

DOI 10.26886/2414-634X.6(33)2019.4

UDC: 519.8

ABOUT THE PROBLEMS OF THE SCHEDULE THEORY IN EDUCATION

L. V. Zinchenko

National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Ukraine, Kyiv

In the article, a number of statements of the problems of the theory of schedules with accentuation are considered and emphasized for those that reflect the peculiarities of the modern enterprising education of persons who are interested in teachers (mentors), which does not require the presence and analysis of groups and audiences. A classification of such tasks was carried out and a new task of composing schedules from the sphere of education was proposed, it uses such basic parameters as the coefficient of similarity between the mentor and the student, as well as the importance of the subject area for the mentor. The methods of solving the optimization problems of the theory of decompositions are considered. Since the considered problem of scheduling training sessions is NP-complicated, an overview of existing applied methods and task statements is necessary to determine the relevance of this task statement. Further research should be aimed at formalizing and solving similar scheduling tasks using information technology.

Keywords: education, online education, theory of schedules, mentor, student, NP-complex.

Зінченко Л. В. До задач теорії розкладів у сфері освіти / Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Україна, Київ

В статті розглянуто ряд постановок задач теорії розкладів з акцентуацією на такі, які відображають особливості ініціативної освіти бажаючих у вчителів (менторів), де не вимагається наявності та аналізу груп і аудиторій. Проведена класифікація таких задач і запропоновано постановку нової задачі складання розкладів із сфери освіти, у якій використовуються такі принципові параметри, як коефіцієнт схожості ментора та учня, а також важливість предметної області для ментора. Розглянуто методи розв'язування оптимізаційних задач теорії розкладів. Оскільки розглянута задача складання розкладу навчальних занять є NP-складною, актуальним є огляд наявних прикладних методів і постановок задач для визначення актуальності цієї постановки задачі. Подальші дослідження доцільно спрямувати на формалізацію та розв'язування подібних задач складання розкладів з використанням інформаційних технологій.

Ключові слова: освіта, онлайн освіта, теорія розкладів, ментор, учень, NP-складність

Постановка проблеми. В умовах глобалізації та діджиталізації актуальною стає онлайн освіта. Ми живемо в інформаційному суспільстві, у світі, у якому все змінюється з величезною швидкістю, у якому кожній людині щодня доводиться опрацьовувати великі потоки інформації. Багато процесів автоматизуються, і отримання освіти не виключення. Нині у всьому світі здійснюється автоматизація підприємств, і в Україні також. У зв'язку з цим, багато людей втрачають роботу і змушені отримувати ще одну кваліфікацію, і оскільки часу і грошей на отримання нової освіти у ВНЗ немає, краще зайнятися самоосвітою.

Тисячі людей щодня не тільки перекваліфікуються, а і паралельно з існуючою роботою прагнуть освоїти ще одне або декілька

занять (чи підняти рівень знань в області, що безпосередньо їх торкається кожного дня), чи то для власного розвитку, чи для більшого заробітку, чи від поклику серця або від нудьги. Найлегше це зробити в Інтернеті. Проте не завжди є підходящі ресурси для цього.

Так, наприклад, учні, які роблять домашні завдання, чи студенти, які виконують лабораторні та практичні завдання в університетах, потребують допомоги, людину, яка пояснить їм незрозумілий матеріал. Для цього є репетитори або ментори. Ментор – це людина, що має теоретичні чи практичні знання будь-якого рівня у певній предметній області, виявляє бажання ділитися своїми знаннями та навчати інших в тій чи іншій формі, допомагає навчатися та консультує у питаннях, які відповідають його предметній області. самостійно встановлює інтенсивність та напрямок навчання. Ментор – це людина, яка самостійно встановлює інтенсивність та напрямок навчання. Завдання ментора – допомогти, коли потрібна допомога. Питання в тому, як знайти менторів? Якщо раніше пошук здійснювався в газетах чи оголошеннях на зупинках, тепер пошук повністю перейшов на Інтернет.

Нині в Україні існує дуже мало Інтернет-ресурсів, які надають можливість викладачам та менторам викладати безкоштовні оголошення про готовність надавати допомогу. Проте ці існуючі аналоги мають ряд недоліків, такі як: незручний та несучасний інтерфейс, мало функціоналу, заточені під одну справу.

Метою роботи є підвищення якості інформування потенційних споживачів та інтелектуалізація процесів надання освітніх послуг шляхом розробки оригінального програмно-алгоритмічного забезпечення та реалізації його у вигляді спеціалізованої програмної системи.

Досягнення мети базується на розробці оригінальних математичних методів та алгоритмів та їх реалізації у спеціальній

системі, яка призначена для формування та модифікації чи уточнення розкладів занять учні з менторами.

