

ТОМАШЕВСЬКИЙ В.М.,
НОВИКОВ Ю.Л.,
КАМІНСЬКА П.А.

СКЛАДАННЯ РОЗКЛАДІВ ЗАНЯТЬ У ДИСТАНЦІЙНИХ СИСТЕМАХ НАВЧАННЯ

У статті розглянуто особливості побудови розкладів для дистанційної форми освіти з огляду на відмінності процесу класичного та дистанційного навчання. Подано математичну постановку загальної задачі побудови розкладу. Розглянуто методи та підходи до вирішення цієї задачі. Обґрунтовано вибір методу генетичного алгоритму та розроблено алгоритм побудови розкладу для дистанційного навчання, реалізований у якості підсистеми для системи дистанційного навчання «Віртуальний Університет».

The features of construction schedules for the distance form of education according to differences of classical and distance learning are described in the article. The mathematical formulation of the general problem of constructing the schedule is represented. Methods and approaches for solving this problem are considered. The choice of the method of genetic algorithm is grounded and the algorithm for constructing a schedule for distance learning, implemented as a subsystem for the distance learning system "Virtual University", is developed.

Вступ

Якість підготовки спеціалістів у вузах і особливо ефективність використання науково-педагогічного потенціалу залежать в значній мірі від рівня організації навчального процесу, який в значній мірі залежить від розкладу занять, що регламентує трудовий режим і впливає на творчу віддачу викладачів. Тому розклад занять можна розглядати як фактор оптимізації використання обмежених трудових ресурсів – викладацького складу.

Метою даного дослідження є автоматизація складання розкладу занять для системи дистанційного навчання „Віртуальний університет” з врахуванням завантаженості викладацького складу. Якість навчання, його економічна ефективність, зручність навчання студентів і роботи професорсько-викладацького складу залежить від вдало складання розкладу.

Потреба в розробці механізму складання розкладів виникла в зв'язку з необхідністю спрощення й прискорення розв'язання складної задачі – складання розкладу занять для системи дистанційного навчання „Віртуальний університет”. Задача складання розкладу є NP-складною, тобто такою, для якої досі невідомий алгоритм розв'язку, час виконання якого виражався б поліноміальною функцією. Складність задачі виявляється також у тому, що до розкладу занять, як правило, залучена велика кількість людей,

причому кожен зі своїми інтересами та вимогами, які часто є суперечливі.

Тому для розв'язку таких задач застосовуються наближені методи, які дозволяють скласти субоптимальний розклад занять.

1. Аналіз умов і предметного середовища

Складаючи розклад занять, необхідно врахувати такі умови:

- забезпечення найбільш рівномірного навантаження на учасників навчального процесу;
- забезпечення гнучкості графіку з навантаженням та з розміщенням занять з урахуванням базових вимог,
- виключення різного роду „накладок” в розкладі,
- максимальна відповідність вихідним даним з урахуванням обмежень.

Обмеженнями в даному випадку можуть бути наступні:

- 1) жорсткі обмеження:
 - режим роботи викладачів у системі дистанційного навчання;
 - виключення святкових і неробочих днів.
- 2) відносно гнучкі обмеження:
 - план-графік тестових контролів;
 - план-графік інших занять;
 - періоди підвищеної зайнятості у деяких викладачів або груп студентів.

Для складання розкладу важливою вхідною інформацією є організаційна структура підрозділів навчального закладу. Наприклад, візьмемо типову структуру університету, що включає:

- факультети (потрібна наявність інформації про назву, декана);
- кафедри (потрібна наявність інформації про назву, завідуючого);
- спеціальності (потрібна наявність інформації про назви);
- групи (потрібна наявність інформації про назви, дату утворення, поточний семестр навчання і курс);
- дисципліни (потрібна наявність інформації про викладача, максимальний бал, номер навчального семестру);
- викладацький склад;
- студентський склад.

Для складання реального розкладу занять для дистанційного способу навчання необхідно мати наступну інформацію.

1. Кількість годин роботи викладача у семестрі, що відведені на певну дисципліну. Цей час включає:

- відеореєстрації (лекції, віртуальні заняття, тренінги, конференції, презентації, доклади, он-лайн семінари)
- консультаційні заняття (пояснення щодо пройденого матеріалу дисципліни, передтестові консультації, відповіді на запитання);
- практичні (лабораторні) заняття (віртуальна присутність викладача як спостерігача і координатора та консультування через інтерактивне спілкування);
- перевірку тестових (контрольних та екзаменаційних) робіт.

2. Орієнтовний обсяг академічного навантаження для студента (кількість годин у семестрі) по певній дисципліні. Цей час включає самостійну роботу з:

- вивчення лекційного матеріалу;
- виконання практичних, лабораторних і контрольних робіт;
- тестування (здавання екзаменів і заліків).

3. Розподілення видів освітньої діяльності за періодами:

- послідовність вивчення матеріалів і перевірки знань по них згідно графіку навчання;
- час на опанування тієї або іншої дисципліни (підготовка до тестування, обробка результатів тестування, робота над помилками, тощо).

2. Математична постановка задачі складання розкладу

Задача складання розкладу є однією з найбільш розповсюджених задач, які вирішує кожна людина (свідомо або ні) практично кожен день. У загальній постановці вона є процесом упорядкування деякою скінченою множиною подій в часі за умов ресурсних та інших обмежень. Мета полягає в тому, щоб при заданих властивостях для кожної події (завдання) і обмеженнях на ресурси знайти ефективний алгоритм впорядкування завдань, який оптимізував би певний критерій ефективності. За критерій оцінки розкладу звичайно беруть середні, середні зважені або максимальні значення:

- моментів часу закінчення робіт;
- тривалостей виконання робіт;
- часового зміщення;
- запізнювання;
- випередження;
- тривалості очікування.

Математичні моделі для цих задач є дискретними, детермінованими тому, що вся інформація, на основі якої приймаються рішення щодо упорядкування, відома заздалегідь. Більшість задач складання розкладу є багатокритеріальними, тому що до навчального процесу залучена велика кількість людей, кожен з яких має свої інтереси та вимоги, задоволення яких часто призводить до суперечливих ситуацій.

