УДК 624.07::519.83+004.42

В.В. РОМАНЮК
Хмельницький національний університет

ОБЧИСЛЕННЯ ОПТИМАЛЬНИХ ПЛОЩ ПОПЕРЕЧНИХ ПЕРЕРІЗІВ У КОНСТРУКЦІЇ З ТРЬОМА ОПОРАМИ ЗА УМОВ ЧАСТКОВОЇ НЕВИЗНАЧЕНОСТІ СТИСКАЮЧИХ ЗУСИЛЬ НА ЕЛЕМЕНТИ З ПОЗДОВЖНЬОЮ СТІЙКІСТЮ

Систематизовано дані про оптимальні стратегії проектувальника у моделі дії нормованого одиничного навантаження на конструкцію з трьома опорами за умов часткової невизначеності стискаючих зусиль. Побудовано програмний засіб для визначення оптимальних площ поперечних перерізів опор такої конструкції.

There have been systematized data on projector optimal strategies in the model of the normed unit load action on three supports construction within partial uncertainties of compression forces. A program tool has been developed for determining the optimal squares of cross-sections in supports of such construction.

Ключові слова: часткова невизначеність, конструкція з опорами, нормоване (одиничне) навантаження, стискаюче зусилля, ігрове антагоністичне моделювання, антагоністична гра, оптимальна стратегія (проектувальника).

Про актуальність проблеми у загальному виді

Задачі та труднощі, котрі виникають у будівельній механіці і машинобудуванні, є завжди актуальними, потребують швидкого вирішення, а також часто вимагають безкомпромісних рішень, що пов'язано з безпекою. Проектування будівельних або машинних конструкцій-опор є саме тією задачею, де недопустимі розв'язки у формі імовірнісних розподілів чи відповідних математичних (геометричних) середніх. Прийнятним є лише представлення рішень до такою задачі з гарантованим (максимінним) результатом, де оптимізація витрат проектування відбувається на фоні забезпечення цілковитої надійності. Такий результат отримується часто за допомогою ігрового антагоністичного моделювання, де першого гравця персоніфікують випадкові обставини (природні або ті, які важко врахувати у моделі). Й отримання чистих оптимальних стратегій для другого гравця (проектувальника) є актуальною проблемою при вирішенні описаних задач проектування.

Аналіз останніх досліджень і публікацій за предметом дослідження та окреслення питання щодо усунення часткових невизначеностей у задачах будівельної механіки і машинобудування

Задача оптимального використання і розподілу будівельних ресурсів у конструкції-опорі, що складається з декількох елементів однакової геометричної форми, на які діє відоме навантаження, є відомою за джерелами [1 – 4]. Ефект стискаючого зусилля або навантаження, що діє на опори, можна моделювати за допомогою антагоністичної гри наступним чином. По-перше, вважається, що загальне навантаження, котре діятиме на конструкцію-опору, є відомим, і воно нормується до одиничного [2 – 4]. По-друге, частка від граничного навантаження на i-й елемент будівельної конструкції з площею поперечного перерізу y_i, на який діє стискаюче зусилля x_i, знаходиться як

$$T_i\left(x_i,\, y_i\right) = \beta \frac{x_i}{y_i^2}, \tag{1}$$

де β є коефіцієнтом, куди включено механічні властивості матеріалу i-ї колони [1, с. 144]. Усі можливі нормовані навантаження, які можуть стискувати i-ту опору, знаходяться у межах сегмента

$$\left[a_i;\, b_i\right] \subset \left(0;\, 1\right) \subset \left[0;\, 1\right]. \tag{2}$$

Ненульова міра сегмента $\left[a_i;\, b_i\right]$ й означає часткову невизначеність того стискаюче зусилля x_i, котре діятиме на i-ту опору. У роботі [2] представлено модель задачі про розрахунок повздовжньої стійкості двох опор будівельної конструкції при дії на них нормованого стискаючого зусилля у формі строго опуклої антагоністичної гри з функцією виграшу на підмножині одиничного квадрата, де для кожного випадку співвідношення кінців сегмента чистих стратегій другого гравця, у ролі якого виступає проектувальник, знайдено його чисту оптимальну стратегію. У роботі [3] у формі антагоністичної гри на підмножині одиничного гіперкуба в $\mathbb{R}^4$ представлено модель визначення оптимальних витрат будівельних ресурсів при проектуванні опорної конструкції з трьома колонами. Там же обґрунтовано строгу опуклість представленої гри і визначено оптимальну стратегію проектувальника за усіх можливих варіантів співвідношень із мінімальним та максимальним навантаженням на кожну з двох колон. Твердження про ситуації рівноваги у чистих стратегіях та деякі особливі випадки розглянуті у [4]. Утім, існує потреба у побудові програмного засобу для швидкого визначення результатів усунення часткової невизначеності типу (2) для i-го елемента конструкції-опори. Це пов'язано насамперед з тим, що кінці сегмента $\left[a_i;\, b_i\right]$ теж неможливо зафіксувати однозначно.

Формулювання мети і постановка завдань статті

Програмування результатів усунення часткових невизначеностей в конструкції з двома опорами є недоцільним, оскільки відповідні обчислення є практично "калькуляторними" Тому будуватимемо програмний засіб для визначення оптимальних площ поперечних перерізів у конструкції з трьома опорами. Для цього спочатку систематизуємо дані про оптимальні стратегії проектувальника у моделі дії нормованого одиничного навантаження на триколонну будівельну конструкцію, після чого розроблятимемо обчислювальний MATLAB-засіб.