У сфері освіти наявні задачі, що зазвичай належать до теорії розкладів. Проте якщо мова йде про онлайн освіту, то постановка задачі може відрізнитися від класичних. Отже, дослідимо наявні постановки задач, класифікуємо їх та методи їх розв'язування. На основі цього аналізу зможемо визначити актуальність формулювань нових постановок задачі або видозмінення існуючих для врахування різних вимог проектів, які стосуються онлайн освіти. Відзначимо, що близькими до задач, на яких акцентується розгляд в статті, та алгоритмів їх розв'язування є задачі та підходи до їх розв'язування, викладені рядом авторів, зокрема, в [1-4].

Мета роботи. Дослідити існуючі методи розв'язування задач теорії розкладів із сфери освіти та існуючі постановки низки задач теорії розкладів для подальшого визначення актуальності нової постановки задачі із сфери освіти та її місця серед наявних постановок.

Огляд постановок задач. У загальній постановці задача складання розкладу полягає в наступному: використовуючи деяку множину ресурсів або обслуговуючих пристроїв, повинна бути виконана деяка фіксована множина завдань із заданим набором характеристик. У багатьох випадках розглядаються такі характеристики як: перелік операцій, які входять до складу кожного із завдань; обмеження на послідовність виконання операцій кожного завдання й часткові послідовності термінів їх завершення, а також терміни виконання завдань; ресурси, які необхідні для виконання кожної з операцій завдань; час та вартість виконання робіт кожної із операцій завдань при використанні різних видів ресурсів; директивні терміни початку та завершення виконання деяких чи всіх завдань; перелік та

характеристики ресурсів, необхідних для виконання всіх операцій кожного із завдань [1].

Коло розв'язуваних у рамках теорії розкладів задач можна розділити на наступні групи:

- задачі розподілу завдань, вибір ресурсів (пристроїв) для їх виконання;
- задачі упорядкування виконання множини завдань на кожному із пристроїв;
- задачі узгодження часу виконання виділеної підмножин завдань, що виконуються;
- задачі вибору складу та розміщення обладнання, визначення кількості, об'ємів та розміщення проміжних буферних накопичувачів;
- маршрутизація руху виробничого транспорту;
- побудова розкладів проведення занять, руху автомобільного, залізничного, міського транспорту та руху літаків [2].

Задача, яка розглядається у даній роботі, належить до останнього групи. Багато науковців приділяють увагу задачам побудови розкладів проведення занять. Наприклад, у [3] запропоновано розроблення математичної моделі розкладу занять у ВНЗ, яка дозволяла б ефективно розв'язувати задачу автоматичного складання розкладу занять і була б гнучкою у випадку адаптації системи для розв'язання конкретної практичної задачі. Пріоритетне право викладачів на розподіл занять в алгоритмі складання розкладу визначається за методом ранжування посад. У [4] розглядається метод автоматизованого створення розкладу, що забезпечує вимоги Болонського процесу щодо складання індивідуальних планів магістрантів. Класична задача навчального планування оперує поняттями учень, група, викладач, заняття, аудиторія.

Відмінність наукових робіт на цю тематику полягає у розгляді різних обмежень відносно вищевказаних понять. Можна виділити такі:

- один викладач у кожен момент часу проводить не більше одного заняття [5-8];
- у кожен момент часу в одній аудиторії може проводитися не більше одного заняття [5-8];
- у однієї навчальної групи у кожен момент часу може бути не більше одного заняття [5-8];
- умова відсутності “вікон” для навчальних груп [9,10];
- обмеження на максимальну кількість пар для учнів, викладача [9];
- відповідність типу аудиторії і виду заняття [9];
- часові обмеження на день [11];
- пріоритетне проведення певного типу занять (наприклад, лекцій) на перших парах [11];
- врахування побажань викладачів [7,11];
- зменшення переходів між аудиторіями [7,11].

У ряді досліджень подібні обмеження можуть комбінуватися, крім того, також може вводитися пріоритизація обмежень [12].

Огляд методів розв’язування задач теорії розкладів. Задача складання розкладу навчальних занять є NP-складною, тобто такою, для якої невідомий алгоритм її розв’язування із поліноміальною складністю в залежності від розмірності. Задачі теорії розкладів мають широке застосування, наприклад, при оптимізації роботи підприємств, де виконується велика кількість операцій в різні проміжки часу, чи при оптимізації процесів обробки великих обсягів інформації.

Багато задач теорії розкладів зводяться до задач комбінаторної оптимізації (ЗКО).

Індивідуальною задачею оптимізації називається пара $(f(x|c), X)$, де $f(x|c)$ – цільова функція, c – набір даних задачі (вхід задачі), а X – простір розв'язків задачі (простір пошуку).

Під задачею оптимізації розуміють сукупність усіх можливих індивідуальних задач.