Задача складання розкладу може бути представлена як задача цілочисельного математичного програмування. При даному підході інтереси суб'єктів навчального процесу враховуються у вигляді обмежень або редукованих критеріїв оптимальності. Наведемо математичну модель задачі складання розкладу навчальних занять студентів дистанційної форми навчання. Введемо наступні позначення.

1. У вузі є N учбових груп, що навчаються дистанційно, які об'єднані в R потоків, r – номер потоку ($r = \overline{1, R}$); k_r – номер учбової групи на потоці r , ($k_r = \overline{1, G_r}$), G_r – кількість груп на потоці r .

2. Заняття проводяться в робочі дні. t – номер робочого дня тижня, T – загальна кількість робочих днів учбового закладу ($t = \overline{1, T}$). Для кожної групи k_r визначається множина номерів

робочих днів цієї групи T_{k_r} , яка є підмножиною всіх робочих днів учбового закладу.

На відміну від класичного аудиторного навчання, дистанційна форма навчання дозволяє проводити заняття у найбільш зручній для учасників навчального процесу час, тому класичним розподілом навчального часу, прив'язаним до часових інтервалів, які прийнято називати „парами”, можна знехтувати. Отже, кожен учбовий день розбивається на години, що починаються з J_s і закінчуються годиною J_f . Значення J_s і J_f можуть залежати від робочого часу викладачів (деяким викладачам зручніше проводити заняття у першій або у другій половині дня); j – конкретна учбова година ($j = \overline{J_s, J_f}$).

3. На основі учбового плану для кожного потоку складається список із S_r лекційних занять, з номером конкретної дисципліни в списку $s_r = \overline{1, S_r}$. Для кожної групи k_r складається список із Q_{k_r} запланованих консультаційних занять, q_{k_r} – номер дисципліни в списку ($q_{k_r} = \overline{1, Q_{k_r}}$). Загальний список занять W_{k_r} для кожної групи буде складатися із усіх дисциплін, що присутні в даних списках ($W_{k_r} = S_r + Q_{k_r}$). Крім того, якщо по деякій дисципліні протягом тижня проводиться більше одного заняття, то вона вказується в списку лекцій або консультаційних занять стільки раз, скільки це передбачено навчальним планом.

4. Нехай p – номер (ім'я) викладача, який проводять лекційні або консультаційні заняття, P – множина всіх викладачів ($P = \overline{1, P}$).

Для кожного викладача p введемо наступні булеві змінні:

$$\delta_{s_r}^p = \begin{cases} 1, & \text{якщо на потоці } r \text{ лекцію } s_r \text{ читає викладач } p, \\ 0, & \text{в протилежному випадку,} \end{cases}$$

$$\Delta_{r k_r q_{k_r}}^p = \begin{cases} 1, & \text{якщо в групі } k_r \text{ заняття } q_{k_r} \text{ проводить викладач } p, \\ 0, & \text{в протилежному випадку,} \end{cases}$$

Оскільки учбове навантаження викладачів планується до складання розкладу занять, то

ці величини вважаються заданими. Також вражається заданою для кожного викладача p його навантаження – N_p годин за тиждень ($p = \overline{1, P}$).

Дистанційне навчання не прив'язане до аудиторії, тому це дозволяє не обмеження на аудиторії, що є дуже критичним класичному навчанні. Це дозволить скласти розклад, який задовольняє потребам і побажанням учасників навчального процесу.

При даному представленні початкових даних задача складання розкладу занять полягає у визначенні для кожного заняття дня тижня і пари в ньому з урахуванням накладених на розклад обмежень. Вводяться наступні групи булевих змінних, які необхідно знайти.

$$\eta_j^{k_r} = \begin{cases} 1, & \text{якщо в день } t \text{ група } k_r \text{ закінчує заняття о } j\text{-й годині} \\ 0, & \text{в протилежному випадку,} \end{cases}$$

$$\xi_j^{k_r} = \begin{cases} 1, & \text{якщо в день } t \text{ група } k_r \text{ починає заняття о } j\text{-й годині} \\ 0, & \text{в протилежному випадку,} \end{cases}$$

$$y_{rtj}^{s_r} = \begin{cases} 1, & \text{якщо на потоці } r \text{ в день } t \text{ у час } j \text{ читається лекція } s_r \\ 0, & \text{в протилежному випадку,} \end{cases}$$

$$x_{rk_r,tj}^{q_{k_r}} = \begin{cases} 1, & \text{якщо в групі } k_r \text{ в день } t \text{ у час } j \text{ проводиться консультація } q_{k_r} \\ 0, & \text{в протилежному випадку.} \end{cases}$$

Обмеження «кожного дня на кожній парі для кожної групи може проводитися не більше одного заняття» має наступний вигляд:

$$\sum_{q_{k_r}=1}^{Q_{k_r}} x_{rk_r,tj}^{q_{k_r}} + \sum_{s_r=1}^{S_r} y_{rtj}^{s_r} \leq 1,$$

де $r = \overline{1, R}$, $k_r = \overline{1, G_r}$, $t \in T_{k_r}$, $j = \overline{J_s, J_f}$.

Інша обов'язкова умова складання розкладу для учбових закладів «для кожної учбової групи повинні виконуватися всі передбачені планом види робіт протягом тижня» має наступний вигляд:

$$\sum_{t \in T_{k_r}} \sum_{j=1}^J \left(\sum_{q_{k_r}=1}^{Q_{k_r}} x_{rk_r,tj}^{q_{k_r}} + \sum_{s_r=1}^{S_r} y_{rtj}^{s_r} \right) = W_{k_r},$$

де $r = \overline{1, R}$, $k_r = \overline{1, G_r}$.

Ще одним важливим обмеженням є відсутність «накладок», тобто розклад дистанційних занять у кожного викладача не повинен перетинатися з розкладом його аудиторних занять.

У конкретних задачах список обмежень може бути продовжений.

