УДК 004.75
DOI https://doi.org/10.32782/tnv-tech.2024.3.6

ТЕОРЕТИЧНІ ОЦІНКИ СКЛАДНОСТІ АЛГОРИТМІВ ІТЕРАЦІЙНИХ МЕТОДІВ ТИПУ БРАУНА-РОБІНСОН ДЛЯ РОЗВ’ЯЗУВАННЯ КОМБІНАТОРНИХ ОПТИМІЗАЦІЙНИХ ЗАДАЧ ІГРОВОГО ТИПУ

Ольховська О. В. – *кандидат фізико-математичних наук,*
завідувач кафедри комп’ютерних наук та інформаційних технологій
Полтавського університету економіки і торгівлі
ORCID ID: 0000-0001-5366-5995

Кошова О. П. *– кандидат педагогічних наук,*
доцент кафедри комп’ютерних наук та інформаційних технологій
Полтавського університету економіки і торгівлі
ORCID ID: 0000-0003-0794-6774

Гаркуша С. В. – *доктор технічних наук,*
професор кафедри комп’ютерних наук та інформаційних технологій
Полтавського університету економіки і торгівлі
ORCID ID: 0000-0002-6473-4324

Тур В. М. – *аспірант*
Полтавського університету економіки і торгівлі
ORCID ID: 0009-0003-2825-1434

Комбінаторні оптимізаційні задачі ігрового типу є однією з найважливіших категорій задач в області теорії ігор і оптимізації. Вони охоплюють широкий спектр проблем, від розподілу ресурсів до стратегічного планування в умовах конфлікту. Ітераційні методи, зокрема метод Брауна-Робінсон, є класичними підходами до розв’язання таких задач.

Потреба в розв'язуванні комбінаторних оптимізаційних задач ігрового типу на множинах перестановок та розміщень підкреслює важливість дослідження можливості модифікації існуючих методів та оцінки їхньої складності. Виконано розширення ітераційного методу для задач комбінаторної оптимізації ігрового типу з різними комбінаторними конфігураціями.

Стаття починається з опису алгоритмів методів типу Брауна-Робінсон, що використовуються для розв’язування комбінаторних оптимізаційних задач ігрового типу на множинах перестановок та розміщень. Далі основний акцент робиться на теоретичних оцінках складності цих алгоритмів. Розглядається кількість ітерацій, необхідних для досягнення оптимального розв’язання, а також асимптотична складність алгоритмів у залежності від розмірності задачі та параметрів. Досліджуються умови збіжності алгоритмів. Визначаються межі складності для найгірших сценаріїв, де алгоритми можуть демонструвати максимальну складність. Для ітераційних методів типу Брауна-Робінсон, що розв'язує комбінаторні оптимізаційні задачі ігрового типу з обмеженнями у вигляді перестановок, накладеними на стратегії обох гравців, та для розв’язування ігрових задач з обмеженнями-розміщеннями на стратегії одного гравця. отримано теоретичні оцінки складності, сформульовані у вигляді теорем.

Результати дослідження підтверджують, що ітераційні методи типу Брауна-Робінсон є ефективними інструментами для розв’язання комбінаторних оптимізаційних задач ігрового типу. Проте їх обчислювальна складність значною мірою залежить від специфіки задачі. Подальші дослідження у цьому напрямку можуть значно покращити розуміння і застосування цих методів у практиці, сприяючи розвитку теорії ігор і комбінаторної оптимізації.

Ключові слова: *мінімальний програш, стратегії гравців, обмеження-перестановки, максимальний виграш.*

Olkhovska O. V., Koshova O. P., Harkusha S. V., Tour V. M. Theoretical assessments of the complexity of algorithms of iterative methods of the Brown-Robinson type for solving combinatory optimization problems of the game type

Game-type combinatorial optimization problems are one of the most important categories of problems in the field of game theory and optimization. They cover a wide range of issues, from resource allocation to strategic planning in conflict. Iterative methods, in particular the Brown-Robinson method, are classic approaches to solving such problems.

The need to solve game-type combinatorial optimization problems with multiple permutations and placements emphasizes the importance of studying the possibility of modifying existing methods and assessing their complexity. An extension of the iterative method for game-type combinatorial optimization problems with various combinatorial configurations is performed.