Залежно від типу простору розв'язків X розрізняють задачі неперервної, комбінаторної та змішаної оптимізації [13,14]. Перший клас утворюють задачі, у яких простір X є неперервним (континуальним). Щодо другого класу, то тут існують різні підходи до означення як самого поняття ЗКО, так і його підкласів – власне комбінаторних задач, а також задач дискретного й цілочислового програмування [13].

Існує багато підходів до класифікації алгоритмів комбінаторної оптимізації (АКО) – за точністю, типом використаних просторів, структурою обчислювальної схеми тощо.

За складністю структури АКО можна виділити [13]:

- прості алгоритми;
- комбіновані алгоритми;
- метаевристики;
- гібридні метаевристики;
- гіперевристики.

Комбіновані алгоритми утворюються шляхом послідовного застосування двох чи більше ітераційних алгоритмів з передаванням розв'язків від одного до іншого. У метаевристиках, про що піде далі, здійснюється вкладення одного алгоритму/процедури в іншу стратегію.

Гіперевристикою (гіперевристичним алгоритмом) називають метод пошуку, орієнтований на автоматизацію процесів вибору, комбінування або адаптації чи налаштування кількох простіших алгоритмів (евристик або метаевристик) для ефективного розв'язання

ЗКО чи їх класів. Це може досягатися як вибором наявних евристик чи їх фрагментів, так і генеруванням нових.

Таким чином, якщо метаевристики та інші алгоритми здійснюють переважно пошук у просторі розв'язків ЗКО, то простором пошуку для гіперевристик є множина евристик (простіших алгоритмів чи їх частин).

За впливом на ландшафт пошуку більшість АКО можна віднести до таких, що залишають його незмінним. Проте є алгоритми, які модифікують цей ландшафт шляхом [13]:

- зміни простору розв'язків (наприклад, послідовні алгоритми);
- зміни цільової чи оцінкової функції (алгоритми керованого локального пошуку);
- варіації системи околів, що використовується при пошуку (алгоритми локального пошуку (ЛП) зі змінними околами, метод вектора спаду з пульсуючими околами).

Якщо робота алгоритму базується на безпосередніх даних ЗКО, то такі АКО належать до *задаче-орієнтованих* алгоритмів. У деяких нових АКО використовуються не стільки прями дані ЗКО, скільки спеціальна модель задачі, що розв'язується (наприклад феромонна матриця та матриця маршрутів у ОМК), – такі алгоритми отримали назву *моделе-орієнтованих*.

Точні алгоритми можна поділити на загальні методи та спеціальні алгоритми.

До загальних методів відносяться [13]:

- повний перебір (вичерпний пошук);
- метод гілок і меж (МГіМ);
- метод гілок і відтинань;
- послідовний аналіз варіантів (ПАВ, "київський віник");
- динамічне програмування (метод Беллмана).

Спеціальні алгоритми будуються на основі врахування специфіки конкретної задачі оптимізації, що розв'язується, тому мають вузький спектр застосування.

Далі розглянемо наближені АКО, бо саме вони, у більшості випадків, застосовуються при розв'язуванні практичних ЗКО. Для цього зведемо їх в таблицю, де використано такі скорочення: GRASP (англ. greedy randomized adaptive search procedure) — послідовний жадібний алгоритм із стохастичною процедурою вибору варіантів побудови розв'язку [14]; ЛП – локальний пошук.

Таблиця 1

Класифікація основних наближених методів комбінаторної оптимізації

| № | Загальна назва алгоритму | Назва алгоритму |
|---|--------------------------|---|
| 1 | Послідовні алгоритми | Конструктивні (у т.ч. “жадібні”) |
| | | Алгоритми на базі точних методів (МГІМ, ПАВ тощо) |
| | | GRASP |
| 2 | Детермінований ЛП | Стандартний ЛП |
| | | Пошук зі змінюваними околами |
| | | Керований ЛП |
| | | Табу-пошук |
| 3 | Стохастичний ЛП | Алгоритми імітаційного відпалу |
| | | G-алгоритми |
| | | Повторюваний ЛП |
| | | Квантовий відпал |
| 4 | Еволюційні алгоритми | Генетичні алгоритми |
| | | Міметичні алгоритми |
| | | Імунні алгоритми |
| | | Диференціальна еволюція |

| | | |
|---|-------------------|----------------------------------|
| 5 | Ройові алгоритми | Мурашині алгоритми |
| | | Оптимізація потоком частинок |
| | | Бджолині алгоритми |
| | | Алгоритми світлячків |
| 6 | Методи сканування | Розсіяний пошук (Scatter Search) |
| | | Перекомпоновка маршрутів |
| | | МДБ |
| | | <i>H</i> -алгоритми |
| 7 | Спеціальні методи | “Іди в найближче місто” |
| | | Алгоритм Ліна-Кернігана |
| | | ϵ -наближені алгоритми |

Враховуючи тематику, що розглядається, далі піде мова про деякі групи наближених алгоритмів.

