УДК 519.83

РОЗВ'ЯЗУВАННЯ КОМБІНАТОРНИХ ЗАДАЧ ІГРОВОГО ТИПУ З ОБМЕЖЕННЯМИ-ПЕРЕСТАВЛЕННЯМИ У ОБОХ ГРАВЦІВ: ІТЕРАЦІЙНИЙ МЕТОД

О.О. ЄМЕЦЬ, О.В. ОЛЬХОВСЬКА

Розглянуто постановку та математичну модель ігрової задачі сільськогосподарського виробництва з обмеженнями-переставленнями, що накладаються на стратегії обох гравців. Поширено ітераційний метод на задачі комбінаторної оптимізації ігрового типу з обмеженнями, що визначаються переставленнями на стратегії обох гравців. Метод ґрунтується на розігруванні гри, за умови, що кожен гравець прагне досягнути своєї мети. Запропоновано критерії зупинки та процедури визначення результату. На основі розробленого програмного продукту проведено обчислювальні експерименти, які показують наближення платежів до ціни гри, що дає можливість наближеного знаходження мішаних стратегій гравців. Проведено теоретичну та експериментальну оцінку кількості операцій запропонованого ітераційного алгоритму.

ВСТУП

Різноманітні комбінаторні оптимізаційні задачі часто виникають під час моделювання на виробництві, у сільськогосподарській діяльності тощо. Ці задачі потребують розв'язання, а тому розробки необхідних методів. Теорія комбінаторної оптимізації містить алгоритми та методи розв'язання задач такого типу [1–13]. Праці [7–13] присвячені дослідженню задач комбінаторної оптимізації ігрового типу. У [7–13] розглянуто розв'язування цього класу задач ітераційним методом за умови, коли на стратегії тільки одного гравця накладаються обмеження, що визначені переставленнями [7], або розміщеннями [10, 13]. У [7] сформульовано математичні моделі у вигляді задач комбінаторної оптимізації ігрового типу, на які накладаються обмеження, що визначаються переставленнями, на стратегії обох гравців. Такі задачі, як відомо з [7], зустрічаються на виробничих підприємствах, у сфері обслуговування, а тому потребують методів для свого розв'язання. Поки що таких методів не існує.

Мета роботи — поширення ітераційного методу [11, 13] для розв'язування задач комбінаторної оптимізації ігрового типу за умови, що на стратегії накладаються обмеження, які визначені переставленнями для обох гравців.

ПОСТАНОВКА ЗАДАЧІ

Розглянемо задачу з роботи [7]. Нехай є два підприємства, які займаються вирощуванням сільськогосподарських культур. Перше господарство на своїх m полях вирощує m різних культур. Поля різної площі, тому кількість кожної вирощеної культури залежить від того, на якому полі її буде посаджено. У другого господарства є відповідно l полів, на яких вирощується l різних культур. Кожне поле в обох господарствах засівають повністю однією культурою. Восени вирощену культуру реалізують, саме тому прибутки обох підприємств залежать від кількості вирощеної продукції кожним господарством. Потрібно скласти оптимальні плани вирощування культур в обох господарствах.

Складемо математичну модель задачі, що розглядається, пояснивши одночасно оптимальність плану.

Позначимо P_i^x — відношення площі i-го поля до суми площ усіх полів господарства. Вектор $P^x = (P_1^x, P_2^x, ..., P_m^x)$ показує, яку частину від загальної суми площ складає площа кожного з полів. Оскільки P^x — це частини від загальної суми площ, то $P_i^x \geq 0;\ \forall i \in J_m:\ \sum_{i=1}^{m} P_i^x = 1$, де m — натуральні числа $\{1,2,...,m\} = J_m$. Покладемо $X = (x_1, x_2, ..., x_m)$. Компонента вектора з номером i показує, яку частину від максимально можливої кількості продукції i-го типу виростили, тобто цей вектор характеризує обсяги вирощування продукції першим господарством.

На кожному з полів вирощується тільки одна культура, тобто компонента i вектора X буде відповідати тій компоненті з P^x, що відповідає полю, на якому вирощено культуру i-го типу. Отже, компоненти вектора X є переставленням компонент вектора $P^x: X = (x_1, x_2, ..., x_m) \in E_m(P^x)$, де $E_m(P^x)$ — множина переставлень із елементів вектора P^x. Для другого господарства введемо аналогічні вектори: $P^y = (P_1^y, P_2^y, ..., P_l^y)$ — вектор для якого виконується: $P_j^y \geq 0;\ \forall j \in J_l;\ \sum_{j=1}^{l} P_j^y = 1.$

Вектор $Y = (y_1, y_2, ..., y_l)$ характеризує обсяги вирощування продукції другим господарством, а отже компоненти цього вектора є переставленнями з $P^y: Y = (y_1, y_2, ..., y_l) \in E_l(P^y)$.

Складемо матрицю $A' = (a'_{ij})$, елемент a'_{ij} — показує перевищення (різницю) прибутків другого підприємства порівняно з першим тоді, коли перше господарство вирощувало на всіх своїх полях тільки i-ту культуру, а друге — тільки j-ту культуру. Позначимо $F(X,Y)$ функцію:

$$F(X,Y) = \sum_{i=1}^{m} x_i \sum_{j=1}^{l} a'_{ij} y_j = \sum_{i=1}^{m} \sum_{j=1}^{l} a'_{ij} x_i y_j . \qquad (1)$$

Функція прибутку є функцією переставлень X та Y.

Перший гравець — завдяки своїм можливостям вибору вектора X прагне мінімізувати свій максимальний програш, отже треба знайти таке X^* за якого досягається величина $\alpha = \min\limits_{X \in E_m(P_x)} \max\limits_{Y \in E_l(P_y)} F(X,Y)$, яку називатимемо нижньою ціною гри. Другий гравець, у свою чергу, завдяки вибору Y^* прагне максимізувати свій мінімальний виграш, тобто визначає величину $\beta = \max\limits_{Y \in E_l(P_y)} \min\limits_{X \in E_m(P_x)} F(X,Y)$, яку називатимемо верхньою ціною гри.