The article begins with a description of algorithms of Brown-Robinson type methods used to solve combinatorial optimization problems of game type on sets of permutations and placements. Further, the main emphasis is placed on theoretical estimates of the complexity of these algorithms. The number of iterations required to achieve the optimal solution is considered, as well as the asymptotic complexity of algorithms depending on the dimension of the problem and parameters. The conditions of convergence of algorithms are studied. Complexity bounds are defined for worst-case scenarios, where algorithms can exhibit maximum complexity. For iterative methods of the Brown-Robinson type, which solve combinatorial optimization problems of the game type with permutation constraints imposed on the strategies of both players, and for solving game problems with placement constraints on the strategies of one player. obtained theoretical estimates of complexity, formulated in the form of theorems.

The research results confirm that iterative methods of the Brown-Robinson type are effective tools for solving combinatorial game-type optimization problems. However, their computational complexity largely depends on the specifics of the problem. Further research in this direction can significantly improve the understanding and application of these methods in practice, contributing to the development of game theory and combinatorial optimization.

Key words: *minimum loss, player strategies, permutation restrictions, maximum winning.*

Постановка проблеми. Теорія ігор є важливою галуззю математики, що знаходить широке застосування в економіці, військових стратегіях, біології, соціальних науках та інших сферах [1-11]. Ітераційний метод типу Брауна-Робінсон є одним з класичних підходів до розв'язання ігрових задач, що дозволяє знаходити рівноважні стратегії у двохособових іграх з нульовою сумою.

Багато реальних задач можна моделювати як ігрові задачі з комбінаторними обмеженнями. Це стосується таких сфер, як оптимізація виробничих процесів, розподіл ресурсів, управління проектами, де один із гравців може мати обмеження на стратегії через фізичні або інші фактори.

Ітераційні методи, такі як метод Брауна-Робінсон, мають переваги у вигляді відносної простоти реалізації та здатності до обробки великих обсягів даних. Вивчення ефективності цих методів у контексті задач з комбінаторними обмеженнями є важливим для покращення алгоритмів та розробки нових, більш ефективних підходів.

Аналіз ітераційних методів дозволяє поглибити розуміння теоретичних аспектів збіжності алгоритмів, їхньої стійкості та точності. Це, в свою чергу, сприяє розвитку математичного апарату та методів розв'язання складних ігрових задач.

В сучасних умовах розвитку технологій, алгоритми для розв’язування ігрових задач знаходять нові області застосування, зокрема в сфері штучного інтелекту, машинного навчання та аналізу великих даних. Вивчення алгоритмів, що враховують специфічні обмеження, дозволяє створювати більш адаптивні та потужні системи.

Виклад основного матеріалу. Розглянемо ітераційний метод типу Брауна-Робінсон (ІМТБР) для розв’язування ігрових задач з обмеженнями-розміщеннями на стратегії одного гравця. В [1] розроблено наближений ІМТБР для знаходження оптимальної стратегії гравців. В основі ідеї ІМТБР лежить принцип

поетапного розігрування гри, в якій гравці вважаються розумними сторонами, і, при виборі власної стратегії, роблять ходи за принципом «майбутнє схоже на минуле», враховуючи всі проведені раніше ходи. Такий спосіб є моделлю реального практичного «взаємного навчання» гравців, коли кожен з них на досвіді досліджує спосіб поведінки супротивника і вчиться на його і своїх помилках.

Розглянемо алгоритм запропонованого методу.

Зауважимо, що процес обчислень зручно оформлювати у вигляді таблиці, куди заносяться всі результати проведених обчислень.

У перший стовбець N таблиці заноситься номер поточного етапу розіграшу гри, на якому кожен з гравців по черзі робить свій крок (обирають по одному разу свої стратегії).

Крок 0. Встановлюється номер N ітерації рівний 1: $N = 1$.

Крок 1. Першу стратегію i перший гравець обирає випадковим чином з множини розміщень $E_M^{m-1}\left(P^x\right)$.

Записуємо її покоординатно в стовпець X таблиці.

Крок 2. Обчислюються скалярні добутки векторів стратегій другого гравця на вектор обраної стратегії першого гравця.