Після складання обмежень, необхідно обрати критерій оптимальності розкладу. Це складне питання через принципову багатокритеріаль-

ність задач складання розкладу. Зокрема, можна розглядати які критерій максимальне сукуне ущільнення навантаження викладачів. У такому випадку критерій має вигляд:

$$\sum_{t=1}^T \sum_{p=1}^P \xi_p z_t^p \rightarrow \max ,$$

де ξ_p – ваговий коефіцієнт, який визначається статусом викладача; z_t^p – булева змінна, яка приймає значення 0, якщо даний викладач p має заняття в день t , і 1 – в протилежному випадку.

Дана модель відображає основні фактори, які необхідно враховувати при складанні розкладу і відноситься до класу задач лінійного булевого програмування. Однак, розв'язання таких задач навіть відносно малої розмірності (для невеликих навчальних закладів) звичайно вимагає значних витрат часу. Побудована таким чином модель, при дотриманні основних вимог до розкладу, повинна включати тисячі обмежень і десятки тисяч змінних. Стає зрозуміло, що не лише цілочисельна, а й проста задача лінійного програмування подібної розмірності не має перспектив на оптимізацію точними методами. Цей факт цілком пояснює, чому при зіставленні розкладу занять на практиці користуються зазвичай евристичними методами [1].

Задача складання розкладу в дистанційному навчанні є NP -повною. Для розв'язання NP -повних задач досі невідомо алгоритмів, час роботи яких виражався б поліноміальною функцією, але й не доведено, що для якоїсь із них такого алгоритму не існує. Тому розв'язувати дану задачу методом повного перебору немає сенсу і слід використовувати наближені методи:

- імітації відпалу;
- розфарбування графу;
- імітаційного моделювання;
- логічного програмування з обмеженнями;
- генетичного алгоритму.

Розглянемо подані методи

3. Метод імітації відпалу

Ідея алгоритму імітації відпалу [2] запозичена із досліджень поведінки атомів металу в процесі його відпалу. В металі, нагрітому до температури, яка перевищує точку його плавлення, атоми знаходяться в безладному русі. При цьому, як і в усіх фізичних системах, вони прагнуть до стану мінімуму енергії (єдиного кристалу), але при високих температурах енергія атомного руху перешкоджає цьому. В процесі поступового охолодження металу виникають все більш низькоенергетичні стани, поки не буде досягнуто найнижчий з можливих станів – глобальний мінімум.

Переходячи до математичної моделі, визначимо, що в задачі складання розкладу може грати роль енергії, і стан якого об'єкту така «енергія» може характеризувати. Один із можливих варіантів – це розгляд такої енергії у вигляді цільової функції, яка основана на штрафних, що додаються до поточного розкладу за кожний невдалий розклад, а коректний (хоча і невідомий) розклад – розглядати як низькоенергетичний стан.

Загальну ідею алгоритму імітації відпалу для задачі складання розкладу можна представити наступною схемою.

1. На першій ітерації генерується деякий коректний початковий розклад X^0 , який вважається поточним розв'язком задачі ($X = X^0$).

2. Задається початкове, високе значення температури T^0 і операція мутації розкладу. Мутація може бути реалізована наступними діями:

- зміна часу проведення занять;
- обмін місцями в розкладі двох занять.

3. На основі введених операцій мутації та поточного розв'язку генерується новий коректний розклад X' , який дещо відрізняється від попереднього.

4. Визначається зміна цільової функції $\Delta f = f(X) - f(X')$:

– якщо $\Delta f < 0$ (розв'язок не погіршився), то новий варіант розкладу стає поточним ($X = X'$);

– якщо $\Delta f > 0$ (розв'язок погіршився), то новий розклад стає поточним лише з ймовірністю $p = e^{-\Delta f / T}$. Відповідно, з ймовірністю $(1-p)$ попередній розклад зберігається як поточ-

ний.

Таким чином, допускається збільшення цільової функції, але ймовірність цього експоненціально зменшується з ростом Δf . Такий підхід дозволяє долати локальні екстремуми.

5. Виконується функція зміни поточної температури. Температура, i , відповідно, ймовірність прийняти поточного розкладу з більшим значенням цільової функції зменшується на кожній ітерації. Величину зменшення температури можна зв'язати з числом виконаних ітерацій, або з величиною зміни цільової функції. При високих температурах спостерігається стрибкоподібна зміна цільової функції. Поступово зі зменшенням температури її значення прагне до мінімуму.

6. Якщо не виконується критерій зупину, то перейти до п. 3. Для критерію зупину можуть бути прийняті наступні умови:

- виконання заданого числа ітерацій;
- виконання заданого числа ітерацій без покращення значення цільової функції на задане значення.

Для підвищення ефективності алгоритму можна реалізувати збереження m кращих варіантів, а також n останніх згенерованих розкладів. Це дозволить запобігти зацикленню процесу складання розкладу.

Точність розв'язку, отриманого за даним алгоритмом, можна збільшити за рахунок більш повільної зміни температури. В такому випадку алгоритм більш детально досліджує простір пошуку, але час його роботи суттєво збільшується. Тому функцію температури доводиться підбирати для кожної конкретної задачі індивідуально. Незважаючи на зовнішню простоту, такий підхід може виявитися цілком ефективним для складання невеликих розкладів.

4. Метод розфарбування графу

Задачу складання розкладу можна розглядати як задачу розфарбування графу. Задачею розфарбування графу називають пошук хроматичного числа графу або, іншими словами, пошук мінімального числа кольорів, необхідних для розфарбування вершин деякого графу таким чином, щоб для кожної пари сусідніх вершин використовувалися різні кольори [4]. Сама задача пошуку

хроматичного числа є NP -повною, для розв'язання якої в більшості випадків використовують різного роду жадібні алгоритми.

Для постановки задачі складання розкладу як задачі розфарбування графу будується граф, в якому кожна вершина представляє собою заплановане навчальним планом заняття. У випадку, коли між деякими двома вершинами можливі конфлікти (наприклад, обидва заняття проводяться в однаковий час з одним викладачем), то вони з'єднуються ребром. Це еквівалентно забороні одночасного проведення цих занять. Тоді задача складання розкладу представляється як мінімізація числа кольорів, необхідних для розфарбування графа. Кожен колір відповідає одному періоду занять.