Попередні відомості про оптимальні стратегії проектувальника у моделі дії нормованого одиничного навантаження на триколонну будівельну конструкцію

Нехай має місце (2) $\forall\ i = \overline{1,\ 3}$. Тоді достатньо мати дані про невизначеності у стискаючих зусиллях на дві з трьох опор –

$$x_1 \in [a_1;\, b_1] \text{ та } x_2 \in [a_2;\, b_2], \tag{3}$$

звідки

$$x_3 = 1 - (x_1 + x_2). \tag{4}$$

Сумарна площа поперечних перерізів трьох опор теж нормована й

$$y_1 \in [a_1;\, b_1],\ y_2 \in [a_2;\, b_2],\ y_3 = 1 - (y_1 + y_2). \tag{5}$$

Мають місце умови

$$0 < a_1 < b_1 < 1,\ 0 < a_2 < b_2 < 1,\ b_1 + b_2 < 1 \tag{6}$$

завдяки тому, що міра

$$\mu_{\mathbb{R}}\left([a_i;\, b_i]\right) > 0 \quad \forall\ i = \overline{1,\ 3}. \tag{7}$$

Ядром побудованої в [3] антагоністичної гри є гіперповерхня

$$\begin{aligned} T(\mathbf{X},\, \mathbf{Y}) &= T(x_1,\, x_2;\, y_1,\, y_2) = \max\left\{T_1(x_1,\, y_1),\, T_2(x_2,\, y_2),\, T_3(x_3,\, y_3)\right\} = \\ &= \max\left\{\beta\frac{x_1}{y_1^2},\, \beta\frac{x_2}{y_2^2},\, \beta\frac{1-x_1-x_2}{(1-y_1-y_2)^2}\right\} = \beta\max\left\{\frac{x_1}{y_1^2},\, \frac{x_2}{y_2^2},\, \frac{1-x_1-x_2}{(1-y_1-y_2)^2}\right\}, \end{aligned} \tag{8}$$

котра задається на паралелепіпеді

$$\boldsymbol{X} \times \boldsymbol{Y} = \left\{[a_1;\, b_1] \times [a_2;\, b_2]\right\} \times \left\{[a_1;\, b_1] \times [a_2;\, b_2]\right\} = \prod_{s=1}^{2}[a_1;\, b_1] \times [a_2;\, b_2] \subset \prod_{k=1}^{4}(0;\, 1) \subset \prod_{k=1}^{4}[0;\, 1] \subset \mathbb{R}^4 \tag{9}$$

з умовами (3) – (6). Перед проектувальником стоїть задача вибрати площу поперечного перерізу кожної з трьох опор так, щоб мінімізувати максимально можливий дисбаланс [1, с. 130, с. 144] у навантаженні (стискаючому зусиллі) на них. І саме обмеження по верхньому значенню площі поперечного перерізу кожної опори або обмеження по сумарній кількості наявного матеріалу, з якого виготовляють опори, і породжує антагоністичну гру [2, 3].

Обґрунтування строгої опуклості гри з ядром (8) на паралелепіпеді (9) відбулось в [3, с. 19]. Тому гра з ядром (8) на паралелепіпеді (9) з умовами (3) – (6) є строго опуклою, й у ній проектувальник має єдину чисту оптимальну стратегію

$$\mathbf{Y}_* = \begin{bmatrix} y_1^* & y_2^* \end{bmatrix} \in [a_1;\, b_1] \times [a_2;\, b_2]. \tag{10}$$

Компонентами оптимальної стратегії проектувальника є [3, с. 20]

$$y_1^* = \frac{\sqrt{b_1}}{\sqrt{b_1} + \sqrt{b_2} + \sqrt{1 - a_1 - a_2}}, \tag{11}$$

$$y_2^* = \frac{\sqrt{b_2}}{\sqrt{b_1} + \sqrt{b_2} + \sqrt{1 - a_1 - a_2}}, \tag{12}$$

проте (11) і (12) мають місце лише за умов

$$\frac{\sqrt{b_1}}{\sqrt{b_1} + \sqrt{b_2} + \sqrt{1 - a_1 - a_2}} \in [a_1; b_1], \tag{13}$$

$$\frac{\sqrt{b_2}}{\sqrt{b_1} + \sqrt{b_2} + \sqrt{1 - a_1 - a_2}} \in [a_2; b_2]. \tag{14}$$

Завдяки опуклості гіперповерхні (8) по змінним y_1 та y_2 при

$$\frac{\sqrt{b_1}}{\sqrt{b_1} + \sqrt{b_2} + \sqrt{1 - a_1 - a_2}} < a_1 \tag{15}$$

буде $y_1^* = a_1$. Так само при

$$\frac{\sqrt{b_2}}{\sqrt{b_1} + \sqrt{b_2} + \sqrt{1 - a_1 - a_2}} < a_2 \tag{16}$$

буде $y_2^* = a_2$, при

$$\frac{\sqrt{b_1}}{\sqrt{b_1} + \sqrt{b_2} + \sqrt{1 - a_1 - a_2}} > b_1 \tag{17}$$