Таким чином, можна записати математичну модель цієї задачі, яка вперше була розглянута в [7].

Знайти оптимальні стратегії гравців X^* та Y^*, де $X^* = \arg\min\limits_{X \in E_m(P^x)} F_x(X)$; $F_x(X) = \max\limits_{Y \in E_l(P^y)} F(X,Y)$; $Y^* = \arg\max\limits_{Y \in E_l(P^y)} F_y(Y)$; $F_y(Y) = \min\limits_{X \in E_m(P^x)} F(X,Y)$. За обмежень:

- $X = (x_1, x_2, ..., x_m) \in E_m(P^x)$, $X = (x_1, ... x_m) : \sum\limits_{i=1}^{m} x_i = 1$;
- вектор $P^x = (P_1^x, P_2^x, ..., P_m^x)$, задовольняє умовам $P_i^x \geq 0 \;\; \forall i \in J_m$;
- $Y = (y_1, y_2, ..., y_l) \in E_l(P^y)$, $Y = (y_1, y_2, ..., y_l) : \sum\limits_{j=1}^{l} y_j = 1$;
- вектор $P^y = (P_1^y, P_2^y, ..., P_l^y)$, задовольняє умовам $P_j^y \geq 0 \;\; \forall j \in J_l$, де функція $F(X,Y)$ має вигляд (1), a_{ij} $(\forall i \in J_m \;\; \forall j \in J_l)$ — задані дійсні числа.

Задачі, які мають описану математичну модель, назвемо задачами комбінаторної оптимізації ігрового типу з переставними обмеженнями на стратегії обох гравців. Якщо існують X^* та Y^*, то вважатимемо, що задача зв'язок у чистих стратегіях. Інакше, як і у випадку класичної матричної гри, введемо в розгляд мішані стратегії. Для цього складемо нову платіжну матрицю $A = (a_{ij})$ вимірності $k \times n$, де $k = m!$, $n = l!$. Платіж a_{ij} у цій матриці нехай обчислюється так: $a_{ij} = \sum\limits_{j=1}^{n} \sum\limits_{t=1}^{m} a'_{tj} x_{it} y_{jt}$, $\forall i \in J_k$, $\forall j \in J_n$, де i — номер відповідного вектора $X_i = (x_{i1}, x_{i2}, ..., x_{im})$ — це переставлення з множини $E_m(P^x)$, а j — номер відповідного вектора $Y_j = (y_1, y_2, ..., y_l)$ — переставлення з множини $E_l(P^y)$. Таким чином, a_{ij} — платіж (перевищення прибутку) першого гравця над другим, коли перший гравець обирає стратегію-переставлення X_i, а другий — стратегію-переставлення Y_j.

Позначимо

$$S_k = \left\{ P = (p_1, p_2, ..., p_k), \; \sum_{i=1}^{k} p_i = 1, \; p_i \geq 0 \right\};$$

$$S_n = \left\{ Q = (q_1, q_2, ..., q_n), \ \sum_{j=1}^{n} q_j = 1, \ q_j \geq 0 \right\}.$$

Мішаною стратегією першого гравця є елемент $p \in S_k$, мішаною стратегією гравця 2 — елемент $q \in S_n$. Якщо гравець 1 застосовує свою мішану стратегію $P = (p_1, ..., p_k)$, а 2 — $Q = (q_1, ..., q_n)$, то платою гравця 1 гравцю 2 є величина $F(P,Q)$, яка є математичним сподіванням випадкової величини, що приймає значення a_{ij} $\forall i \in J_k$, $\forall j \in J_n$: $F\left(P,Q\right) = \sum_{j=1}^{n} \ \sum_{i=1}^{k} a_{ij} p_i q_j$, де добуток $p_i q_j$ — ймовірність випадкової події, що полягає в одночасному настанні випадкової події X_i для першого гравця та Y_j — другого. Гравець 1 може забезпечити собі програш не більше $\min\limits_{P \in S_k} \max\limits_{Q \in S_n} \sum_{j=1}^{n} \ \sum_{i=1}^{k} a_{ij} p_i q_j$, а гравець 2 — не менше $\max\limits_{P \in S_k} \min\limits_{Q \in S_n} \sum_{j=1}^{n} \sum_{i=1}^{k} a_{ij} p_i q_j$. Якщо (P^*, Q^*) — сідлова точка функції $F(P,Q)$, що визначається $F(P,Q)$, тобто виконуються нерівності $F(P^*, Q) \leq \ \leq F(P^*, Q^*) \leq F(P,Q^*)$, то P^*, Q^* називають оптимальними мішаними стратегіями гравців 1 та 2 відповідно. У цьому випадку, як відомо [14], значення платіжної функції в сідловій точці: $F(P^*, Q^*) = = \max\limits_{P \in S_n} \min\limits_{Q \in S_k} F(P,Q) = \min\limits_{Q \in S_k} \max\limits_{P \in S_n} F(P,Q)$. При цьому задача комбінаторної оптимізації ігрового типу з переставними обмеженнями на стратегії обох гравців має розв'язок у мішаних стратегіях, а $F(P^*, Q^*)$ — ціна гри. Таким чином, можна ставити задачу знаходження ймовірностей застосувань стратегій–переставлень першим та другим гравцями. Зауважимо, що ця постановка включає в себе і знаходження чистих стратегій (коли всі ймовірності крім однієї є нулями, а одна — одиницею).

ІТЕРАЦІЙНИЙ НАБЛИЖЕНИЙ МЕТОД ЗНАХОДЖЕННЯ МІШАНИХ СТРАТЕГІЙ

Далі пропонується ітераційний метод для розв'язування в мішаних стратегіях задач комбінаторної оптимізації ігрового типу, в яких на чисті стратегії накладаються обмеження, що визначені переставленнями для обох гравців.