Застосування цього підходу для розв'язку реальних задач, очевидно, малоефективне. Але задача розфарбування графа при складанні розкладу може бути корисною у випадку її комбінації з іншими алгоритмами.

5. Метод імітаційного моделювання

Враховуючи NP -повноту задачі, для її розв'язання можна застосувати імітацію дій диспетчера при складанні розкладу. В цьому випадку алгоритм оперує безпосередньо розкладом і списком занять, які необхідно включити до розкладу (навчальним планом). Процес складання розкладу розпочинається з порожнього розкладу, коли всі заняття знаходяться в списку неврахованих занять. Далі алгоритм переходить від одного незавершеного розкладу до другого, намагаючись найліпшим чином розставити усі заняття, які внесені до списку. Процес триває доти, поки не буде сформований повний розклад або виконається фіксоване число ітерацій.

Наведемо схему цього алгоритму більш детально. Даний алгоритм складається з трьох основних дій [4].

1. Вибирається ще не включене в розклад заняття. Вибір робиться на підставі аналізу «вузьких місць». В даному випадку вузькими місцями є найбільш дефіцитні ресурси: викладачі, студенти та час. В першу чергу зіставляється розклад для найбільш дефіцитних ресурсів. Це можуть бути заняття, що проводяться викладачами, які ставлять жорсткі умови щодо часу їх проведення та інше.

2. Для обраного заняття визначаються всі можливі варіанти його розміщення в розкладі, які задовольняють усі жорсткі обмеження. Далі

кожна така позиція оцінюється за допомогою спеціальної евристичної цільової функції і заняття розміщується в найкращу із можливих позицій.

3. Якщо у випадку п. 2 в розкладі виникла конфліктна ситуація, то «конфліктуючі» заняття видаляють з розкладу і вносять знову у список неврахованих занять.

При реалізації алгоритму, який оснований на принципах імітаційного моделювання, особлива увага приділяється розробці евристичних правил вибору наступного заняття зі списку, визначенні найкращої для нього позиції в розкладі та оцінці отриманого розкладу з виконанням обов'язкових обмежень.

До позитивних сторін такого підходу можна віднести можливість детального врахування специфіки задачі, що розв'язується, у випадку складання розкладу для конкретного вузу. Однак, у такому випадку сильно обмежується можливість застосування розробленої системи в інших учбових закладах. Крім цього, очевидно, доведеться вносити суттєві зміни до алгоритму при незначних внутрішніх змінах у вузі. Оскільки алгоритм оснований на діях, які виконує диспетчер при складанні розкладу, то існує можливість організувати діяльний діалог між користувачем і системою при пошуку найкращого розкладу. Хоча це справедливо лише для відносно невеликих задач, так як в протилежному випадку значна залежність алгоритму від дій користувача може зробити такий діалог малоефективним.

6. Метод логічного програмування з обмеженнями

Складання розкладу можна представити як задачу задоволення обмежень. Для розв'язку таких задач розроблено достатньо алгоритмів, починаючи з класичного методу Гаусса і закінчуючи складними методами, які застосовуються в системах доведення теорем і в системах символічних обчислень. Виник навіть цілий напрямок в програмуванні – програмування з обмеженнями.

Програмування з обмеженнями тісно пов'язано з традиційним логічним програмуванням, в рамках якого воно сформувалося. Більшість систем програмування в обмежен

ням представляють собою звичайний інтерпретатор мови Пролог з вбудованим механізмом для розв'язку певного класу задач задоволення обмеженням. Програмування в таких системах називають логічним програмуванням з обмеженнями (Constraint Logic Programming або CLP).

Ідея розв'язання задач полягає в тому, що програміст визначає деяку множину змінних x_1, \dots, x_n та область їх значень X_1, \dots, X_n , описує додаткові обмеження, яким повинні задовольняти змінні, а система повинна знайти такі значення цих змінних, які б задовольняли одночасно усім заданим обмеженням.

Наведемо невеликий приклад вищезгаданого методу. Розглянемо вуз, в якому працює M викладачів, в учбовому плані зафіксовано N дисциплін. Визначимо множину $P = 1, 2, \dots, D$, елементами якої є всі періоди навчання (години) у вузі протягом робочого тижня, а D – число всіх періодів навчання протягом тижня. Нехай y_{ij} - період навчання, в якому i -й викладач веде предмет j , $y_{ij} \in P$. Тоді обмеження „кожен викладач в кожен момент часу може вести не більше одного заняття” має вигляд:

$$y_{ij} \neq y_{ij'}, 1 \leq i \leq M, 1 \leq j \leq N, 1 \leq j' \leq N, \\ j \neq j'.$$

Аналогічним чином складаються інші обмеження.

Результатом роботи алгоритму є множина значень кожної змінної, що не суперечать вказаним обмеженням. При цьому область визначення змінних, що входять до строгих обмежень, може суттєво зменшитися або навіть матиме єдине значення. Основна перевага, яку можна отримати при використанні CLP, – скорочення простору пошуку. Це досягається не шляхом оцінки кожного варіанту розкладу, а за рахунок того, що система сама виключає з розгляду неперспективні варіанти.

7. Метод генетичного алгоритму

Усі розглянуті вище методи в своїй основі використовують ітераційну техніку для покращення результатів. На протязі однієї ітерації вони шукають розв'язок, який буде кращий в околі даного. Якщо таке рішення знайдено, воно стає поточним і починається нова ітерація. Так триває доти, поки приріст цільової функції

не зменшиться практично до нуля або не виконається задане число ітерацій. Очевидно, що такі методи орієнтовані на пошук лише локальних оптимумів, до того ж розташування знайденого оптимуму залежить від стартової точки. Глобальний оптимум може бути знайдений лише випадково. Для збільшення ймовірності знаходження глобального оптимуму використовується множинний експеримент з різними початковими точками, що суттєво збільшує час пошуку.

В зв'язку з цим представляє інтерес розробка алгоритмів, які б зберігали переваги описаних методів і не мали б вказаного недоліку. Саме до таких відносяться генетичні алгоритми. Тому вони були обрані як метод розв'язування задачі складання розкладу в даній дипломній роботі.