буде $y_1^* = b_1$, а при

$$\frac{\sqrt{b_2}}{\sqrt{b_1} + \sqrt{b_2} + \sqrt{1 - a_1 - a_2}} > b_2 \tag{18}$$

буде $y_2^* = b_2$. Отже, за умов (13) і (14) оптимальною стратегією проектувальника є (її компоненти можна називати незміщеними або регулярними)

$$\mathbf{Y}_* = \begin{bmatrix} \dfrac{\sqrt{b_1}}{\sqrt{b_1} + \sqrt{b_2} + \sqrt{1 - a_1 - a_2}} & \dfrac{\sqrt{b_2}}{\sqrt{b_1} + \sqrt{b_2} + \sqrt{1 - a_1 - a_2}} \end{bmatrix}. \tag{19}$$

При невиконанні хоча б одного зі співвідношень (13) і (14) в [3] враховано наступні випадки. Для випадку $y_1^* = a_1$ при (14) та (15) за умови $y_2^{*\langle 1 \rangle} \in [a_2; b_2)$, де (зміщена компонента)

$$y_2^{*\langle 1 \rangle} = \frac{(1 - a_1)\sqrt{b_2}}{\sqrt{b_2} + \sqrt{1 - a_1 - a_2}}, \tag{20}$$

буде

$$\mathbf{Y}_* = \begin{bmatrix} a_1 & \dfrac{(1 - a_1)\sqrt{b_2}}{\sqrt{b_2} + \sqrt{1 - a_1 - a_2}} \end{bmatrix}. \tag{21}$$

При $y_2^{*\langle 1 \rangle} < a_2$ оптимальною стратегією проектувальника буде

$$\mathbf{Y}_* = [a_1 \quad a_2]. \tag{22}$$

При одночасному виконанні (13) і (16) за умови $y_1^{*\langle 1 \rangle} \in [a_1; b_1)$, де (зміщена компонента)

$$y_1^{*\langle 1\rangle} = \frac{(1-a_2)\sqrt{b_1}}{\sqrt{b_1}+\sqrt{1-a_1-a_2}}, \tag{23}$$

буде

$$\mathbf{Y}_* = \begin{bmatrix} \dfrac{(1-a_2)\sqrt{b_1}}{\sqrt{b_1}+\sqrt{1-a_1-a_2}} & a_2 \end{bmatrix}. \tag{24}$$

А при $y_1^{*\langle 1\rangle} < a_1$ оптимальною стратегією проектувальника буде (22). При одночасному виконанні (15) і (16) стратегія (22) залишається оптимальною, а при одночасному виконанні (17) і (18) оптимальна стратегія проектувальника

$$\mathbf{Y}_* = \begin{bmatrix} b_1 & b_2 \end{bmatrix}. \tag{25}$$

При одночасному виконанні (14) і (17) оптимальна стратегія проектувальника

$$\mathbf{Y}_* = \begin{bmatrix} b_1 & \dfrac{\sqrt{b_2}}{\sqrt{b_1}+\sqrt{b_2}+\sqrt{1-a_1-a_2}} \end{bmatrix}. \tag{26}$$

У симетричному випадку, коли одночасно виконані (13) і (18), оптимальна стратегія проектувальника

$$\mathbf{Y}_* = \begin{bmatrix} \dfrac{\sqrt{b_1}}{\sqrt{b_1}+\sqrt{b_2}+\sqrt{1-a_1-a_2}} & b_2 \end{bmatrix}. \tag{27}$$

При одночасному виконанні (15) і (18) за умови

$$\frac{1}{b_2} \geqslant \frac{1-a_1-a_2}{(1-a_1-b_2)^2} \tag{28}$$

оптимальною стратегією проектувальника є

$$\mathbf{Y}_* = \begin{bmatrix} a_1 & b_2 \end{bmatrix}. \tag{29}$$

У випадку виконання нерівності

$$\frac{1}{b_2} < \frac{1-a_1-a_2}{(1-a_1-b_2)^2} \tag{30}$$

при $y_2^{*\langle 1\rangle} \in [a_2;\, b_2)$ оптимальною стратегією проектувальника буде (21), а при $y_2^{*\langle 1\rangle} < a_2$ вектор (22) є його оптимальною стратегією. При одночасному виконанні (16) і (17) міркування є симетричними. При

$$\frac{1}{b_1} \geqslant \frac{1-a_1-a_2}{(1-b_1-a_2)^2} \tag{31}$$

оптимальна стратегія проектувальника

$$\mathbf{Y}_* = \begin{bmatrix} b_1 & a_2 \end{bmatrix}. \tag{32}$$

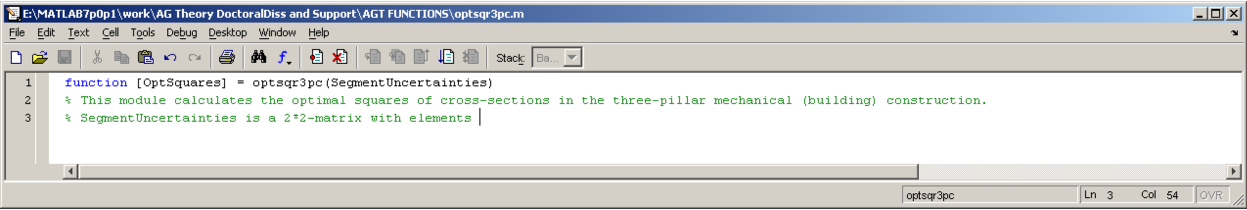
У випадку виконання нерівності

$$\frac{1}{b_1} < \frac{1-a_1-a_2}{(1-b_1-a_2)^2} \tag{33}$$

при $y_1^{*\langle 1\rangle} \in [a_1;\, b_1)$ буде (24), а при $y_1^{*\langle 1\rangle} < a_1$ оптимальною стратегією проектувальника буде (22).