Розглянемо матрицю $A = (a_{ij})$, елемент a_{ij} якої показує перевищення (різницю) прибутків другого гравця порівняно з першим. Вимірності цієї матриці — $m \times n$, де m — кількість рядків, n — кількість стовпців матриці.

На стратегії першого гравця накладаються обмеження, що визначені переставленнями. Ці переставлення є переставленнями елементів із мультимножини P^x; на стратегії другого гравця накладаються обмеження, які теж визначені переставленнями з елементів мультимножини P^y.

Ідея ітераційного методу, як і для класичної матричної гри, зводиться до наступного [14]. Розігрується гра, в якій супротивники застосовують один проти одного свої стратегії. Експеримент складається з послідовності ходів. Починається з того, що один із гравців обирає довільно одну зі своїх стратегій, інший на це відповідає своєю стратегією, котра найменш вигідна для першого гравця тощо. У кожній партії, коли настає черга гравця обирати стратегію, інший відповідає своєму опоненту тією ж чистою стратегією, яка є найгіршою для противника з урахуванням усіх його попередніх виборів. Ходи розглядаються, як своєрідна «мішана стратегія», де чисті стратегії змішані в пропорціях, які відповідні частоті їх застосування в минулому. Такий спосіб є моделлю реального практичного «взаємного навчання» гравців, коли кожен із них на досвіді досліджує спосіб поведінки супротивника і вчиться на його та своїх помилках.

АЛГОРИТМ МЕТОДУ

Всі наступні обчислення представлені табл. 1. Перший стовбець N таблиці — це номер одного етапу розіграшу гри, на якому гравці по черзі роблять свої кроки (обирають по черзі свої стратегії). Курсивом в описі алгоритму зображено табличну реалізацію методу.

Т а б л и ц я 1. Розв’язування комбінаторної задачі ігрового типу ітераційним методом

№	X	B_1	B_1X	B_2	B_1X	$N\overline{v}$	$\overline{v}$	Y	A_1	A_1Y	A_2	A_2Y	A_3	A_3Y	$N\underline{v}$	$\underline{v}$	υ^*
1	0,1	7	0,7	3	0,3			0,2	7	1,4	1	0,2	3	0,6			
	0,1	1	0,3	5	1,5			0,8	3	2,4	5	4	4	3,2			
	0,6	3	1,8	4	2,4			*sum_r*		3,8		4,2		3,8			
	sum_l		2,8		4,2			*SUM_R*		3,8		4,2		3,8			
	SUM_L		2,8		4,2			*NextX*		0,3		0,1		0,6	3,84	3,84	3,88
	NextY		0,2		0,8	3,92	3,92										
2	0,3	7	2,1	3	0,9			*0,2*	7	1,4	1	0,2	3	0,6			
	0,1	1	0,1	5	0,5			*0,8*	3	2,4	5	4	4	3,2			
	0,6	3	1,8	4	2,4			*sum_r*		3,8		4,2		3,8			
	Sum_l		4		3,8			*SUM_R*		7,6		8,4		7,6			
	SUM_L		6,8		8			*NextX*		0,3		0,1		0,6	7,68	3,84	3,86
	NextY		0,2		0,8	7,76	3,88										
… … …																	
12	0,1	7	0,7	3	0,3			0,2	7	1,4	1	0,2	3	0,6			
	0,1	1	0,3	5	1,5			0,8	3	2,4	5	4	4	3,2			
	0,6	3	1,8	4	2,4			*sum_r*		3,8		4,2		3,8			
	sum_l		2,8		4,2			*SUM_R*		48		48		45			
	SUM_L		43,2		47,2			*NextX*		0,3		0,1		0,6	46,2	3,85	3,85
	NextY		0,2		0,8	46,4	3,87										

Крок 1. Першу стратегію (переставлення X) 1-й гравець обирає випадково із множини переставлень $E_m(P^x)$. *Запишемо її покоординатно в стовпець X таблиці*. (Тут і далі курсивом викладено дії алгоритму в табличній формі).

Крок 2. У таблиці в стовпці B_j, де b_j ($\forall j \in J_n$) записуємо вектор-стовпець із номером j з матриці A. Обчислимо скалярні добутки векторів-стратегій B_j другого гравця та вектора обраної стратегії першого гравця. *У таблиці це зручно оформлювати, ввівши стопці: B_jX — вектор, що складається з поелементних добутків векторів X та B_j, $j \in J_n$. Ці вектори (X, B_j, B_jX) займають m рядків табл. 1. У наступному рядку sum_l у стовпцях B_jX записуємо скалярні добутки векторів B_j та X (як сума елементів стовпця B_jX табл.1).*

Крок 3. Знаходимо значення SUM_L — накопичені суми скалярних добутків (у лівій частині табл. 1). *У рядку SUM_L табл.1 записуємо суму значень елементів рядка sum_l та рядка SUM_L із попереднього ($(N-1)$-го) етапу. На першому кроці (коли $N=1$) рядок SUM збігається з рядком sum_l цього етапу.*

Крок 4. Стратегія $Next\,Y$ другого гравця обирається з умови отримання ним максимального сумарного за N етапів платежу, як розв'язок задачі максимізації цільової функції на множині переставлень [2]. Практично це означає впорядкування елементів вектора P^x так же як і елементи рядка SUM_L (найменшому значенню з рядка SUM_L — відповідне найменше значення з P^x, тощо).

Крок 5. Знаходимо значення $N\bar{v}$ — максимальний накопичений виграш другого гравця. *Максимальне значення $N\bar{v}$ обчислюється як скалярний добуток рядка SUM та стратегії $Next\,Y$. Записуємо його в цьому ж рядку в стовпці $N\bar{v}$.*

Крок 6. Знаходимо значення $\bar{v}$. Обчислюємо значення за формулою $\bar{v} = \frac{N\bar{v}}{N}$. *Результат записуємо в цьому ж рядку в стовпці $\bar{v}$.*

Крок 7. У стовпець Y записуємо значення стратегії другого гравця. У стовпець Y поокординатно заносимо значення рядка $Next\,Y$ лівої частини табл. 1.

