VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

FINANSŲ IR DRAUDIMO MATEMATIKOS MAGISTRANTŪROS STUDIJŲ PROGRAMA

Magistro baigiamasis darbas

Jungtinis dviejų žalų diskretaus laiko rizikos modelis su premija lygia dviem

Bi-risk discrete time risk model with premium rate two

Artur Nakliuda
artur.nakliuda@mif.stud.vu.lt

Darbo vadovas: **doc. dr. Andrius Grigutis**

Vilnius
2022

Turinys

Santrauka/Abstract

Jungtinis dviejų žalų diskretaus laiko rizikos modelis su premija lygia dviem

Santrauka

Šiame darbe nagrinėjamas jungtinis dviejų žalų diskretaus laiko rizikos modelis su premija lygia dviem. Tiriamojo modelio žalos yra nepriklausomi ir neneigiami diskretieji sveikareikšmiai atsitiktiniai dydžiai, pasirodantys laike tokia tvarka: X, $X+Y$, X, $X+Y$, $\ldots$. Šį rizikos modelį, su premija lygia vienam, 2015 metais nagrinėjo ir A. Grigutis, A. Korvel bei J. Šiaulys, o analogišką rizikos modelį, tik su kitokia žalų išsidėstymo tvarka X, Y, X, Y, $\ldots$, 2021 metais nagrinėjo A. Alencenovič ir A. Grigutis. Šio darbo rezultatas yra rekursinės formulės skirtos baigtinio ir begalinio laiko išgyvenamumo tikimybėms apskaičiuoti. Šiame darbe taip pat pateikiami pavyzdžiai iliustruojantys išvestų formulių taikymą.

Raktiniai žodžiai : jungtinis rizikos modelis; diskretaus laiko rizikos modelis; baigtinio laiko išgyvenamumo tikimybė; begalinio laiko išgyvenamumo tikimybė; rekursinės formulės.

Bi-risk discrete time risk model with premium rate two

Abstract

This work focuses on investigating the bi-risk discrete time risk model with premium rate two. The considered risk model has claim amounts consisting of independent and nonnegative discrete integer valued random variables which occur in such order: X, $X+Y$, X, $X+Y$, $\ldots$. The same model with income rate equal to one was investigated in 2015 by A. Grigutis, A. Korvel and J. Šiaulys. Furthermore, the same model but with different order in which the random variables representing the claim amounts occur X, Y, X, Y, $\ldots$ was investigated in 2021 by A. Alencenovič and A. Grigutis. In this article, a recursive formula for calculation of both finite and ultimate time survival probabilities in the case of bi-risk model is presented. Additionally, several numerical examples illustrating the application of the derived formulas are presented.

Key words : bi-risk risk model; discrete time risk model; finite time survival probability; ultimate time survival probability; recursive formula.

1 Įvadas

Vienas iš šio darbo siekių yra susipažinimas su diskretaus laiko rizikos modeliais bei jų išgyvenamumo tikimybių išvedimo būdais, dėl to, kaip tyrimo pagrindas paimti keli anksčiau parašyti darbai: 2015 metų A. Grigučio, A. Korvel ir J. Šiaulio [2] bei 2021 metų A. Alencenovič ir A. Grigučio [3]. Šiame darbe laikomės panašios struktūros kaip A. Alencenovič ir A. Grigučio straipsnis [3], o idėjiškai tęsiama jungtinio dviejų žalų diskretaus laiko rizikos modelio analizė pateikta A. Grigučio, A. Korvel ir J. Šiaulio parašytame straipsnyje [2]. Šiame darbe tyrinėjamas minėtuose straipsniuose apibrėžtų diskretaus laiko rizikos modelių atskiras atvejis, jam gaunamos naujos išgyvenamumo tikimybių išraiškos, taip siekiant patvirtinti naudotų metodų teisingumą bei prisidėti prie krypties ieškojimo norint tirti bendresnius modelius.

Diskretaus laiko ir panašiems rizikos modeliams aprašyti tipiškai naudojamas atsitiktinis klajojimas, siekiant apskaičiuoti tikimybes, kad atsitiktinio klajojimo reikšmės viršys tam tikrą slenkstį. Šiame darbe toks slenkstis yra $u + \kappa t$, o atsitiktinį klajojimą apibrėžiame kaip atsitiktinių dydžių sumą $\sum_{i=1}^{t} Z_i$, $t \in \mathbb{N}$. Tuomet, jungtinis dviejų žalų rizikos modelis $W(t)$ su apibendrinta premija apibrėžiamas taip

$$W(t) = u + \kappa t - \sum_{i=1}^{t} Z_i, \tag{1}$$

kai

- $t, \kappa \in \mathbb{N}$ ir $u \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$,
- $Z_{2i-1} \overset{d}{=} X$, $Z_{2i} \overset{d}{=} X + Y$, visiems $i \in \mathbb{N}$, čia X, Y - nepriklausomi sveikieji neneigiami atsitiktiniai dydžiai.

Taip apibrėžti bei kiti panašūs ir nebūtinai diskretūs rizikos modeliai, pavyzdžiui klasikinis Sparre Andersen modelis [1], dažnai naudojami draudime, kai $W(t)$ suprantamas kaip draudiko turtas momentu t, parametras $u \in \mathbb{N}_0$ yra pradinis turtas, $\kappa \in \mathbb{N}$ - premija arba kitaip pajamos, o atsitiktiniai dydžiai Z_i laikomi atsitiktinai patiriamomis žalomis momentais $i \in \mathbb{N}$. Pagrindinis klausimas, rūpintis tiriant tokius atsitiktinius procesus, modeliuojamus pagal (1), tai ar deterministinė modelio dalis $u + \kappa t$ yra didesnė už atsitiktinę dalį $\sum_{i=1}^{t} Z_i$ visiems natūraliesiems t iki tam tikro $T \in \mathbb{N}$ arba net $T \to \infty$. Tam tikrame finansiniame kontekste tai gali būti suprantama, kaip vertinimas, kokia tikimybė, jog pradinio turto ir gaunamų premijų užteks padengti patiriamoms žaloms. Tokias tikimybes apibrėžiame taip

$$\varphi(u, T) := \mathbb{P}\left(\bigcap_{t=1}^{T} \{W(t) > 0\}\right) = \mathbb{P}\left(\max_{1 \leqslant t \leqslant T} \left\{\sum_{i=1}^{t} (Z_i - \kappa)\right\} < u\right), \tag{2}$$

$$\varphi(u) := \mathbb{P}\left(\bigcap_{t=1}^{\infty} \{W(t) > 0\}\right) = \mathbb{P}\left(\sup_{t \geqslant 1} \left\{\sum_{i=1}^{t} (Z_i - \kappa)\right\} < u\right), \tag{3}$$

kai pirmoji $\varphi(u, T)$ vadinama baigtinio laiko išgyvenamumo tikimybe, o antroji $\varphi(u)$ begalinio laiko išgyvenamumo tikimybe. Išgyvenamumo tikimybės $\varphi(u, T)$ apskaičiavimo būdai yra daug paprastesni nei $\varphi(u)$. Dėl $\varphi(u)$ sudėtingumo, kai κ yra bet koks natūrinis skaičius, nagrinėsime modelį (1), kai $\kappa = 2$. Atvejis, kai $\kappa = 1$ buvo aprašytas [2], o atvejis $\kappa = 2$, su kitokiu žalų išsidėstymu, aprašytas [3]. Net tokie maži pakeitimai gali turėti reikšmingą poveikį begalinio laiko išgyvenamumo tikimybių išraiškoms. Pavyzdžiui, jau minėtame darbe [2], kai $\kappa = 1$, reikėjo tirti du galimus žalų Z_i pasiskirstymo atvejus, lemiančius begalinio laiko išgyvenamumo tikimybių formas, tačiau, šiame darbe, perėjus prie $\kappa = 2$, tokių atvejų skaičius išaugo iki 6. Didinant κ toliau, reikalingų tirti atvejų skaičius reikšmingai auga, kas stipriai apsunkina apibendrintų $\varphi(u)$ formų išvedimą visoms $\kappa \in \mathbb{N}$.

Toliau parodysime iš kur atsiranda skirtingos $\varphi(u)$ išraiškos. Bet kokiam $u \in \mathbb{N}_0$ apsibrėžiame

$$x_u := \mathbb{P}(X = u),\ y_u := \mathbb{P}(Y = u),\ s_u := \mathbb{P}(X + Y = u),\ a_u := \mathbb{P}(X + X + Y = u),$$
$$X(u) := \sum_{i=0}^{u} x_i,\ Y(u) := \sum_{i=0}^{u} y_i,\ S(u) := \sum_{i=0}^{u} s_i,\ A(u) := \sum_{i=0}^{u} a_i,$$
$$\overline{X}(u) := 1 - X(u),\ \overline{Y}(u) := 1 - Y(u),\ \overline{S}(u) := 1 - S(u),\ \overline{A}(u) := 1 - A(u).$$

Naudojantis formule $\mathbb{P}(A \cap B) = \mathbb{P}(A) - \mathbb{P}(A \cap B^c)$, pertvarkymais ir kitais elementariais tikimybių teorijos metodais, diskrečiam dviejų žalų rizikos modeliui su premija lygia dviem, gauname

$$\begin{aligned}
\varphi(u) &= \mathbb{P}\left(\bigcap_{t=1}^{\infty}\left\{u + 2t - \sum_{i=1}^{t} Z_i > 0\right\}\right) = \mathbb{P}\left(\bigcap_{t=2}^{\infty}\left\{u + 2t - \sum_{i=1}^{t} Z_i > 0\right\},\ Z_1 < u + 2\right)\\
&= \mathbb{P}\left(\bigcap_{t=2}^{\infty}\left\{u + 2t - \sum_{i=1}^{t} Z_i > 0\right\}\right) - \mathbb{P}\left(\bigcap_{t=2}^{\infty}\left\{u + 2t - \sum_{i=1}^{t} Z_i > 0\right\},\ Z_1 \geqslant u + 2\right)\\
&= \mathbb{P}\left(\bigcap_{t=3}^{\infty}\left\{u + 2t - Z_1 - Z_2 - \sum_{i=3}^{t} Z_i > 0\right\},\ Z_1 + Z_2 < u + 4\right)\\
&- \mathbb{P}\left(\bigcap_{t=3}^{\infty}\left\{u + 2t - Z_1 - Z_2 - \sum_{i=3}^{t} Z_i > 0\right\},\ Z_1 \geqslant u + 2,\ Z_1 + Z_2 < u + 4\right)\\
&= \sum_{k=0}^{u+3} \mathbb{P}\left(\bigcap_{t-2=1}^{\infty}\left\{u + 2(t-2) + 4 - Z_1 - Z_2 - \sum_{i=1}^{t-2} Z_i > 0\right\},\ Z_1 + Z_2 = k\right)\\
&- (x_{u+3}s_0 + x_{u+2}s_1)\mathbb{P}\left(\bigcap_{t=1}^{\infty}\left\{1 + 2t - \sum_{i=1}^{t} Z_i > 0\right\}\right) - x_{u+2}s_0\mathbb{P}\left(\bigcap_{t=1}^{\infty}\left\{2 + 2t - \sum_{i=1}^{t} Z_i > 0\right\}\right)\\
&= \sum_{k=0}^{u+3} \varphi(u + 4 - k)a_k - (x_{u+3}s_0 + x_{u+2}s_1)\varphi(1) - x_{u+2}s_0\varphi(2)\\
&= \sum_{k=1}^{u+4} \varphi(k)a_{u+4-k} - (x_{u+3}s_0 + x_{u+2}s_1)\varphi(1) - x_{u+2}s_0\varphi(2).
\end{aligned}$$

