

Мулява Ігор Ярославович

студент

Навчально-наукового комплексу

«Інститут прикладного системного аналізу»

Національного технічного університету України

«Київський політехнічний інститут імені Ігоря Сікорського»

Мулява Игорь Ярославович

студент

Учебно-научного комплекса

«Институт прикладного системного анализа»

Национального технического университета Украины

«Киевский политехнический институт имени Игоря Сикорского»

Mulyava Igor

Student of the

National Technical University of Ukraine

«Igor Sikorsky Kyiv Polytechnic Institute»

**ВИРІШЕННЯ ЗАДАЧІ АВТОМАТИЗОВАНОГО ФОРМУВАННЯ
РОЗКЛАДУ НАВЧАЛЬНОГО ЗАКЛАДУ ЗА ДОПОМОГОЮ
ГЕНЕТИЧНИХ АЛГОРИТМІВ**

**РЕШЕНИЕ ЗАДАЧИ АВТОМАТИЗИРОВАННОГО ФОРМИРОВАНИЯ
РАСПИСАНИЯ УЧЕБНЫХ ЗАВЕДЕНИЙ С ПОМОЩЬЮ
ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ**

**THE SOLUTION OF THE PROBLEM OF AUTOMATED FORMATION
OF SCHEDULE OF EDUCATIONAL INSTITUTIONS
WITH GENETIC ALGORITHMS**

Анотація. Дана стаття присвячена алгоритму для формування розкладу навчальних занять за допомогою еволюційних алгоритмів.

Ключові слова: розклад занять, еволюційний алгоритм, урахування суб'єктивних вимог.

Аннотация. Данная статья посвящена алгоритму для формирования расписания учебных занятий с помощью эволюционных алгоритмов.

Ключевые слова: расписание занятий, эволюционный алгоритм, учет субъективных требований.

Summary. This article is devoted to the algorithm for forming the schedule of training sessions using evolutionary algorithms.

Key words: schedule of classes, evolutionary algorithm, taking into account subjective requirements.

Складання розкладу є важливим завданням для університету. Задача складання розкладу розв'язується у багатьох галузях. Насамперед, при плануванні дискретного виробництва, організації пасажирських та товарних перевезень, проектуванні та проведенні навчальних занять у середній,

професійно-технічній та вищій школі. В її основу покладено необхідність забезпечення оптимального розподілу робіт серед виконавців, враховуючи просторові та часові обмеження.

Розклад сам по собі залежить від багатьох факторів. Їх можна розділити на об'єктивні(жорсткі) та

суб'єктивні(непостійні) параметри. Об'єктивні — це база даних університету, в якій зберігається інформація про аудиторії та предмети. Суб'єктивні — це побажання студентів та викладачів.

Тому метою даної статті є розробка і дослідження алгоритму, який автоматизує цей процес.

Основні поняття

Розклад — це саме по собі поняття тривіальне з точки зору сучасного життя, а от задачу його формування важко назвати тривіальною. За класичним означенням, розклад — це документ підприємства, який регламентує робочий ритм, визначає часові обмеження всіх робочих процесів і формує оптимальне розділення такого важливого ресурсу як час.

В даній роботі ми будемо зосереджувати свою увагу саме на розкладі занять навчального закладу, для якого жорсткими умовами будуть:

- предметні години, які треба відпрацювати,
- викладачі, які ці предмети ведуть,
- студенти,
- аудиторії, де ці заняття будуть проходити.

Також не таким і очевидним ресурсом буде час, який буде розділений між робочими днями та па-рами.

До нежорстких умов треба віднести:

- вимоги і побажання викладачів,
- вимоги і побажання студентів.

Жорсткі умови повинні виконуватись завжди, бо інакше розклад є хибним і збитковим. Нежорсткі умови можуть і не виконуватись, але їх виконання напряму впливає на ефективність розкладу з психологічної точки зору.

Предмет — це певна наукова дисципліна, яку проводить певний викладач певній групі студентів.