У таблицю це зручно вносити, ввівши стопці: B_j ($b_j \ \ \forall j \in J_n$) – вектор стратегії j другого гравця з матриці A_{ij} ; $B_j X$ – вектор, що складається з поелементних добутків векторів X та B_j , $j \in J_n$. Ці вектори (X , B_j , $B_j X$) займають n рядків таблиці. В наступному рядку sum в стовпцях $B_j X$ записуємо скалярні добутки векторів B_j та X (як сума елементів стовпця $B_j X$ таблиці).

Крок 3. Знаходяться SUM_L – накопичені суми скалярних добутків (в лівій частині таблиці).

У наступному рядку SUM_L таблиці записується сума значень елементів рядка sum_l та рядка SUM_L з попереднього ($\left(N-1\right)$-го) етапу. На першій ітерації алгоритму ($N = 1$) рядок SUM_L збігається з рядком sum_l цього етапу.

Крок 4. Обирається стратегія другого гравця за критерієм отримання максимального виграшу (стратегію з максимальною накопиченою сумою), знаходячи $\overline{Nv}$.

Обирається максимальне значення з рядка SUM_L, яке записується в цьому ж рядку в стовпці $\overline{Nv}$. Стратегія B_j, якій відповідає знайдене максимальне значення з рядку SUM є стратегією другого гравця при наступному виборі. У стовпець j таблиці записується номер стратегії B_j, якій відповідає максимальне значення з рядка SUM_L.

Вибраний стовпець B_j покоодинатно заноситься в рядок у відповідні стовпці A_i.

На першій ітерації ($N = 1$) рядок SUM_R правої частини таблиці збігається з попереднім рядком цієї (правої) частини таблиці, де записана стратегія другого гравця, тобто значеннями елементів стовпців A_i. На наступних етапах в рядок SUM_R правої частини таблиці записується сума значень елементів рядка SUM_R з попереднього ($\left(N-1\right)$-го) етапу та рядка стратегії другого гравця (попередній перед SUM_R рядок правої частини таблиці).

Крок 5. Стратегія $NextX$ першого гравця обирається з умови отримання ним мінімального сумарного за N етапів платежу (програшу). При цьому на кожному етапі розв'язується така лінійна умовна задача на розміщеннях:

$$c^* = \min_{x \in R^m} \sum_{j=1}^{m} c_j x_j \ ; \ x^* = \arg\min_{x \in R^m} \sum_{j=1}^{m} c_j x_j \qquad (1)$$

за обмежень:

$$x \in E_M^m\left(P^x\right), \tag{2}$$

$$\sum_{j=1}^{m} x_j = 1, \tag{3}$$

де $c = (c_1, ..., c_m)$ – вектор SUM_L; $E_M^m\left(P^x\right)$ – множина m-розміщень з M елементів вектора P^x.

Зауважимо, що $NextX$ – це і є x^* з (1).

Значення $N\underline{v}$ – мінімальний накопичений програш – це скалярний добуток елементів векторів накопиченої суми SUM_R платежів (програшів) і обраної стратегії $NextX$ першого гравця. Зауважимо, що $N\underline{v}$ знаходиться при розв'язанні задачі типу (1)-(3).

Отримане значення c^* заноситься у стовпець таблиці $N\underline{v}$.

Крок 6. За формулами $\overline{v} = \frac{N\overline{v}}{N}$, $\underline{v} = \frac{N\underline{v}}{N}$, $\upsilon^* = \frac{\overline{v} + \underline{v}}{2}$ обчислюються $\overline{v}$, $\underline{v}$ та υ^*.

Обчислені значення заносяться у стовпці $\overline{v}$, $\underline{v}$ та v^* таблиці відповідно.

Крок 7. Перевіряється критерій завершення роботи алгоритму: рівність мінімального зі знайдених значень $\overline{v}$ (позначка $\min\overline{v}$) максимальному зі знайдених значень $\underline{v}$ (позначка $\max\underline{v}$):

$$\min\overline{v} = \max\underline{v}.$$

Якщо цей критерій виконується, то роботу алгоритму завершено. За ціну гри приймають $V^* = \underline{v} = v$. Інакше виконується перехід на пункт 2 алгоритму, обравши за стратегію першого гравця стратегію $NextX$, та збільшивши номер N ітерації на 1: $N = N + 1$.