Генетичні алгоритми. Базується на тому, що на кожному кроку обчислень розглядається множина розв'язків (популяція) і певним чином вибираються пари розв'язків, які рекомбінуються, і таким чином створюються нащадки. Цим створюється розширена популяція, із якої за правилами селекції формується наступна популяція. Таким чином, можна розділити алгоритм на три етапи:

- 1) генерація проміжної популяції шляхом відбору поточного покоління;
- 2) схрещування особин проміжної популяції шляхом кросовера, у результаті чого формується нове покоління;
- 3) мутація нового покоління.

Для кожної задачі, що розв'язується, необхідно визначити функцію пристосованості та спосіб кодування, в іншому алгоритм універсальний і етапи алгоритму для всіх задач виконуються однаково.

Не рекомендується використовувати ці методи, коли необхідно знайти всі розв'язки задачі, а не один із них. Мінусом цих алгоритмів є те, що вони ігнорують специфіку задачі, а час розрахунку функції пристосованості часто буває достатньо великим.

Мурашині алгоритми. Алгоритм базується на поведінці мурах, що шукають шлях від мурашника до їжі. Проте для кожного конкретного типу задач потребують визначення параметрів, що можуть бути отримані лише експериментально.

Метод імітаційного відпалу є одним із представників поширеного класу методів стохастичного локального пошуку. Характеризується тим, що при пошуку оптимального розв'язку на кожному кроці має ймовірність переходу у стан з більш високим значенням цільової функції (для задач на мінімізацію), що дозволяє виходити з локальних оптимумів. Чим більша ітерація алгоритму, тим більше зменшується ця ймовірність на кожному кроці.

Серед наближених оптимізаційних методів розв'язання ЗКО окремий клас утворюють *метаевристики*. За своєю природою метаевристики об'єднують простіші алгоритми чи техніки в межах обчислювальних схем вищого рівня, які спрямовані на ефективне вивчення простору пошуку. Точніше, метаевристика — це метод розв'язання широкого класу обчислювальних задач шляхом такого комбінування існуючих процедур, при якому одна є провідною, а інша (чи кілька) - підлеглою (підлеглими). Як провідними, так і підлеглими процедурами часто стають відомі евристики чи алгоритми, зазвичай рандомізовані. Якщо складовою метаевристики є певний математичний метод, то вживають термін *матеевристика*.

Серед еволюційних методів, окрім генетичних алгоритмів, які призначені для розв'язання ЗКО, визначених у бінарних просторах і комбінаторних просторах зі складнішими конфігураціями (наприклад у

просторі перестановок), виділяють ще метаевристики на основі штучних імунних систем і методи диференціальної еволюції.

Пошук концепцій для розробки алгоритмів комбінаторної оптимізації зумовив появу в останній час нових класів алгоритмів, які нав'язані природою. Помітне місце серед таких алгоритмів займають методи ройового інтелекту. Найпоширеніші серед таких обчислювальних схем - оптимізації мурашиними колоніями (Ant Colony Optimization) та потоком частинок (Particle Swarm Optimization), бджолині алгоритми й методи штучних бджолиних колоній (Bee Algorithms, Artificial Bee Colony) [13].

Отже, серед зазначених методів на практиці перевагу надають метаевристичним алгоритмам, оскільки вони характеризуються властивостями запобігання потрапляння в локальні оптимуми і гарантують отримання наближеного розв'язку за прийнятний час. Перспективним є використання мурашиних алгоритмів розв'язування задач, а для пошуку початкових наближень – методів локального пошуку.

Подальші дослідження доцільно спрямувати на постановку та формалізацію таких задач складання розкладів, які відображають особливості ініціативної освіти бажаючих у вчителів (менторів), де не вимагається наявності та аналізу груп і аудиторій, а також використовуватимуться такі принципові параметри, як коефіцієнт схожості ментора та учня, а також важливість предметної області для ментора.

Математична постановка задачі

Є множина учителів (менторів), що навчають одному чи декільком предметам; учні подають заявки на вивчення предмету, який відноситься до однієї із областей знань. Таким чином, маємо множини учнів, менторів, предметних областей, заявок на навчання та часові

інтервали, в яких ментори доступні для навчання. Перенумеруємо кожну з цих множин.

Введемо ряд позначень:

P_n – множина предметних областей;

N – кількість предметних областей;

P_m – множина менторів;

M – кількість менторів;

P_s – множина учнів;

S – кількість учнів;

P_r – множина заявок від учнів про допомогу;

R – кількість заявок;

P_t – множина часових інтервалів;

T – кількість часових інтервалів.