Предмети мають декілька форм занять. До класичних треба віднести лекції, практики, лабораторні, семінари, самостійні заняття, екзамени, факультативи.

Викладачі — це співробітники навчального закладу, які проводять заняття і є рушійною силою навчального процесу. Кожен викладач має свою посаду, наукове звання та ступінь, які прямо відповідають їх важливості та досягненням на кафедрі. Зрозуміло, чим вища посада, тим більш пріоритетні є вимоги даного викладача.

У кожного викладача є свої вимоги до розкладу і пріоритети цих вимог, тому їх всі треба враховувати. А також кожен викладач має свою важливість, яка вираховується через його грошовий оклад, для спрощення обрахунків.

Також треба відвітити, що не всі форми занять можуть проводити всі викладачі. Лекції можуть вести тільки лектори-доценти, практики та лабораторні можуть вести аспіранти, чи нижчі за званням особи.

Студенти — такі самі важливі учасники процесу, як і викладачі, проте їх набагато більше, тому у розкладі будемо враховувати лише групи студентів, а не кожного окремо.

Жорсткі вимоги є такі самі як і для викладачів, бо студенти — теж люди, проте є одна відмінність, яка зумовлена тим, що їх багато. А саме те, що якщо аудиторія мала, то вся група в неї не поміститься, тому це треба обов'язково враховувати.

Загальний опис алгоритму

Ця задача буде реалізовано завдяки генетичному алгоритму та цільовій функції, яка відповідатиме побажанням. Сама цільова функція буде складатися з двох частин: переваги голосування студентів та викладачів. Записано це буде в формі матриці переваг, яка буде побудована через голосування. За допомогою даної функції ми зможемо оцінити розклад.

Сам розклад буде генеруватися за допомогою певного алгоритму, після чого буде оцінений описаною вище функцією та, у разі необхідності, буде удосконалений в наступній генерації та оцінений знову. І так до тих пір поки не буде знайдений оптимальний варіант.

Цільова функція

Формуючи цільову функцію треба врахувати багато факторів, які визначають на оцінюють розклад як ефективний, валідний та оптимальний з точки зору навчального процесу. Давайте визначимо основні вимоги до функції:

- Функція повинна бути відображенням виконання вимог навчального процесу.
- Функція повинна давати більше значення, тоді, коли її аргумент є вигіднішим і кращим за гірші аргументи як і з об'єктивної так і з суб'єктивної точки зору.
- Функція може мати однакові значення для різних розкладів, з цього випливає, що класичні методи оптимізації до неї застосувати не можна.
- Значення, які вона повертає не можуть бути від'ємні.
- Функція повинна мати потенціал до розширення на випадок, якщо кількість викладачів, чи груп зросте.

З огляду на ці вимоги можна сформулювати формулу (1), яка буде діяти на множині (2). Ця функція є дискретною з великою кількістю розривів. Залежить вона на пряму від виконання вимог розкладом. Гарантуються це завдяки двом індикаторним функціям.

$$F(r) = a_s \sum_{j=1}^l x_j \chi\{Z_j^v\} + \tag{1}$$

$$+ a_L \sum_{j=1}^K y_j \sum_{i=1}^M \chi\{L_i \in T_i\} \sum_{j=1}^K d_{ij}^j \chi\{Z_{ij}^T\} \rightarrow \max$$

$$r \in \Omega(P, S, L, A) \tag{2}$$

де r — розклад, a_s, a_L — вагові коефіцієнти, що вказують на пріоритети викладачів і студентів, як суб'єктів навчального процесу, x_j, y_j — пріори-

тети вимог студентів і викладачів, Z_j^v — вимоги груп студентів, L_i — викладачі, T_j — групи викладачів, Z_u^T — переваги викладачів, d_u^j — пріоритети таких побажань, l — кількість вимог студентів, K — кількість груп викладачів, які розподілені посадами, науковими ступенями та вченими званнями, M — кількість викладачів, і n — кількість викладачів в i -й групі, Ω — область обмежень, P, L, A — множина навчальних дисциплін, викладачів і аудиторій, відповідно [2, с. 89–90].