Крок 8. У правій частині табл. 1 у стовпці A_i $\forall i \in J_m$, записуємо вектор стратегії першого гравця — рядок i з матриці A. Обчислюємо скалярні добутки вектора-стратегії другого гравця на вектори стратегій першого гравця — стовпці A_i $\forall i \in J_m$. *У табл. 1 це зручно оформлювати, ввівши стопці: A_iY — вектори, які складаються з поелементних добутків векторів Y та A_i, $i \in J_m$. Вектори Y, A_i, A_iY займають n рядків табл. 1.*

Крок 9. Обчислюємо значення sum_r та SUM_R. *У наступному рядку sum_r в стовпцях A_iY записуємо скалярні добутки векторів A_i та Y (як сума елементів стовпця A_iY табл.1). На першому етапі (коли $N=1$) рядок SUM_R правої частини табл. 1 збігається з попереднім рядком sum_r правої частини табл.1. У наступному рядку SUM_R табл.1 записуємо суму значень елементів рядка sum та рядка SUM_R із попереднього ($(N-1)$-го) етапу.*

Крок 10. Стратегія $NextX$ (права частина табл.1) першого гравця обирається з умови отримання ним мінімального сумарного за N етапів платежу (програшу), тобто розв'язок задачі мінімізації цільової функції на множині переставлень [2]. Практично це означає впорядкування елементів вектора P^y так само як і елементи рядка SUM_R (найбільшому значенню з рядка SUM_R — відповідає найменшому значенню з вектора P^y, тощо). Обрана стратегія записується в рядок $Next\,X$ правої частини табл.1 та стовпець X.

Крок 11. Знаходимо значення $N\underline{v}$ — мінімальний накоплений програш другого гравця. *Мінімальне значення* $N\underline{v}$ *правої частини табл.1 обчислюється як скалярний добуток рядка* SUM_R *та стратегії* $NextX$ *(правої частини табл. 1), його записуємо в цьому ж рядку в стовпці* $N\underline{v}$.

Крок 12. За формулою $\underline{v} = \dfrac{N\underline{v}}{N}$ обчислюємо $\underline{v}$. *Обчислене значення заносимо у стовпець* $\underline{v}$ *табл. 1.*

Крок 13. За формулою $\upsilon^* = \dfrac{\overline{v} + \underline{v}}{2}$ обчислюємо υ^*. *Отримане значення заносимо у стовпець* υ^* *табл. 1.*

Крок 14. Перевіряємо критерій завершення роботи алгоритму. Перевіряється умова, що мінімальне зі знайдених значень $\overline{v}$ дорівнює максимальному зі знайдених значень $\underline{v}$: $\min \overline{v} = \max \underline{v}$ (рівність максимального виграшу першого гравця мінімальному програшу другого гравця). Якщо цей критерій завершення виконується, то алгоритм зупиняється. Інакше переходимо на крок 10 алгоритму, обравши за стратегію 1-го гравця стратегію $Next\,X$. За ціну гри приймають $\upsilon^* = \overline{v} = \underline{v}$.

Роботу алгоритму також можна завершити, якщо проведено вказану кількість ітерацій, або коли досягнута задана точність $\Delta = \min \overline{v} - \max \underline{v} \le \varepsilon$, де $\varepsilon > 0$ — задана величина, а $\min \overline{v}$ — мінімум зі знайдених $\overline{v}$, $\max \underline{v}$ — максимум зі знайдених $\underline{v}$ або при $\delta = \dfrac{\Delta}{(1/2)(|\min \overline{v}| + |\max \underline{v}|)}$. При цьому ціна гри $V^* = (1/2)(\min \overline{v} + \max \underline{v})$.

Наближені значення P^*, Q^* (оптимальні мішані стратегії першого та другого гравців) — це вектори частот застосування своїх чистих стратегій–перставлень гравцями. Логічно вважати, що для забезпечення більшої точності знаходження чистих стратегій гравцями, проводиться якомога більша кількість ітерацій.

ІЛЮСТРАТИВНИЙ ПРИКЛАД РОБОТИ АЛГОРИТМУ

Розглянемо роботу ітераційного методу розв'язування комбінаторних оптимізаційних задач ігрового типу на переставленнях, коли на стратегії обох гравців накладаються комбінаторні обмеження.

Нехай $A = \begin{pmatrix} 7 & 3 \\ 1 & 5 \\ 3 & 4 \end{pmatrix}$, на стратегії першого гравця накладаються обмеження, що визначаються переставленнями з множини $P^x = \{0{,}1; 0{,}3; 0{,}6\}$, тобто $E_3(P^x) = \{(0{,}1; 0{,}3; 0{,}6), (0{,}3; 0{,}1; 0{,}6), (0{,}3; 0{,}6; 0{,}1), (0{,}6; 0{,}3; 0{,}1), (0{,}1; 0{,}6; 0{,}3)$ $(0{,}6; 0{,}1; 0{,}3)\}$. На стратегії другого гравця теж накладаються комбінаторні обмеження з множини $P^y = \{0{,}2; 0{,}8\}$, що визначені розміщеннями, тобто $E_2(P^y) = \{(0{,}2; 0{,}8), (0{,}8; 0{,}2)\}$.

У табл. 1 вноситимемо результати розрахунків на кожній ітерації методу.

