

ДОДАТОК А

ПРОГРАМНИЙ КОД

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Xml;
using System.IO;
using Timetable_classes;

namespace WindowsFormsApplication1
{

    public partial class Form1 : Form
    {

        SqlConnection sqlConnection;
        List<string> days = newList<string>() { "Понеділок", "Вівторок", "Середа", "Четвер",
        "П'ятниця", "Субота", "Неділя" };

        static int r_max = 0; //загальна кількість аудиторій
        static int s_max = 0; //загальна кількість занять

        int[,] conflict_matrix = newint[200, 200];
        int[,] conflict_matrix1 = newint[200, 200];
        int[,] union_matrix = newint[200, 200];
        int[,] candidate_list = newint[8, 8, 100];

        // Create a list of parts.
        List<Room> rooms = newList<Room>();
        List<Subject> subjects = newList<Subject>();
        List<Stream1> subjects1 = newList<Stream1>();

        struct Position
        {
            public int column, row;
            public Position(int column, int row)
            {
                this.column = column;
                this.row = row;
            }
        }

        public class Room
        {
            public string n { get; set; }
            public int building { get; set; }
            public int depart { get; set; }
            public string r_type { get; set; }
            public int capacity { get; set; }

            public Room(string n, int building, int depart, string r_type, int capacity)
            {

```

```

this.n = n;
this.building = building;
this.depart = depart;
this.r_type = r_type;
this.capacity = capacity;
    }
};

publicclassSubject
{
publicstringpr { get; set; }
publicstringgr { get; set; }
publicstrings_type { get; set; }
publicstringdisc { get; set; }
publicstringdisc_short { get; set; }
publicintsvoboda { get; set; }
publicintkurs { get; set; }
publicstringspec { get; set; }
publicstringdep { get; set; }
publicintn_modul { get; set; }
publicintpodgr { get; set; }
publicintcapacity { get; set; }
publicintstream { get; set; }
publicboollocated { get; set; }

publicSubject(stringpr, stringgr, strings_type, stringdisc, stringdisc_short, intsvoboda,
intkurs, stringspec, stringdep,
intpodgr = 0, intn_modul = 1, intcapacity = 1, intstream = -1, boollocated=false)
{
this.pr = pr;
this.gr = gr;
this.s_type = s_type;
this.disc = disc;
this.disc_short = disc_short;
this.svoboda = svoboda;
this.kurs = kurs;
this.spec = spec;
this.dep = dep;
this.n_modul = n_modul;
this.podgr = podgr;
this.capacity = capacity;
this.stream = stream;
this.located = located;
    }
}

publicclassStream1
{
publicstrings_type { get; set; }
publicintsvoboda { get; set; }
publicintcapacity { get; set; }

public Stream1(strings_type, intcapacity, intsvoboda)
{
this.s_type = s_type;
this.capacity = capacity;
this.svoboda = svoboda;
    }
}

publicclassLesson
{
publicintpr { get; set; }
publicintdisc { get; set; }
publicints_type { get; set; }

```

```

public int room { get; set; }
public Lesson(int pr = -1, int disc = -1, int s_type = -1, int room = -1)
{
    this.pr = pr;
    this.disc = disc;
    this.s_type = s_type;
    this.room = room;
}

public Form1()
{
    InitializeComponent();

    //allowdrop
    this.dataGridView1.AllowDrop = true;
    this.dataGridView_timetable.AllowDrop = true;
}

private void ACO()
{
    SqlDataReader sqlReader = null;
    string query = "SELECT * from [Rooms]";
    SqlCommand command = new SqlCommand(query, sqlConnection);
    try
    {
        sqlReader = command.ExecuteReader();
        while (sqlReader.Read())
        {
            Room room = new Room(Convert.ToString(sqlReader["n"]),
                Convert.ToInt32(sqlReader["building"]),
                Convert.ToInt32(sqlReader["id_dep"]),
                Convert.ToString(sqlReader["r_type"]),
                Convert.ToInt32(sqlReader["capacity"]));

            rooms.Add(room); // добавление элемента
        }

        sqlReader.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(), MessageBoxButtons.OKCancel);
    }
    r_max = rooms.Count;