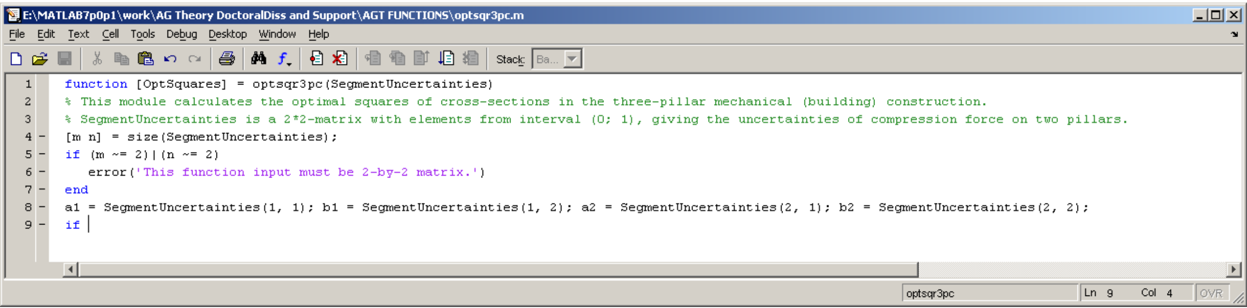
Програмний засіб для визначення оптимальних площ поперечних перерізів у конструкції з трьома опорами за умов часткової невизначеності стискаючих зусиль

Обираємо програмне середовище MATLAB для побудови засобу обчислення одного з дев'яти випадків оптимальної стратегії проектувальника (19), (21), (22), (24) – (27), (29), (32). Цей MATLAB-засіб назвемо **`optsqr3pc`** (optimal squares in three-pillar construction). Його вхідним аргументом будуть, очевидно, чотири числа a_1, b_1, a_2, b_2, котрі окреслюють умови часткової невизначеності. Ці числа доцільно уводити як 2×2 -матрицю $\begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix}$. На рис. 1 – 4 показано початок запису програмного коду до моменту обчислення значень (11) і (12).



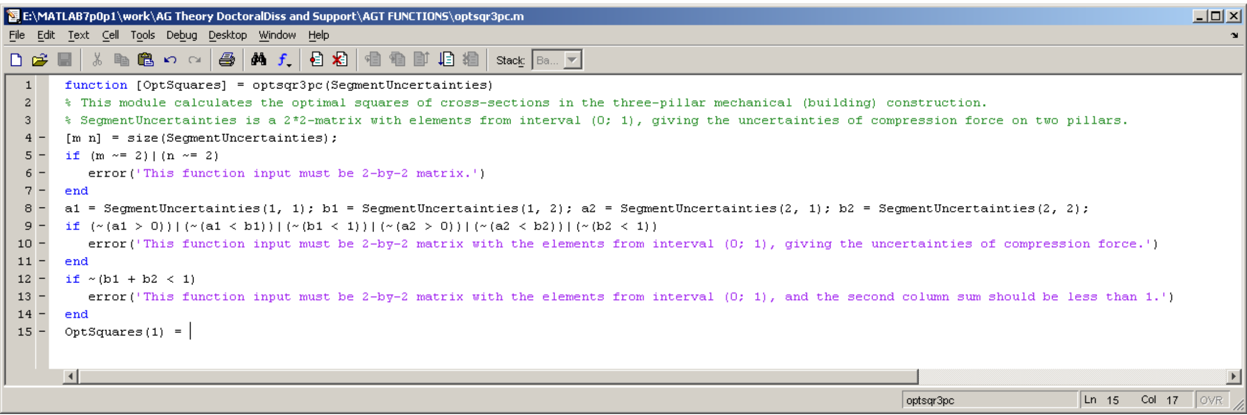
```
1  function [OptSquares] = optsqr3pc(SegmentUncertainties)
2  % This module calculates the optimal squares of cross-sections in the three-pillar mechanical (building) construction.
3  % SegmentUncertainties is a 2*2-matrix with elements
```

Рис. 1. Заголовок MATLAB-засобу (MATLAB-функції) `optsqr3pc` і набирання опису (пояснення) до мети його використання