Робота алгоритму також може бути завершена, якщо проведена певна кількість ітерацій, або у разі, коли досягнена задана точність:

$$\Delta = \left|\min\overline{v} - \max\underline{v}\right| \le \varepsilon,$$

де $\varepsilon > 0$ – задана величина або при

$$\delta = \frac{\Delta}{\frac{1}{2}\left(\min\overline{v} + \max\underline{v}\right)} < \varepsilon.$$

При цьому за розв'язок задачі приймається: ціна гри – значення $V^* = \frac{1}{2}\left(\min\overline{v} + \max\underline{v}\right)$, оптимальна стратегією першого гравця X^* – його стратегія з максимальною частотою, оптимальна стратегією другого гравця Y^* – мішана стратегія-вектор, що складається з частот застосування чистих стратегій другого гравця.

Для виконання оцінки складності алгоритму ітераційного методу типу Брауна-Робінсон для розв'язування ігрових задач з обмеженнями-розміщеннями на стратегії одного гравця складемо таблицю 1, у якій згідно до алгоритму методу наведені основні етапи (у стовпці «Алгоритм»), час виконання кожного етапу (стовпець «Час c_j ») та кількість повторень кожного етапу алгоритму (стовпець «Кількість раз»).

При розрахунку складності алгоритму потрібно визначити асимптотичну верхню границю з точністю до постійного множника [11]. Для функції $g(n)$

позначка $O(g(n)) = f(n)$ [11] означає множину функцій таких, що існує додатна константа c і натуральне n_0 такі, що $0 \le f(n) \le cg(n)$ для всіх $n \ge n_0$.

Для визначення складності алгоритму згідно таблиці 1 необхідно обчислити $T = \sum_{j=0}^{7} c_j \tau_j$:

$$T = 1(c_1 + c_3 + c_4) + nc_2 + O(n \cdot m) + O(m) + O(m) + S(m,n).$$

Таблиця 1

Оцінка складності алгоритму ітераційного методу (з обмеженнями-розміщенями на стратегії одного гравця)

№ кроку	Алгоритм	Час c_j	Кількість раз τ_j
0	Встановлення номеру ітерації N	c_1	1
1	Визначення першої стратегії першим гравцем	c_2	n
2	Обчислення скалярних добутків векторів стратегій другого гравця на вектор стратегії першого гравця	1	$O(n \cdot m)$
3	Обчислення накопичених сум скалярних добутків	1	$O(m)$
4	Вибір стратегії другого гравця	1	$O(m)$
5	Вибір стратегії першого гравця	1	$S(m,n)$
6	Обчислення значень $\overline{v}$, $\underline{v}$ та v^*	c_3	1
7	Перевірка критерію завершення роботи алгоритму	c_4	1

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

$$T = O(n \cdot m) + O(m) + S(m,n).$$

Відомо [11], що якщо $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$, то $O(f(n)) + O(g(n)) = O(f(n))$. З огляду на це, складність алгоритму набуде вигляду:

$$T = O(n \cdot m) + S(m,n). \qquad (4)$$

У виразі (4) функція $S(m,n)$ набуває різного вигляду в залежності від початкових умов задачі. Доцільно розглядати такі випадки:

1. У разі, коли кількість елементів, з яких обираються елементи розміщення становить $n = M - 1$, то знаходження розв'язку полягає в розв'язуванні задачі на перестановках, для чого, як відомо [3], достатньо виконати сортування елементів множини, тобто $S(m,n) = n \cdot \log n$ [11].

2. У іншому випадку, для знаходження оптимального розв'язку задачі необхідно виконати направлений перебір розміщень, що вимагатиме в найгіршому випадку $\frac{n!}{m!(n-m)!}$ операцій, що відповідає оцінці $\frac{n!}{m!}$: $S(m,n) = \frac{n!}{m!}$.

Таким чином, доведено теореми.

Теорема 1. Час роботи ітераційного методу типу Брауна-Робінсон для розв'язування ігрових задач з обмеженнями-розміщеннями на стратегії одного гравця має оцінку $T = O(n \cdot \log n + n \cdot m)$ у разі, коли кількість елементів, з яких обираються елементи розміщення, становить $n = M - 1$.