Генетичні алгоритми – це стохастичні евристичні оптимізаційні методи, основна ідея яких взята з теорії еволюційного розвитку видів [5]. Основним механізмом еволюції є природний відбір, суть якого полягає в тому, що більш пристосовані особини мають більше шансів на виживання та розмноження і, відповідно, приносять більше потомства, ніж менш пристосовані особини. При цьому завдяки передачі генетичної інформації нащадки успадковують від батьків основні їх якості. Носіями генетичної інформації індивіда виступають молекули ДНК. При розмноженні живих організмів відбувається злиття двох батьківських статевих клітин. Їх ДНК взаємодіють, утворюючи ДНК нащадка. Основний спосіб взаємодії – кросовер. При кросовері ДНК предків діляться на дві частини, а потім обмінюються своїми половинками. При спадкуванні можливі мутації через радіоактивність або інші впливи, в результаті яких можуть змінитися деякі гени в статевих клітинах одного з батьків. Змінені гени передаються нащадку і надають йому нових властивостей. Якщо ці нові властивості є корисними, то вони, швидше за все, зберуться в даному виді і при цьому відбудеться стрибкоподібне підвищення пристосованості виду.

Першим кроком при розробці математичної моделі, основаної на генетичному алгоритмі, є розробка структури хромосоми, в

якій буде зберігатися розв'язок. У нашому випадку такою „хромосомою” є розклад.

Обрана структура повинна враховувати всі особливості і обов'язкові обмеження, які висуваються до шуканого розкладу, а також те, що від її вибору напряму залежить реалізація алгоритмів кросоверу та мутації. Загалом, вибір „хромосоми” впливає не лише на швидкість, але й на збіжність алгоритму в цілому.

На наступному кроці алгоритму створюється початкова популяція, розмір якої залежить від розмірності задачі та складає звичайно декілька сотень рішень.

Для організації оптимізаційного процесу необхідно створити направляючу силу розвитку популяції. Такою силою може виступати вимога мінімізації цільової функції або, в термінах генетичних алгоритмів, функція придатності. Зазвичай як функцію придатності використовують адитивний показник оптимальності, який оснований на штрафах, що встановлюються кожному розв'язку за будь-який незручний момент в розкладі. Перевагою такого вибору є можливість налаштування алгоритму під конкретну задачу шляхом варіювання коефіцієнтів і, тим самим, зміни пріоритету при пошуку оптимального розкладу.

Таким чином, розмістивши початкову популяцію в створене штучне середовище та реалізувавши процеси селекції, кросоверу та селекції, ми отримуємо ітераційний алгоритм пошуку оптимального рішення, на кожній ітерації якого виконуються такі дії:

- 1) кожна особина популяції оцінюється за допомогою функції придатності;
- 2) кращі рішення (звичайно близько 5 %) копіюються в нову популяцію без змін. Такий принцип (принцип елітизму) не дозволяє втратити кращі рішення і забезпечує підвищену збіжність алгоритму;
- 3) на основі пропорційного відбору з поточної популяції вибираються два рішення, які піддаються рекомбінації. Для цього хромосоми батьків обмінюються відповідними частинами;
- 4) отриманий на попередньому етапі розклад може виявитися некоректним, наприклад, може не відповідати навчальному плану. В цьому випадку можна повторювати операцію рекомбінації до того часу, коли не буде отриманий коректний розклад, у такому випадку краще передбачити евристичний механізм виведення розкладу;

5) якщо нова популяція сформована, то стара видаляється, після чого переходимо до п. 1. У протилежному випадку переходимо до п. 3.

Розглянутий алгоритм є не лише стійким до локальних мінімумів, але й завдяки внутрішньому паралелізму, який працює не з окремими розв'язками, а з цілими класами розв'язків, забезпечується відносно швидкий пошук оптимального рішення.

8. Обґрунтування доцільності використання генетичного алгоритму

Серед існуючих методів розв'язку задач складання розкладів занять навчальних закладів необхідно вибрати такий, який дозволить врахувати специфіку дистанційного навчання. Для таких закладів проблема навчальних аудиторій вирішена, адже студентам і викладачам не має необхідності бути присутніми у одному й тому самому місті в один і той же час. Це, в свою чергу, накладає нові вимоги до вибору методу розв'язування задачі складання учбового розкладу. Також необхідно вибрати оптимальний метод з точки зору якості його роботи. Найбільш важливим є вибір критеріїв оптимізації, а також накладання різного роду обмежень на результати роботи методу. Вибраний метод повинен задовольняти вимогам швидкодії, бажана відсутність нестійкості до локальних оптимумів.

Не менш складною є задача реалізації обраного методу в рамках системи дистанційного навчання «Віртуальний Університет». Також необхідно внести зміни у саму постановку задачі, що в свою чергу потребує модифікації алгоритму.

Врахувавши специфіку задачі та переваги і недоліки методів, метод генетичного алгоритму найбільш пристосований та ефективний для розв'язання задачі складання розкладів.

Ідея генетичних алгоритмів (ГА) перейнята у самої природи. Розробник ГА повинен правильно встановити закони еволюції, щоб досягти бажаної мети якомога швидше. Сама ідея здається досить цікавою, а багаточисельні позитивні результати ще більше розпалюють інтерес до цього методу.

Незважаючи на цей факт, існує чимало думок, які ставлять під сумнів успішне використання ГА. Наведемо обґрунтування дієздатності ГА [3] та його спроможність ефективно вирішувати поставлені задачі.

Звісно, ГА не претендує на роль універсального засобу розв'язання задач: в одному випадку він хороший, а в іншому – не дуже. В загальному випадку ГА не знаходить оптимального розв'язку дуже складних задач, але з іншого боку, ГА може знайти непогане рішення. Особливо, якщо розмірність задачі є значною. Існує багато прикладів, коли в дуже складних задачах за допомогою ГА були отримані хороші результати.