Отже, до k -ї предметної області, $k \in P_n$ відноситься певна кількість менторів $m_{kj}, j = 1, \dots, M, j$ – номер ментора. І навпаки, кожен j -й ментор може мати відношення до декількох предметних областей із вказанням їх пріоритету у вигляді строго ранжування, унікального для кожної k -ї предметної області, де максимально можливе значення – це кількість таких предметних областей N .

Припускається, що раз на добу будуть опрацьовуватися нові заявки учнів про допомогу, що з'явилися за останню добу, а також ті, що були взяті раніше, але не підтверджені учнем та ментором. Учні, що не мають заявок, до розгляду, не братимуться до уваги.

Маємо, що i -й учень, $i = 1, \dots, S$, може створити певну кількість запитів на навчання із певної предметної області.

Позначимо v_{ij} – коефіцієнт схожості між i -м учнем та j -м ментором, значення якого знаходяться в межах $0 \leq v_{ij} \leq 1$, $i = 1, \dots, S$,

$j = 1, \dots, M$. Тут 1 означає, що ментор та учень максимально схожі за вибором предметної області, 0 – навпаки.

Пронумерувавши предметні області j -го ментора, отримаємо пріоритет кожної предметної області, що його стосується. Відобразимо ці пріоритети на відрізок $[0, 1]$ так, що 1 означатиме, що дана предметна область являється найвищою за пріоритетом для ментора, а 0 – найнижчою.

Нехай w_{kj} – коефіцієнт важливості k -ї предметної області для j -ого ментора, який буде задовольняти нерівності:

$$0 \leq w_{kj} \leq 1, \quad k = 1, \dots, N, \quad j = 1, \dots, M.$$

Використовуючи введені поняття, сформуємо розклад ментора. Подамо обмеження на розклад ментора у вигляді матриці Y розмірності $T \times M$, елемент y_{tj} якої визначається так:

$$y_{tj} = \begin{cases} 1, & \text{якщо в } t\text{-й часовий інтервал ментор зайнятий,} \\ 0, & \text{у іншому разі} \end{cases}$$

де t – часовий інтервал, $t = 1, \dots, T$,

j – номер ментора, $j = 1, \dots, M$.

Для подання розв'язку введемо матрицю X , розмірності $M \times T$, елемент x_{tj} якої визначається так:

$$x_{tj} = \begin{cases} r, & \text{якщо в } t\text{-й часовий інтервал заявка призначена } j\text{-му ментору, } r \in P_r, \\ -1, & \text{якщо ментор зайнятий в } t\text{-й часовий інтервал,} \\ 0, & \text{в іншому разі} \end{cases}$$

де t – часовий інтервал, $t = 1, \dots, T$,

j – ментор, підібраний на заявку певного учня в даному варіанті розкладу, $j = 1, \dots, M$.

Для матриці X виконується наступні обмеження.

1. Одна заявка повинна бути назначена лише один раз одному ментору:

$$\exists t_1, t_2 \in \{1, \dots, T\} : \forall j_1, j_2 \in \{1, \dots, M\}, x_{t_1 j_1} \neq 0 : x_{t_1 j_1} = x_{t_2 j_2} .$$

2. Кожна заявка повинна бути оброблена:

$$\forall r \in P_r \Rightarrow \exists (t \in \{1, \dots, T\}, j \in \{1, \dots, M\}) : x_{tj} = r .$$

3. Заявка може бути назначена ментору лише у тому випадку, якщо ментор вільний у період доби, вказаний у заявці про допомогу:

$$\forall y_{tj} \in V, y_{tj} = 0 \Rightarrow \exists (t \in \{1, \dots, T\}, j \in \{1, \dots, M\}) : x_{tj} = r .$$

Нехай маємо матрицю заявок C розмірності $2 \times R$. Перший рядок цієї матриці буде містити номери учнів, що подали відповідну заявку. Другий рядок цієї матриці буде визначати предметну область, до якої належить ця заявка. Для спрощення припускаємо, що одна заявка може мати відповідати лише одній предметній області. У подальших дослідженнях ця умова може бути замінена чи розвинена.

Таким чином маємо наступні елементи матриці C : у стовпчику j елемент c_{1j} визначає номер учня, який створив заявку з номером j , а c_{2j} – до якої відноситься ця заявка.

Тоді цільову функцію задачі можна буде подати так:

$$F(X) = \sum_{t=1}^T \sum_{j=1}^M (\alpha (1 - v_{c_{1,x_{tj}} j}) + (1 - \alpha) (1 - w_{c_{2,x_{tj}} j})) \rightarrow \min ,$$

де α – ваговий коефіцієнт, який балансує фактори схожості і важливості,

$v_{c_{1,x_{tj}} j}$ – коефіцієнт схожості між учнем $c_{1,x_{tj}}$ та j -м ментором,

$w_{c_{2,x_{tj}} j}$ – коефіцієнт важливості $c_{2,x_{tj}}$ -ї предметної області для j -го ментора.