Gauta lygybė

$$\varphi(u) = \sum_{k=1}^{u+4} \varphi(k)a_{u+4-k} - (x_{u+3}s_0 + x_{u+2}s_1)\varphi(1) - x_{u+2}s_0\varphi(2) \tag{4}$$

yra vienas iš esminių šio darbo tyrinėjimo objektų. Įrašydami $u = 0, 1, \ldots, n-4$ į (4) gauname lygčių sistemą

$$\begin{cases}
a_0\varphi(4) = \varphi(0) + (-a_3 + x_3s_0 + x_2s_1)\varphi(1) + (-a_2 + x_2s_0)\varphi(2) - a_1\varphi(3),\\
a_0\varphi(5) = (1 - a_4 + x_4s_0 + x_3s_1)\varphi(1) + (-a_3 + x_3s_0)\varphi(2) - a_2\varphi(3) - a_1\varphi(4),\\
\vdots\\
a_0\varphi(n) = (x_{n-1}s_0 + x_{n-2}s_1)\varphi(1) + x_{n-2}s_0\varphi(2) + \varphi(n-4) - \sum\limits_{k=1}^{n-1} \varphi(k)a_{n-k}.
\end{cases} \tag{5}$$

Gauta lygčių sistema (5) leidžia išreikšti $\varphi(n)$ per $\varphi(0), \ldots, \varphi(3)$, kai $n = 4, 5, \ldots$. Tad pagrindinė užduotis, su kuria susiduriame, yra pradinių reikšmių $\varphi(0), \ldots, \varphi(3)$ radimas. Kadangi lygybės (5) yra priklausomos nuo atsitiktinių dydžių X ir Y sąsūkos $A := X + X + Y$ skirstinio, tai ir pradinių φ reikšmių išraiškos yra priklausomos nuo A struktūros. Šiame darbe nenagrinėjame bendresnių modelio (1) atvejų nei pasirinktomis sąlygomis: $\kappa = 2$ ir $Z_{2i-1} \stackrel{d}{=} X$, $Z_{2i} \stackrel{d}{=} X + Y$, $i \in \mathbb{N}$. Neapsiribojus minėtomis sąlygomis, begalino laiko išgyvenamumo tikimybių išraiškų išvedimas tampa labai abstraktus ir sudėtingas. Tad šis ir [2] bei [3] darbai gali būti tik pradiniai žingsniai patvirtinantys metodo teisingumą bei suteikiantys kryptį bandymams nagrinėti bendrą modelio (1) atvejį, kai κ yra bet koks natūrinis skaičius.

Kaip jau minėjome anksčiau, norėdami rasti $\varphi(n)$ išraišką visiems $n \in \mathbb{N}_0$, visų pirma turime rasti pradines reikšmes $\varphi(0), \ldots, \varphi(3)$. Šių nežinomųjų skaičių galime sumažinti vienu, pasinaudodami toliau pateikta begalinio laiko išgyvenamumo tikimybės savybe (11), kuri galioja, kuomet tenkinama grynojo pelno sąlyga. Sakome, kad nagrinėjamas modelis (1), kai $\kappa = 2$, tenkina grynojo pelno sąlygą, kai $\mathbb{E}A < 4$. Ši sąlyga yra būtina, norint išvengti garantuoto bankroto (išgyvenamumo tikimybės lygios nuliui) laikui augant į begalybę, žiūrėti teoremą 6 skyriuje 2. Intuityviai suprasti grynojo pelno sąlygą galima pažiūrėjus į nagrinėjamo modelio (1) vidutinio turto $\mathbb{E}W(t)$ išraišką lyginiams ir nelyginiams $t \in \mathbb{N}$

$$\begin{aligned}
\mathbb{E}W(2t) &= u + t(2\kappa - \mathbb{E}A), \\
\mathbb{E}W(2t-1) &= u - \kappa + \mathbb{E}S + t(2\kappa - \mathbb{E}A).
\end{aligned}$$

Šiose išraiškose akivaizdžiai matosi, jog nuo $2\kappa - \mathbb{E}A$ ženklo priklauso $\mathbb{E}W(t)$ ženklas bei $W(t)$ galimybė įgyti neteigiamas reikšmes, tam tikriems $t \in \mathbb{N}$.

Toliau parodysime prieš tai minėtą begalinio laiko išgyvenamumo tikimybės savybę (11). Sumuodami abi lygybės (4) puses, gauname

$$\begin{aligned}
\sum_{u=0}^{v} \varphi(u) = \sum_{u=0}^{v} \sum_{k=1}^{u+4} \varphi(k) a_{u+4-k} - \left((X(v+3) - X(2)) s_0 + (X(v+2) - X(1)) s_1 \right) \varphi(1) \\
- (X(v+2) - X(1)) s_0 \varphi(2), \qquad (6)
\end{aligned}$$

čia

$$\begin{aligned}
\sum_{u=0}^{v} \sum_{k=1}^{u+4} \varphi(k) a_{u+4-k} &= \sum_{k=1}^{3} \varphi(k) \sum_{u=0}^{v} a_{u+4-k} + \sum_{k=4}^{v+4} \varphi(k) \sum_{u=k-4}^{v} a_{u+4-k} \\
&= \sum_{k=1}^{3} \varphi(k) \left(A(v+4-k) - A(3-k) \right) + \sum_{k=4}^{v+4} \varphi(k) A(v+4-k). \qquad (7)
\end{aligned}$$

Įstatę (7) į (6) ir pertvarkę, gauname

$$\begin{aligned}
\sum_{k=0}^{v+4} \varphi(k) \overline{A}(v+4-k) - \sum_{u=v+1}^{v+4} \varphi(u) = \sum_{k=1}^{3} \varphi(k) \left(A(v+4-k) - A(3-k) \right) - \sum_{k=0}^{3} \varphi(k) A(v+4-k) \\
- \left((X(v+3) - X(2)) s_0 + (X(v+2) - X(1)) s_1 \right) \varphi(1) - (X(v+2) - X(1)) s_0 \varphi(2). \qquad (8)
\end{aligned}$$

Toliau bus reikalinga pagalbinė lema 1, sumoms (8) tirti. Pažymėkime $\varphi(\infty) := \lim_{u \to \infty} \varphi(u)$.

Lema 1. *Jungtinio dviejų žalų rizikos modelio su premija lygia dviem, begalinio laiko išgyvenamumo tikimybei teisingos šios savybės:*

$$\varphi(\infty) = 1, \ \text{jei } \mathbb{E}A < 4, \qquad (9)$$

$$\lim_{v \to \infty} \sum_{k=0}^{v+4} \varphi(k) \overline{A}(v+4-k) = \varphi(\infty) \cdot \mathbb{E}A. \qquad (10)$$

Šios lemos pirmoji lygybė (9) įrodoma analogiškai, kaip [4, p. 935–936], o antroji (10) analogiškai, kaip [5, p. 4] arba [6, p. 13].

Tada, kai $v \to \infty$ ir $\mathbb{E}A < 4$, iš lygybės (8) ir lemos 1, gauname

$$\varphi(0) + \left(\overline{X}(2)s_0 + \overline{X}(1)s_1\right)\varphi(1) + \overline{X}(1)s_0\varphi(2) + \sum_{k=1}^{3}\varphi(k)A(3-k) = 4 - \mathbb{E}A. \tag{11}$$

Turėdami (11) ir (5) lygybes galime išsireikšti $\varphi(n)$ per $\varphi(0), \varphi(1)$ ir $\varphi(2)$ visiems $n \in \mathbb{N}_0$.

2 Rezultatai

Šiame skyriuje suformuluojame teoremas, kuriomis naudojantis galima gauti $\varphi(u,T)$ ir $\varphi(u)$ - baigtinio ir begalinio laiko išgyvenamumo tikimybių išraiškas modeliui (1) su $\kappa = 2$. Pirmoji teorema skirta baigtinio laiko išgyvenamumo tikimybei $\varphi(u,T)$. Išgyvenamumo tikimybės $\varphi(u,T)$ išraiška gali būti gauta naudojantis [7, Teoremos 1–4], tačiau pateikta teorema 1 specialiai pritaikyta nagrinėjamam rizikos modeliui. Vėliau ši teorema bus reikalinga skyriuje 4, pateikiant pavyzdžius baigtinio ir begalinio laiko išgyvenamumo tikimybių palyginimams. Teoremos 2 – 5 skirtos begalinio laiko išgyvenamumo tikimybėms, kai tenkinama grynojo pelno sąlyga $\mathbb{E}A < 4$. Teoremose 2 – 5 aprašytos $\varphi(u)$ išraiškos priklauso nuo $A = X + X + Y$ pasiskirstymo įgyjamų mažiausių reikšmių. Paskutinė teorema 6 skirta parodyti, jog neskaitant kelių trivialių atvejų, begalinio laiko išgyvenamumo tikimybė lygi nuliui, kai grynojo pelno sąlyga netenkinama $\mathbb{E}A \geqslant 4$. Visos šiame skyriuje pateiktos teoremos vėliau įrodomos tolimesniame skyriuje 3.

Teorema 1. *Dviejų žalų rizikos modelio su premija lygia dviem, baigtinio laiko išgyvenamumo tikimybei $\varphi(u,T)$ galioja:*

$$\varphi(u,1) = X(u+1),$$
$$\varphi(u,2) = \sum_{k=0}^{u+1} x_k S(u+3-k),$$
$$\varphi(u,T) = \sum_{k=0}^{u+3}\varphi(u+4-k,T-2)a_k - (x_{u+2}s_1 + x_{u+3}s_0)\varphi(1,T-2) - x_{u+2}s_0\varphi(2,T-2),\ T \geqslant 3.$$

Toliau pereiname prie begalinio laiko. Begalinio laiko išgyvenamumo tikimybės φ išraiškos yra stipriai priklausomos nuo mažiausių sąsūkos $A = X + X + Y$ reikšmių. Apibrėžiame keturias rekurentines sekas α_n, β_n, γ_n, δ_n, kuomet $a_0 > 0$. Kai $n = 0,1,2,3$, tai

n	α_n	β_n	γ_n	δ_n
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	$-1/a_0$	$-\left(A(2) + \overline{X}(2)s_0 + \overline{X}(1)s_1\right)/a_0$	$-\left(A(1) + \overline{X}(1)s_0\right)/a_0$	$1/a_0$

ir, kai $n = 4, 5, \ldots$, tai

$$\alpha_n = \frac{1}{a_0}\left(\alpha_{n-4} - \sum_{k=1}^{n-1} a_{n-k}\alpha_k\right),\ \beta_n = \frac{1}{a_0}\left(\beta_{n-4} - \sum_{k=1}^{n-1} a_{n-k}\beta_k + x_{n-1}s_0 + x_{n-2}s_1\right),$$
$$\gamma_n = \frac{1}{a_0}\left(\gamma_{n-4} - \sum_{k=1}^{n-1} a_{n-k}\gamma_k + x_{n-2}s_0\right),\ \delta_n = \frac{1}{a_0}\left(\delta_{n-4} - \sum_{k=1}^{n-1} a_{n-k}\delta_k\right).$$

Teorema 2. *Jei $a_0 > 0$ ir $\mathbb{E}A < 4$, jungtinio dviejų žalų rizikos modelio su premija lygia dviem, begalinio laiko išgyvenamumo tikimybei galioja*

$$\begin{pmatrix} \alpha_{n+1}-\alpha_n & \beta_{n+1}-\beta_n & \gamma_{n+1}-\gamma_n \\ \alpha_{n+2}-\alpha_n & \beta_{n+2}-\beta_n & \gamma_{n+2}-\gamma_n \\ \alpha_{n+3}-\alpha_n & \beta_{n+3}-\beta_n & \gamma_{n+3}-\gamma_n \end{pmatrix} \times \begin{pmatrix} \varphi(0) \\ \varphi(1) \\ \varphi(2) \end{pmatrix} + \begin{pmatrix} \delta_{n+1}-\delta_n \\ \delta_{n+2}-\delta_n \\ \delta_{n+3}-\delta_n \end{pmatrix} \times (4-\mathbb{E}A)$$
$$= \begin{pmatrix} \varphi(n+1)-\varphi(n) \\ \varphi(n+2)-\varphi(n) \\ \varphi(n+3)-\varphi(n) \end{pmatrix}, \; n \in \mathbb{N}_0, \qquad (12)$$

$$\varphi(3) = \frac{-\varphi(0) - (A(2) + \overline{X}(2)s_0 + \overline{X}(1)s_1)\varphi(1) - (A(1) + \overline{X}(1)s_0)\varphi(2) + 4 - \mathbb{E}A}{a_0},$$
$$\varphi(u) = \frac{1}{a_0}\left(\varphi(u-4) + (x_{u-1}s_0 + x_{u-2}s_1)\varphi(1) + x_{u-2}s_0\varphi(2) - \sum_{k=1}^{u-1} a_{u-k}\varphi(k)\right), \; u = 4, 5, \ldots$$