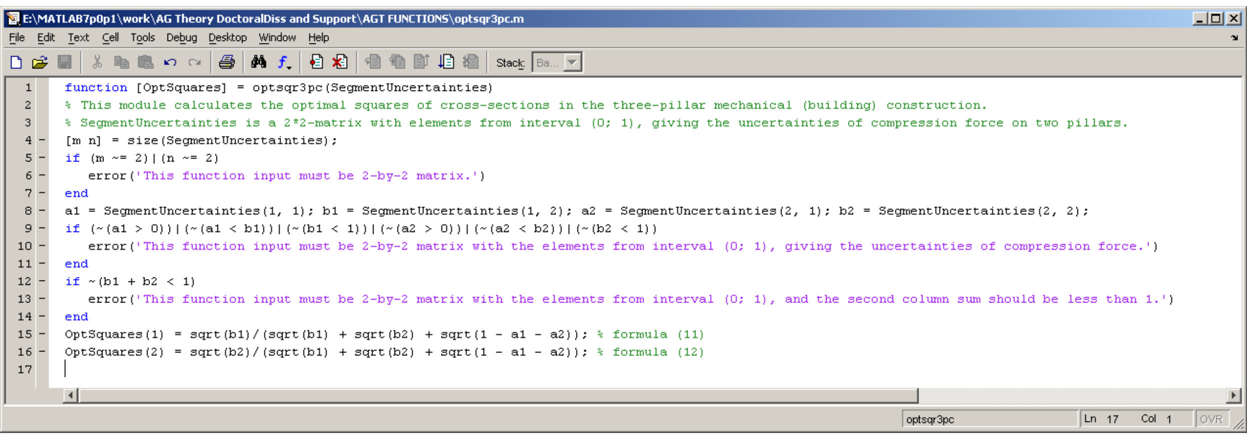
```
1  function [OptSquares] = optsqr3pc(SegmentUncertainties)
2  % This module calculates the optimal squares of cross-sections in the three-pillar mechanical (building) construction.
3  % SegmentUncertainties is a 2*2-matrix with elements from interval (0; 1), giving the uncertainties of compression force on two pillars.
4  [m n] = size(SegmentUncertainties);
5  if (m ~= 2)|(n ~= 2)
6     error('This function input must be 2-by-2 matrix.')
7  end
8  a1 = SegmentUncertainties(1, 1); b1 = SegmentUncertainties(1, 2); a2 = SegmentUncertainties(2, 1); b2 = SegmentUncertainties(2, 2);
9  if
```

Рис. 2. Аналіз розміру вхідного аргументу MATLAB-засобу `optsqr3pc` як матриці з кінцями сегментів $[a_1; b_1]$ й $[a_2; b_2]$ (рядки 4 – 7) та запис цих кінців у зручній формі (восьмий рядок)



```
1  function [OptSquares] = optsqr3pc(SegmentUncertainties)
2  % This module calculates the optimal squares of cross-sections in the three-pillar mechanical (building) construction.
3  % SegmentUncertainties is a 2*2-matrix with elements from interval (0; 1), giving the uncertainties of compression force on two pillars.
4  [m n] = size(SegmentUncertainties);
5  if (m ~= 2)|(n ~= 2)
6     error('This function input must be 2-by-2 matrix.')
7  end
8  a1 = SegmentUncertainties(1, 1); b1 = SegmentUncertainties(1, 2); a2 = SegmentUncertainties(2, 1); b2 = SegmentUncertainties(2, 2);
9  if (~(a1 > 0))|(~(a1 < b1))|(~(b1 < 1))|(~(a2 > 0))|(~(a2 < b2))|(~(b2 < 1))
10    error('This function input must be 2-by-2 matrix with the elements from interval (0; 1), giving the uncertainties of compression force.')
11 end
12 if ~(b1 + b2 < 1)
13    error('This function input must be 2-by-2 matrix with the elements from interval (0; 1), and the second column sum should be less than 1.')
14 end
15 OptSquares(1) =
```

Рис. 3. Перевірка (рядки 9 – 14) уведених кінців сегментів $[a_1; b_1]$ й $[a_2; b_2]$ на виконання умов (6)



```
1  function [OptSquares] = optsqr3pc(SegmentUncertainties)
2  % This module calculates the optimal squares of cross-sections in the three-pillar mechanical (building) construction.
3  % SegmentUncertainties is a 2*2-matrix with elements from interval (0; 1), giving the uncertainties of compression force on two pillars.
4  [m n] = size(SegmentUncertainties);
5  if (m ~= 2)|(n ~= 2)
6     error('This function input must be 2-by-2 matrix.')
7  end
8  a1 = SegmentUncertainties(1, 1); b1 = SegmentUncertainties(1, 2); a2 = SegmentUncertainties(2, 1); b2 = SegmentUncertainties(2, 2);
9  if (~(a1 > 0))|(~(a1 < b1))|(~(b1 < 1))|(~(a2 > 0))|(~(a2 < b2))|(~(b2 < 1))
10    error('This function input must be 2-by-2 matrix with the elements from interval (0; 1), giving the uncertainties of compression force.')
11 end
12 if ~(b1 + b2 < 1)
13    error('This function input must be 2-by-2 matrix with the elements from interval (0; 1), and the second column sum should be less than 1.')
14 end
15 OptSquares(1) = sqrt(b1)/(sqrt(b1) + sqrt(b2) + sqrt(1 - a1 - a2)); % formula (11)
16 OptSquares(2) = sqrt(b2)/(sqrt(b1) + sqrt(b2) + sqrt(1 - a1 - a2)); % formula (12)
17
```

Рис. 4. Обчислення (рядки 15 і 16) значень (11) і (12)

Після того, як значення (11) і (12) обчислені у рядках 15 і 16, відбувається систематизація умов й обчислення (рис. 5, рис. 6) одного з дев'яти випадків оптимальної стратегії проектувальника (19), (21), (22), (24) – (27), (29), (32). Відповідні коментарі дозволяють легко ідентифікувати частини коду з цими умовами і формулою для (10). Приклади (рис. 7) є демонстрацією працездатності розробленого MATLAB-засобу.