    XmlDocument xDoc = new XmlDocument();
    using (var sr = new StreamReader("input.xml", Encoding.UTF8))
    xDoc.Load(sr);

    // получим корневой элемент
    XmlElement xRoot = xDoc.DocumentElement;
    // обход всех узлов в корневом элементе
    foreach (XmlNode group in xRoot)
    {
        // если узел - group
        if (group.Name == "group")
        {
            string group_id = group.Attributes.GetNamedItem("id").Value.ToString();
            string dep = group.SelectSingleNode("department").InnerText;
            string spec = group.SelectSingleNode("speciality").InnerText;
            int kurs = Convert.ToInt32(group.SelectSingleNode("kurs").InnerText);
            XmlNode schedule = group.SelectSingleNode("schedule");
            foreach (XmlNode course in schedule.ChildNodes)
            {

```



```

union_matrix[i, j] = 1;
union_matrix[j, i] = 1;
    }
if ((union_matrix[i, j] != 1) && (i != j) &&
    ((subjects[i].pr == subjects[j].pr) ||
    ((subjects[i].gr == subjects[j].gr) && (subjects[i].podgr ==
subjects[j].podgr)) ||
    ((subjects[i].gr == subjects[j].gr) && ((subjects[i].podgr == 0)
    || (subjects[j].podgr == 0))))))
    {
conflict_matrix[i, j] = 1;
conflict_matrix[j, i] = 1;
    }
    }
//формирование потоков
int temp = 0, s1_max = -1;
for (int i = 0; i < s_max; i++)
    {
if (subjects[i].stream == -1)
    {
temp = 0;
for (int j = i; j < s_max; j++)
if (union_matrix[i, j] == 1)
    {
temp++;
if (temp == 1)
        {
            s1_max++;
subjects[i].stream = s1_max;
subjects[j].stream = s1_max;
Stream1 stream_temp = new Stream1(subjects[i].s_type, temp, 1);
// добавление элемента
            subjects1.Add(stream_temp);
        }
else
        {
subjects[j].stream = s1_max;
            subjects1[s1_max].capacity = temp;
            subjects1[s1_max].s_type = subjects[j].s_type;
        }
    }
    }
    }
s1_max++;
for (int i = 0; i < s_max; i++)
for (int j = i + 1; j < s_max; j++)
    {
temp = subjects[i].stream;
int temp1 = subjects[j].stream;
conflict_matrix1[temp, temp1] = conflict_matrix1[temp, temp1] +
conflict_matrix[i, j];
if (conflict_matrix1[temp, temp1] > 1) conflict_matrix1[temp, temp1] = 1;
conflict_matrix1[temp1, temp] = conflict_matrix1[temp, temp1];
    }
    }
    }
textBox2.Text = Convert.ToString(s1_max + subjects1.Count);
    }

private void treeview_upload()
    {
SqlDataReader sqlReader = null;
// очистка дерева
    treeView1.Nodes.Clear();

// установка источника изображений

```

```

        treeView1.ImageList = imageList1;
        treeView1.BeginUpdate();

Stringquery;
query = "SELECT * from [Departments],[Staff],[Teachers] WHERE Departments.Id_dep=Staff.Id_dep
AND Teachers.Id_teacher=Staff.Id_teacher";
SqlCommandcommand = newSqlCommand(query, sqlConnection);
try
    {
    sqlReader = command.ExecuteReader();
    TreeNodecurrNode = null;
    TreeNodeteachers_Category = newTreeNode { Text = "Викладачі",
    Tag = 0, ImageIndex = 1, SelectedImageIndex = 1 };
        treeView1.Nodes.Add(teachers_Category);// Додавляемэлемент, какузел

while (sqlReader.Read())
    {
    stringdep = Convert.ToString(sqlReader["name_dep"]);

if (currNode == null || currNode.Text != dep)
    {
    TreeNodenewNode = newTreeNode { Text = dep, Tag = sqlReader["Id_dep"],
    ToolTipText = "кафедра", ImageIndex = 1, SelectedImageIndex = 1 };
    teachers_Category.Nodes.Add(newNode);// Додавляемэлемент, какузел
    currNode = newNode;
    }