Отже, для знаходження розв'язку поставленої задачі потрібно знайти найкращий варіант розкладу.

Формалізуємо алгоритму знаходження одного варіанту розкладу у вигляді псевдокоду:

$F := 0$

for $j = 1, \dots, M$ do { j – ментор }

for $t = 1, \dots, T$ do { x_{tj} – номер заявки }

Беремо схожість ментора j з учнем $c_{1,x_{tj}}$:

$$v_{c_{1,x_{tj}}j}.$$

Беремо важливість для ментора j предметної області $c_{2,x_{tj}}$:

$$w_{c_{2,x_{tj}}j}.$$

$$F := F + \alpha \left(1 - v_{c_{1,x_{tj}}j} \right) + (1 - \alpha) \left(1 - w_{c_{2,x_{tj}}j} \right)$$

endfor t

endfor j

Одним з найвідоміших і найпростіших, проте потужних, евристичних алгоритмів є локальний пошук. Спробуємо застосувати цей алгоритм для пошуку найкращого варіанту розкладу. Точкою в нашому випадку буде варіант розкладу, а околom точки – модифікація цього розкладу.

Специфіка поставленої задачі полягає у тому, що на точку (варіант розкладу) накладено багато обмежень, тому модифікація варіанту розкладу повинна відбуватися за певними правилами.

Я вивела два правила знаходження околу точки.

Допустимим ментором вважаємо такого, який має таку предметну область, яка вказана в заявці, та який вільний в проміжку ± 1 година від вказаного часу в заявці.

Правило 1.

1. Взяти заявку випадковим чином.
2. Визначити назначеного ментора на цю заявку.
3. Визначити допустимих менторів для заявки.
4. Назначити заявці одного з допустимих менторів, відмінного від того, що зараз назначений у розкладі.

Правило 2.

1. Взяти 2 заявки випадковим чином.
2. Визначити для 1-ї заявки, чи є допустимим ментор 2-ї заявки.

Якщо так – крок 3. Якщо ні – вибрати випадковим чином ще одну заявку і повторити поточний крок.

3. Поміняти заявки місцями.

Оскільки незначна зміна в розкладі може призвести до різкого збільшення значення функції, вважаю недоцільним брати за критерій завершення умову відсутності поліпшуючої точки в околі поточного розв'язку, що забезпечило б швидке завершення роботи алгоритму через високу ймовірність виконання цієї умови, використовуючи будь-яке з двох правил.

Тому критерієм завершення оберемо досягнення певної кількості ітерацій.

Формалізуємо алгоритм локального пошуку, використовуючи правило 1.

Крок 1. Обирати початкову заявку випадковим чином.

Крок 2. Для обраної заявки знайти всіх допустимих менторів.

Крок 3. Випадковим чином обрати 1 ментора з допустимих і назначити йому заявку на час вказаний у заявці, якщо цей час зайнятий, то з зсувом ± 1 годину.

Крок 4. Позначити заявку як оброблену.

Крок 5. Перейти до наступної необробленої заявки, вибраної випадковим чином. Повторити крок 2-4 для цієї заявки. Виконувати, поки всі заявки не будуть назначені менторам.

Крок 6. Порахувати цільову функцію (ЦФ) для складеного варіанту розкладу.

Крок 7. Обрати випадковим чином заявку, для якої допустимих менторів більше ніж 1.

Крок 8. Обрану заявку назначити допустимому для неї ментору, вибраного випадковим чином із наявних, при цьому для поточного назначеного ментора звільнити час у розкладі.

Крок 9. Порахувати ЦФ для створеного варіанту розкладу.

Крок 10. Порівняти ЦФ для розкладу, створеного на кроці 6 і на кроці 9. Якщо на кроці 9 значення ЦФ менше, то записати варіант розкладу і значення ЦФ, як поточні найкращі значення.

Крок 11. Повторювати кроки 8-10, поки не переберемо всіх допустимих менторів заявки.

Крок 12. Повторювати кроки 7-11, поки не настане умова завершення алгоритму.

Результати досліджень

Запропонований алгоритм реалізовано на мові C#. Для оцінювання його ефективності проведено обчислювальний експеримент на ПК з наступними характеристиками: процесор: Intel(R) Core(TM) i5-7200U (2.5 -3.1 ГГц), об'єм оперативної пам'яті: 8 ГБ.

Результати експерименту наведені на рисунку 1.