Pastaba 1. *Begalinio laiko išgyvenamumo tikimybėms galioja $\varphi(n) \leqslant \varphi(n+1) \leqslant 1$ visiems $n \in \mathbb{N}_0$. Be to, iš lemos 1, matome, kad $\varphi(n) \approx 1$, kai n – pakankamai didelis. Dėl šių savybių, praktiniuose teoremos 2 taikymuose, lygčių sistemos (12) dešinę pusę laikome lygia $(0,0,0)^T$ ir reikalingas pradines reikšmes $\varphi(0), \varphi(1)$ ir $\varphi(2)$ apskaičiuojame iš (12).*

Pastaba 2. *Įrodyti, kad sistemos (12) matrica nėra išsigumusi visiems $n \in \mathbb{N}_0$ negalime, tačiau tokios matricos, kuri būtų apibrėžta pagal teoremą 2 ir būtų išsigimusi, rasti nepavyko.*

Toliau nagrinėjame begalinio laiko išgyvenamumo tikimybės φ išraiškas, kai $a_0 = 0$ ir $a_1 > 0$. Apibrėžiame tris rekurentines sekas $\overline{\alpha}_n$, $\overline{\beta}_n$, $\overline{\delta}_n$. Kai $n = 0, 1, 2$, tai

n	$\overline{\alpha}_n$	$\overline{\beta}_n$	$\overline{\delta}_n$
0	1	0	0
1	0	1	0
2	$-1/a_1$	$-(A(2) + \overline{X}(1)s_1)/a_1$	$1/a_1$

ir, kai $n = 3, 4, \ldots$, tai

$$\overline{\alpha}_n = \frac{1}{a_1}\left(\overline{\alpha}_{n-3} - \sum_{k=1}^{n-1} a_{n+1-k}\overline{\alpha}_k\right),$$
$$\overline{\beta}_n = \frac{1}{a_1}\left(\overline{\beta}_{n-3} - \sum_{k=1}^{n-1} a_{n+1-k}\overline{\beta}_k + x_{n-1}s_1\right)$$
$$\overline{\delta}_n = \frac{1}{a_0}\left(\overline{\delta}_{n-3} - \sum_{k=1}^{n-1} a_{n+1-k}\overline{\delta}_k\right).$$

Teorema 3. *Jei $a_0 = 0$, $a_1 > 0$ ir $\mathbb{E}A < 4$, jungtinio dviejų žalų rizikos modelio su premija lygia dviem, begalinio laiko išgyvenamumo tikimybei galioja*

$$\begin{pmatrix} \overline{\alpha}_{n+1}-\overline{\alpha}_n & \overline{\beta}_{n+1}-\overline{\beta}_n \\ \overline{\alpha}_{n+2}-\overline{\alpha}_n & \overline{\beta}_{n+2}-\overline{\beta}_n \end{pmatrix} \times \begin{pmatrix} \varphi(0) \\ \varphi(1) \end{pmatrix} + \begin{pmatrix} \overline{\delta}_{n+1}-\overline{\delta}_n \\ \overline{\delta}_{n+2}-\overline{\delta}_n \end{pmatrix} \times (4-\mathbb{E}A)$$
$$= \begin{pmatrix} \varphi(n+1)-\varphi(n) \\ \varphi(n+2)-\varphi(n) \end{pmatrix}, \; n \in \mathbb{N}_0, \qquad (13)$$

$$\varphi(2) = \frac{-\varphi(0) - (A(2) + \overline{X}(1)s_1)\varphi(1) + 4 - \mathbb{E}A}{a_1},$$
$$\varphi(u) = \frac{1}{a_1}\left(\varphi(u-3) + x_{u-1}s_1\varphi(1) - \sum_{k=1}^{u-1} a_{u+1-k}\varphi(k)\right), \; u = 3, 4, \ldots$$

Pastaba 3. *Teoremos 3 praktinis taikymas yra analogiškas teoremai 2, kuris aprašytas pastaboje 1.*

Pastaba 4. *Įrodyti, kad sistemos (13) matrica nėra išsigumusi visiems $n \in \mathbb{N}_0$ negalime, tačiau tokios matricos, kuri būtų apibrėžta pagal teoremą 3 ir būtų išsigimusi, rasti nepavyko.*

Toliau nagrinėjame begalinio laiko išgyvenamumo tikimybės φ išraiškas, kai $a_0 = 0$, $a_1 = 0$ ir $a_2 > 0$. Apibrėžiame dvi rekurentines sekas $\hat{\alpha}_n$, $\hat{\delta}_n$.

$$\hat{\alpha}_0 = 1,\; \hat{\alpha}_1 = -\frac{1}{\overline{X}(1)s_1 + a_2},\; \hat{\alpha}_n = \frac{1}{a_2}\left(\hat{\alpha}_{n-2} - \sum_{k=1}^{n-1} a_{n+2-k}\hat{\alpha}_k + x_n s_1 \hat{\alpha}_1\right), \; n = 2, 3, \ldots$$
$$\hat{\delta}_0 = 0,\; \hat{\delta}_1 = \frac{1}{\overline{X}(1)s_1 + a_2},\; \hat{\delta}_n = \frac{1}{a_2}\left(\hat{\delta}_{n-2} - \sum_{k=1}^{n-1} a_{n+2-k}\hat{\delta}_k + x_n s_1 \hat{\delta}_1\right), \; n = 2, 3, \ldots$$

Teorema 4. *Kai $a_0 = 0$, $a_1 = 0$, $a_2 > 0$ ir $\mathbb{E}A < 4$, jungtinio dviejų žalų rizikos modelio su premija lygia dviem, begalinio laiko išgyvenamumo tikimybei galioja*

$$(\hat{\alpha}_{n+1} - \hat{\alpha}_n)\varphi(0) + (\hat{\delta}_{n+1} - \hat{\delta}_n)(4 - \mathbb{E}A) = \varphi(n+1) - \varphi(n), \; n \in \mathbb{N}_0, \tag{14}$$

$$\varphi(1) = \hat{\alpha}_1\varphi(0) + \hat{\delta}_1(4 - \mathbb{E}A),$$
$$\varphi(u) = \frac{1}{a_2}\left(\varphi(u-2) - \sum_{k=1}^{u-1} a_{u+2-k}\varphi(k) + x_u s_1 \varphi(1)\right), \; u = 2, 3, \ldots$$

Be to, $\hat{\alpha}_{n+1} - \hat{\alpha}_n \neq 0$ visiems $n \in \mathbb{N}_0$.

Pastaba 5. *Yra du X, Y skirstinių atvejai, išpildantys teoremoje 4 esančias sąlygas $a_0 = 0$, $a_1 = 0$ ir $a_2 > 0$, nuo kurių priklauso φ išraiškos forma.*

- *Pirmu atveju $x_0 > 0$ ir $y_0 = y_1 = 0$, $y_2 > 0$, tada $s_0 = x_0 y_0 = 0$ ir $s_1 = x_1 y_0 + x_0 y_1 = 0$. Šiuo atveju teoremoje 4 esančios išgyvenamumo tikimybės $\varphi(1)$ vardiklis ir anksčiau aprašytų išraiškų $\hat{\alpha}_n$, $\hat{\delta}_n$ vardikliai lygūs $a_2 > 0$ pagal teoremos 4 sąlygas.*
- *Antru atveju $x_0 = 0$, $x_1 > 0$ ir $y_0 > 0$, tada $s_0 = x_0 y_0 = 0$ ir $s_1 = x_1 y_0 + x_0 y_1 = x_1 y_0 > 0$. Šiuo atveju teoremoje 4 esančios išgyvenamumo tikimybės $\varphi(1)$ vardiklis ir anksčiau aprašytų išraiškų $\hat{\alpha}_n$, $\hat{\delta}_n$ vardikliai lygūs $a_2 + \overline{X}(1)s_1 = x_0^2 y_2 + x_1^2 y_0 + (1 - x_0 - x_1)x_1 y_0 = x_1 y_0 > 0$ pagal teoremos 4 sąlygas.*

Pastaba 6. *Teoremos 4 praktinis taikymas yra analogiškas teoremoms 2 ir 3, minimas pastabose 1 ir 3.*

Galiausiai pereiname prie paskutinio φ priklausomybės nuo A atvejo, kai $a_0 = 0$, $a_1 = 0$, $a_2 = 0$ ir $a_3 > 0$.

Teorema 5. *Kai $a_0 = 0$, $a_1 = 0$, $a_2 = 0$, $a_3 > 0$ ir $\mathbb{E}A < 4$, jungtinio dviejų žalų rizikos modelio su premija lygia dviem, begalinio laiko išgyvenamumo tikimybei galioja*

$$\varphi(0) = 4 - \mathbb{E}A,$$
$$\varphi(1) = \frac{\varphi(0)}{a_3},$$
$$\varphi(u) = \frac{1}{a_3}\left(\varphi(u-1) - \sum_{k=1}^{u-1} a_{u+3-k}\varphi(k)\right),\ u = 2, 3, \ldots$$

Lengva pastebėti, kad, kai $a_0 = a_1 = a_2 = a_3 = 0$, grynojo pelno sąlyga netenkinama, nes $\mathbb{E}A \geqslant 4$. Tokiu atveju, teisinga tokia teorema

Teorema 6. *Kai grynojo pelno sąlyga netenkinama, jungtinio dviejų žalų rizikos modelio su premija lygia dviem, begalinio laiko išgyvenamumo tikimybei galioja*

i. $\varphi(u) = 0$ *visiems* $u \in \mathbb{N}_0$*, kai* $\mathbb{E}A > 4$*,*

ii. $\varphi(u) = 0$ *visiems* $u \in \mathbb{N}_0$*, kai* $\mathbb{E}A = 4$ *ir* $a_4 < 1$*,*

iii. $\varphi(0) = 0$ *ir* $\varphi(u) = 1$ *visiems* $u \in \mathbb{N}$*, kai* $\mathbb{E}A = 4$ *ir* $a_4 = 1$*.*

3 Įrodymai

Toliau pateikiame ankstesniame skyriuje 2 suformuluotų teoremų įrodymus.