Визначимо загальні переваги ГА:

- він не потребує ніякої інформації про поверхню відгуку;
- розриви, що присутні на поверхні відгуку, мають несуттєвий вплив на повну ефективність оптимізації;
- він стійкий до попадання в локальні оптимуми;
- добре працює при вирішенні крупномасштабних проблем оптимізації;
- використовується для широкого класу задач;
- простий та прозорий в реалізації;
- може використовуватися в задачах зі змінним середовищем.
- До недоліків ГА можна віднести:
 - не бажано використовувати його, якщо необхідно знайти точний глобальний оптимум;
 - не бажано використовувати його, якщо необхідно знайти всі розв'язки задачі, а не один;
 - значний час роботи алгоритму.

Необхідність пошуку всіх рішень також не є проблемою для цього алгоритму. Якщо дослідженню простору пошуку приділити достатньо уваги, то ГА зможе знайти декілька (або навіть всі) екстремуми.

В обох випадках простори пошуку є одноіменними. Хромосома описує одне число, яке є розв'язком задачі. Розмірність простору пошуку для більш складних задач може сягати кілька сотень. У такому випадку зображати будь-які графіки неможливо, а програми пошуку рішень є досить складними.

Розглянемо один з простих методів пошуку – метод сходження. Він працює наступним чи-

ном: спочатку генерується випадковий розв'язок, далі в нього вносять одиничне збурення і такий збурений розв'язок порівнюють з вихідним. Якщо новий розв'язок кращий, то він зберігається, а старий відкидається, в протилежному випадку – зберігається попередній (старий) розв'язок. Після цього процес повторюється зі збереженим розв'язком, але вже з іншим одиничним збуренням і так у подальшому. Якщо, наприклад, на протязі тисячі ітерацій розв'язок не покращився, то він вважається оптимальним і процес зупиняється. При сходженні завжди рухаються вгору (в розумінні функціоналу якості розв'язку). Якщо простір відгуку простий і містить лише один пік, то методом сходження можна знайти оптимальний розв'язок. Якщо ж він більш складний, і сходження відбувається поблизу одного з невеликих піків (локальних максимумів), то глобальний оптимум навряд або буде знайдений.

Генетичний алгоритм діє більш тонко. Генетична популяція містить багато розв'язків, які розосереджені всюди на просторі пошуку. Мутація діє аналогічно одиничному збуренню в методі сходження і приводить, в основному, до блукання поблизу того піку, на якому знаходиться розв'язок. Однак процедура кросоверу породжує нові точки, які можуть попадати на віддалені піки.

У цьому полягає основна відмінність генетичних алгоритмів від більш простих, таких як сходження. ГА ефективно проглядає простір пошуку, зберігаючи хороших „батьків”, які частіше, ніж погані, дають хороше потомство (кращі розв'язки). Тому такий алгоритм з більшою ймовірністю знайде кращий розв'язок задачі.

Існує два випадки, коли ГА дуже хороші. Перший випадок: коли не відомий спосіб точного розв'язання задачі. Якщо відомо, як оцінити пристосованість хромосом, то завжди можна примусити ГА розв'язувати дану задачу. Другий випадок: коли спосіб для точного розв'язання існує, але він дуже складний в реалізації і потребує значних витрат часу.

9. Застосування методу генетичного алгоритму для розв'язку задачі складання розкладу занять в системі дистанційного навчання «Віртуальний Університет»

Для задачі складання розкладу відобразимо у блоці занять навчальний план, перелік викладачів, перелік учбових груп, перелік учбових дисциплін. Блок занять представлений у вигляді двовимірного масиву $Z[i,j]$. Стовпець масиву містить інформацію про викладача, дисципліну, що він викладає, групу, для якої викладає, та тип дисципліни. Таким чином, блок занять являє собою агрегований об'єкт (рис. 1).

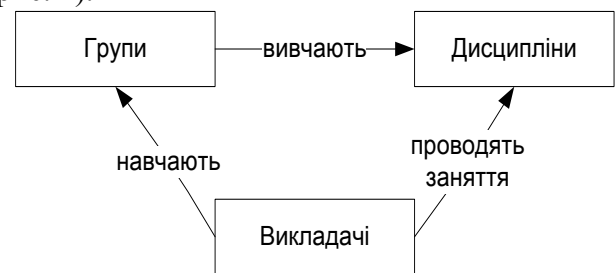


Рис. 1. Структура об'єкта «блок занять»

Множину блоків занять представимо у такому вигляді:

$$Z = \{z_i\}, \{z_i\} = \{z_i^p, z_i^d, z_i^g, z_i^t\},$$

де z_i^p – викладач, що викладає даний блок занять, $z_i^p = \overline{1, N_{pr}}$, N_{pr} – загальне число викладачів; z_i^d – дисципліна, що викладається в даному блоці занять, $z_i^d = \overline{1, N_{dsc}}$, N_{dsc} – загальне число дисциплін; z_i^g – навчальна група, для якої проводяться заняття даного блоку, $z_i^g = \overline{1, N_{grp}}$, N_{grp} – загальне число груп; $z_i^t \in \{1, N_{type}\}$ – тип, до якого відноситься дисципліна даного блоку занять (тип дисципліни відображає її приналежність до математичних, гуманітарних або іншої групи дисциплін (вводиться в модель для усунення випадків, коли в один і той самий день групі студентів будуть викладатися дисципліни одного типу, що значно зменшить ефективність навчання і сприйяття матеріалу); N_{type} – загальне число типів дисциплін.

Часова сітка занять, кількість пар у дні тижня, кількість учбових днів у тижні задані у блоці часових інтервалів. Блок часових інтервалів представлений у вигляді двовимірного масиву $\$T[i,j]$. Стовпець масиву містить інформацію про тиждень, день і час.

Множину часових інтервалів представимо у такому вигляді:

$$T = \{t_k\}, \{t_k\} = \{t_k^w, t_k^d, t_k^p\},$$

де t_k^w – номер тижня, $t_k^d = \overline{1, N_{wps}}$, N_{wps} – число навчальних тижнів у семестрі; t_k^d – номер дня тижня, $t_k^d = \overline{1, N_{dpw}}$, N_{dpw} – число навчальних днів у тижні; t_k^p – номер години протягом дня, $t_k^p = \overline{N_S, N_F}$, N_S – початок навчального дня, N_F – кінець навчального дня. Доцільно використовувати найбільший відрізок часу, наприклад, з 8:00 до 21:00, адже потрібно призначити заняття у найбільш зручний для учасників навчального процесу час, а деякі викладачі можуть надавати перевагу другій половині дня.