    TreeNodeChildNode = newTreeNode { Text = sqlReader["name"].ToString(),
    Tag = sqlReader["Id_teacher"], ToolTipText = "викладач", ImageIndex = 2,
    SelectedImageIndex = 2 };
    currNode.Nodes.Add(ChildNode);
    }

    sqlReader.Close();
    }
catch (Exceptionex)
    {
    MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(), MessageBoxButtons.OKCancel);
    }

query = "SELECT * from [Rooms], [Buildings] WHERE Buildings.Id_building=Rooms.building ORDER
BY Id_building, r_type";
command = newSqlCommand(query, sqlConnection);
try
    {
    sqlReader = command.ExecuteReader();
    TreeNodcurrbuild = null, currtype = null;
    TreeNoderooms_Category = newTreeNode { Text = "Аудиторний фонд",
    Tag = 0, ImageIndex = 1, SelectedImageIndex = 1 };
        treeView1.Nodes.Add(rooms_Category);// Додавляемэлемент, какузел

while (sqlReader.Read())
    {
    stringbuilding = (sqlReader["name_building"] + "-корпус").ToString();

if (currbuild == null || currbuild.Text != building)
    {
    TreeNodenewNode = newTreeNode { Text = building, Tag = sqlReader["Id_building"],
    ToolTipText = "корпус", ImageIndex = 1, SelectedImageIndex = 1 };
    rooms_Category.Nodes.Add(newNode);// Додавляемэлемент, какузел
    currbuild = newNode;
    }
    stringr_type = sqlReader["r_type"].ToString();

if (currtype == null || currtype.Text != r_type)

```

```

        {
TreeNodeChildNode = newTreeNode { Text = r_type, Tag = 0,
ToolTipText = "тип", ImageIndex = 1, SelectedImageIndex = 1 };
currbuild.Nodes.Add(ChildNode);// Добавляемэлемент, какузел
currtype = ChildNode;

        }
TreeNode ChildNode1 = newTreeNode { Text = sqlReader["n"].ToString(),
Tag = sqlReader["Id_room"], ToolTipText = "аудитория", ImageIndex = 3,
SelectedImageIndex = 3 };

currtype.Nodes.Add(ChildNode1);

        }
        treeView1.EndUpdate();
sqlReader.Close();
    }
    catch (Exceptionex)
    {
        MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(), MessageBoxButtons.OKCancel);
    }
}

privatevoidroom_calendar_clear()
{
    for (introw = 0; row<Parameters.p_max; row++)
    for (intcol = 0; col<Parameters.day_max; col++)
        {
        room_calendar_week_odd.Rows[row].Cells[col].Style.BackColor = Color.LawnGreen;
        room_calendar_week_even.Rows[row].Cells[col].Style.BackColor = Color.LawnGreen;
        }
    room_calendar_week_odd.ClearSelection();
    room_calendar_week_even.ClearSelection();
    ignor.Checked = false;
    available.Checked = true;
}

PositionStart_point;
privatevoid Form1_Load(objectsender, EventArgs e)
{
    Parameters.day_max = 6;
    Parameters.p_max = 6;
    // TODO: даннаястрокакодапозволяет загрузить данные в таблицу "database1DataSet3.Departments".
    При необходимостинаможетбытьперемещенаилиудалена.
    this.departmentsTableAdapter.Fill(this.database1DataSet3.Departments);
    StringconnectionString = @"DataSource = (LocalDB)\MSSQLLocalDB; AttachDbFilename =
c:\users\user\documents\visual
studio2015\Projects\WindowsFormsApplication1\WindowsFormsApplication1\Database1.mdf";
    sqlConnection = newSqlConnection(connectionString);
    sqlConnection.Open();
        treeView1.Height = splitContainer1.Panel1.Height - panel2.Height;
    treeView_studies.Height = splitContainer2.Panel1.Height-panel_header.Height;

    for (int i = 0; i < 8; i++)
    for (int j = 0; j < 8; j++)
    for (int k = 0; k < 100; k++)
    candidate_list[i,j,k] = -1;

```

```
treeview_upload();