Всі розрахунки (ітерації) будемо заносити в табл. 1, в якій N — номер кроку; X — вибрана стратегія першого гравця; $B_1, B_2, ..., B_n$ — стратегії другого гравця; $B_1X, B_2X, ..., B_nX$ — скалярний добуток векторів; $N\bar{v}$ — максимальний накопичений виграш (максимальний із накопичених скалярних добутків SUM_L та $Next\,X$); $\bar{v} = \dfrac{N\bar{v}}{N}$; Y — вибрана стратегія другого гравця; $A_1, A_2, ...A_m$ — стратегії другого гравця; числа в рядку з назвою SUM_L — сума значень елементів рядка sum_l та рядка SUM_L із попереднього ($(N-1)$-го) кроку; SUM_R — сума значень елементів рядка sum_r та рядка SUM_R із попереднього ($(N-1)$-го) кроку. Коли $N = 1$ рядок SUM_R збігається з попереднім рядком цієї правої (лівої) частини табл.1 $Next\,X$ — обрана стратегія першого гравця; $Next\,Y$ — обрана стратегія другого гравця; $N\underline{v}$ — мінімальний накопичений платіж (програш) (скалярний добуток елементів векторів SUM_R та $Next\,Y$); $\bar{v} = \dfrac{N\bar{v}}{N}$, $\underline{v} = \dfrac{N\underline{v}}{N}$, $\upsilon^* = \dfrac{\bar{v} + \underline{v}}{2}$.

Першу чисту стратегію першого гравця виберемо довільно — нехай це, $(0{,}1; 0{,}3; 0{,}6)$. Перші дві та останній 120-й етап показано в табл. 1.

Після 12 ітерації виконався критерій зупинки алгоритму рівності максимального виграшу першого гравця мінімальному програшу другого гравця. Отримаємо $F_x(X^*) = F_y(Y^*) = 3{,}85$. Стратегія $(0{,}1; 0{,}3; 0{,}6)$ першого гравця використовувалась 3 рази, $(0{,}3; 0{,}1; 0{,}6)$ — 9 разів. Для другого гравця: перша стратегія $(0{,}2; 0{,}8)$ використовувалась 11 разів, а $(0{,}8; 0{,}2)$ — 1 раз.

Наближені значення мішаних стратегій P^*, Q^* такі: для першого гравця частота використання стратегії $(0{,}1; 0{,}3; 0{,}6)$ становить $\dfrac{3}{12} = 0{,}25$, отже, $p_1 \approx 0{,}25$, $(0{,}1; 0{,}3; 0{,}6)$ дорівнює $\dfrac{9}{12} = 0{,}75$, $p_2 \approx 0{,}75$ частоти застосувань усіх інших можливих стратегій — переставлень рівні 0 ($p_3 = ... = p_6 = 0$). Для другого гравця частоти застосування стратегії (0,2;0,8) — становить

$q_1 = \dfrac{11}{12}$, а стратегії (0,8;0,2) — $q_2 = \dfrac{1}{12}$.Частоти приймаються за наближене значення ймовірностей.

РЕЗУЛЬТАТИ ЧИСЛОВИХ ЕКСПЕРИМЕНТІВ

За допомогою розробленої програмної реалізації цього методу, проведено числовий експеримент з метою виявлення залежності часу обчислень від вимірності задачі. Результати експерименту внесено в табл. 2, де: t — загальний час серії в секундах (с); $t_{\text{сер}}$ — середній час задачі в секундах (с) або мілісекундах (мс); t_{im} — середній час ітерації в мілісекундах (мс); $\Delta = \min \overline{v} - \max \underline{v}$; $\delta = \dfrac{\Delta}{(1/2)(|\min \overline{v}| + |\max \underline{v}|)}$; $\delta_{\min}$ — мінімальне δ в серії обчислень; $\delta_{\max}$ — максимальне δ в серії обчислень.

Т а б л и ц я 2. Результати першої серії експериментів

№	$m = n$	$[a,b]$	t	$t_{\text{сер}}$	t_{im} (мс)	Δ сере	$\delta_{\min}$	$\delta_{\max}$	$\delta_{\text{сер}}$	10^{-4}	10^{-5}
1	10	[1,200]	< 1 с	470мс	<1	0,0119	1,06E–5	0,000354	0,000129	100	39
2	20	[1,400]	1 с	14 мс	<1	0,0177	2,52E–5	0,000164	8,99E–5	100	66
3	30	[1,600]	3 с	30 мс	<1	0,0246	3,27E–5	0,000164	8,26E–5	100	82
4	40	[1,800]	5 с	50 мс	<1	0,0279	3,8E–5	0,000114	7,02E–5	100	94
5	50	[1,1000]	7 с	74 мс	<1	0,0306	4,22E–5	9,85E–5	6,12E–5	100	100
6	60	[1,1200]	10 с	101 мс	<1	0,0334	3,37E–5	8,33E–5	5,58E–5	100	100
7	70	[1,1400]	13 с	136 мс	<1	0,0366	3,19E–5	7,12E–5	5,24E–5	100	100
8	80	[1,1600]	17 с	167мс	<1	0,0394	3,44E–5	6,76E–5	4,93E–5	100	100
9	90	[1,1800]	21 с	214 мс	<1	0,0419	3,27E–5	6,66E–5	4,65E–5	100	100
10	100	[1,2000]	25 с	255 мс	<1	0,0451	3,12E–5	6E–5	4,51E–5	100	100
11	200	[1,4000]	3 хв,21с	1 с	1	0,0651	2,38E–5	4,13E–5	3,26E–5	100	100
12	300	[1,6000]	5 хв,4 2с	4 с	4	0,0831	2,21E–5	3,49E–5	2,77E–5	100	100
13	400	[1,8000]	10 хв,51с	6 с	6	0,106	2,18E–5	3,83E–5	2,65E–5	100	100
14	500	[1,10000]	17 хв,32с	10 с	10	0,129	2,06E–5	4,24E–5	2,6E–5	100	100
15	600	[1,12000]	24 хв,44с	14 с	15	0,206	2,07E–5	7,1E–5	3,48E–5	100	100
16	700	[1,14000]	35 хв,24с	21 с	21	0,201	2,05E–5	5,73E–5	2,88E–5	100	100

Для цього класу задач розв’язано по 100 задач вимірності $m \times n$, причому $m = n$. Елементи a_{ij} матриці генерувалися як рівномірно розподілені цілі числа в інтервалі $[a,b]$ вказаному в стовпці інтервалів табл. 2. Елементи мультимножин P^x, P^y генерувалися за рівномірним розподілом за умов $\sum_{i=1}^{m} x_i = 1$, $\sum_{j=1}^{n} y_j = 1$. Максимальна кількість ітерації 1000. Вимірність задач задавалась в інтервалі від 10 до 700 (від 10 до 100 через 10; від 100 до 700 через 100). У кожній серії проведено 100000 ітерацій. У стовпцях 10^{-3} (10^{-4}, 10^{-5}) вказано відсоток розв'язаних задач із зазначеною точністю по δ 10^{-3}, 10^{-4}, 10^{-5} відповідно.