Пріоритетні вектори будуть формуватися вручну оператором через обмеження в ресурсах і часі під час виконання цієї роботи, але в майбутньому залишає простір до розширення. Кожен вектор буде множитись на вектор індикатор, який у відповідних позиціях буде мати «1» якщо вимога виконується, і «0», якщо — ні, як на формулі (4). Це забезпечить простоту реалізації, що дозволить програмі просто перевіряти певні умови і передати обрахунок якомусь іншому методу [4, с. 6–9].

$$\chi\{Z_j^v\} = \begin{cases} 1, & \text{якщо } Z_j^v \text{ виконується} \\ 0, & \text{в іншому випадку} \end{cases} \quad (3)$$

Дана функція є великою і може спантеличити з першого погляду, тому давайте приведемо спрощений приклад, щоб було ясно зрозуміло як вона працює. Вихідні дані для неї є вектор вимог студентських груп, вектори вимог кожного викладача, та вектор пріоритетів самих викладачів, який відповідних позиціях має коефіцієнт пріоритети кожного викладача. Спочатку іде перевірка виконання вимог, під час якої ці вектори множаться на відповідні індикаторні функції.

Оскільки дана функція має неперервний характер, то застосовувати до неї методи класичного математичного аналізу не можна. Тому було запропоновано використовувати генетичний алгоритм. Але перед тим треба сформулювати як буде побудований розклад у пам'яті.

Математична модель розкладу

З огляду на інформацію описану в першому розділі ми можемо створити математичну модель розкладу. Напишемо розклад у такій формі, як показано на таблиці 1.

Дана модель описана на таблиці 1 потребує спрощення, оптимізації на вдосконалення. Всю інформацію про розклад ми будемо розміщувати в паралелепіпед. Спочатку ми розділимо її на 4 групи, в які ми об'єднаємо певні поля як показано у формулах 4, 5, 6, 7.

$$X_1 = \langle \text{День} \rangle \quad (4)$$

$$X_2 = \langle \text{Пара} \rangle \quad (5)$$

$$X_3 = \langle \text{Аудиторія} \rangle \quad (6)$$

$$Z = \langle \text{Викладач} - (\text{Предмет} - \text{Тип}) - (\text{Курс} - \text{Група}) \rangle \quad (7)$$

Тоді X_1, X_2, X_3 — координати вузла у паралелепіпеді, а Z — значення вузла. Поля Група і Курс (далі просто Група) можна об'єднати в одне поле, бо вони однозначно відрізняють групу на факультеті. Предмет і Тип (далі просто Предмет) об'єднуються, бо їх суміщення однозначно визначаються одиницю навчального процесу з точки зору дисциплін. День, Пара і Аудиторія не розділяються, бо вони відображають фізичні жорсткі умови, які не можна порушити і це буде гарантувати нам те, що в одній аудиторії в той самий час не буде проходити два заняття [1, с. 67–69].

Поля Викладач — предмет — група будуть міститись у вузлах паралелепіпеда, що виходить з точки зору звичайної логіки. Тому запис у списку розкладу у нашій моделі буде виглядати так як у формулі (8).

$$r_n = (D_i, T_j, R_k, N_l) \quad (8)$$

де D_i — день тижня, T_j — номер пари, R_k — аудиторія, N_l — ланка, яка з'єднує в собі ключ <Викладач — предмет — група>, r_n — Розклад. Звідси можна зробити висновок, про розмірність кубу, який буде мати 3 виміри з четвертим у вузлах. Розміри кубу будуть статичні, оскільки в тижні всього 6 робочих днів, та в день може бути лише 6 пар, а кількість аудиторій буде братись з даних про кафедру, але про це в наступному розділі. Графічно можна зобразити розклад як показано на рисунку 1.

Формуючи розклад таким чином, ми зможемо забезпечувати жорсткі умови та мати зручний доступ до будь-якого значення, чи предмету в розкладі. Треба зауважити, що обмеження, яке полягає в тому, що один викладач та одна група може бути лише в одній ланці одночасно гарантуватись даною схемою не може, тому це буду гарантувати алгоритм генерації розкладу.