Побажання викладачів стосовно часу проведення занять задані у блоці переваг викладачів. Блок переваг викладачів представлений у вигляді двовимірного масиву $\$P[i,j]$. Для кожного викладача є стовпець, який містить інформацію про бажані часові інтервали (години для навчання).

Розклад дистанційних занять повинен враховувати аудиторне навантаження викладачів, оскільки вони здебільшого спершу викладають за класичною схемою аудиторні заняття, а вже потім дистанційні. Тому потрібно знати розклад занять викладачів, щоб уникнути «накладок» дистанційних занять на аудиторні. Тому розклад аудиторних занять представимо у вигляді двовимірного масиву $\$PA[i,j]$. Для кожного викладача є стовпець, який містить інформацію про часові інтервали, зайняті аудиторними заняттями.

Окрім цього важливою вхідною умовою, що може бути врахована у дистанційному навчанні, це часові інтервали, у які викладачі не хотіли б (не можуть) проводити заняття.

Зберігатимемо ці умови у вигляді двовимірного масиву $\$PX[i,j]$. Для кожного викладача є стовпець, який містить інформацію про небажані часові інтервали.

У результаті роботи буде отримано варіант розкладу, що відповідним чином враховує задані умови. Розклад зберігається у хромосомі.

Хромосома представлена у вигляді асоціативного масиву $\$Chromosome[i]$, що зв'язує блок занять та блок часових інтервалів. Індеси масиву – номери блоків занять (посилання на індекс). Значення масиву – номери блоків часових інтервалів.

Кожна хромосома особини складається із числа генів, рівних числу блоків занять. Інформаційним наповненням хромосоми є часові інтервали. Для кожної хромосоми визначена оцінка (штраф).

Побудову хромосоми проілюструє приклад, показаний на рис. 2.

Для заданої множини об'єктів T , Z потрібно знайти варіант розкладу, що забезпечує мінімальне значення критерію P втрат “якості” розкладу:

$$P = f(\tau) = \sum_{i=1}^N c_i w_i(\tau) \rightarrow \min .$$

де w_i значення коефіцієнта штрафу за невиконання i -ї частки критерію, c_i оцінка, що визначає ступінь невиконання i -ї частки критерію, τ – елемент з множини часових інтервалів T .

При цьому необхідно забезпечити виконання всіх заданих обмежень: відсутність накладок всіх типів (для викладачів, навчальних груп, аудиторних і дистанційних занять викладачів), необхідність проведення всіх запланованих на семестр і передбачених навчальним планом занять, врахування часових інтервалів, у які викладачі не можуть проводити заняття.

Вимоги при яких досягається найкращий розклад можуть бути наступні:

- дотримання рівномірності розподілу занять протягом тижня;
- дотримання необхідної відповідності між характером проведених занять і часовим інтервалом його проведення (наприклад, не варто проводити поспіль заняття одного й того ж типу, зокрема, лабораторні);
- урахування побажань викладачів щодо свого розкладу занять;
- вимоги, зв'язані із забезпеченням комфортності умов роботи викладачів і навчання сту-

дентів даного вузу.

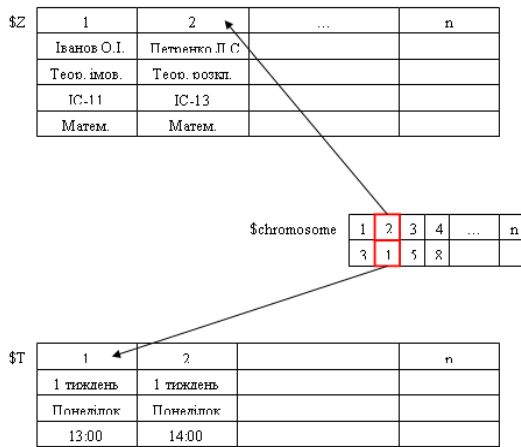


Рис. 2. Зв'язок між хромосомою, блоками занять і блоками часових інтервалів

Для запропонованої структури особин генетичного алгоритму реалізовані модифіковані процедури схрещування й мутації. Процедура схрещування в генетичному алгоритмі відбувається в такий спосіб: для двох обраних особин провадиться обмін частками генетичного коду між відповідними хромосомами (рис. 3). Точка поділу L вибирається випадково.

Оператор мутації (рис. 4) з деякою ймовірністю змінює значення декількох генів у хромосомі “нової” особини на інші значення, що входять у число припустимих значень даного гена. Наприклад, у результаті мутації i -того гена хромосоми, де i – визначає номер навчальної пари, i -тому гену буде привласнений номер навчальної пари із припустимої підмножини навчальних пар, випадково обраний з підмножини навчальних пар, призначених для проведення саме даного виду заняття.

Запропоновані процедури схрещування й мутації дозволяють враховувати ряд вимог, запропонованих до розкладу занять. Для роботи генетичного алгоритму також необхідним є використання ефективного методу відбору особин у наступну популяцію (популяцію нащадків). Для цього в алгоритмі використаний метод елітарного відбору, що сприяє збереженню кращих рішень.