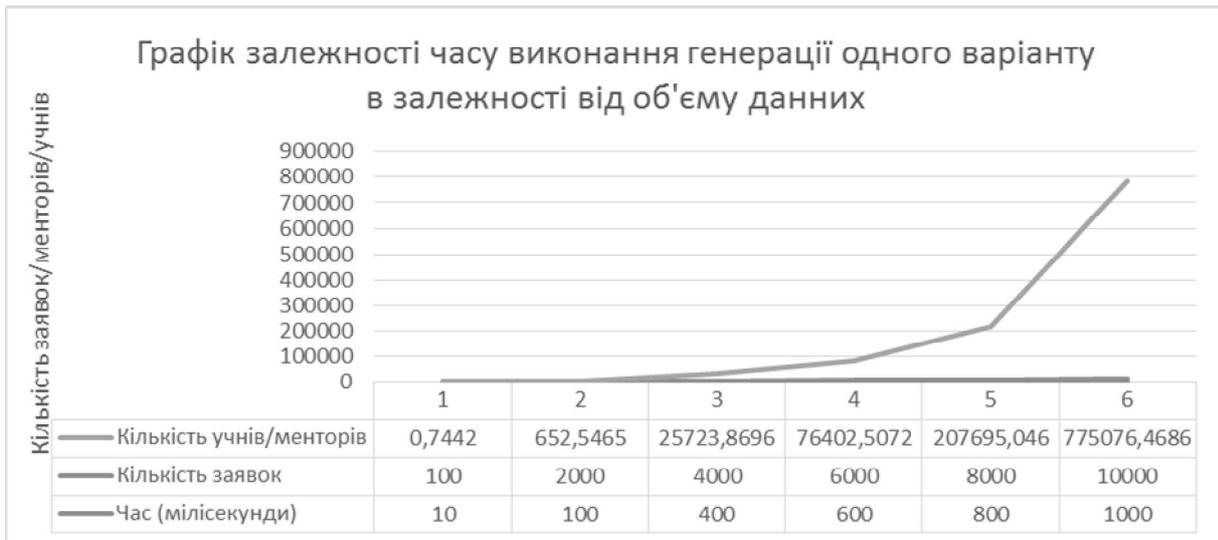


Рисунок 1. Графік залежності часу виконання генерації одного варіанту в залежності від об'єму даних

З проведених експериментів видно, що час виконання значно зростає зі збільшенням розмірності даних.

Для порівняння ефективності цього алгоритму в подальшому необхідно реалізувати його модифікацію (правило 2) і порівняти результати експериментів.

Висновки

Зроблений постановка проблеми та її огляд. Показана актуальність підняття даної постановки задачі. Зроблена математична постановка задачі та на основі неї прийнято рішення застосувати алгоритм локального пошуку як відправну точку і дослідженні даного типу задач. Виведено два правила, якими можливо користуватися, для реалізації евристичних алгоритмів для даної постановки задач. Зроблені заміри часу виконання алгоритму ЛП відповідно до збільшення розмірності даних.

Подальші дослідження варто спрямувати на можливу модифікацію алгоритму локального пошуку використовуючи правило 2, а також застосування мурашиного алгоритму як альтернативи алгоритму локального пошуку.

Література:

1. Зак Ю.А. (2012). *Прикладные задачи теории расписаний и маршрутизации перевозок*. Книжный дом «ЛИБРОКОМ», 394.
2. Сперкач М.О. (2016). *Інформаційна технологія оперативно-календарного планування дрібносерійного виробництва за концепцією «точно в строк»*. 39.
3. Бойко О.М. (2006). Еволюційна теорія розв'язування задачі складання розкладів навчальних занять. *Штучний інтелект*, 3, 341-348.
4. Бевз С.В. (2008). Розробка автоматизованої системи формування розкладу магістратури. *Наукові праці ВНТУ*, 4.
5. Безгинов А.Н. (2011). Комплекс алгоритмов построения расписания вуза. Ч.1: Система оценки качества расписания на основе нечетких множеств, алгоритм поиска оптимального расписания. *Вестник Балтийского федерального университета им. И. Канта*, 5, 127-135.
6. Бабкина Т.С. (2008). Задача составления расписаний: решение на основе многоагентного подхода. *Бизнес-информатика*, 1, 23-28.
7. Шостак И.В. (2012). Автоматизация процесса составления расписания занятий на основе тензорного исчисления в учебном комплексе. *Авиационно-космическая техника и технология*, 9, 263-266.
8. Ризун Н.О. (2012). Применение методов декомпозиции при решении многокритериальной задачи автоматизации составления расписания учебных занятий в ВУЗе. *Східно-Європейський журнал передових технологій*, 2, 59-70.
9. Кабальнов Ю.С. (2006). Композиционный генетический алгоритм составления расписания учебных занятий. *Вестник Уфимского*

государственного авиационного технического университета, 2, 99-107.