Метод оптимізації цільової функції

Для виконання даної задачі було обрано еволюційний генетичний алгоритм, який полягає в тому, що кожна наступна генерація розкладів буде формуватися з минулих ітерацій. Таким чином цільова функція буде рости без жодного втручання з сторони користувача, оскільки ця модель є закритою.

Загальні кроки, які треба буде реалізувати алгоритмом такі [1, с. 83–84]:

1. Створення першої генерації розкладів випадковим чином.
2. Оцінка цих розкладів цільовою функцією.

Таблиця 1

Початкова модель розкладу

День	Пара	Курс	Група	Предмет	Викладач	Тип	Аудиторія
...

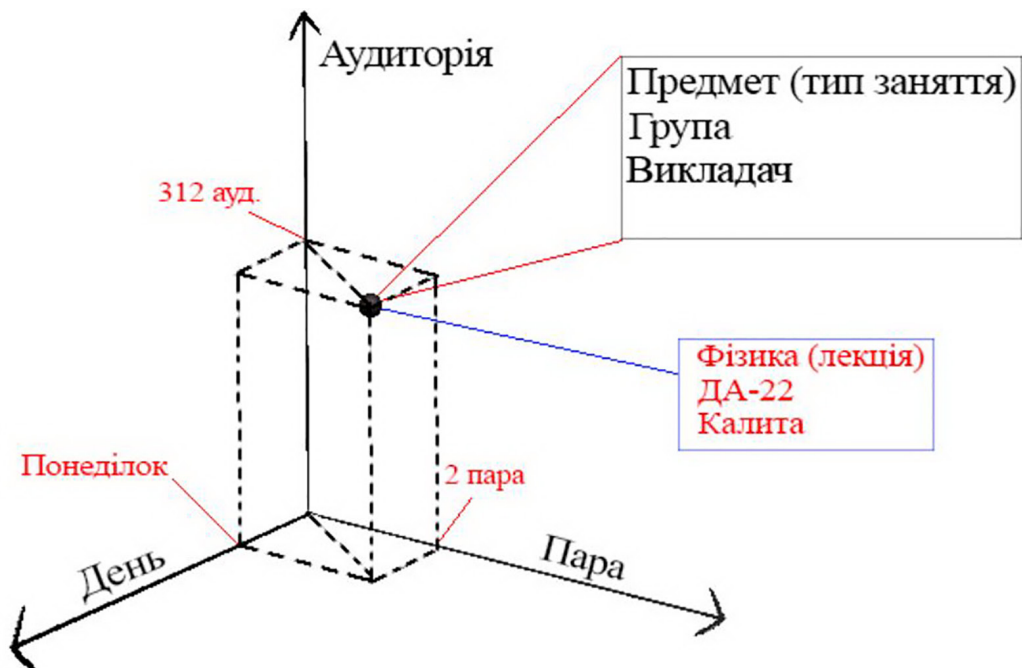


Рис. 1. Графічне уявлення розкладу

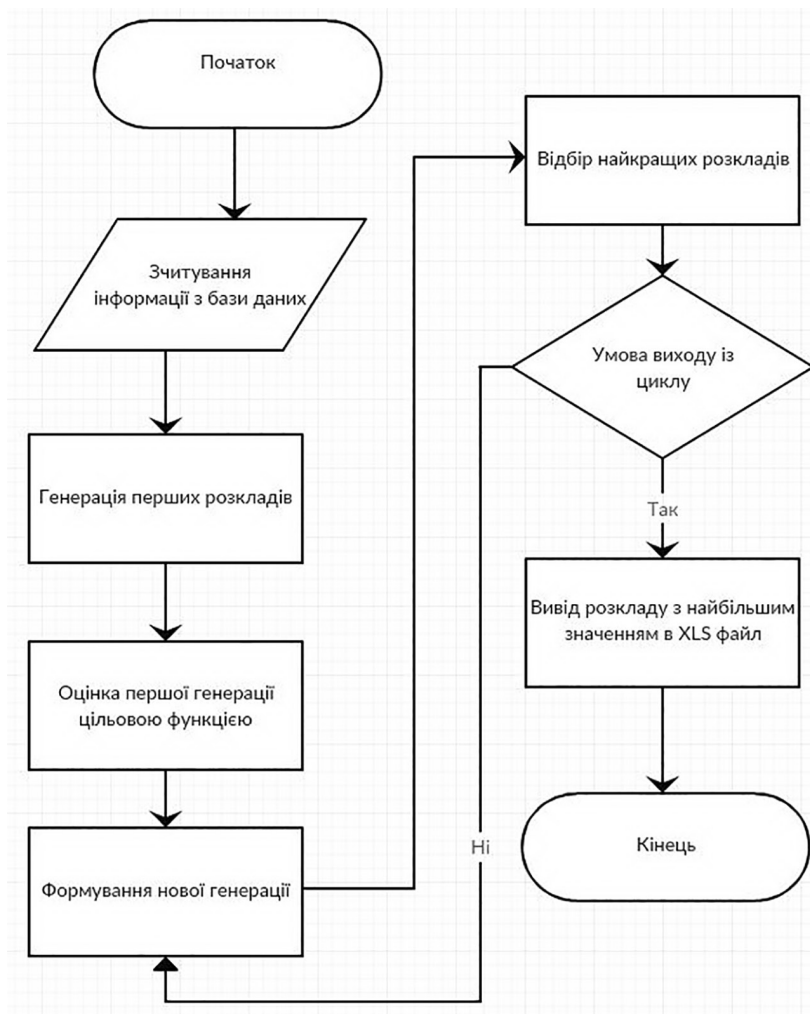


Рис. 2. Блок-схема алгоритму формування розкладу

		Понеділок						Вівторок																				
Група 21	1 пара	SA лекція Kiselov 304	2 пара	Economics практика Roshchyna 310	3 пара	TZI лабораторна Grechko 101	4 пара	TZI лекція Karshuk 307	5 пара	СРР лабораторна Khaichenko 206	6 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара											
Група 22	1 пара	SA лекція Kiselov 304	2 пара	Economics практика Roshchyna 310	3 пара	TZI лабораторна Grechko 101	4 пара	TZI лекція Karshuk 307	5 пара	СРР лабораторна Khaichenko 206	6 пара	1 пара	SA лабораторна Chekaluk 206	2 пара	СРР лекція Khaichenko 303	3 пара	SA лекція Kiselov 304	4 пара	Economics практика Roshchyna 310	5 пара	6 пара							