Teoremos 1 įrodymas. Iš dviejų žalų rizikos modelio su premija lygia dviem baigtinio laiko išgyvenamumo tikimybės apibrėžimo (2) seka, kad

$$\varphi(u, 1) = \mathbb{P}\left(u + 2 - Z_1 > 0\right) = X(u+1).$$

Kai $T = 2$, analogiškai iš apibrėžimo (2) seka, kad

$$\begin{aligned}
\varphi(u, 2) &= \mathbb{P}(\{W(1) > 0\} \cap \{W(2) > 0\}) \\
&= \mathbb{P}(\{Z_1 < u + 2\} \cap \{Z_1 + Z_2 < u + 4\}) \\
&= \mathbb{P}(\{X \leqslant u + 1\} \cap \{X + S \leqslant u + 3\}) \\
&= \sum_{k=0}^{u+1} \mathbb{P}(X = k)\mathbb{P}(S \leqslant u + 3 - k) \\
&= \sum_{k=0}^{u+1} x_k S(u + 3 - k).
\end{aligned}$$

Kai $T \geqslant 3$, naudojantis panašiais argumentais, kaip begalinio laiko lygybės (4) įrodyme, seka, kad

$$\begin{aligned}
\varphi(u,T) &= \mathbb{P}\left(\bigcap_{t=1}^{T}\left\{u+2t-\sum_{i=1}^{t}Z_i>0\right\}\right) = \mathbb{P}\left(\bigcap_{t=2}^{T}\left\{u+2t-\sum_{i=1}^{t}Z_i>0\right\}\cap\{Z_1<u+2\}\right)\\
&= \mathbb{P}\left(\bigcap_{t=2}^{T}\left\{u+2t-\sum_{i=1}^{t}Z_i>0\right\}\right) - \mathbb{P}\left(\bigcap_{t=2}^{T}\left\{u+2t-\sum_{i=1}^{t}Z_i>0\right\}\cap\{Z_1\geqslant u+2\}\right)\\
&= \mathbb{P}\left(\bigcap_{t=3}^{T}\left\{u+2t-Z_1-Z_2-\sum_{i=3}^{t}Z_i>0\right\}\cap\{Z_1+Z_2<u+4\}\right)\\
&- \mathbb{P}\left(\bigcap_{t=3}^{T}\left\{u+2t-Z_1-Z_2-\sum_{i=3}^{t}Z_i>0\right\}\cap\{Z_1\geqslant u+2\}\cap\{Z_1+Z_2<u+4\}\right)\\
&= \sum_{k=0}^{u+3}\mathbb{P}(Z_1+Z_2=k)\mathbb{P}\left(\bigcap_{t-2=1}^{T-2}\left\{u+2(t-2)+4-k-\sum_{i=1}^{t-2}Z_i>0\right\}\cap\{Z_1+Z_2=k\}\right)\\
&- (x_{u+2}s_1+x_{u+3}s_0)\mathbb{P}\left(\bigcap_{t=1}^{T-2}\left\{1+2t-\sum_{i=1}^{t}Z_i>0\right\}\right) - x_{u+2}s_0\mathbb{P}\left(\bigcap_{t=1}^{T-2}\left\{2+2t-\sum_{i=1}^{t}Z_i>0\right\}\right)\\
&= \sum_{k=0}^{u+3}\varphi(u+4-k,T-2)a_k - (x_{u+2}s_1+x_{u+3}s_0)\varphi(1,T-2) - x_{u+2}s_0\varphi(2,T-2).
\end{aligned}$$

□

Teoremos 2 įrodymas. Laikykime, kad α_n, β_n, γ_n ir δ_n yra rekurentinės sekos, apibrėžtos prieš teoremą 2. Norime parodyti, kad visiems $n \in \mathbb{N}_0$

$$\varphi(n) = \alpha_n\varphi(0)+\beta_n\varphi(1)+\gamma_n\varphi(2)+\delta_n(4-\mathbb{E}A). \tag{15}$$

Kai $n = 0$, 1 ir 2, lygybės akivaizdžios. Kai $n = 3$, išraišką gauname iš (11)

$$\begin{aligned}
\varphi(3) &= -\frac{1}{a_0}\varphi(0) - \frac{\overline{X}(2)s_0+\overline{X}(1)s_1+A(2)}{a_0}\varphi(1) - \frac{\overline{X}(1)s_0+A(1)}{a_0}\varphi(2) + \frac{1}{a_0}(4-\mathbb{E}A)\\
&= \alpha_3\varphi(0)+\beta_3\varphi(1)+\gamma_3\varphi(2)+\delta_3(4-\mathbb{E}A).
\end{aligned}$$

Kai $n \geqslant 4$, naudojame indukciją. Tada, iš (4) ir indukcijos hipotezės, seka

$$
\begin{aligned}
\varphi(n) &= \frac{1}{a_0}\left(\varphi(n-4) + (x_{n-1}s_0 + x_{n-2}s_1)\varphi(1) + x_{n-2}s_0\varphi(2) - \sum_{k=1}^{n-1}\varphi(k)a_{n-k}\right)\\
&= \frac{1}{a_0}\Big(\alpha_{n-4}\varphi(0) + \beta_{n-4}\varphi(1) + \gamma_{n-4}\varphi(2) + \delta_{n-4}(4-\mathbb{E}A)\Big)\\
&+ \frac{1}{a_0}\Big(\big(x_{n-1}s_0 + x_{n-2}s_1\big)\big(\alpha_1\varphi(0) + \beta_1\varphi(1) + \gamma_1\varphi(2) + \delta_1(4-\mathbb{E}A)\big)\Big)\\
&+ \frac{1}{a_0}\Big(x_{n-2}s_0\big(\alpha_2\varphi(0) + \beta_2\varphi(1) + \gamma_2\varphi(2) + \delta_2(4-\mathbb{E}A)\big)\Big)\\
&- \frac{1}{a_0}\left(\sum_{k=1}^{n-1}a_{n-k}\big(\alpha_k\varphi(0) + \beta_k\varphi(1) + \gamma_k\varphi(2) + \delta_k(4-\mathbb{E}A)\big)\right)\\
&= \frac{1}{a_0}\left(\alpha_{n-4} - \sum_{k=1}^{n-1}a_{n-k}\alpha_k\right)\varphi(0)\\
&+ \frac{1}{a_0}\left(\beta_{n-4} - \sum_{k=1}^{n-1}a_{n-k}\beta_k + x_{n-1}s_0 + x_{n-2}s_1\right)\varphi(1)\\
&+ \frac{1}{a_0}\left(\gamma_{n-4} - \sum_{k=1}^{n-1}a_{n-k}\gamma_k + x_{n-2}s_0\right)\varphi(2)\\
&+ \frac{1}{a_0}\left(\delta_{n-4} - \sum_{k=1}^{n-1}a_{n-k}\delta_k\right)(4-\mathbb{E}A)\\
&= \alpha_n\varphi(0) + \beta_n\varphi(1) + \gamma_n\varphi(2) + \delta_n(4-\mathbb{E}A).
\end{aligned}
$$

Iš to seka, kad lygtis (15) teisinga visiems $n \in \mathbb{N}_0$. Atimdami $\varphi(n+1) - \varphi(n)$, $\varphi(n+2) - \varphi(n)$ ir $\varphi(n+3) - \varphi(n)$ gauname sistemą (12), o likusios lygtys teoremoje 2 seka iš (11) ir (4). □

Teoremos 3 įrodymas. Laikykime, kad $\overline{\alpha}_n, \overline{\beta}_n$ ir $\overline{\delta}_n$ yra rekurentinės sekos, apibrėžtos prieš teoremą 3. Tada, naudojantis analogiška argumentacija kaip teoremoje 2, parodome

$$\varphi(n) = \overline{\alpha}_n\varphi(0) + \overline{\beta}_n\varphi(1) + \overline{\delta}_n(4-\mathbb{E}A),\ n \in \mathbb{N}_0. \tag{16}$$

Kai $n = 0$ arba 1, lygybės akivaizdžios. Kai $n = 2$, iš (11) gauname, kad

$$
\begin{aligned}
\varphi(2) &= -\frac{1}{a_1 + \overline{X}(1)s_0}\varphi(0) - \frac{\overline{X}(2)s_0 + \overline{X}(1)s_1 + A(2)}{a_1 + \overline{X}(1)s_0}\varphi(1) + \frac{1}{a_1 + \overline{X}(1)s_0}(4-\mathbb{E}A)\\
&= \overline{\alpha}_2\varphi(0) + \overline{\beta}_2\varphi(1) + \overline{\delta}_2(4-\mathbb{E}A).
\end{aligned}
$$

Kai $n \geqslant 3$, naudojame indukciją. Tada, iš (4) ir indukcijos hipotezės seka

$$\begin{aligned}
\varphi(n) &= \frac{1}{a_1}\left(\varphi(n-3) + x_{n-1}s_1\varphi(1) - \sum_{k=1}^{n-1}\varphi(k)a_{n+1-k}\right) \\
&= \frac{1}{a_1}\left(\overline{\alpha}_{n-3}\varphi(0) + \overline{\beta}_{n-3}\varphi(1) + \overline{\delta}_{n-3}(4-\mathbb{E}A)\right) \\
&+ \frac{1}{a_1}\left(x_{n-1}s_1\Big(\overline{\alpha}_1\varphi(0) + \overline{\beta}_1\varphi(1) + \overline{\delta}_1(4-\mathbb{E}A)\Big)\right) \\
&- \frac{1}{a_1}\left(\sum_{k=1}^{n-1}a_{n+1-k}\Big(\overline{\alpha}_k\varphi(0) + \overline{\beta}_k\varphi(1) + \overline{\delta}_k(4-\mathbb{E}A)\Big)\right) \\
&= \frac{1}{a_1}\left(\overline{\alpha}_{n-3} - \sum_{k=1}^{n-1}a_{n+1-k}\overline{\alpha}_k\right)\varphi(0) \\
&+ \frac{1}{a_1}\left(\overline{\beta}_{n-3} - \sum_{k=1}^{n-1}a_{n+1-k}\overline{\beta}_k + x_{n-1}s_1\right)\varphi(1) \\
&+ \frac{1}{a_1}\left(\overline{\delta}_{n-3} - \sum_{k=1}^{n-1}a_{n+1-k}\overline{\delta}_k\right)(4-\mathbb{E}A) \\
&= \overline{\alpha}_n\varphi(0) + \overline{\beta}_n\varphi(1) + \overline{\delta}_n(4-\mathbb{E}A).
\end{aligned}$$

Iš to seka, kad lygtis (15) teisinga visiems $n \in \mathbb{N}_0$. Atimdami $\varphi(n+1) - \varphi(n)$ ir $\varphi(n+2) - \varphi(n)$ gauname sistemą (13), o likusios lygtys teoremoje 3 seka iš (11) ir (4). $\square$

Teoremos 4 įrodymas. Laikykime, kad $\hat{\alpha}_n$ ir $\hat{\delta}_n$ yra rekurentinės sekos, apibrėžtos prieš teoremą 4. Tada, naudojantis analogiška argumentacija kaip teoremose 2 ir 3, parodome

$$\varphi(n) = \hat{\alpha}_n\varphi(0) + \hat{\delta}_n(4-\mathbb{E}A),\ n \in \mathbb{N}_0. \tag{17}$$

Kai $n = 0$, lygybė akivaizdi. Kai $n = 1$, iš (11) gauname, kad

$$\begin{aligned}
\varphi(1) &= -\frac{1}{a_2 + \overline{X}(1)s_1}\varphi(0) + \frac{1}{a_2 + \overline{X}(1)s_1}(4-\mathbb{E}A) \\
&= \hat{\alpha}_2\varphi(0) + \hat{\delta}_2(4-\mathbb{E}A).
\end{aligned}$$

Kai $n \geqslant 2$, naudojame indukciją. Tada, iš (4) ir indukcijos hipotezės seka

$$\begin{aligned}
\varphi(n) &= \frac{1}{a_2}\left(\varphi(n-2) + x_n s_1 \varphi(1) - \sum_{k=1}^{n-1} \varphi(k) a_{n+2-k}\right) \\
&= \frac{1}{a_2}\Big(\hat{\alpha}_{n-2}\varphi(0) + \hat{\delta}_{n-2}(4 - \mathbb{E}A)\Big) \\
&+ \frac{1}{a_2}\left(x_n s_1 \Big(\hat{\alpha}_1 \varphi(0) + \hat{\delta}_1 (4 - \mathbb{E}A)\Big)\right) \\
&- \frac{1}{a_2}\left(\sum_{k=1}^{n-1} a_{n+2-k}\Big(\hat{\alpha}_k \varphi(0) + \hat{\delta}_k (4 - \mathbb{E}A)\Big)\right) \\
&= \frac{1}{a_2}\left(\hat{\alpha}_{n-2} - \sum_{k=1}^{n-1} a_{n+2-k}\hat{\alpha}_k + x_n s_1 \hat{\alpha}_1\right)\varphi(0) \\
&+ \frac{1}{a_2}\left(\hat{\delta}_{n-2} - \sum_{k=1}^{n-1} a_{n+2-k}\hat{\delta}_k + x_n s_1 \hat{\delta}_1\right)(4 - \mathbb{E}A) \\
&= \hat{\alpha}_n \varphi(0) + \hat{\delta}_n (4 - \mathbb{E}A).
\end{aligned}$$

Iš to seka, kad lygtis (15) teisinga visiems $n \in \mathbb{N}_0$. Atimdami $\varphi(n+1) - \varphi(n)$ gauname lygtį (14), o likusios lygtys teoremoje 4 seka iš (11) ir (4).

