, 0.01, 0.0, 0.0, 0.0, 0.0, 0.002]
3
 :
[0.0, 0.0, 0.918, 0.0, 0.0, 0.04, 0.03, 0.01, 0.0, 0.0, 0.0, 0.002]
4
 :
[0.0, 0.0, 0.0, 0.918, 0.0, 0.0, 0.04, 0.03, 0.01, 0.0, 0.0, 0.002]
5
 :
[0.0, 0.0, 0.0, 0.0, 0.918, 0.0, 0.0, 0.04, 0.03, 0.01, 0.0, 0.002]
6
 :
[0.0, 0.0, 0.0, 0.0, 0.0, 0.918, 0.0, 0.0, 0.04, 0.03, 0.01, 0.002]
7
 :
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.918, 0.0, 0.0, 0.04, 0.03, 0.012]
8
 :
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.918, 0.0, 0.0, 0.04, 0.042]
9

:
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.918, 0.0, 0.0, 0.082]
10
:
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.918, 0.0, 0.082]
11
:
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.918, 0.082]

**Анализираме кои състояния си комуникират:**

0 -> 2 -> 1 -> 0 => 0, 1 и 2 образуват комуникиращ клас // в частност за 3 стъпки можем да стигнем от 0 до 0 **(1)**
0 -> 3 -> 2 -> 1 -> 0 => и 3 заедно с предходните състояния образува комуникиращ клас // в частност за 4 стъпки можем да стигнем от 0 до 0 **(2)**
0 -> 4 -> 3 => и 4 заедно с предходните състояния образува комуникиращ клас (включихме го в цикъла)
1 -> 5 -> 4 => и 5 заедно с предходните състояния образува комуникиращ клас
2 -> 6 -> 5 => и 6 заедно с предходните състояния образува комуникиращ клас
3 -> 7 -> 6 => и 7 заедно с предходните състояния образува комуникиращ клас
4 -> 8 -> 7 => и 8 заедно с предходните състояния образува комуникиращ клас
5 -> 9 -> 8 => и 9 заедно с предходните състояния образува комуникиращ клас
6 -> 10 -> 9 => и 10 заедно с предходните състояния образува комуникиращ клас
7 -> 11 -> 10 => и 11 заедно с предходните състояния образува комуникиращ клас

Тоест цялото множество от състояния образува комуникиращ клас => веригата е **неразложима**.
В крайна неразложима верига всяко състояние е възвратно => веригата е **възвратна**.
В крайна възвратна верига всяко възвратно състояние е положително-възвратно => веригата е **положително-възвратна**.
**От (1) и (2)** => периодът на 0 е gcd(3, 4) = 1 и веригата е неразложима => това е периодът на цялата верига => веригата е **апериодична**.

Неразложима и положително-възвратна има единствено стационарно разпределение $\pi=(\pi_{0,}\pi_{1,}\pi_{2,}\ldots,\pi_{11})$. Намираме го от $\pi P=\pi$ и $\pi_0+\pi_1+\ldots+\pi_{11}=1$, като, за да получим решение с numpy.linalg(a, b), трябва да подаваме определена система (а $\pi P=\pi$ не е такава), затова подменяме едно от уравненията, които дава условието $\pi P=\pi$ (нека подменим последното), с условието $\pi_0+\pi_1+\ldots+\pi_{11}=1$ и така получаваме стационарния вектор:

Stationary distribution:
[ 0.75289756 0.06725229 0.07325957 0.04699746 0.02366061 0.01218267
0.00809636 0.0054547 0.00412593 0.00348582 0.00152192 0.00106511 ]

**За застрахователния проблем**

Търсим ограничена монотонно растяща функция $f: I \to R$, такава че $E_\pi f(X_1)=1$, тоест $\sum \pi_i f(i)=1$ (3)

Ще се опитаме да намерим решение, при което стимулиращите коефициенти за първите 5 групи са съответно 0.75, 0.82, 0.88, 0.94 и 1 – приблизително на равни разстояния едно от друго.