Із використанням пакета CurveExpert Professional 1.2.1 [15] побудовано аналітичний вигляд та графік регресійної залежності часу обчислень від вимірності задачі та величини δ від вимірності задачі для кожного з експериментів. Для першої серії експериментів залежності часу обчислень від вимірності задачі аналітичний вигляд функції такий: $T(m) = 0{,}00007m^2 - 0{,}0024m$, при цьому коефіцієнт кореляції $r = 0{,}99$. Аналітичний вигляд залежності величини δ від вимірності задачі такий: $\delta(m) = 0{,}00004 \cdot 2929^{\left(\frac{1}{m}\right)}$; коефіцієнт кореляції $r = 0{,}91$.

У другій серії експериментів елементи a_{ij} матриці генерувалися як рівномірно розподілені цілі числа на проміжках від $[-200, 200]$ до $[-1400, 1400]$, параметри $m = n$ (вимірність) приймали значення від 10 до 700 відповідно. Час обчислень задач збільшувався зі збільшенням вимірності задачі від 1 сек до 35 хв, значення Δ (середнє по 100 задачам вимірності) зростало від 0,026 до 0,418. Аналітичний вигляд залежності часу обчислень від вимірності задачі такий: $T(m) = 0{,}00007m^2 - 0{,}006m + 0{,}179$, коефіцієнт кореляції $r = 0{,}98$.

У третій серії експериментів вимірність задач $m \times n$ $(m = n)$ задавалась в інтервалі від 10 до 700 (від 10 до 100 через 10; від 100 до 700 через 100). Для кожної серії генерувалося по 100 задач. Елементи a_{ij} генерувалися, як рівномірно розподілені цілі числа в інтервалі $[-100, 100]$. Для кожної вимірності розв'язано по 100 задач. Час обчислень задач збільшувався зі збільшенням вимірності задачі від 1 сек до 5 хв, значення Δ (середнє по 100 задачах вимірності) зростало від 0,013 до 0,00287. Аналітичний вигляд залежності часу обчислень від вимірності задачі одержано такий: $T(m) = 0{,}00007m^2 - 0{,}004m + 0{,}094$, коефіцієнт кореляції $r = 0{,}99$.

У четвертій серії експериментів елементи a_{ij} матриці генерувалися, як рівномірно розподілені цілі числа в інтервалі $[-10000, 10000]$, при чому параметри $m = n$ приймали значення від 10 до 700 відповідно (від 10 до 100 через 10; від 100 до 700 через 100). Для кожної серії генерувалося по 100 задач. Час обчислень задач збільшувався зі збільшенням параметрів задачі від 1 сек до 35 хв, значення Δ (середнє по 100 задачах вимірності) змінювалося від 1,21 до 0,289. Аналітична функція залежності часу обчислень від вимірності задачі одержана така: $T(m) = 0{,}00007m^2 - 0{,}006m + 0{,}195$, коефіцієнт кореляції $r = 0{,}99$.

У п'ятій серії експериментів розв'язано по 100 задач вимірності $m \times n$, причому $m = n$, вимірність задач задавалась в інтервалі від 10 до 700 відповідно (від 10 до 100 через 10; від 100 до 700 через 100). Елементи a_{ij} матриці генерувалися нормально розподілені цілі числа з параметрами $[M_x, \sigma_x]$, де M_x — математичне сподівання, σ_x — середнє квадратичне відхилення. Параметри генерації $[M_x, \sigma_x]$ становить [0,3]. Час обчислень задач збільшувався зі збільшенням параметрів задачі від 1 сек до 35 хв, значення Δ (середнє по 100 задачах вимірності) змінювалося від 0,000485 до 0,000128. Значення величини δ змінювалося від 0,0083 до 0,1430. Кількість розв'язаних задач із точністю 10^{-3} по δ у відсотковому співвідношенні для першої серії експериментів становила 86 % та зменшувалася до 5% розв'язаних задач при $m = n = 700$. Кількість розв'язаних задач із точністю 10^{-4} по δ у відсотковому співвідношенні було зафіксовано лише для $m = n = 10$ та становила 21 %, $m = n = 20$ — 5 %, $m = n = 30$ — 2 %. Для всіх інших серій цього експерименту не було розв'язаних задач. Аналітичний вигляд залежності часу обчислень від вимірності задачі для п'ятої серії експериментів одержано такий: $T(m) = 0{,}00007m^2 + 0{,}0005m - 0{,}192$, коефіцієнт кореляції $r = 0{,}98$.

Проведено також експерименти з параметрами генерації $[M_x, \sigma_x]$, що становлять [0,30]. Розв'язано по 100 задач вимірності $m \times n$, причому $m = n$. Вимірність задач задавалась в інтервалі від 10 до 700 відповідно (від 10 до 100 через 10; від 100 до 700 через 100). Час обчислень задач збільшувався зі збільшенням параметрів задачі від 1 сек до 35 хв, значення Δ (середнє по 100 задачах вимірності) змінювалося від 0,00686 до 0,0015. Значення величини δ змінювалося від 0,00895 до 0,1430. Кількість розв'язаних задач із точністю 10^{-3} по δ у відсотковому співвідношенні для першої серії експериментів становила 83 % та зменшувалася до 1 % розв'язаних задач при $m = n = 700$. Кількість розв'язаних задач із точністю 10^{-4} по δ у відсотковому співвідношенні була зафіксована лише для $m = n = 10$ та становила 14 %, $m = n = 20$ — 6%, $m = n = 30$ — 1 %, для всіх інших серій цього експерименту не було розв'язано задач із точністю 10^{-4} по δ. Аналітичний вигляд залежності часу обчислень від вимірності задачі для п'ятої серії експериментів отримано такий: $T(m) = 0{,}00008m^2 - 0{,}006m + 0{,}188$, коефіцієнт кореляції $r = 0{,}99$.