E:\MATLAB7p0p1\work\AG Theory DoctoralDiss and Support\AGT FUNCTIONS\optsqr3pc.m

```
1    function [OptSquares] = optsqr3pc(SegmentUncertainties)
2    % This module calculates the optimal squares of cross-sections in the three-pillar mechanical (building) construction.
3    % SegmentUncertainties is a 2*2-matrix with elements from interval (0; 1), giving the uncertainties of compression force on two pillars.
4    [m n] = size(SegmentUncertainties);
5    if (m ~= 2)|(n ~= 2)
6       error('This function input must be 2-by-2 matrix.')
7    end
8    a1 = SegmentUncertainties(1, 1); b1 = SegmentUncertainties(1, 2); a2 = SegmentUncertainties(2, 1); b2 = SegmentUncertainties(2, 2);
9    if (~(a1 > 0))|(~(a1 < b1))|(~(b1 < 1))|(~(a2 > 0))|(~(a2 < b2))|(~(b2 < 1))
10      error('This function input must be 2-by-2 matrix with the elements from interval (0; 1), giving the uncertainties of compression force.')
11   end
12   if ~(b1 + b2 < 1)
13      error('This function input must be 2-by-2 matrix with the elements from interval (0; 1), and the second column sum should be less than 1.')
14   end
15   OptSquares(1) = sqrt(b1)/(sqrt(b1) + sqrt(b2) + sqrt(1 - a1 - a2)); % formula (11)
16   OptSquares(2) = sqrt(b2)/(sqrt(b1) + sqrt(b2) + sqrt(1 - a1 - a2)); % formula (12)
17   if (OptSquares(1) >= a1) & (OptSquares(1) <= b1) & (OptSquares(2) >= a2) & (OptSquares(2) <= b2) % checking (13) and (14)
18      OptSquares(3) = 1 - OptSquares(1) - OptSquares(2);
19      disp('   The optimal squares are regular.')
20      return
21   else
22      y1 = (1 - a2)*sqrt(b1)/(sqrt(b1) + sqrt(1 - a1 - a2)); % formula (23)
23      y2 = (1 - a1)*sqrt(b2)/(sqrt(b2) + sqrt(1 - a1 - a2)); % formula (20)
24      if (OptSquares(1) < a1) & ((OptSquares(2) >= a2) & (OptSquares(2) <= b2)) % case with true (14) and (15)
25         OptSquares(1) = a1;
26         if (y2 >= a2) & (y2 < b2) % checking whether (20) belongs to [a2; b2)
27            OptSquares(2) = y2; % projector optimal strategy is (21)
28            OptSquares(3) = 1 - OptSquares(1) - OptSquares(2);
29            disp('   The first optimal square is moved to the left uncertainty end, and the second is less than regular.')
30            return
31         else
32            OptSquares(2) = a2; % projector optimal strategy is (22)
33            OptSquares(3) = 1 - OptSquares(1) - OptSquares(2);
34            disp('   Both first and second optimal squares are moved to the left ends of the given uncertainties.')
35            return
36         end
37      end
38      if ((OptSquares(1) >= a1) & (OptSquares(1) <= b1)) & (OptSquares(2) < a2) % case with true (13) and (16)
39         OptSquares(2) = a2;
40         if (y1 >= a1) & (y1 < b1) % checking whether (23) belongs to [a1; b1)
41            OptSquares(1) = y1; % projector optimal strategy is (24)
42            OptSquares(3) = 1 - OptSquares(1) - OptSquares(2);
43            disp('   The first optimal square is less than regular, and the second is moved to the left uncertainty end.')
44            return
45         else
46            OptSquares(1) = a1; % projector optimal strategy is (22)
47            OptSquares(3) = 1 - OptSquares(1) - OptSquares(2);
48            disp('   Both first and second optimal squares are moved to the left ends of the given uncertainties.')
49            return
```