10. Конькова И.С. (2012). Генетические алгоритмы в решении задачи составления расписания занятий в вузе. *Проблемы информатики в образовании, управлении, экономике и технике : Сб. Стетей XII Междунар. научно-техн. конф*, 26-29.

11. Лебединская Е.Н. (2003). Формализация задачи построения рационального расписания занятий в системе АСУ ВУЗ. *Наука и техника транспорта*, 1, 67-73.

12. Попов Г.А. (2006). Формализация задачи составления расписания в высшем учебном заведении. *Вестник АЕТУ*, 1, 118-139.

13. Зінченко Л.В. (2019). До задач теорії розкладів у сфері освіти. *Polish journal of science*, 16, 37.

14. Talbi E-G. (2009). Metaheuristics: from design to implementation. *John Wiley & Sons*, 74.

References:

1. Zak Yu.A. (2012). *Prikladnyye zadachi teorii raspisaniy i marshrutizatsii perevozok*, Knizhnyy dom "LIBROKOM", 394. [in Russian].

2. Sperkach M.O. (2016). *Informatsiyana tekhnolohiya operatyvno-kalendarnoho planuvannya dribnoseriynoho vyrobnytstva za kontseptsiyeyu «tochno v strok»*. 39. [in Ukrainian].

3. Boyko O.M. (2006). Evolyutsiyana teoriya rozv'yazuvannya zadachi skladannya rozkladiv navchal'nykh zanyat'. *Shtuchnyy intelekt*, no. 3, 341-348. [in Ukrainian].

4. Bevz S.V. (2008). Rozrobka avtomatizovanoї sistemi formuvannya rozkladu magistraturi, *Naukovi pratsi VNTU*, no. 4. [in Russian].

5. Bezginov A.N. (2011). Kompleks algoritmov postroyeniya raspisaniya vuza. Ch.1: Sistema otsenki kachestva raspisaniya na osnove nechetkikh

mnozhestv. algoritm poiska optimalnogo raspisaniya. *Baltiyskogo federalnogo universiteta im. I. Kanta, no. 5*, 127-135. [in Russian].

6. Babkina T.S. (2008). Zadacha sostavleniya raspisaniy: resheniye na osnove mnogoagentnogo podkhoda. *Biznes-informatika, no. 1*, 23-28. [in Russian].

7. Shostak I.V. (2012). Avtomatizatsiya protsessa sostavleniya raspisaniya zanyatiy na osnove tenzornogo ischisleniya v uchebnom komplekse. *Aviatsionno-kosmicheskaya tekhnika i tekhnologiya, no. 9*, 263-266. [in Russian].

8. Rizun N.O. (2012). Primeneniye metodov dekompozitsii pri reshenii mnogokriterialnoy zadachi avtomatizatsii sostavleniya raspisaniya uchebnykh zanyatiy v VUZe, *Ckhdno-Evropeyskiy zhurnas peredovikh tekhnologiy, no. 2*, 59-70. [in Russian].

9. Kabalnov Yu.S. (2006). Kompozitsionnyy geneticheskiy algoritm sostavleniya raspisaniya uchebnykh zanyatiy. *Vestnik Ufimskogo gosudarstvennogo aviatsionnogo tekhnicheskogo universiteta. no. 2*, 99-107. [in Russian].

10. Konkova I.S. (2012). Geneticheskiye algoritmy v reshenii zadachi sostavleniya raspisaniya zanyatiy v vuze. *Problemy informatiki v obrazovanii. upravlenii. ekonomike i tekhnike : Sb. Stetey XII Mezhdunar. nauchno-tekhn. Konf*, 26-29. [in Russian].

11. Lebedinskaya E.N. (2003). Formalizatsiya zadachi postroyeniya ratsionalnogo raspisaniya zanyatiy v sisteme ASU VUZ. *Nauka i tekhnika transporta, no. 1*, 67-73. [in Russian].

12. Popov G.A. (2006). Formalizatsiya zadachi sostavleniya raspisaniya v vysshem uchebnov zavedenii. *Vestnik AyeTU*. [in Russian].

13. Zinchenko L.V. (2019). Do zadach teorii rozkladiv u sferi osvity. *Polish journal of science, no. 16*, 37. [in Ukrainian].

14. Talbi E-G. (2009). *Metaheuristics: from design to implementation*, John Wiley & Sons, no. 74. [in Ukrainian].

Citation: L. V. Zinchenko. (2019). About the problems of the schedule theory in education. *Innovative Solutions in Modern Science*. 6(33). doi: 10.26886/2414-634X.6(33)2019.4

Copyright: © 2019 L. V. Zinchenko. This is an openaccess article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.