Įrodymas, kad $\hat{\alpha}_{n+1} - \hat{\alpha}_n \neq 0$ visiems $n \in \mathbb{N}_0$, yra analogiškas pateiktiems darbuose [4, p. 937] ir [5, p. 16]. □

Teoremos 5 įrodymas. Kai $n = 0$, iš (11) gauname, kad

$$\varphi(0) = 4 - \mathbb{E}A.$$

Tada, kai $n \geqslant 1$, iš (4) seka, kad

$$\begin{aligned}
\varphi(1) &= \frac{4 - \mathbb{E}A}{a_3}, \\
\varphi(n) &= \frac{1}{a_3}\left(\varphi(n-1) - \sum_{k=1}^{n-1} \varphi(k) a_{n+3-k}\right), \ n \geqslant 2.
\end{aligned}$$

□

Teoremos 6 įrodymas.

i. Pirmu atveju, $\varphi(u) = 0$ visiems $u \in \mathbb{N}_0$, kai $\mathbb{E}S > 4$, seka iš lygybių (8) ir (10)

$$\varphi(0) + \Big(\overline{X}(2)s_0 + \overline{X}(1)s_1\Big)\varphi(1) + \overline{X}(1)s_0\varphi(2) + \sum_{k=1}^{3}\varphi(k)A(3-k) = (4 - \mathbb{E}A)\cdot\varphi(\infty). \tag{18}$$

Kadangi kairė lygties (18) pusė yra neneigiama ir $4 - \mathbb{E}A < 0$, tai $\varphi(\infty) = 0$, iš ko seka, kad $\varphi(u) = 0$ visiems $u \in \mathbb{N}_0$, nes φ - nemažėjanti.

ii. Antru atveju, $\varphi(u) = 0$ visiems $u \in \mathbb{N}_0$, kai $\mathbb{E}A = 4$ ir $a_4 < 1$, matome, kad (18) tampa

$$\varphi(0) + (x_0 s_2 + s_1 + s_0)\varphi(1) + (x_0 s_1 + s_0)\varphi(2) + x_0 s_0 \varphi(3) = 0. \tag{19}$$

Tada, panašiu principu kaip teoremose 2 – 5, nagrinėjame scenarijus, kai

$\mathcal{A}_1$ $a_0 > 0$,

$\mathcal{A}_2$ $a_0 = 0$, $a_1 > 0$,

$\mathcal{A}_3$ $a_0 = 0$, $a_1 = 0$, $a_2 > 0$,

$\mathcal{A}_4$ $a_0 = 0$, $a_1 = 0$, $a_2 = 0$, $a_3 > 0$.

Scenarijaus $\mathcal{A}_1$ atveju, $a_0 > 0 \Longrightarrow x_0 > 0$, $y_0 > 0 \Longrightarrow s_0 > 0$, dėl to lygtyje (19) koeficientai prie φ teigiami. Iš to seka, kad lygybė (19) teisinga tik tada, kai $\varphi(0) = \varphi(1) = \varphi(2) = \varphi(3) = 0$, tada iš (4) matome, kad ir $\varphi(n) = 0$, $n \geqslant 4$.

Scenarijaus $\mathcal{A}_2$ atveju, $a_0 = 0$, $a_1 > 0 \Longrightarrow x_0 > 0$, $y_0 = 0$, $y_1 > 0 \Longrightarrow s_0 = 0\, s_1 > 0$, dėl to lygtyje (19) koeficientai prie φ teigiami. Iš to seka, kad lygybė (19) teisinga tik tada, kai $\varphi(0) = \varphi(1) = \varphi(2) = 0$, tada iš (4) matome, kad ir $\varphi(n) = 0$, $n \geqslant 3$.

Scenarijaus $\mathcal{A}_3$ atveju yra dvi galimybės:

$\mathcal{A}_{3.1}$ $a_0 = 0$, $a_1 = 0\, a_2 > 0 \Longrightarrow x_0 = 0$, $x_1 > 0$, $y_0 > 0 \Longrightarrow s_0 = 0\, s_1 > 0$,

$\mathcal{A}_{3.2}$ $a_0 = 0$, $a_1 = 0\, a_2 > 0 \Longrightarrow x_0 > 0$, $y_0 = 0$, $y_1 = 0\, y_2 > 0 \Longrightarrow s_0 = 0\, s_1 = 0\, s_2 > 0$.

Abiem atvejais, $\mathcal{A}_{3.1}$ ir $\mathcal{A}_{3.2}$, lygtyje (19) esantys koeficientai prie φ teigiami. Iš to seka, kad lygybė (19) teisinga tik tada, kai $\varphi(0) = \varphi(1) = 0$, tada iš (4) matome, kad ir $\varphi(n) = 0$, $n \geqslant 2$.

Scenarijaus $\mathcal{A}_4$ atveju taip pat yra dvi galimybės:

$\mathcal{A}_{4.1}$ $a_0 = 0$, $a_1 = 0\, a_2 = 0\, a_3 > 0 \Longrightarrow x_0 = 0$, $x_1 > 0$, $y_0 = 0\, y_1 > 0 \Longrightarrow s_0 = 0\, s_1 > 0$,

$\mathcal{A}_{4.2}$ $a_0 = 0$, $a_1 = 0\, a_2 = 0\, a_3 > 0 \Longrightarrow x_0 > 0$, $y_0 = 0$, $y_1 = 0\, y_2 = 0\, y_3 > 0 \Longrightarrow s_0 = 0\, s_1 = 0\, s_2 > 0$,

Abiem atvejais, $\mathcal{A}_{4.1}$ ir $\mathcal{A}_{4.2}$, lygtyje (19) esantys koeficientai prie φ lygūs nuliui. Iš to seka, kad $\varphi(0) = 0$, tada iš (4) matome, kad ir $\varphi(n) = 0$, $n \geqslant 1$.

iii. Trečiasis ir paskutinysis atvejis, kai $\mathbb{E}A = 4$ ir $a_4 = x_0^2 y_4 + x_1^2 y_2 + x_2^2 y_0 = 1$, seka iš jungtinio dviejų žalų rizikos modelio su premija lygia dviem ($\kappa = 2$) apibrėžimo (1). Tada funkcija $W(t)$ lygi

- $W(t) = u + 2 \cdot \mathbb{1}_{\{t \text{ nelyginis}\}}$, jei $x_0 = y_4 = 1$,
- $W(t) = u + 1 \cdot \mathbb{1}_{\{t \text{ nelyginis}\}}$, jei $x_1 = y_2 = 1$,
- $W(t) = u$, jei $x_2 = y_0 = 1$.

Iš to matosi, kad $W(t) = 0$ su kai kuriais $t \in \mathbb{N}$, kai $u = 0$, o kai $u > 0$, tai $W(t) > 0$ visiems $t \in \mathbb{N}$. Dėl to

$$\varphi(u) = \begin{cases} 0, & \text{kai } u = 0 \\ 1, & \text{kai } u > 0 \end{cases}, \quad u \in \mathbb{N}_0.$$

□

4 Pavyzdžiai

Šiame skyriuje pateikiame konkrečius teoremų, aprašytų skyriuje 2, taikymo pavyzdžius, sugeneruotus *Python* aplinkoje (programos kodas pateiktas priede 4). Laikome, kad atsitiktiniai dydžiai, generuojantys jungtinį dviejų žalų diskretaus laiko riziko modelį su premija lygia dviem, pasiskirstę pagal paslinktojo Puasono skirstinį $\mathcal{P}(\lambda, \xi)$ su parametrais $\lambda > 0$ ir $\xi \in \mathbb{N}_0$, kurio tikimybių pasiskirstymo funkcija lygi

$$\mathbb{P}(X = m) = e^{-\lambda} \frac{\lambda^{m-\xi}}{(m-\xi)!}, \; m = \xi, \xi + 1, \ldots$$

Tokiam atsitiktiniam dydžiui galioja:

- $\mathbb{E}X = \lambda + \xi$,
- $X + Y \sim \mathcal{P}(\lambda_1 + \lambda_2, \xi_1 + \xi_2)$,

jei $X \sim \mathcal{P}(\lambda_1, \xi_1)$, $Y \sim \mathcal{P}(\lambda_2, \xi_2)$ ir X, Y yra nepriklausomi. Detalesnis paslinktojo Puasono savybių aprašymas pateiktas [8].

Toliau pateiktose $\varphi(u, T)$ ir $\varphi(u)$ pavyzdžių lentelėse:

- visos reikšmės suapvalintos iki 3 skaitmenų po kablelio, išskyrus 0 ir 1;
- kintamieji T ir u pasirinkti taip, kad geriausiai atspindėtų išgyvenamumo tikimybių pasikeitimus;
- taikius teoremas 2 – 4, pasirinktas $n = 20$, kaip pakankamai didelis pakankamam išgyvenamumo tikimybių tikslumui gauti.

Pavyzdys 4.1. *Laikykime, kad* $X \sim \mathcal{P}(1, 0)$ *ir* $Y \sim \mathcal{P}(1, 0)$. *Naudojantis teorema 1 ir teorema 2 gaunama toliau pateikta lentelė*

1 lentelė: Išgyvenamumo tikimybės, kai $X \sim \mathcal{P}(1, 0)$ ir $Y \sim \mathcal{P}(1, 0)$.

T	$\boldsymbol{u=0}$	$\boldsymbol{u=1}$	$\boldsymbol{u=2}$	$\boldsymbol{u=3}$	$\boldsymbol{u=4}$	$\boldsymbol{u=5}$	$\boldsymbol{u=10}$	$\boldsymbol{u=15}$
1	0.736	0.920	0.981	0.996	0.999	1	1	1
2	0.564	0.788	0.909	0.965	0.988	0.996	1	1
3	0.547	0.771	0.898	0.959	0.985	0.995	1	1
4	0.505	0.727	0.863	0.936	0.972	0.989	1	1
5	0.499	0.720	0.857	0.932	0.969	0.987	1	1
6	0.481	0.699	0.838	0.918	0.960	0.982	1	1
7	0.478	0.695	0.834	0.915	0.958	0.980	1	1
8	0.468	0.683	0.823	0.906	0.952	0.976	1	1
9	0.466	0.681	0.821	0.904	0.950	0.975	1	1
10	0.460	0.673	0.813	0.898	0.946	0.972	0.999	1
11	0.459	0.672	0.812	0.897	0.945	0.971	0.999	1
12	0.455	0.667	0.807	0.892	0.941	0.969	0.999	1
13	0.454	0.666	0.806	0.891	0.941	0.968	0.999	1
14	0.452	0.662	0.803	0.888	0.938	0.967	0.999	1
15	0.451	0.662	0.802	0.888	0.938	0.966	0.999	1
∞	0.442	0.650	0.790	0.876	0.928	0.958	0.997	1

Pavyzdys 4.2. *Laikykime, kad* $X \sim \mathcal{P}(1,0)$ *ir* $Y \sim \mathcal{P}(9/10,1)$. *Naudojantis teorema 1 ir teorema 3 gaunama toliau pateikta lentelė*

2 lentelė: Išgyvenamumo tikimybės, kai $X \sim \mathcal{P}(1,0)$ ir $Y \sim \mathcal{P}(9/10,1)$.