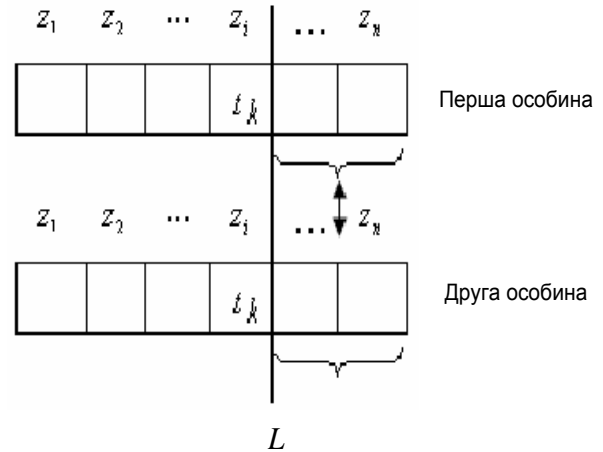


Рис. 3. Схема схрещування двох особин
Хромосома

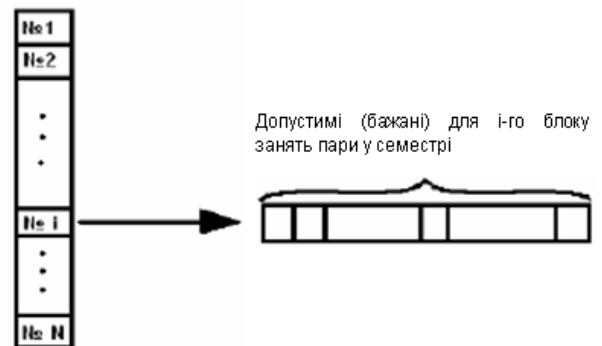


Рис. 4. Схема мутації особин

Отже, генетичний алгоритм складання розкладу занять включає наступні кроки.

1. Формування множини блоків занять.
2. Формування множини часових інтервалів.
3. Формування матриці переваг викладачів.
4. Генерація початкової популяції хромосом.

4.1. Генерація екземпляру хромосоми.

Генерація хромосоми – це процес співставлення блоків занять, що попередньо визначені, з часовими інтервалами. Тобто для кожної зв'язки «викладач-дисципліна-група» признається відповідний часовий інтервал у відповідний день відповідного тижня. Цей процес проходить випадково для кожного блоку.

4.2. Перевірка згенерованої хромосоми на допустимість

Тобто визначається, чи існують ситуації типу:

- «одна група одночасно знаходиться на заняттях у різних викладачів»;
- «викладач одночасно проводить аудиторне та дистанційне заняття»;

– «сформовано заняття у неприйнятний для викладача час»;

– «один викладач одночасно веде пару у різних групах».

4.3. Якщо таких ситуацій не існує, то маємо допустимий екземпляр розкладу і додаємо його до початкової популяції.

4.4. Перехід до п. 4.1.

5. Визначення оцінки (штрафу) хромосом у початковій популяції.

Оцінка формується виходячи з того, наскільки хромосома не задовольняє умовам, що накладаються. Цими умовами можуть бути переваги кожного викладача, рівномірність розподілу занять протягом тижня, відсутність «вікон» у розкладі, відсутність дисциплін однакового типу, що викладаються однієї групі підряд протягом дня.

6. Вибір найкращої хромосоми з популяції (з меншою оцінкою), а також вибір ще однієї з 10 кращих.

7. Виконання операції кросингування для обраних двох хромосом.

Застосування кросоверу – це обмін ділянками хромосом між хромосомами. Тобто, випадковим чином вибирається точка поділу хромосом та проходить обмін.

8. Перевірка на допустимість отриманих результаті кросингування хромосом.

9. Якщо отримано допустимі хромосоми, оцінюємо їх.

9.1. Якщо отримані хромосоми кращі існуючих, додаємо їх у популяцію замість найгірших.

9.2. Якщо ж отримані екземпляри не кращі, тоді:

9.2.1. Виконується мутація.

Мутація – це внесення випадкових змін у хромосому. Випадковим чином обирається елемент хромосоми, що відповідає за співвідношення «тиждень – день - година» і змінюється у відповідності з перевагами викладачів. Тобто замість випадково обраної зв'язки «тиждень - день - година» на початковому етапі генерації хромосоми,

ставиться зв'язка, що найкращим чином враховує побажання відповідного викладача.

9.2.2. Перевірка на допустимість: якщо отримали допустимі хромосоми, оцінюємо їх.

9.2.3. Якщо отримані хромосоми кращі існуючих, додаємо їх у популяцію замість найгірших.

10. Повторення пп. 6-9.

11. У будь-який момент можна припинити виконання алгоритму. Результатом роботи буде хромосома з найменшою оцінкою (штрафом).

Висновки

У поданому дослідженні були розглянуті питання створення підсистеми складання розкладу для системи дистанційного навчання «Віртуальний Університет» для організації найбільш зручного та ефективного навчального процесу. Розглянуто умови функціонування та вимоги складання розкладу для дистанційної форми навчання. Проведено аналіз існуючих методів розв'язання задачі складання розкладу, виявлені їх позитивні та негативні сторони, та обґрунтовано метод генетичного алгоритму. Запропонований алгоритм, реалізований мовою програмування РНР, враховує особливості системи дистанційного навчання «Віртуальний Університет», а саме: побажання викладачів стосовно часу проведення їх занять; часові інтервали, у які викладачі не можуть проводити заняття; обмеження на відсутність накладок усіх типів (для викладачів, навчальних груп, аудиторних і дистанційних занять викладачів); необхідність проведення всіх запланованих на семестр і передбачених навчальним планом занять.

Результати роботи алгоритму зберігаються у формі, придатній для подальшого оброблення. Керування параметрами роботи алгоритму винесені у конфігураційні файли, що дає можливість ефективнішого використання та внесення оптимізаційних змін у його роботу.

Закладена можливість подальшого удосконалення алгоритму та розширення його функціональності.

Список літератури

1. Логоша Б. А., Петропавловская А. В. Комплекс моделей и методов оптимизации расписания занятий в вузе // Экономика и математические методы, Т. 29, 1993, № 4.
2. Уоссермен Ф. Нейрокомпьютерная техника // Теория и практика. М. Мир. 1992.
3. Клемент Р. Генетические алгоритмы: почему они работают? Когда их применять? // Компьютерра. – 1999. – № 11.
4. Безгинов А. Н. Обзор существующих методов составления расписаний / Безгинов А. Н., Трегубов С. Ю. // Информационные технологии и программирование. Межвузовский сборник статей, М. – 2005. – Выпуск 2 (14).
5. Низамова Г.Ф. Автореферат диссертации «Математическое и программное обеспечение составления расписания учебных занятий на основе агрегативных генетических алгоритмов» // Уфа – 2006.