Теорема 2. Час роботи ітераційного методу типу Брауна-Робінсон для розв'язування ігрових задач з обмеженнями-розміщеннями на стратегії одного гравця має оцінку: $T = O\left(\frac{n!}{m!}\right)$ у разі, коли кількість елементів, з яких обираються елементи розміщення, становить $n \neq M - 1$.

Ітераційний метод типу Брауна-Робінсон для розв'язування ігрових задач з обмеженнями-перестановками у обох гравців

Опишемо даний ІМТБР. За аналогією з попереднім методом, обчислення зручно заносити в таблицю.

Крок 0. Встановлюється номер N ітерації рівний 1: $N = 1$.

Крок 1. Перша стратегія (перестановки X) першого гравця обирається випадковим чином з множини перестановок $E_m\left(P^x\right)$. Вона записується покоординатно в стовпець X таблиці.

Крок 2. У таблиці в стовпці B_j з елементами b_j $\forall j \in J_n$ записується вектор-стовпець з номером j з матриці A. Обчислюються скалярні добутки векторів-стратегій B_j другого гравця і 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 таблиці. У наступному рядку sum_l в стовпцях B_jX записуємо скалярні добутки векторів B_j та X (як сума елементів стовпця B_jX таблиці).

Крок 3. Знаходяться значення SUM_L – накопичені суми скалярних добутків (в лівій частині таблиці). У рядку SUM_L таблиці записується сума значень елементів рядка sum_l та рядка SUM_L з попереднього ($(N-1)$-го) етапу. Перший раз ($N = 1$) рядок SUM_L збігається з рядком sum_l цього етапу.

Крок 4. Стратегія $NextY$ другого гравця обирається з умови отримання ним максимального сумарного за N етапів платежу, як розв'язок задачі максимізації цільової функції на множині перестановок [14]. Практично це означає впорядкування елементів вектора P^x відповідно до порядку елементів рядка SUM_L.

Крок 5. Обчислюється значення $N\overline{v}$ – максимальний накопичений виграш другого гравця як скалярний добуток рядка SUM_R та стратегії $NextY$, воно записується в цьому ж рядку в стовпці Nv.

Крок 6. Обчислюється значення $\overline{v}$ за формулою $\overline{v} = \frac{N\overline{v}}{N}$ та записується в цьому ж рядку в стовпці $\overline{v}$.

Крок 7. У стовпець Y записується значення стратегії другого гравця. У стовпець Y поокординатно заноситься значення рядка $NextY$ лівої частини таблиці.

Крок 8. У праву частину таблиці у стовпці A_i $\forall i \in J_m$, записується вектор стратегії першого гравця – рядок i з матриці A. Обчислюється скалярні добутки вектора-стратегії другого гравця на вектори стратегій першого гравця – стовпці A_i $\forall i \in J_m$. У таблиці це зручно оформлювати, ввівши стопці: A_iY – вектор, що складається з поелементних добутків векторів Y та A_i, $i \in J_m$. Вектори Y, A_i, A_iY займають n рядків таблиці.

Крок 9. Обчислюються значення sum_r та SUM_R. У наступному рядку sum_r в стовпцях A_iY записуються скалярні добутки векторів A_i та Y (як сума елементів стовпця A_iY таблиці). У наступному рядку SUM_R таблиці записується сума значень елементів рядка sum_r та рядка SUM_R з попереднього (($N-1$)-го) етапу.

Крок 10. Стратегія $Next\,X$ (права частина таблиці) першого гравця обирається з умови отримання ним мінімального сумарного за N етапів платежу (програшу), тобто розв'язок задачі мінімізації цільової функції на множині перестановок [3] аналогічно до кроку 4. Обрана стратегія записується в рядок $NextX$ правої частини таблиці та стовпець X.

Крок 11. Знаходиться значення $N\underline{v}$ – мінімальний накоплений програш другого гравця. Мінімальне значення $N\underline{v}$ правої частини таблиці обчислюється як скалярний добуток рядка SUM_R та стратегії $NextX$ (правої частини таблиці) та записується у цьому ж рядку в стовпці $N\underline{v}$.

Крок 12. За формулою $\underline{v} = \dfrac{N\underline{v}}{N}$ обчислюється $\underline{v}$ та заноситься у стовпець $\underline{v}$ таблиці.