//количество строк и столбцов календаря загрузки
room_calendar_week_odd.RowCount = Parameters.p_max;
room_calendar_week_odd.ColumnCount = Parameters.day_max;
room_calendar_week_even.RowCount = Parameters.p_max;
room_calendar_week_even.ColumnCount = Parameters.day_max;
room_calendar_week_odd.AutoSizeColumns();
room_calendar_week_even.AutoSizeColumns();
room_calendar_clear();

dataGridView_timetable.AutoGenerateColumns = false;
    }

private void Form1_FormClosing(object sender, FormClosingEventArgs e)
    {
if (sqlConnection != null) sqlConnection.Close();
    }
}
```

ДОДАТОК Б
СЛАЙДИ ПРЕЗЕНТАЦІЇ

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Атестаційна робота магістра

**Дослідження адаптивних методів
складання розкладу
навчального процесу в університеті**

Науковий керівник:
к.т.н., доц.

Назаров О.С.

Виконала:
студентка групи ІПЗмзд-18-1

Гайтан О.М.

2020

2

Актуальність роботи

Складання розкладу навчального процесу у закладах вищої освіти є важливою складовою системи забезпечення навчального процесу, оскільки від якості розкладу залежить комфорт учасників навчального процесу та його якість і ефективність.

Завдання складання розкладу в університеті характеризується:

- значним обсягом різноманітної оперативної інформації, отриманої з різних структурних підрозділів ЗВО;
- складністю ідентифікації та формалізації параметрів та обмежень розкладу;
- конфліктом інтересів основних учасників навчального процесу;
- трудомісткістю адаптації універсальних алгоритмів складання розкладу до потреб конкретного навчального закладу.

3

Об'єкт, предмет та мета досліджень

- **Об'єкт дослідження** – організація навчального процесу в університеті, розклад навчального процесу.
- **Предмет дослідження** – моделі та методи комп'ютерного складання розкладу навчального процесу.
- **Мета і задачі дослідження** – розроблення та дослідження гібридного підходу до розв'язання задачі складання розкладу з метою підвищення якості та швидкості створення розкладу та програмна реалізація системи складання розкладу навчального процесу в університеті з використанням розробленого методу.
- **Методи дослідження.** Системний аналіз, матричний аналіз, теорія графів, математична логіка та теорія прийняття рішень. Технології об'єктно-орієнтованого програмування, теорія баз даних, Drag-and-drop технології.

4

Постановка задачі

- Розробити гібридний алгоритм складання розкладу навчального процесу в університеті.
- Спроекувати та програмно реалізувати систему складання розкладу навчального процесу, яка повинна забезпечувати виконання таких функцій:
 - імпорт даних із xml-файлу; редактор вхідних xml-файлів;