Середа																												
Четвер																												
Група 21	1 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара			
Група 22	1 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	1 пара	СРР лабораторна Golubova 102	2 пара	СРР лабораторна Grechko 102	3 пара	4 пара	5 пара	6 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	
П'ятниця																												
Субота																												
Група 21	1 пара	СРР лекція Khaichenko 307	2 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	
Група 22	1 пара	СРР лекція Khaichenko 307	2 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	1 пара	SA лабораторна Chekaluk 101	2 пара	3 пара	4 пара	5 пара	6 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара	1 пара	2 пара	3 пара	4 пара	5 пара	6 пара

Рис. 3. Згенерований розклад

3. Генерація нового «потомства» з минулої ітерації.

4. Відкидання гірших розкладів.

5. Повторювати кроки 2–4 до тих пір, поки не буде отримана задана точність, або кількість кроків перебільшить допустиму.

Для зручності також алгоритм показаний на рисунку 2:

Перша генерація буде формуватися випадково. Спочатку алгоритмом вибирається випадковий день тижня та час, вибирається аудиторія, щоб було достатньо місць на групу. Потім іде перевірка, чи нема в цей час в цій аудиторії якогось заняття, якщо нема, то назначаємо заняття, яке вибрали раніше зі списку занять, якщо є, то шукаємо інші координати [3].