Крок 13. За формулою обчислюється та заноситься у стовпець таблиці.

Крок 14. Перевіряється критерій завершення роботи алгоритму – 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 виконується, то зупинка алгоритму, інакше – перехід на крок 2 алгоритму, обравши за стратегію першого гравця стратегію $NextX$. За ціну гри приймається значення $\upsilon^* = \overline{v} = \underline{v}$.

Іншими критеріями зупинки алгоритму можуть бути: проведення заданої кількості ітерацій або досягнення заданої точності $\Delta = \min\overline{v} - \max\underline{v} \le \varepsilon$, де $\varepsilon > 0$ – задана величина, $\min\overline{v}$ – мінімум зі знайдених $\overline{v}$, $\max\underline{v}$ – максимум зі знайдених $\underline{v}$, або при $\delta = \dfrac{\Delta}{\frac{1}{2}\left(\left|\min\overline{v}\right| + \left|\max\underline{v}\right|\right)}$. При цьому значення ціни гри обчислюється як $V^* = \dfrac{1}{2}\left(\min\overline{v} + \max\underline{v}\right)$.

Наближені значення p^*, q^* (оптимальні мішані стратегії першого та другого гравців) – це вектори частот застосування чистих стратегій–перестановок гравцями. Логічно вважати, що для забезпечення більшої точності знаходження чистих стратегій гравцями, необхідно проводити якомога більшу кількість ітерацій.

Для розрахунку складності [11] алгоритму складемо таблицю (табл. 2), в якій в стовпці «Алгоритм» записана програмна реалізація ітераційного методу (одна ітерація) для розв'язування комбінаторних оптимізаційних задач ігрового типу, де накладаються обмеження, що визначаються перестановками, 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.

Таблиця 2

Оцінка складності алгоритму

№	Алгоритм	Час c_j	Кількість раз τ_j
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	$O(n\log n)$ [11]
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		-
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

Продовження таблиці 2

№	Алгоритм	Час c_j	Кількість раз τ_j
	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	$O(m\log m)$ [11]
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 / cIterNum;	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	$T_0(m)$ [11]
36	CheckStrat(1);	1	$T_1(n)$ [11]
37	Result := CheckEval(aBreakType, aMaxIter);	c_{33}	1

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 $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = c < \infty$, 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 $\lim\limits_{n\to\infty} \frac{f(n)}{g(n)} = \infty,$ то

$$O(f(n)) + O(g(n)) = O(f(n)).$$

Отже:

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

Таким чином, складність алгоритму розв'язування комбінаторних оптимізаційних задач ігрового типу на комбінаторних конфігураціях, у разі, коли обмеження накладаються на стратегії двох гравців і визначені перестановками, становить: $T = O(nm + n\log n + m\log m)$.

Висновки. Необхідність розв'язування комбінаторних оптимізаційних задач ігрового типу на множинах перестановок та розміщень зумовлює актуальність дослідження можливості модифікації існуючих методів розв'язування цих задач та визначення оцінки їх складності. Здійснено поширення ітераційного методу на задачі комбінаторної оптимізації ігрового типу з різними комбінаторними конфігураціями. Для ітераційного методу типу Брауна-Робінсон, який розв'язує комбінаторні оптимізаційні задачі ігрового типу з обмеження-перестановками, що накладаються на стратегії обох гравців, отримано теоретичну оцінку складності. Подальші дослідження можуть зосередитися на удосконаленні алгоритмів і розробці нових підходів для підвищення їхньої ефективності.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Ємець О. О. Ольховська О. В. Монотонний ітераційний метод для розв'язування задач комбінаторної оптимізації ігрового типу на переставленнях. *Доповіді Національної академії наук України.* 2014. №8. С. 48-52.

2. Слабінога М. О., Чабан С. В. Розробка веб-додатків в контексті оптимізації їх швидкодії. *Таврійський науковий вісник. Серія: Технічні науки*, 2022, (3), 63-69. https://doi.org/10.32851/tnv-tech.2022.3.7

3. Стоян Ю. Г., Ємець О. О. Теорія і методи евклідової комбінаторної оптимізації. К., ІСДО, 1993. 188 с.