optsqr3pc Ln 49 Col 16

E:\MATLAB7p0p1\work\AG Theory DoctoralDiss and Support\AGT FUNCTIONS\optsqr3pc.m

```
49            return
50         end
51      end
52      if (OptSquares(1) < a1) & (OptSquares(2) < a2) % case with true (15) and (16)
53         OptSquares(1) = a1;
54         OptSquares(2) = a2; % projector optimal strategy is (22)
55         OptSquares(3) = 1 - OptSquares(1) - OptSquares(2);
56         disp('   Both first and second optimal squares are moved to the left ends of the given uncertainties.')
57         return
58      end
59      if (OptSquares(1) > b1) & (OptSquares(2) > b2) % case with true (17) and (18)
60         OptSquares(1) = b1;
61         OptSquares(2) = b2; % projector optimal strategy is (25)
62         OptSquares(3) = 1 - OptSquares(1) - OptSquares(2);
63         disp('   Both first and second optimal squares are moved to the right ends of the given uncertainties.')
64         return
65      end
66      if (OptSquares(1) > b1) & ((OptSquares(2) >= a2) & (OptSquares(2) <= b2)) % case with true (14) and (17)
67         OptSquares(1) = b1; % projector optimal strategy is (26)
68         OptSquares(3) = 1 - OptSquares(1) - OptSquares(2);
69         disp('   The first optimal square is moved to the right uncertainty end, and the second is regular.')
70         return
71      end
72      if ((OptSquares(1) >= a1) & (OptSquares(1) <= b1)) & (OptSquares(2) > b2) % case with true (13) and (18)
73         OptSquares(2) = b2; % projector optimal strategy is (27)
74         OptSquares(3) = 1 - OptSquares(1) - OptSquares(2);
75         disp('   The first optimal square is regular, and the second is moved to the right uncertainty end.')
76         return
77      end
78      if (OptSquares(1) < a1) & (OptSquares(2) > b2) % case with true (15) and (18)
79         if 1/b2 >= (1 - a1 - a2)/((1 - a1 - b2)^2) % checking whether (28) is true
80            OptSquares(1) = a1;
81            OptSquares(2) = b2; % projector optimal strategy is (29)
82            OptSquares(3) = 1 - OptSquares(1) - OptSquares(2);
83            disp('   The first optimal square is moved to the left uncertainty end, and the second is moved to the right uncertainty end.')
84            return
85         else
86            if (y2 >= a2) & (y2 < b2) % checking whether (20) belongs to [a2; b2)
```

optsqr3pc Ln 86 Col 79

Рис. 5. Систематизація умов й обчислення (починаючи з 17-го рядка) одного з дев'яти випадків оптимальної стратегії (10)

```
E:\MATLAB7p0p1\work\AG Theory DoctoralDiss and Support\AGT FUNCTIONS\optsqr3pc.m
            if (y2 >= a2) & (y2 < b2) % checking whether (20) belongs to [a2; b2)
                OptSquares(1) = a1;
                OptSquares(2) = y2; % projector optimal strategy is (21)
                OptSquares(3) = 1 - OptSquares(1) - OptSquares(2);
                disp('   The first optimal square is moved to the left uncertainty end, and the second is less than regular.')
                return
            else
                OptSquares(1) = a1;
                OptSquares(2) = a2; % projector optimal strategy is (22)
                OptSquares(3) = 1 - OptSquares(1) - OptSquares(2);
                disp('   Both first and second optimal squares are moved to the left ends of the given uncertainties.')
                return
            end
        end
    end
    if (OptSquares(1) < a2) & (OptSquares(2) > b1) % case with true (16) and (17)
        if 1/b1 >= (1 - a1 - a2)/((1 - b1 - a2)^2) % checking whether (31) is true
            OptSquares(1) = b1;
            OptSquares(2) = a2; % projector optimal strategy is (32)
            OptSquares(3) = 1 - OptSquares(1) - OptSquares(2);
            disp('   The first optimal square is moved to the right uncertainty end, and the second is moved to the left uncertainty end.')
            return
        else
            if (y1 >= a1) & (y1 < b1) % checking whether (23) belongs to [a1; b1)
                OptSquares(1) = y1;
                OptSquares(2) = a2; % projector optimal strategy is (24)
                OptSquares(3) = 1 - OptSquares(1) - OptSquares(2);
                disp('   The first optimal square is less than regular, and the second is moved to the left uncertainty end.')
                return
            else
                OptSquares(1) = a1;
                OptSquares(2) = a2; % projector optimal strategy is (22)
                OptSquares(3) = 1 - OptSquares(1) - OptSquares(2);
                disp('   Both first and second optimal squares are moved to the left ends of the given uncertainties.')
                return
            end
        end
    end
end
```

Рис. 6. Систематизація умов й обчислення (заключна частина) одного з дев'яти випадків оптимальної стратегії (10)

```
MATLAB
Current Directory: E:\MATLAB7p0p1\work
>> OptSquares = optsqr3pc([0.01 0.6;0.1 0.3])
   The optimal squares are regular.
OptSquares =
    0.3419    0.2417    0.4164
>> OptSquares = optsqr3pc([0.2 0.4;0.1 0.3])
   The optimal squares are regular.
OptSquares =
    0.3136    0.2716    0.4148
>> OptSquares = optsqr3pc([0.2 0.21;0.4 0.41])
   The first optimal square is moved to the right uncertainty end, and the second is moved to the left uncertainty end.
OptSquares =
    0.2100    0.4000    0.3900
>> OptSquares = optsqr3pc([0.2 0.21;0.4 0.45])
   The first optimal square is moved to the right uncertainty end, and the second is moved to the left uncertainty end.
OptSquares =
    0.2100    0.4000    0.3900
>> OptSquares = optsqr3pc([0.2 0.21;0.4 0.6])
   The first optimal square is moved to the right uncertainty end, and the second is regular.
OptSquares =
    0.2100    0.4153    0.3747
>> OptSquares = optsqr3pc([0.3 0.6;0.4 0.5])
??? Error using ==> optsqr3pc
This function input must be 2-by-2 matrix with the elements from interval (0; 1), and the second column sum should be less than 1.

>> OptSquares = optsqr3pc([0.3 0.6;0.3 0.35])
   The first optimal square is less than regular, and the second is moved to the left uncertainty end.
OptSquares =
    0.3854    0.3000    0.3146
>> OptSquares = optsqr3pc([0.45 0.47;0.44 0.52])
   Both first and second optimal squares are moved to the left ends of the given uncertainties.
OptSquares =
    0.4500    0.4400    0.1100
>> OptSquares = optsqr3pc([0.45 0.47;0.44 0.47])
   Both first and second optimal squares are moved to the left ends of the given uncertainties.
OptSquares =
    0.4500    0.4400    0.1100
>> OptSquares = optsqr3pc([0.45 0.47;0.1 0.47])
   The first optimal square is moved to the left uncertainty end, and the second is less than regular.
OptSquares =
    0.4500    0.2780    0.2720
>> OptSquares = optsqr3pc([0.15 0.47;0.1 0.47])
   The optimal squares are regular.
OptSquares =
    0.3064    0.3064    0.3871
>> OptSquares = optsqr3pc([0.25 0.4;0.1 0.58])
   The optimal squares are regular.
OptSquares =
    0.2874    0.3461    0.3664
>>
```