Під час дослідження проведених експериментів із параметрами генерації $[M_x, \sigma_x]$, що становлять $[0{,}300]$, розв'язано по 100 задач вимірності $m \times n$, причому $m = n$. Вимірність задач задавалась в інтервалі від 10 до 700 відповідно (від 10 до 100 через 10; від 100 до 700 через 100). Час обчислень задач збільшувався зі збільшенням параметрів задачі від 1 сек до 36 хв, значення Δ (середнє по 100 задачах вимірності) змінювалося від 0,0642 до 0,0147. Значення величини δ змінювалося від 0,0108 до 0,0966. Кількість розв'язаних задач із точністю 10^{-3} по δ у відсотковому співвідношенні для першої серії експериментів становила 92 % та зменшувалася до 1 % розв'язаних задач при $m = n = 700$. Кількість розв'язаних задач із точністю 10^{-4} по δ у відсотковому співвідношенні була зафіксована лише для $m = n = 10$ та становила 15 %, $m = n = 20$ — 5%, для всіх інших серій цього експерименту задачі не розв'язувалися. Аналітичний вигляд залежності часу обчислень від вимірності задачі для п'ятої серії експериментів отримано такий: $T(m) = 0{,}00008m^2 - 0{,}007m + 0{,}187$, коефіцієнт кореляції $r = 0{,}99$.

Елементи мультимножини P^x, P^y генерувалися за рівномірним розподілом за умов $\sum_{i=1}^{m} x_i = 1$, $\sum_{j=1}^{n} y_j = 1$. Максимальна кількість ітерації 1000. У серії проведено 100000 ітерацій.

Отже, усі проведені експерименти показують наближення платежів до ціни гри, що дає можливість наближеного знаходження P^*, Q^*.

ОЦІНКА СКЛАДНОСТІ АЛГОРИТМУ

Для розрахунку складності алгоритму [16] складемо табл. 3, де у стовпці «Алгоритм» записано програмну реалізацію ітераційного методу (одна ітерація) для розв'язування комбінаторних оптимізаційних задач ігрового типу на перестановках, в яких накладаються обмеження, що визначаються переставленнями, на стратегії обох гравців. Час виконання різних рядків алгоритму різний, але один і той рядок i виконується за час c_i, де c_i — константа. Позначимо через τ_j — кількість раз виконання умови.

Під час розрахунку складності алгоритму потрібно визначити асимптотичну верхню границю з точністю до постійного множника [16]. Для функції $g(n)$ позначка $O(g(n)) = f(n)$ з [16] означає множину функцій таких, що існує додатня константа c і n_0 така, що $0 \le f(n) \le cg(n)$ для всіх $n \ge n_0$. Підрахуємо $T = \sum_{i=1}^{39} c_j \tau_j$:

$$T = 1(c_1 + c_8 + c_{11} + c_{15} + c_{16} + c_{25} + c_{30} + c_{31} + c_{32} + c_{33}) +$$
$$+ n(c_2 + c_3 + c_6 + c_7 + c_9 + c_{10} + c_{12} + c_{13} + c_{14}) +$$
$$+ m(c_{17} + c_{18} + c_{21} + c_{22} + c_{23} + c_{24} + c_{26} + c_{27} + c_{28} + c_{29}) +$$
$$+ nm(c_4 + c_5 + c_{19} + c_{20}) + O(n\log n) + O(m\log m) + T_0(m) + T_1(n)$$

або

$$T = O(1) + O(n) + O(m) + O(nm) + O(n\log n) + O(m\log m) + T_0(m) + T_0(m),$$

якщо $T_0(m) = O(m)$, $T_1(n) = O(n)$, то

$$T = O(1) + O(n) + O(m) + O(nm) + O(n\log n) + O(m\log m) + O(m) + O(n).$$

Враховуючи, що $\forall c > 0 : cO(f(n)) = O(f(n))$, то

$$T = O(n) + O(m) + O(nm) + O(n\log n) + O(m\log m),$$

якщо $\lim_{n\to\infty} \frac{f(n)}{g(n)} = c < \infty$, то $O(f(n)) + O(g(n)) = O(f(n) + g(n))$, отже

$$T = O(n+m) + O(nm) + O(n\log n) + O(m\log m),$$

якщо $\lim\limits_{n\to\infty} \dfrac{f(n)}{g(n)} = \infty$, то $O(f(n)) + O(g(n)) = O(f(n))$, отже

$$T = O(nm + n\log n + m\log m).$$

Таблиця 3. Оцінка складності алгоритму

№	Алгоритм	Час c_j	Кількість раз τ_i
1	inc(cIterNum);	c_1	1
2	for *i*:=1 to *cN* do	c_2	n
	begin		–
3	*s* := 0;	c_3	n
4	for *j*:=1 to *cM* do	c_4	nm
	begin		–
5	*s* := *s* + cBx[*j*, *i*];	c_5	nm
	end;		–
6	*cSumX*[*i*] := *s*;	c_6	n
7	*cSumXn*[*i*] := *cSumXn*[*i*] + *s*;	c_7	n
	end;		–
8	*SetLength*(*IndArr*, max (*cN*, *cM*) + 1);	c_8	1
9	for *i*:=1 to *cN* do	c_9	n
10	*IndArr*[*i*] := *i*;	c_{10}	n
11	*MakeInd*(*IndArr*, *cSumXn*, *cN*, 1);	1	$O(n\log n)$ [16]
12	*cNv_* := 0;	c_{11}	1
13	for *i*:=1 to *cN* do	c_{12}	n
	begin		
14	*cNextY*[*IndArr*[*i*]] := cPy[i];	c_{13}	n
15	*cNv_* := *cNv_* + *cNextY*[*IndArr*[*i*]]* *SumXn*[*IndArr*[*i*]];	c_{14}	n
	end;		–
16	*cv_* := *cNv_*/*cIterNum*;	c_{15}	1
17	*cminv_* := min (*cv_*, *cminv_*);	c_{16}	1
18	for *i*:=1 to *cM* do	c_{17}	m
	begin		–