4. Черненко Н. Штучний інтелект в управлінні персоналом. *Таврійський науковий вісник. Серія: Економіка,* 2022, (12), 76-83. https://doi.org/10.32851/2708-0366/2022.12.11

5. Avinash K Dixit, Susan Skeath, David McAdams Games of Strategy. Fifth Edition, 2020, 1853 p.

6. Christos H. Papadimitriou, Kenneth Steiglitz. Combinatorial Optimization: Algorithms and Complexity, Courier Corporation, 1998. 496 p.

7. Dexter C Kozen. The Design and Analysis of Algorithms, 1990, 25 p.

8. Fraenkel A. Combinatorial games: Selected bibliography with a succinct gourmet introduction. *Games of No Chance 3*. Vol. 56. 2009. P. 491–575.

9. Jon Kleinberg, Eva Tardos Algorithm design. 1st ed., 2006. 864 p.

10. Nowakowski, Richard J., Landman, Bruce M., Luca, Florian, Nathanson, Melvyn B., Nešetřil, Jaroslav and Robertson, Aaron. Combinatorial Game Theory: A Special Collection in Honor of Elwyn Berlekamp, John H. Conway and Richard K. Guy, Berlin, Boston: De Gruyter, 2022. https://doi.org/10.1515/978311075541

11. Thomas H Cormen, Charles E Leiserson, Ronald L, Clifford Stein Rivest Introduction to Algorithms, 3rd Edition, 2009. 1292 p.

REFERENCES:

1. Yemets O. O. Olkhovska O. V. (2014). Monotonnyi iteratsiinyi metod dlia rozviazuvannia zadach kombinatornoi optymizatsii ihrovoho typu na perestavlenniakh [Monotone iterative method for solving game-type combinatorial optimization problems on permutations]. *Dopovidi Natsionalnoi akademii nauk Ukrainy – Reports of the National Academy of Sciences of Ukraine.* 2014. No. 8. P. 48-52 [in Ukrainian]

2. Slabinoha, M. O., Chaban, S. V. (2022) Rozrobka veb-dodatkiv v konteksti optymizatsii yikh shvydkodii [Development of web applications in the context of optimizing their performance]. *Tavriiskyi naukovyi visnyk. Seriia: Tekhnichni nauky – Taurian Scientific Bulletin. Series: Technical sciences*, (3), 63-69. https://doi.org/10.32851/tnv-tech.2022.3.7 [in Ukrainian]

3. Stoian Yu. H., Yemets O. O. (1993). Teoriia i metody evklidovoi kombinatornoi optymizatsii [Theory and methods of Euclidean combinatorial optimization]. K., ISDO, 188 p. [in Ukrainian]

4. Chernenko, N. (2022) Shtuchnyi intelekt v upravlinni personalom [Artificial intelligence in personnel management]. *Tavriiskyi naukovyi visnyk. Seriia: Ekonomika – Taurian Scientific Bulletin. Series: Economy*, (12), 76-83. https://doi.org/10.32851/2708-0366/2022.12.11 [in Ukrainian]

5. Avinash K Dixit, Susan Skeath, David McAdams (2020). Games of Strategy. Fifth Edition, 1853 p. [in English]

6. Christos H. Papadimitriou, Kenneth Steiglitz. (1998). Combinatorial Optimization: Algorithms and Complexity, Courier Corporation, 496 p. [in English]

7. Dexter C Kozen. (1990). The Design and Analysis of Algorithms, 25 p. [in English]

8. Fraenkel A. (2009). Combinatorial games: Selected bibliography with a succinct gourmet introduction. *Games of No Chance 3*. Vol. 56. P. 491–575. [in English]

9. Jon Kleinberg, Eva Tardos. (2006). Algorithm design. 1st ed. 864 p. [in English]

10. Nowakowski, Richard J., Landman, Bruce M., Luca, Florian, Nathanson, Melvyn B., Nešetřil, Jaroslav and Robertson, Aaron. (2022). Combinatorial Game Theory: A Special Collection in Honor of Elwyn Berlekamp, John H. Conway and Richard K. Guy, Berlin, Boston: De Gruyter. https://doi.org/10.1515/978311075541 [in English]

11. Thomas H Cormen, Charles E Leiserson, Ronald L, Clifford Stein Rivest (2009). Introduction to Algorithms, 3rd Edition. 1292 p. [in English]