T	$\boldsymbol{u=0}$	$\boldsymbol{u=1}$	$\boldsymbol{u=2}$	$\boldsymbol{u=3}$	$\boldsymbol{u=4}$	$\boldsymbol{u=5}$	$\boldsymbol{u=10}$	$\boldsymbol{u=15}$	$\boldsymbol{u=20}$
1	0.736	0.920	0.981	0.996	0.999	1	1	1	1
2	0.418	0.660	0.829	0.925	0.971	0.990	1	1	1
3	0.399	0.638	0.811	0.914	0.965	0.987	1	1	1
4	0.318	0.535	0.715	0.841	0.918	0.961	1	1	1
5	0.310	0.524	0.703	0.831	0.912	0.957	1	1	1
6	0.269	0.465	0.639	0.774	0.867	0.926	0.998	1	1
7	0.265	0.458	0.631	0.767	0.861	0.922	0.998	1	1
8	0.239	0.419	0.586	0.722	0.822	0.892	0.995	1	1
9	0.236	0.414	0.580	0.716	0.817	0.887	0.995	1	1
10	0.219	0.386	0.545	0.680	0.784	0.860	0.991	1	1
11	0.216	0.382	0.541	0.675	0.779	0.856	0.990	1	1
12	0.203	0.360	0.513	0.645	0.751	0.831	0.985	0.999	1
13	0.201	0.358	0.510	0.641	0.747	0.827	0.984	0.999	1
14	0.191	0.340	0.487	0.617	0.722	0.805	0.978	0.999	1
15	0.190	0.338	0.484	0.613	0.719	0.802	0.977	0.999	1
∞	0.062	0.113	0.168	0.221	0.272	0.319	0.514	0.653	0.759

Pavyzdys 4.3. *Laikykime, kad* $X \sim \mathcal{P}(1/2, 1)$ *ir* $Y \sim \mathcal{P}(9/10, 0)$*. Naudojantis teorema 1 ir teorema 4 gaunama toliau pateikta lentelė*

3 lentelė: Išgyvenamumo tikimybės, kai $X \sim \mathcal{P}(1/2, 1)$ ir $Y \sim \mathcal{P}(9/10, 0)$.

T	$\boldsymbol{u = 0}$	$\boldsymbol{u = 1}$	$\boldsymbol{u = 2}$	$\boldsymbol{u = 3}$	$\boldsymbol{u = 4}$	$\boldsymbol{u = 5}$	$\boldsymbol{u = 10}$	$\boldsymbol{u = 15}$	$\boldsymbol{u = 20}$
1	0.607	0.91	0.986	0.998	1	1	1	1	1
2	0.359	0.685	0.872	0.956	0.987	0.997	1	1	1
3	0.338	0.658	0.852	0.945	0.982	0.995	1	1	1
4	0.274	0.563	0.769	0.892	0.954	0.982	1	1	1
5	0.265	0.549	0.756	0.882	0.948	0.979	1	1	1
6	0.233	0.493	0.698	0.836	0.918	0.962	1	1	1
7	0.228	0.485	0.689	0.828	0.912	0.958	1	1	1
8	0.208	0.447	0.646	0.790	0.883	0.939	0.999	1	1
9	0.205	0.441	0.639	0.783	0.878	0.935	0.999	1	1
10	0.190	0.414	0.606	0.751	0.852	0.916	0.998	1	1
11	0.188	0.410	0.600	0.746	0.847	0.912	0.997	1	1
12	0.177	0.389	0.574	0.719	0.823	0.894	0.996	1	1
13	0.176	0.385	0.569	0.714	0.819	0.891	0.995	1	1
14	0.167	0.368	0.548	0.691	0.798	0.874	0.993	1	1
15	0.166	0.365	0.544	0.687	0.795	0.870	0.993	1	1
∞	0.064	0.145	0.225	0.300	0.367	0.429	0.656	0.793	0.878

Pavyzdys 4.4. *Laikykime, kad* $X \sim \mathcal{P}(1/4, 1)$ *ir* $Y \sim \mathcal{P}(2/5, 1)$. *Naudojantis teorema 1 ir teorema 5 gaunama toliau pateikta lentelė*

4 lentelė: Išgyvenamumo tikimybės, kai $X \sim \mathcal{P}(1/4, 1)$ ir $Y \sim \mathcal{P}(2/5, 1)$.

T	$\boldsymbol{u=0}$	$\boldsymbol{u=1}$	$\boldsymbol{u=2}$	$\boldsymbol{u=3}$	$\boldsymbol{u=4}$	$\boldsymbol{u=5}$	$\boldsymbol{u=10}$	$\boldsymbol{u=15}$	$\boldsymbol{u=20}$
1	0.779	0.974	0.998	1	1	1	1	1	1
2	0.407	0.772	0.937	0.987	0.998	1	1	1	1
3	0.396	0.762	0.932	0.985	0.997	1	1	1	1
4	0.314	0.664	0.871	0.959	0.989	0.997	1	1	1
5	0.310	0.658	0.867	0.957	0.988	0.997	1	1	1
6	0.270	0.597	0.818	0.929	0.976	0.992	1	1	1
7	0.267	0.593	0.815	0.927	0.975	0.992	1	1	1
8	0.243	0.551	0.775	0.901	0.961	0.986	1	1	1
9	0.241	0.548	0.773	0.899	0.959	0.985	1	1	1
10	0.224	0.517	0.741	0.875	0.945	0.978	1	1	1
11	0.223	0.515	0.738	0.873	0.944	0.977	1	1	1
12	0.210	0.490	0.712	0.852	0.930	0.969	1	1	1
13	0.209	0.489	0.710	0.850	0.929	0.968	1	1	1
14	0.199	0.469	0.688	0.831	0.915	0.960	1	1	1
15	0.199	0.467	0.686	0.830	0.914	0.959	1	1	1
∞	0.100	0.246	0.384	0.499	0.592	0.669	0.882	0.958	0.985

Pavyzdys 4.5. *Laikykime, kad $X \sim \mathcal{P}(1,1)$ ir $Y \sim \mathcal{P}(1,0)$. Naudojantis teorema 1 ir teorema 6 gaunama toliau pateikta lentelė*

5 lentelė: Išgyvenamumo tikimybės, kai $X \sim \mathcal{P}(1,1)$ ir $Y \sim \mathcal{P}(1,0)$.

T	$\boldsymbol{u=0}$	$\boldsymbol{u=1}$	$\boldsymbol{u=2}$	$\boldsymbol{u=3}$	$\boldsymbol{u=4}$	$\boldsymbol{u=5}$	$\boldsymbol{u=10}$	$\boldsymbol{u=15}$	$\boldsymbol{u=20}$
1	0.368	0.736	0.920	0.981	0.996	0.999	1	1	1
2	0.149	0.398	0.639	0.813	0.916	0.966	1	1	1
3	0.119	0.333	0.561	0.747	0.870	0.941	1	1	1
4	0.071	0.215	0.396	0.576	0.728	0.840	0.996	1	1
5	0.061	0.187	0.352	0.525	0.679	0.800	0.993	1	1
6	0.041	0.131	0.258	0.404	0.551	0.683	0.977	1	1
7	0.036	0.116	0.232	0.369	0.511	0.643	0.968	0.999	1
8	0.026	0.085	0.174	0.286	0.412	0.538	0.933	0.997	1
9	0.023	0.076	0.157	0.262	0.381	0.504	0.918	0.995	1
10	0.017	0.057	0.120	0.205	0.307	0.418	0.866	0.988	1
11	0.015	0.052	0.110	0.189	0.284	0.391	0.846	0.985	0.999
12	0.012	0.040	0.085	0.149	0.230	0.323	0.783	0.970	0.998
13	0.011	0.036	0.078	0.137	0.213	0.302	0.760	0.963	0.997
14	0.008	0.028	0.061	0.109	0.172	0.249	0.692	0.940	0.994
15	0.007	0.026	0.056	0.101	0.160	0.232	0.669	0.930	0.993
∞	0	0	0	0	0	0	0	0	0

Iš aukščiau pateiktų pavyzdžių 4.1–4.5, galime pastebėti, kad skirtumas tarp $\mathbb{E}A$ ir 4 turi stiprią įtaką išgyvenamumo tikimybei. Iš pavyzdžio 4.1 matome, kad patirtos žalos, kurios vidutiniškai nėra per daug „didelės“ ($\mathbb{E}A = 3$), gali nesunkiai būti padengtos pradiniu turtu u ir užsitikrint išgyvenamumą, kai $u = 15$. Taip pat, iš pavyzdžio 4.5 matome, kad tam tikram laikui pakankamai užtikrinta išgyvenamumo tikimybė gali būti pasiekta su pakankamu pradiniu turtu, netgi kai galimos atsitiktinės žalos labiau „agresyvios“.

Literatūra

[1] E. Andersen, On the collective theory of risk in case of contagion between the claims, Trans. Xvth Int. Actuar. 2 (1957) 219 – 229.

[2] A. Grigutis, A. Korvel, J. Šiaulys, Ruin probabilities of a discrete-time multi-risk model, Information Technology and Control 44 (4) (2015) 367–379. doi:10.5755/j01.itc.44.4.8635.

[3] A. Alencenovič, A. Grigutis, Bi-seasonal discrete time risk model with income rate two (2021). arXiv:2104.14771.

[4] J. Damarackas, J. Šiaulys, Bi-seasonal discrete time risk model, Applied Mathematics and Computation 247 (2014) 930 – 940. doi:10.1016/j.amc.2014.09.040.

[5] A. Grigutis, J. Šiaulys, Recurrent sequences play for survival probability of discrete time risk model, Symmetry 12 (12) (2020). doi:10.3390/sym12122111.

[6] A. Grigutis, J. Šiaulys, Ultimate time survival probability in three-risk discrete time risk model, Mathematics 8 (2) (2020). doi:10.3390/math8020147.

[7] K. Blaževičius, E. Bieliauskienė, J. Šiaulys, Finite-time ruin probability in the inhomogenous claim case, Lith. Math. J. 50 (2010) 260 – 270.

[8] P. J. Staff, The displaced Poisson distribution, Journal of the American Statistical Association 62 (1967) 643–654. doi:10.1080/01621459.1967.10482938.