Продовження табл. 3

№	Алгоритм	Час c_j	Кількість раз τ_i
19	*s* := 0;	c_{18}	m
20	for *j* :=1 to *cN* do	c_{19}	nm
	begin		–
21	*s* := *s* + *cAy*[*j*, *i*];	c_{20}	nm
	end;		–
22	*cSumY*[*i*] := *s*;	c_{21}	m
23	*cSumYn*[*i*] := *cSumYn*[*i*] + *s*;	c_{22}	m
	end;		–
24	for *i*:=1 to *cM* do	c_{23}	m
25	*IndArr*[*i*] := *i*;	c_{24}	m
26	*MakeInd(IndArr*, *cSumYn*, *cM*, 0);	1	$O(m\log m)$ [16]
27	*cN_v* := 0;	c_{25}	1
28	for *i*:=1 to *cM* do	c_{26}	m
	begin		–
29	*cNextXp*[*i*] := *cNextX*[*i*];	c_{27}	m
30	*cNextX*[*IndArr*[*i*]] := *cPx*[*i*];	c_{28}	m
31	*cN_v* := *cN_v* + *cNextX*[*IndArr*[*i*]] **cSumYn*[*IndArr*[*i*]];	c_{29}	m
	end;		–
32	*c_v* := *cN_v* / *cIter-Num*;	c_{30}	1
33	*cmax_v* := max (*c_v*, *cmax_v*);	c_{31}	1
34	*cv_s* := (*cv_* + *c_v*) / 2;	c_{32}	1
35	*CheckStrat*(0);	1	$T_0(m)$ [16]
36	*CheckStrat*(1);	1	$T_1(m)$ [16]
37	*Result* := *CheckEval*(*aBreak Type*, *aMaxIter*);	c_{33}	1

ВИСНОВКИ

Для розв'язання комбінаторних оптимізаційних задач ігрового типу з обмеженнями-переставленнями, що накладаються на стратегії обох гравців, поширено ітераційний метод [8, 11]. Подано експериментальну та теоретичну оцінки кількості операцій та експериментально оцінено точність методу. Доцільним напрямом подальших досліджень є доведення збіжності запропонованого ітераційного методу знаходження ймовірностей для стратегій-переставлень у задачі, що розглядається.

ЛІТЕРАТУРА

1. *Сергиенко И.В.*, *Каспшицкая М.Ф.* Модели и методы решения на ЭВМ комбинаторных задач оптимизации. — К.: Наук. думка, 1981. — 288 с.
2. *Стоян Ю.Г.*, *Ємець О.О.* Теорія і методи евклідової комбінаторної оптимізації. — К.: Інститут системних досліджень освіти, 1993. — 188 с.
3. *Емец О.А.*, *Колечкина Л.Н.* Задачи комбинаторной оптимизации с дробнолинейными целевыми функциями. — К.: Наук. думка, 2005. — 117 с.
4. *Емец О.А.*, *Романова Н.Г.* Оптимизация на полиперестановках. — К.: Наук. думка, 2010. — 105 с.
5. *Емец О.А.*, *Барболина Т.Н.* Комбинаторная оптимизация на размещениях. — К.: Наук. думка, 2008. — 159 с.
6. *Павлов А.А., Павлова Л.А.* Основы методологии проектирования ПДС-алгоритмов для трудноразрешаемых комбинаторных задач // Проблема информатики и управления. — 1995. — № 4. — С. 135–141.
7. *Емец О.А.*, *Устьян Н.Ю.* Исследование математических моделей и методов решения задач на перестановках игрового типа // Кибернетика и сист. анализ. — 2007. — № 6. — С. 103–114.
8. *Ємець О.О., Устьян Н.Ю.* Розв'язування ігрових задач на переставленнях // Наукові вісті НТУУ «КПІ». — 2007. — № 3. — С. 47–52.
9. *Емец О.А.*, *Устьян Н.Ю.* Решение некоторых задач комбинаторной оптимизации на размещениях и перестановках игрового типа // Проблемы управления и информатики. — 2006. — № 3. — С. 37–47.
10. *Емец О.А.*, *Устьян Н.Ю.* Исследование задач комбинаторной оптимизации игрового типа на размещениях // Проблемы управления и информатики. — 2007. — № 1. — С. 26–36.
11. *Ємець О.О., Устьян Н.Ю.* Один ітераційний метод розв'язування ігрових задач на переставленнях // Наук. вісті НТУУ «КПІ». — 2008. — № 3. — С. 5–10.
12. *Емец О.А., Устьян Н.Ю.* Игры с комбинаторными ограничениями // Кибернетика и сист. анализ. — 2008. — № 4 . — С. 134–141.
13. *Емец О.А.*, *Ольховская Е.В.* Итерационный метод решения комбинаторных оптимизационных задач игрового типа на размещениях // Проблемы управления и информатики. — 2011. — № 3. — С. 69–78.
14. *Вентцель Е.С.* Элементы теории игр. — 2-е изд. стереотип. — М.: Физматгиз, 1961. — 68 с.
15. *Hyams* Daniel G. Curve Expert Software. — 2011. — http://www.curveexpert.net.
16. *Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К.* Алгоритмы: построение и анализ: пер. с англ. — 2-е изд. — М.: Вильямс, 2005. — 1296 с.

Надійшла 11.05.2011