Рис. 7. Приклади застосування розробленого MATLAB-засобу `optsqr3pc`

Висновок та перспектива подальшого дослідження

Розглянуті 11 прикладів обчислення оптимальних площ поперечних перерізів у конструкції з трьома опорами за умов часткової невизначеності стискаючих зусиль

$$x_1 \in [0.01; 0.6] \text{ та } x_2 \in [0.1; 0.3], \quad (34)$$

$$x_1 \in [0.2; 0.4] \text{ та } x_2 \in [0.1; 0.3], \quad (35)$$

$$x_1 \in [0.2; 0.21] \text{ та } x_2 \in [0.4; 0.41], \quad (36)$$

$$x_1 \in [0.2; 0.21] \text{ та } x_2 \in [0.4; 0.45], \quad (37)$$

$$x_1 \in [0.2; 0.21] \text{ та } x_2 \in [0.4; 0.6], \quad (38)$$

$$x_1 \in [0.3; 0.6] \text{ та } x_2 \in [0.3; 0.35], \quad (39)$$

$$x_1 \in [0.45; 0.47] \text{ та } x_2 \in [0.44; 0.52], \quad (40)$$

$$x_1 \in [0.45; 0.47] \text{ та } x_2 \in [0.44; 0.47], \quad (41)$$

$$x_1 \in [0.45; 0.47] \text{ та } x_2 \in [0.1; 0.47], \quad (42)$$

$$x_1 \in [0.15; 0.47] \text{ та } x_2 \in [0.1; 0.47], \quad (43)$$

$$x_1 \in [0.25; 0.4] \text{ та } x_2 \in [0.1; 0.58] \quad (44)$$

дають оптимальні стратегії проектувальнику (перші дві площі поперечних перерізів) та, загалом, нормовані оптимальні площі

$$\begin{gathered}
\{0.3419,\ 0.2417,\ 0.4164\}, \\
\{0.3136,\ 0.2716,\ 0.4148\}, \\
\{0.21,\ 0.4,\ 0.39\}, \\
\{0.21,\ 0.4,\ 0.39\}, \\
\{0.21,\ 0.4153,\ 0.3747\}, \\
\{0.3854,\ 0.3,\ 0.3146\}, \\
\{0.45,\ 0.44,\ 0.11\}, \\
\{0.45,\ 0.44,\ 0.11\}, \\
\{0.45,\ 0.278,\ 0.272\}, \\
\{0.3064,\ 0.3064,\ 0.3871\}, \\
\{0.2874,\ 0.3461,\ 0.3664\} \quad (45)
\end{gathered}$$

відповідно невизначеностям (34) – (44) і повністю підтверджують працездатність розробленого MATLAB-засобу **`optsqr3pc`**. Якщо потрібна більша точність, а помилки заокруглень у (45) виключити неможливо, то вона збільшується за допомогою MATLAB-команди “format long”. У подальшому подібні програмні продукти можуть розроблятись для більш складних задач обчислення площ поперечних перерізів, зокрема, для класичної конструкції з чотирма опорами, розташованими симетрично у вершинах прямокутника, але на які діятимуть $[a_i; b_i]$-невизначені стискаючі зусилля з умовами $\mu_{\mathbb{R}}([a_i; b_i]) > 0 \ \ \forall\, i = \overline{1, 4}$.

Література

1. Воробьёв Н. Н. Теория игр для экономистов-кибернетиков / Воробьёв Н. Н. – М. : Наука, Главная редакция физико-математической литературы, 1985. – 272 с.

2. Романюк В. В. Модель визначення оптимального рішення проектувальника у задачі про розрахунок повздовжньої стійкості двох елементів будівельної конструкції при дії на них нормованого стискаючого зусилля / В. В. Романюк // Проблеми трибології. – 2010. – № 1. – С. 42 – 56.

3. Романюк В. В. Моделювання дії нормованого одиничного навантаження на три колони однакової висоти у будівельній конструкції і знаходження оптимальної площі кожної опори / В. В. Романюк // Проблеми трибології. – 2010. – № 3. – С. 18 – 25.

4. Романюк В. В. Доведення тверджень для моделі дії нормованого одиничного навантаження на три колони однакової висоти у будівельній конструкції / В. В. Романюк // Проблеми трибології. – 2010. – № 4. – С. 72 – 81.

Надійшла 13.1.2011 р.