Priedas

Šiame priede pateiktas *Python* programos kodas, kuris buvo naudojamas generuojant išgyvenamumo tikimybes, skyriuje 4 pateiktiems pavyzdžiams.

```
###
#import of required packages
import numpy as np
import math
import pandas as pd
from datetime import timedelta
from timeit import default_timer as timer

#defining the sigma sum function
def sigma_sum(start, end, expression):
return sum(expression(i) for i in range(start, end+1))

#defining P(X=u) function
def x(u,lmbd_x,disp_x):
lmbd = lmbd_x
disp = disp_x
if u>=disp and u>=0:
prob = math.exp(-lmbd) * lmbd**(u-disp)/math.factorial(u-disp)
else:
prob = 0
return prob

#defining P(Y=u) function
def y(u,lmbd_y,disp_y):
lmbd = lmbd_y
disp = disp_y
if u>=disp and u>=0:
prob = math.exp(-lmbd) * lmbd**(u-disp)/math.factorial(u-disp)
else:
prob = 0
return prob

#defining P(X+Y=u) function
def s(u,lmbd_x,lmbd_y,disp_x,disp_y):
lmbd = lmbd_x+lmbd_y
disp = disp_x+disp_y
if u>=disp and u>=0:
prob = math.exp(-lmbd) * lmbd**(u-disp)/math.factorial(u-disp)
else:
prob = 0
return prob

#defining P(X+X+Y=u) function
def a(u,lmbd_x,lmbd_y,disp_x,disp_y):
lmbd = lmbd_x+lmbd_x+lmbd_y
disp = disp_x+disp_x+disp_y
if u>=disp and u>=0:
prob = math.exp(-lmbd) * lmbd**(u-disp)/math.factorial(u-disp)
else:
```

```
prob = 0
return prob

#defining P(X<=u) function
def X(u,lmbd_x,disp_x):
if u<0:
prob = 0
else:
prob = sigma_sum(0, u, lambda i: x(i,lmbd_x,disp_x))
return prob

#defining P(Y<=u) function
def Y(u,lmbd_y,disp_y):
if u<0:
prob = 0
else:
prob = sigma_sum(0, u, lambda i: y(i,lmbd_y,disp_y))
return prob

#defining P(X+Y<=u) function
def S(u,lmbd_x,lmbd_y,disp_x,disp_y):
if u<0:
prob = 0
else:
prob = sigma_sum(0, u, lambda i: s(i,lmbd_x,lmbd_y,disp_x,disp_y))
return prob

#defining P(X+X+Y<=u) function
def A(u,lmbd_x,lmbd_y,disp_x,disp_y):
if u<0:
prob = 0
else:
prob = sigma_sum(0, u, lambda i: a(i,lmbd_x,lmbd_y,disp_x,disp_y))
return prob

#defining P(X>u) function
def X_(u,lmbd_x,disp_x):
prob = 1 - X(u,lmbd_x,disp_x)
return prob

#defining P(Y>u) function
def Y_(u,lmbd_y,disp_y):
prob = 1 - Y(u,lmbd_y,disp_y)
return prob

#defining P(X+Y>u) function
def S_(u,lmbd_x,lmbd_y,disp_x,disp_y):
prob = 1 - S(u,lmbd_x,lmbd_y,disp_x,disp_y)
return prob

#defining P(X+X+Y>u) function
def A_(u,lmbd_x,lmbd_y,disp_x,disp_y):
prob = 1 - A(u,lmbd_x,lmbd_y,disp_x,disp_y)
return prob
```

```
#defining phi(u,T) - finite time surivival probability function
def phi(u,T,lmbd_x,lmbd_y,disp_x,disp_y):
if u<0:
return print("Invalid u value")
if T<=0:
return print("Invalid T value")
else:
if T==1:
return X(u+1,lmbd_x,disp_x)
elif T==2:
return sigma_sum(0, u+1, lambda k: x(k,lmbd_x,disp_x)*S(u+3-k,lmbd_x,lmbd_y,disp_x,disp_y))
else:
return sigma_sum(0,u+3, lambda k: phi(u+4-k,T-2,lmbd_x,lmbd_y,disp_x,disp_y)*a(k,lmbd_x,lmbd_y,di
-(x(u+2,lmbd_x,disp_x)*s(1,lmbd_x,lmbd_y,disp_x,disp_y)+x(u+3,lmbd_x,disp_x)*s(0,lmbd_x,lmbd_y,di
-x(u+2,lmbd_x,disp_x)*s(0,lmbd_x,lmbd_y,disp_x,disp_y)*phi(2,T-2,lmbd_x,lmbd_y,disp_x,disp_y)

#defining recurrent alpha function for CASE 1
def alpha_1(n,lmbd_x,lmbd_y,disp_x,disp_y):
if n<0:
return print("Invalid n value")
elif n==0:
return 1
elif n==1:
return 0
elif n==2:
return 0
elif n==3:
return -1/a(0,lmbd_x,lmbd_y,disp_x,disp_y)
else:
return 1/a(0,lmbd_x,lmbd_y,disp_x,disp_y)\
*(alpha_1(n-4,lmbd_x,lmbd_y,disp_x,disp_y)\
-sigma_sum(1,n-1,lambda k:a(n-k,lmbd_x,lmbd_y,disp_x,disp_y)*alpha_1(k,lmbd_x,lmbd_y,disp_x,disp_

#defining recurrent beta function for CASE 1
def beta_1(n,lmbd_x,lmbd_y,disp_x,disp_y):
if n<0:
return print("Invalid n value")
elif n==0:
return 0
elif n==1:
return 1
elif n==2:
return 0
elif n==3:
return -1/a(0,lmbd_x,lmbd_y,disp_x,disp_y)*(A(2,lmbd_x,lmbd_y,disp_x,disp_y)+X_(2,lmbd_x,disp_x)*
else:
return 1/a(0,lmbd_x,lmbd_y,disp_x,disp_y)\
*(beta_1(n-4,lmbd_x,lmbd_y,disp_x,disp_y)\
-sigma_sum(1,n-1,lambda k:a(n-k,lmbd_x,lmbd_y,disp_x,disp_y)*beta_1(k,lmbd_x,lmbd_y,disp_x,disp_y
+x(n-1,lmbd_x,disp_x)*s(0,lmbd_x,lmbd_y,disp_x,disp_y)+x(n-2,lmbd_x,disp_x)*s(1,lmbd_x,lmbd_y,dis

#defining recurrent gamma function for CASE 1
def gamma_1(n,lmbd_x,lmbd_y,disp_x,disp_y):
```

```
if n<0:
return print("Invalid n value")
elif n==0:
return 0
elif n==1:
return 0
elif n==2:
return 1
elif n==3:
return -1/a(0,lmbd_x,lmbd_y,disp_x,disp_y)*(A(1,lmbd_x,lmbd_y,disp_x,disp_y)+X_(1,lmbd_x,disp_x)*
else:
return 1/a(0,lmbd_x,lmbd_y,disp_x,disp_y)\
*(gamma_1(n-4,lmbd_x,lmbd_y,disp_x,disp_y)\
-sigma_sum(1,n-1,lambda k:a(n-k,lmbd_x,lmbd_y,disp_x,disp_y)*gamma_1(k,lmbd_x,lmbd_y,disp_x,disp_
+x(n-2,lmbd_x,disp_x)*s(0,lmbd_x,lmbd_y,disp_x,disp_y))

#defining recurrent delta function for CASE 1
def delta_1(n,lmbd_x,lmbd_y,disp_x,disp_y):
if n<0:
return print("Invalid n value")
elif n==0:
return 0
elif n==1:
return 0
elif n==2:
return 0
elif n==3:
return 1/a(0,lmbd_x,lmbd_y,disp_x,disp_y)
else:
return 1/a(0,lmbd_x,lmbd_y,disp_x,disp_y)\
*(delta_1(n-4,lmbd_x,lmbd_y,disp_x,disp_y)\
-sigma_sum(1,n-1,lambda k:a(n-k,lmbd_x,lmbd_y,disp_x,disp_y)*delta_1(k,lmbd_x,lmbd_y,disp_x,disp_

#defining phi(0), phi(1), phi(2) and phi(3) finder function for CASE 1
def PHI_123(n,lmbd_x,lmbd_y,disp_x,disp_y):
if 2*(lmbd_x+disp_x)+lmbd_y+disp_y >= 4 or disp_x!=0 or disp_y!=0:
print('Wrong usage')
else:
M=[]
E=[]
for i in range(1,4):
row=[]
row.append(alpha_1(n+i,lmbd_x,lmbd_y,disp_x,disp_y)-alpha_1(n,lmbd_x,lmbd_y,disp_x,disp_y))
row.append(beta_1(n+i,lmbd_x,lmbd_y,disp_x,disp_y)-beta_1(n,lmbd_x,lmbd_y,disp_x,disp_y))
row.append(gamma_1(n+i,lmbd_x,lmbd_y,disp_x,disp_y)-gamma_1(n,lmbd_x,lmbd_y,disp_x,disp_y))
M.append(row)
row=[]
row.append((delta_1(n+i,lmbd_x,lmbd_y,disp_x,disp_y)-delta_1(n,lmbd_x,lmbd_y,disp_x,disp_y))*(2*(
E.append(row)
M=np.matrix(M)
E=np.matrix(E)
M_=np.matrix(np.linalg.inv(M))
res123=np.matmul(M_,E)
return res123
```

```
#defining optimal n finder function for phi(0), phi(1), phi(2) and phi(3) finder function for CAS
def PHI_123_n_finder(precision,lmbd_x,lmbd_y,disp_x,disp_y):
phi_123_last=PHI_123(1,lmbd_x,lmbd_y,disp_x,disp_y)
phi_123_n=PHI_123(2,lmbd_x,lmbd_y,disp_x,disp_y)
i=3
while round(float(phi_123_last[0]),precision)!=round(float(phi_123_n[0]),precision) or\
round(float(phi_123_last[1]),precision)!=round(float(phi_123_n[1]),precision) or\
round(float(phi_123_last[2]),precision)!=round(float(phi_123_n[2]),precision):
phi_123_last=phi_123_n
phi_123_n=PHI_123(i,lmbd_x,lmbd_y,disp_x,disp_y)
print('--- n =',i,'---')
print(np.around(phi_123_n.astype(float),precision))
if i >= 20:
print('---Computing time too long, n =',i,'used.---')
return i
else:
i=i+1
print('---Selection complete, n =',i-1,'used.---')
return phi_123_n

#defining recurrent alpha function for CASE 2
def alpha_2(n,lmbd_x,lmbd_y,disp_x,disp_y):
if n<0:
return print("Invalid n value")
elif n==0:
return 1
elif n==1:
return 0
elif n==2:
return -1/a(1,lmbd_x,lmbd_y,disp_x,disp_y)
else:
return 1/a(1,lmbd_x,lmbd_y,disp_x,disp_y)\
*(alpha_2(n-3,lmbd_x,lmbd_y,disp_x,disp_y)\
-sigma_sum(1,n-1,lambda k:a(n+1-k,lmbd_x,lmbd_y,disp_x,disp_y)*alpha_2(k,lmbd_x,lmbd_y,disp_x,dis

#defining recurrent beta function for CASE 2
def beta_2(n,lmbd_x,lmbd_y,disp_x,disp_y):
if n<0:
return print("Invalid n value")
elif n==0:
return 0
elif n==1:
return 1
elif n==2:
return -1/a(1,lmbd_x,lmbd_y,disp_x,disp_y)*(A(2,lmbd_x,lmbd_y,disp_x,disp_y)+X_(1,lmbd_x,disp_x)*
else:
return 1/a(1,lmbd_x,lmbd_y,disp_x,disp_y)\
*(beta_2(n-3,lmbd_x,lmbd_y,disp_x,disp_y)\
-sigma_sum(1,n-1,lambda k:a(n+1-k,lmbd_x,lmbd_y,disp_x,disp_y)*beta_2(k,lmbd_x,lmbd_y,disp_x,disp
+x(n-1,lmbd_x,disp_x)*s(1,lmbd_x,lmbd_y,disp_x,disp_y))

#defining recurrent gamma (in article delta) function for CASE 2
def gamma_2(n,lmbd_x,lmbd_y,disp_x,disp_y):
```

```
if n<0:
return print("Invalid n value")
elif n==0:
return 0
elif n==1:
return 0
elif n==2:
return 1/a(1,lmbd_x,lmbd_y,disp_x,disp_y)
else:
return 1/a(1,lmbd_x,lmbd_y,disp_x,disp_y)\
*(gamma_2(n-3,lmbd_x,lmbd_y,disp_x,disp_y)\
-sigma_sum(1,n-1,lambda k:a(n+1-k,lmbd_x,lmbd_y,disp_x,disp_y)*gamma_2(k,lmbd_x,lmbd_y,disp_x,dis