Тъй като целта ни е да е изпълнено (3), трябва да успеем да нагласим останалите 7 коефициента, така че едновременно да удовлетворяват (3) и да бъдат нарастващи. Може да продължим да слагаме нарастващи стойности през равен интервал (примерно през 2.0 с идеята наказанието да е по-голямо), но проблемът идва от това, че поне един от коефициентите ни ще бъде „вързан“ за останалите - трябва да му определим стойността от (3) чрез останалите 11 вече фиксирани коефициента. Ако оставим това да е последния коефициент, може да се окаже, че неговата стойност пада под стойността на предпоследния и то доста, и така да нарушим нарастването. Затова след поредица от проби-грешки стигнах до извода, че е добре да изведем от (3) някой от **средните коефициенти** – така успях да получа коефициенти за следващите 7 групи – ~5.4, 6, 7, 9, 11, 13, 15 – по-детайлно може да се види в кода.

Bonus malus (solution 1) coefficients
[0.75, 0.82, 0.88, 0.94, 1.0, 5.405214446273115, 6.0, 7.0, 9.0, 11.0, 13.0, 15.0]

Verify E_pi f(X_1) =
1.0159766911157002

И след това, за да намерим колко години са нужни, докато средният бонус-малус коефициент стане приблизително равен на 1, независимо от това от кое състояние стартираме, можем да подходим по следния начин:

за всяко i, намираме минималното n, такова че $\left(\delta_i P^n\right).f \epsilon \left(1-\epsilon, 1+\epsilon\right)$ (за epsilon съм избрал 0.1).

(тук разглеждаме f като 12-елементен вектор-ред и операцията . е скаларно произведение)

Резултатите са:

Expected number of years for each state:

[4, 3, 1, 4, 10, 13, 16, 18, 19, 21, 22, 23]

Отделно правя и симулация на процеса, като фиксиран брой пъти го стартирам от всяко едно състояние и търся първата година, в която средното аритметично на бонус-малус коефициентите до момента попада в интервала $\left(1-\epsilon, 1+\epsilon\right)$. Накрая намирам средното-аритметично на намерените години от всички „проигравания“ на процеса. Резултатите от симулацията са:

Average number of years for each state (simulated):

[7.793792240300376, 6.379, 4.934723304754482, 1.8622405215302797, 1.4489170021084914, 27.140647378974506, 25.724591890468666, 24.85006195786865, 24.005540166204987, 38.661918328584996, 23.542056074766354, 31.62406015037594]

Доста са различни, но си го обяснявам с това, че около 10000 стартирания от всяко състояние не е достатъчно, за да е правдоподобно.

**Второ решение с по-малки отстъпки и по-малки наказания:**

Bonus malus (solution 2) coefficients:
[0.96, 0.97, 0.98, 0.99, 1.0, 1.0687749010853413, 1.7999999999999998, 2.1999999999999997, 2.5999999999999996, 2.9999999999999996, 3.3999999999999995, 3.7999999999999994]

Verify E_pi f(X_1) =
1.0

Expected number of years for each state:
[1, 1, 1, 1, 1, 4, 7, 9, 11, 12, 14, 15]

**Анализ на скоростта на сходимост:**

Eigenvalues:
[-0.48637309+0.j -0.41038302+0.3348473j -0.41038302-0.3348473j
 -0.12840305+0.5773754j -0.12840305-0.5773754j -0.00106203+0.5825243j
 -0.00106203-0.5825243j 1. +0.j **0.78278446+0.j**
 -0.03811259+0.j 0.48726283+0.j 0.33413458+0.j ]

Скоростта на сходимост се определя от втората по големина собствена стойност. В случая това е 0.78278446. Тоест скоростта на сходимост към стационарност е същата като скоростта на сходимост на $0.78278446^n$ към 0.