І так повторювати для кожного заняття. Коли розклад готовий, перевіряємо його на валідність та записуємо його у список розкладів. Коли список розкладів буде заповнений, виходимо з циклу.

Результати

Сформований таким чином розклад можна побачити на рис 3.

Даним розкладом виконуються всі жорсткі вимоги: нема жодних дублікатів, жоден викладач, чи група студентів не знаходяться у двох місцях одночасно. Та вибрані відповідні аудиторії залежно від кількості студентів та типу заняття.

Час обрахунку 2,356742412 секунди за 7 кроків.

3 точки зору нежорстких вимог розклад чудово реалізує ці потреби, а саме наявність вихідного дня під час тижня, відсутність вікон, мала кількість перших пар, відсутність вікон у викладачів, пари в основному зосереджені в околі 2–3 пари та мінімум пар у суботу.

Виконання цих вимог показує, що генерація розкладу пройшла успішно. Також давайте поглянемо на графік залежності цільової функції від кількості кроків як показано на рис 4.

Також треба розглянути графік, який показує залежність різниці між мінімумом та максимумом цільової функції від ітерації як у формулі 10. Цей графік зображена на рис 5.

Висновок. З огляду на отримані результати можна зробити наступні висновки. Алгоритм формує розклад, який гарантовано відповідає жорстким вимогам. Жодні фізичні умови, чи логічні не є порушені під час генерації. Отриманий розклад відповідає нежорстким вимогам викладачів та студентів лише частково, але достатньо, щоб рівень виконання був задовільний. Це важливий факт, бо задовольнити всіх одночасно неможливо. Вимоги викладачів були виконані в пріоритеті, а вимоги викладачів, які є вищими за званням були виконані навіть краще.

Щодо часу, то алгоритм зайняв дуже мало часу, щоб згенерувати горючий робочий розклад. Цей час компенсує дні, а може навіть тижні роботи оператора факультету, якщо не враховувати інформацію, яку вводить оператор лише один раз. Такі дані як інформація про аудиторії, викладачі, групи не часто підлягають до змін. А плани навчання міняються тільки трохи, але не сильно.

Щодо залежності цільової функції від ітерації видно, що середнє її значення росте з кожною ітерацією, що є чудовим доказом роботи алгоритму. Але значення різниці містить стрибок.

Результуючи все сказане вище, можна зробити висновок, що генерація пройшла успішно і розклад пройшов тестування.

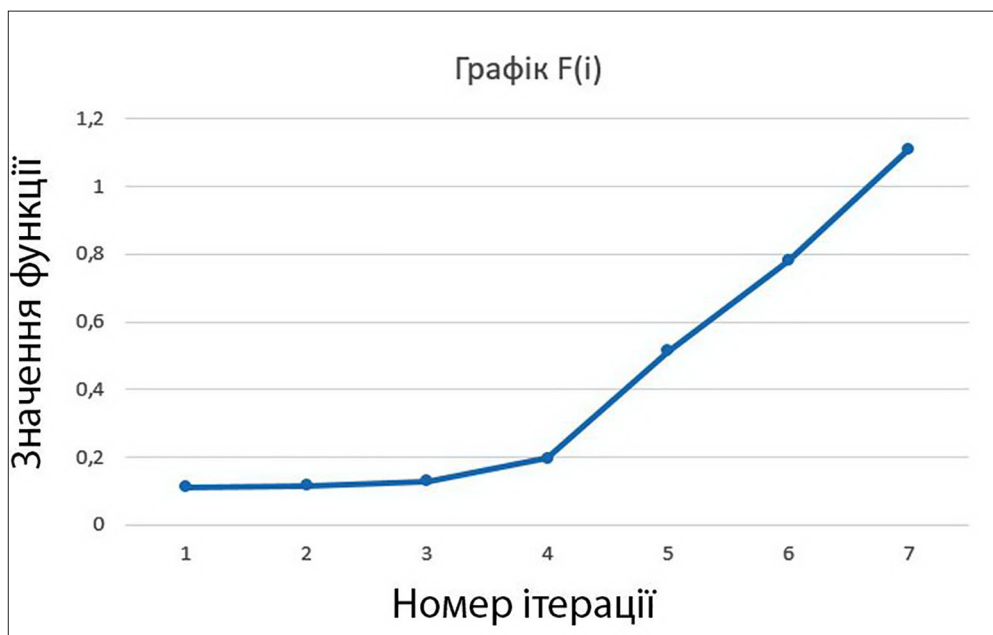


Рис. 4. Графік залежності середнього значення функції по всій генерації від ітерації еволюційного циклу