#defining phi(0), phi(1) and phi(2) finder function for CASE 2
def PHI_12(n,lmbd_x,lmbd_y,disp_x,disp_y):
if 2*(lmbd_x+disp_x)+lmbd_y+disp_y >= 4 or disp_x!=0 or disp_y!=1:
print('Wrong usage')
else:
M=[]
E=[]
for i in range(1,3):
row=[]
row.append(alpha_2(n+i,lmbd_x,lmbd_y,disp_x,disp_y)-alpha_2(n,lmbd_x,lmbd_y,disp_x,disp_y))
row.append(beta_2(n+i,lmbd_x,lmbd_y,disp_x,disp_y)-beta_2(n,lmbd_x,lmbd_y,disp_x,disp_y))
M.append(row)
row=[]
row.append((gamma_2(n+i,lmbd_x,lmbd_y,disp_x,disp_y)-gamma_2(n,lmbd_x,lmbd_y,disp_x,disp_y))*(2*(
E.append(row)
M=np.matrix(M)
E=np.matrix(E)
M_=np.matrix(np.linalg.inv(M))
res12=np.matmul(M_,E)
return res12

#defining optimal n finder function for phi(0), phi(1) and phi(2) finder function for CASE 2
def PHI_12_n_finder(precision,lmbd_x,lmbd_y,disp_x,disp_y):
phi_12_last=PHI_12(1,lmbd_x,lmbd_y,disp_x,disp_y)
phi_12_n=PHI_12(2,lmbd_x,lmbd_y,disp_x,disp_y)
i=3
while round(float(phi_12_last[0]),precision)!=round(float(phi_12_n[0]),precision) or\
round(float(phi_12_last[1]),precision)!=round(float(phi_12_n[1]),precision):
phi_12_last=phi_12_n
phi_12_n=PHI_12(i,lmbd_x,lmbd_y,disp_x,disp_y)
print('--- n =',i,'---')
print(np.around(phi_12_n.astype(float),precision))
if i >= 20:
print('---Computing time too long, n =',i,'used.---')
return i
else:
i=i+1
print('---Selection complete, n =',i-1,'used.---')
return phi_12_n

#defining recurrent alpha function for CASE 3
```

```
def alpha_3(n,lmbd_x,lmbd_y,disp_x,disp_y):
if n<0:
return print("Invalid n value")
elif n==0:
return 1
elif n==1:
return -1/(a(2,lmbd_x,lmbd_y,disp_x,disp_y)+X_(1,lmbd_x,disp_x)*s(1,lmbd_x,lmbd_y,disp_x,disp_y))
else:
return 1/a(2,lmbd_x,lmbd_y,disp_x,disp_y)\
*(alpha_3(n-2,lmbd_x,lmbd_y,disp_x,disp_y)\
-sigma_sum(1,n-1,lambda k:a(n+2-k,lmbd_x,lmbd_y,disp_x,disp_y)*alpha_3(k,lmbd_x,lmbd_y,disp_x,dis
+x(n,lmbd_x,disp_x)*s(1,lmbd_x,lmbd_y,disp_x,disp_y)*alpha_3(1,lmbd_x,lmbd_y,disp_x,disp_y))

#defining recurrent beta (in article delta) function for CASE 3
def beta_3(n,lmbd_x,lmbd_y,disp_x,disp_y):
if n<0:
return print("Invalid n value")
elif n==0:
return 0
elif n==1:
return 1/(a(2,lmbd_x,lmbd_y,disp_x,disp_y)+X_(1,lmbd_x,disp_x)*s(1,lmbd_x,lmbd_y,disp_x,disp_y))
else:
return 1/a(2,lmbd_x,lmbd_y,disp_x,disp_y)\
*(beta_3(n-2,lmbd_x,lmbd_y,disp_x,disp_y)\
-sigma_sum(1,n-1,lambda k:a(n+2-k,lmbd_x,lmbd_y,disp_x,disp_y)*beta_3(k,lmbd_x,lmbd_y,disp_x,disp
+x(n,lmbd_x,disp_x)*s(1,lmbd_x,lmbd_y,disp_x,disp_y)*beta_3(1,lmbd_x,lmbd_y,disp_x,disp_y))

#defining phi(0) and phi(1) finder function for CASE 3
def PHI_1(n,lmbd_x,lmbd_y,disp_x,disp_y):
if 2*(lmbd_x+disp_x)+lmbd_y+disp_y >= 4 or not ((disp_x==1 and disp_y==0) or (disp_x==0 and disp_
print('Wrong usage')
else:
res1=(beta_3(n+1,lmbd_x,lmbd_y,disp_x,disp_y)-beta_3(n,lmbd_x,lmbd_y,disp_x,disp_y))\
/(alpha_3(n+1,lmbd_x,lmbd_y,disp_x,disp_y)-alpha_3(n,lmbd_x,lmbd_y,disp_x,disp_y))\
*(2*(lmbd_x+disp_x)+lmbd_y+disp_y-4)
return res1

#defining optimal n finder function for phi(0) and phi(1) finder function for CASE 3
def PHI_1_n_finder(precision,lmbd_x,lmbd_y,disp_x,disp_y):
phi_1_last=PHI_1(1,lmbd_x,lmbd_y,disp_x,disp_y)
phi_1_n=PHI_1(2,lmbd_x,lmbd_y,disp_x,disp_y)
i=3
while round(float(phi_1_last),precision)!=round(float(phi_1_n),precision):
phi_1_last=phi_1_n
phi_1_n=PHI_1(i,lmbd_x,lmbd_y,disp_x,disp_y)
print('--- n =',i,'---')
print(round(float(phi_1_n),precision))
if i >= 20:
print('---Computing time too long, n =',i,'used.---')
return i
else:
i=i+1
print('---Selection complete, n =',i-1,'used.---')
return phi_1_n
```

```
#defining phi(u) - ultimate time surivival probability function
def phi_inf(u,lmbd_x,lmbd_y,disp_x,disp_y,phi_initial=[]):
if 4<(2*(lmbd_x+disp_x)+lmbd_y+disp_y):#EA>=4
return 0
elif 4==(2*(lmbd_x+disp_x)+lmbd_y+disp_y) and a(4,lmbd_x,lmbd_y,disp_x,disp_y)<1:
return 0
elif 4==(2*(lmbd_x+disp_x)+lmbd_y+disp_y) and a(4,lmbd_x,lmbd_y,disp_x,disp_y)==1:
if u==0:
return 0
else:
return 1
elif disp_x==0 and disp_y==0:#CASE 1
phi_123_n=phi_initial
return float(alpha_1(u,lmbd_x,lmbd_y,disp_x,disp_y)*phi_123_n[0]\
+beta_1(u,lmbd_x,lmbd_y,disp_x,disp_y)*phi_123_n[1]\
+gamma_1(u,lmbd_x,lmbd_y,disp_x,disp_y)*phi_123_n[2]\
+delta_1(u,lmbd_x,lmbd_y,disp_x,disp_y)*(4-(2*(lmbd_x+disp_x)+lmbd_y+disp_y)))
elif disp_x==0 and disp_y==1:#CASE 2
phi_12_n=phi_initial
return float(alpha_2(u,lmbd_x,lmbd_y,disp_x,disp_y)*phi_12_n[0]\
+beta_2(u,lmbd_x,lmbd_y,disp_x,disp_y)*phi_12_n[1]\
+gamma_2(u,lmbd_x,lmbd_y,disp_x,disp_y)*(4-(2*(lmbd_x+disp_x)+lmbd_y+disp_y)))
elif (disp_x==1 and disp_y==0) or (disp_x==0 and disp_y==2):#CASE 3
phi_1_n=phi_initial
return float(alpha_3(u,lmbd_x,lmbd_y,disp_x,disp_y)*phi_1_n\
+beta_3(u,lmbd_x,lmbd_y,disp_x,disp_y)*(4-(2*(lmbd_x+disp_x)+lmbd_y+disp_y)))
elif (disp_x==0 and disp_y==3) or (disp_x==1 and disp_y==1):#CASE 4
if u==0:
return 4-(2*(lmbd_x+disp_x)+lmbd_y+disp_y)
elif u==1:
return (4-(2*(lmbd_x+disp_x)+lmbd_y+disp_y))/a(3,lmbd_x,lmbd_y,disp_x,disp_y)
else:
return (phi_inf(u-1,lmbd_x,lmbd_y,disp_x,disp_y)-sigma_sum(1,u-1,lambda k: phi_inf(k,lmbd_x,lmbd_

#defining example calculation function
def example_table(u_max,T_max,lmbd_x,lmbd_y,disp_x,disp_y,precision=10,n=0):
t1 = timer()
table=[]
phi_initial=[]
if n==0:
if 4<(2*(lmbd_x+disp_x)+lmbd_y+disp_y)\
or (4==(2*(lmbd_x+disp_x)+lmbd_y+disp_y) and a(4,lmbd_x,lmbd_y,disp_x,disp_y)<1)\
or (4==(2*(lmbd_x+disp_x)+lmbd_y+disp_y) and a(4,lmbd_x,lmbd_y,disp_x,disp_y)==1)\
or (disp_x==0 and disp_y==3) or (disp_x==1 and disp_y==1):
phi_initial=[]
elif disp_x==0 and disp_y==0:#CASE 1
phi_initial=PHI_123_n_finder(precision,lmbd_x,lmbd_y,disp_x,disp_y)
elif disp_x==0 and disp_y==1:#CASE 2
phi_initial=PHI_12_n_finder(precision,lmbd_x,lmbd_y,disp_x,disp_y)
elif (disp_x==1 and disp_y==0) or (disp_x==0 and disp_y==2):#CASE 3
phi_initial=PHI_1_n_finder(precision,lmbd_x,lmbd_y,disp_x,disp_y)
else:
phi_initial=[]
```

```
else:
print('---Selection manual, n =',n,'taken.---')
if 4<(2*(lmbd_x+disp_x)+lmbd_y+disp_y)\
or (4==(2*(lmbd_x+disp_x)+lmbd_y+disp_y) and a(4,lmbd_x,lmbd_y,disp_x,disp_y)<1)\
or (4==(2*(lmbd_x+disp_x)+lmbd_y+disp_y) and a(4,lmbd_x,lmbd_y,disp_x,disp_y)==1)\
or (disp_x==0 and disp_y==3) or (disp_x==1 and disp_y==1):
phi_initial=[]
elif disp_x==0 and disp_y==0:#CASE 1
phi_initial=PHI_123(n,lmbd_x,lmbd_y,disp_x,disp_y)
elif disp_x==0 and disp_y==1:#CASE 2
phi_initial=PHI_12(n,lmbd_x,lmbd_y,disp_x,disp_y)
elif (disp_x==1 and disp_y==0) or (disp_x==0 and disp_y==2):#CASE 3
phi_initial=PHI_1(n,lmbd_x,lmbd_y,disp_x,disp_y)
else:
phi_initial=[]
print('test\n',phi_initial)
for T in range(1,T_max+1):
print('[',timedelta(seconds=timer()),'][ T =',T,']')
t_list = []
for u in range(0,u_max+1):
print('\t[',timedelta(seconds=timer()),'][ u =',u,']')
t_list.append(round(phi(u, T, lmbd_x, lmbd_y, disp_x, disp_y),3))
table.append(t_list)
print('[',timedelta(seconds=timer()),'][ T = oo]')
u_inf = []
for u in range(0,u_max+1):
u_inf.append(round(phi_inf(u, lmbd_x, lmbd_y, disp_x, disp_y, phi_initial),3))
table.append(u_inf)
print('{Elapsed time:',timedelta(seconds=timer()-t1),'}')
df=pd.DataFrame(table,columns=['u = '+str(x) for x in range(u_max+1)],index=['T = '+str(x) for x
print(df)
csv_location = 'Examples\\U_'+str(u_max)+'__T_'+str(T_max)+'__X_'+str(lmbd_x)+'_'+str(disp_x)+'__
df.to_csv(csv_location,sep=';')
return df

#calculating survival probabilities for examples:
#case 1 - n=18; lmbd_x=1,lmbd_y=1,disp_x=0,disp_y=0
example_table(u_max=20,T_max=15,lmbd_x=1,lmbd_y=1,disp_x=0,disp_y=0,n=20)
#case 2 - n=19; lmbd_x=1,lmbd_y=0.9,disp_x=0,disp_y=1
example_table(u_max=20,T_max=15,lmbd_x=1,lmbd_y=0.9,disp_x=0,disp_y=1,n=20)
#case 3 - n=17; lmbd_x=0.5,lmbd_y=0.9,disp_x=1,disp_y=0
example_table(u_max=20,T_max=15,lmbd_x=0.5,lmbd_y=0.9,disp_x=1,disp_y=0,n=20)
#case 4 - n=0; lmbd_x=0.25,lmbd_y=0.4,disp_x=1,disp_y=1
example_table(u_max=20,T_max=15,lmbd_x=0.25,lmbd_y=0.4,disp_x=1,disp_y=1)
#case 5 - n=0; lmbd_x=1,lmbd_y=1,disp_x=1,disp_y=0
example_table(u_max=20,T_max=15,lmbd_x=1,lmbd_y=1,disp_x=1,disp_y=0)
###
```