 - зберігання, завантаження, оновлення та видалення даних в довідниках;
 - ручне, автоматичне та автоматизоване формування розкладу;
 - підтримку Drag-and-drop технологій при ручному формуванні розкладу;
 - підтримку занять в підгрупах та об'єднання лекційних занять в потоки;
 - візуалізацію навантаження учасників навчального процесу у вигляді діаграм та графіків;
 - вибірку даних в розрізі факультетів, кафедр, навчальних груп, викладачів, навчальних корпусів та навчальних аудиторій;
 - експорт даних в xml та xlsx файли.

5

Публікації магістранта за темою роботи

Результати атестаційної роботи викладені в 3 публікаціях, у тому числі – 2 статтях у журналах з переліку наукових фахових видань України, а також в матеріалах конференції:

- Гайтан О.М., Назаров О.С. Hybrid approach to solving of the automated timetabling problem in higher educational institution (**«Системи управління, навігації та зв'язку»**, фахове видання, категорія «Б», індексація в Index Copernicus).
- Гайтан О.М. Автоматизація генерації розкладу навчального процесу університету (**Вчені записки Таврійського національного університету імені В.І.Вернадського, серія «Технічні науки»**, фахове видання, категорія «Б», індексація в Index Copernicus).
- Гайтан О.М. Підвищення ефективності методу мурашиних колоній в задачі комп'ютерної генерації розкладу навчального процесу ВНЗ (**72-а наукова конференція професорів, викладачів, наукових працівників, аспірантів та студентів НУ «Полтавська політехніка імені Юрія Кондратюка»**)

6

Формальна постановка задачі складання розкладу

Відповідно до навчальних планів у поточному семестрі необхідно прочитати d навчальних дисциплін S_1, S_2, \dots, S_d : для кожного $i, i = \overline{1 \dots d}$ дисципліна S_i складається з lec_i лекцій, $pract_i$ практичних занять і семінарів й lab_i лабораторних робіт. Кожне заняття s проводиться визначеним викладачем pr_j . Є gr навчальних планів G_1, G_2, \dots, G_{gr} , які являють собою набір навчальних дисциплін, що читаються певній групі студентів Gr_k . Це означає, що дисципліни в $G_{ik}, k = \overline{1 \dots gr}$ повинні бути рознесені у часі.

ts_max – кількість навчальних періодів (пар), r_max_k – максимальна кількість занять, яку можна запланувати в період t_k (кількість аудиторій, доступних у період $t_k, k = \overline{1 \dots ts_max}$).

Необхідно розподілити заняття по всім курсам у межах певної кількості аудиторій і періодів часу, тобто

7

Формальна постановка задачі складання розкладу (продовження)