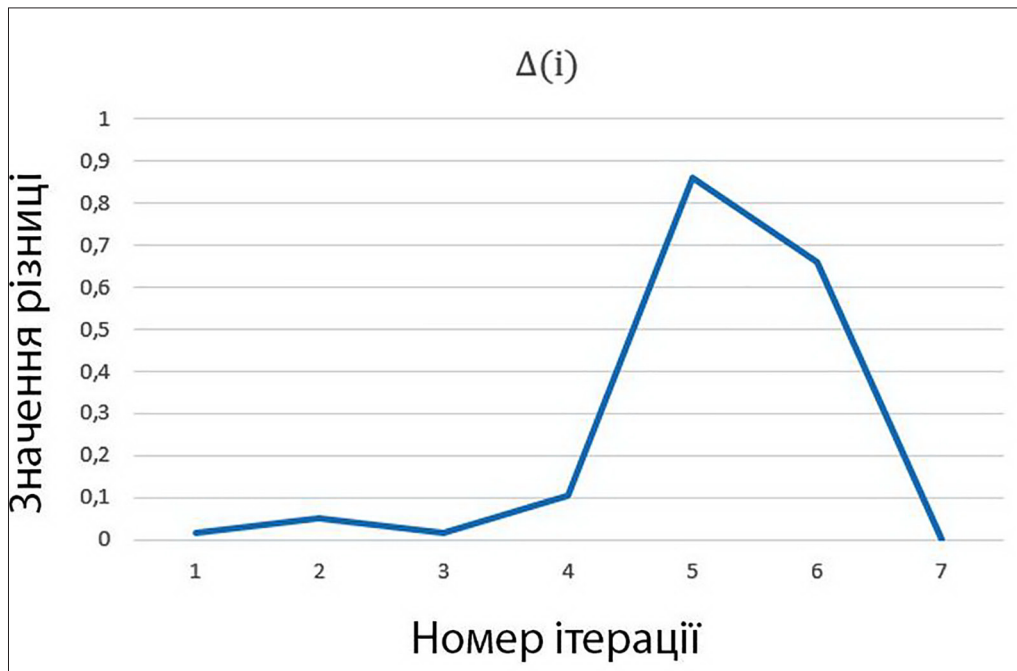


Рис. 5. Графік залежності різниці між мінімумом та максимумом цільової функції від ітерації

Література

1. Снитюк В. Є. Про особливості формування цільової функції та обмежень в задачі складання розкладу занять / Снитюк В. Є., Сіпко Є. Н. // Математичні машини і системи. — 2014. — № 3. — С. 67–76.
2. Снитюк В. Є. Аспекти формування цільової функції в задачі складання розкладу занять у вищих навчальних закладах на основі суб’єктивних переваг / Снитюк В. Є., Сіпко Є. Н. // Автоматика. Автоматизація. Електротехнічні комплекси і системи. — 2013. — № 2. — С. 98–104.
3. Бевз С. В. Розробка автоматизованої системи формування розкладу магістратури / Бевз С. В., Войтко В. В., Бурбело С. М., Шоботенко А. М. // Інформаційні технології та комп’ютерна техніка. — 2009. — № 4. — С. 30–65.
4. Бевз С. В. Автоматизація процесу формування розкладу сесії / Бевз С. В., Войтко В. В., Бурбело С. М., Куба Т. О., Сухоносів О. О. / Принципові концепції та структурування різних рівнів освіти з оптико-електронних інформаційно-енергетичних технологій. — 2009. — № 4. — С. 25–36.