знайти таке $s_{tr}^{ijk} (i = \overline{1 \dots d}, j = \overline{1 \dots pr}, k = \overline{1 \dots gr}, t = \overline{1 \dots ts}, r = \overline{1 \dots r_max})$, що

$\forall i = \overline{1 \dots d} \sum \{s_{tr}^{ijk} \mid j \in Pr, k \in Gr, t \in Ts, r \in R\} = lec_i + pract_i + lab_i$ – розподіл усіх занять;

$\forall t = \overline{1 \dots ts_max} \sum \{s_{tr}^{ijk} \mid i \in D, j \in Pr, k \in Gr, t \in Ts, r \in R\} \leq r_max_t$ – використання обмеженого аудиторного фонду;

$\forall t = \overline{1 \dots ts_max} \forall r = \overline{1 \dots r_max} \sum \{s_{tr}^{ijk} \mid i \in D, j \in Pr, k \in Gr, t \in Ts\} \leq 1$ – запобігання накладок по аудиторії;

$\forall t = \overline{1 \dots ts_max} \forall k = \overline{1 \dots gr} \sum \{s_{tr}^{ijk} \mid i \in D, i \in G_k, j \in Pr, t \in Ts, r \in R\} \leq 1$ – запобігання накладок по групі;

$\forall t = \overline{1 \dots ts_max} \forall j = \overline{1 \dots pr} \sum \{s_{tr}^{ijk} \mid i \in D, k \in Gr, t \in Ts, r \in R\} \leq 1$ – запобігання накладок по викладачу;

$\forall i = \overline{1 \dots d} \forall k = \overline{1 \dots ts} s_{ijk} \in \{0, 1\}$

$s_{tr}^{ijk} = \begin{cases} 1, & \text{якщо } d_i \text{ призначено на пару } t, \\ 0, & \text{в іншому випадку} \end{cases}$

8

Модель предметної області

Змінні моделі розкладу навчального процесу:

S – навчальне заняття $S = \{s_i | s_i = (s_i^{gr}, s_i^{pr}, s_i^d, s_i^{subgr}, s_i^{dep}, s_i^{study_type}, s_i^n), i = \overline{1, s_max}\}$,

Str – множина потоків $Str = \{Str_i \subseteq Gr | i = \overline{1, str_max}\}$

Ts – множина таймслотів (навчальних пар) $Ts = \{t_k | t_k = (t_k^w, t_k^d, t_k^p), k = \overline{1, d_max \cdot t_max}\}$

Ресурси:

Pr – множина викладачів: $Pr = \{Pr_i | i = \overline{1, pr_max}\}$

Gr – множина навчальних груп: $Gr = \{g_i | g_i = (g_i^n, g_i^v, g_i^r), i = \overline{1, gr}\}$

**За номером навчальної групи
однозначно визначаються:**

$\forall i = \overline{1, gr} \exists! (g_i^{spec}, g_i^{kurs}, g_i^{dep}),$

Календар доступності групи:

$$g_i^{w, d, t_k} = \begin{cases} 1, & \text{якщо групі } l \text{ можна призначити заняття} \\ & \text{на } i - \text{тому тижні в день } d_j \text{ на парі } t_k \\ 0, & \text{якщо групі } l \text{ не можна призначити заняття} \\ & \text{на } i - \text{тому тижні в день } d_j \text{ на парі } t_k \end{cases}$$

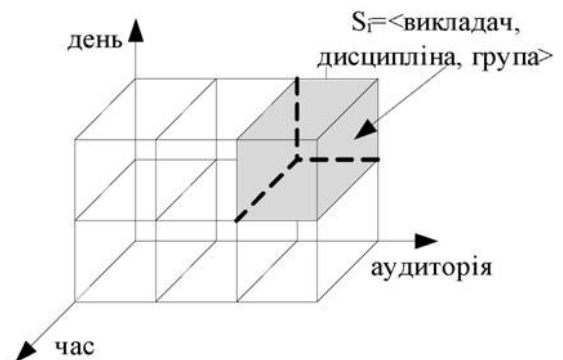
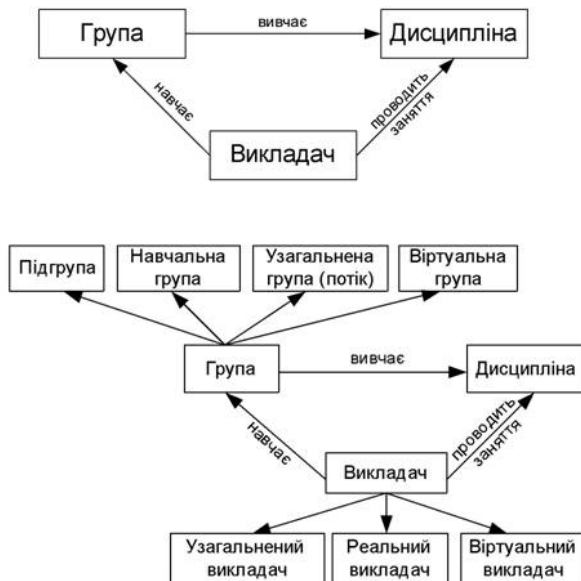
R – множина навчальних аудиторій R: $R = \{r_i | r_i = (r_i^n, r_i^b, r_i^{dep}, r_i^{study_type}, r_i^v), i = \overline{1, r_max}\}$

9

Модель предметної області

Змінні моделі розкладу навчального процесу:

S – навчальне заняття: $S = \{s_i | s_i = (s_i^{gr}, s_i^{pr}, s_i^d, s_i^{subgr}, s_i^{dep}, s_i^{study_type}, s_i^n)\}, i = \overline{1, s_max}$



$candidate_list \{room, day, time\} =$
 $= \begin{cases} i, & \text{якщо } S_i \text{ проводиться в } day, time \text{ в аудиторії } room \\ 0 & \text{в іншому випадку} \end{cases}$

10

Модель предметної області

Змінні моделі розкладу навчального процесу:

Матриця об'єднань $M_union_{s_max \times s_max}$ – це бінарна матриця така, що:

$$M_union_{ij} = \begin{cases} 1, \text{ якщо заняття } S_i \text{ та } S_j \text{ повинні проводитися} \\ \text{одночасно в одній аудиторії;} \\ 0 \text{ в іншому випадку.} \end{cases}$$

$$\forall (s_i, s_j) | s_i \in S, s_j \in S ((s_i^d = s_j^d) \wedge (s_i^{kurs} = s_j^{kurs}) \wedge (s_i^{spec} = s_j^{spec}) \wedge s_i^{lec} \wedge s_j^{lec} \wedge (s_i^{mod} = s_j^{mod})) \rightarrow M_union_{ij} = 1.$$

Матриця конфліктів $M_conflict_{s_max \times s_max}$ – це бінарна матриця така, що:

$$M_conflict_{ij} = \begin{cases} 1, \text{ якщо заняття } S_i \text{ та } S_j \text{ мають спільних студентів} \\ \text{або проводяться одним викладачем та не є потоком;} \\ 0, \text{ в іншому випадку.} \end{cases}$$

$$\forall (s_i, s_j) | s_i \in S, s_j \in S ((i \neq j) \wedge (M_union_{ij} = 0) \wedge \wedge ((s_i^{pr} = s_j^{pr}) \vee ((s_i^{gr} = s_j^{gr}) \wedge (s_i^{subgr} = s_j^{subgr})) \vee \vee ((s_i^{gr} = s_j^{gr}) \wedge (\neg s_i^{subgr} \vee \neg s_j^{subgr})))) \rightarrow M_conflict_{ij} = 1.$$

11

Постановка завдання складання розкладу як задачі багатокритеріальної оптимізації

Задача

NP-повна задача багатокритеріальної комбінаторної оптимізації з обмеженнями

Вектор змінних параметрів:

$$\vec{O} = (\{S_i, R_j, T_k\})$$

$$|X| = s_max \times r_max \times t_max$$

Обмеження:

жорсткі обмеження

$$F_i(X) = 0$$

Критеріальні обмеження:

м'які обмеження

$$F_i(X) \rightarrow \min$$

Вектор критеріїв оптимальності

$$F(X) = (F_1(X), F_2(X), \dots, F_{|F|}(X))$$

Постановка задачі

$$\min_{X \in Ts} F(X) = F(X^*)$$

12

Класифікація обмежень до складання розкладу (критеріїв оптимальності)



13

Формалізація деяких обмежень до складання розкладу

- Відсутність аудиторних накладок:

$$\forall (r_i, t_j) | r_i \in R, t_j \in T (\exists! s_{r,k} | s_{r,k} \in S_{t_j}) \vee (\neg \exists s_{r,k} | s_{r,k} \in S_{t_j})$$

- Відсутність накладок для викладача:

$$\forall (pr_i, t_j) | pr_i \in Pr, t_j \in T (\exists! s_{pr,k} | s_{pr,k} \in S_{t_j}) \vee (\neg \exists s_{pr,k} | s_{pr,k} \in S_{t_j})$$

- Відсутність накладок для групи / підгрупи:

$$\forall gr_i \forall podgr_j \forall t_k | gr_i \in Gr, podgr_j \subseteq gr_i, t_k \in T \sum (s_{gr_i t_k} + s_{gr_i podgr_j t_k}) \leq 1$$

- Відповідність типу аудиторії типу заняття:

$$\forall i | s_{r}^i \in S, r \subseteq R_{pos}^i$$

- Рівномірність навантаження студентів та викладачів

$$\bar{N}_{gr} = \frac{1}{d} \sum_{i=1}^d N_{gr}^{d_i}$$

$$D = \frac{1}{d} \sum_{i=1}^d (\bar{N}_{gr} - N_{gr}^{d_i})^2 \rightarrow \min$$

- Мінімізація «вікон» у розкладі студентів та викладачів

$$W_{gr}^{d_i} = \begin{cases} t_{\max}^{d_i} - t_{\min}^{d_i} - N_{gr}^{d_i} + 1, & \text{якщо } t_{\max}^{d_i} \neq 0 \\ 0 & \text{в іншому випадку} \end{cases}$$

$$W = \sum_{i=1}^d W_{gr}^{d_i} \rightarrow \min,$$

14

Зведення багатокритеріальної задачі складання розкладу до однокритеріальної

- Принцип оптимальності Парето:

$$P_f(X) = \{x^* \in X \mid \text{немає такого } x \in X, \text{ що } f(x) \geq f(x^*)\}$$

- множина Парето

$$F(X) = F_i(X)$$

- Метод головного критерію:

- Методи згортки векторного критерію оптимальності в скалярний:

- метод лінійної адитивної згортки з нормуючими множниками:

$$F(X) = \sum_{i=1}^{|F|} \lambda_i F_i(X)$$

- метод лінійної адитивної згортки з ваговими коефіцієнтами:

$$F(X) = \sum_{i=1}^{|F|} \lambda_i \beta_i F_i(X)$$

- метод мультиплікативної згортки:

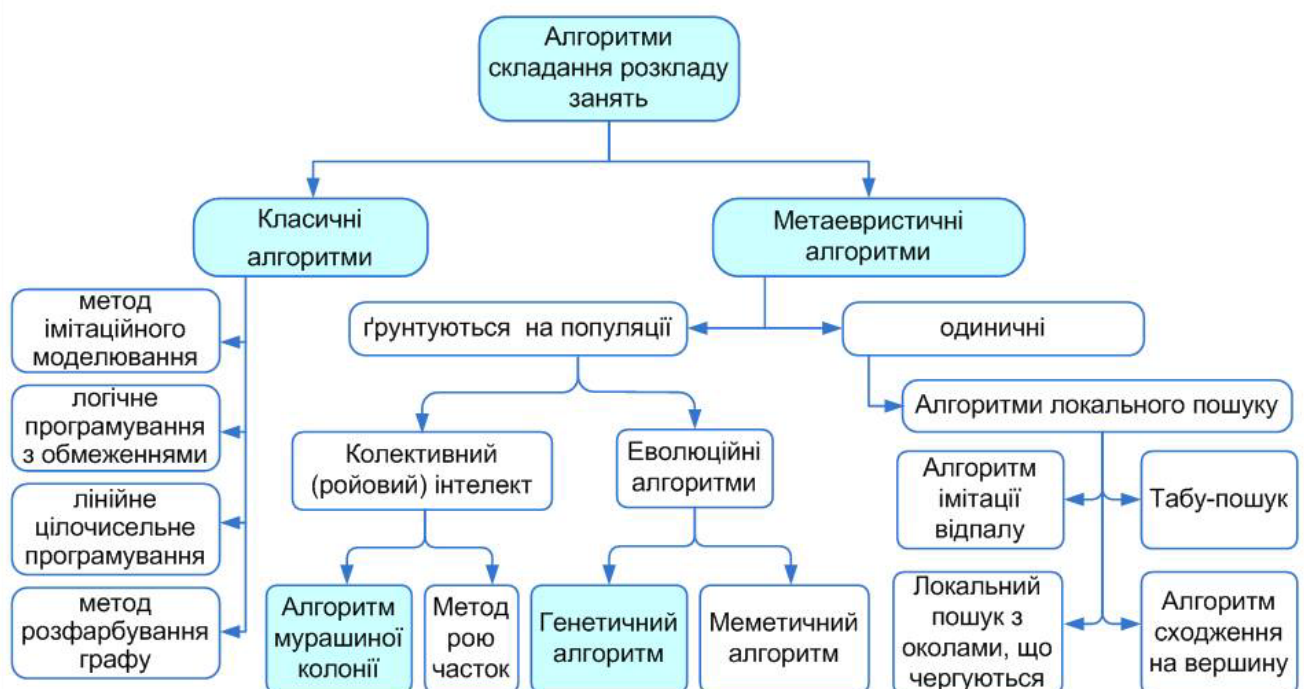
$$F(X) = \prod_{i=1}^{|F|} \lambda_i F_i(X)$$

Постановка задачі складання розкладу у вигляді однокритеріальної задачі:

$$F(X^*) = \min_{X \in Ts} F(X)$$

15

Методи складання розкладу навчального процесу



16

Метод мурашиних колоній

- **Особливості:**
 - ймовірна техніка пошуку кращих шляхів за допомогою графів з використанням штучних мурах, які наслідують поведінку колонії природних мурах.
 - послідовне наближення до оптимального рішення. Кожна ітерація – запуск штучної мурахи, який намагається за деяким правилом вибрати найкращий маршрут до «їжі» (оптимуму функції), використовуючи мітки своїх попередників.
- **Переваги:**
 - гарантування збіжності до оптимального рішення;
 - стохастичність пошуку, за рахунок чого виключається можливість зацікнення в локальному оптимумі.
- **Недоліки:**
 - невизначеність часу збіжності при тому, що збіжність гарантується;
 - сильна залежність результатів роботи методу від початкових параметрів пошуку, які підбираються експериментально.

17

Генетичний алгоритм

- **Особливості:**
 - Основні етапи генетичного алгоритму – послідовний вибір, схрещування та модифікація шуканих параметрів (генів хромосом популяції).
- **Переваги:**
 - стійкість до локальних мінімумів;
 - не потрібна інформація про поверхню відгуку;
 - відносно швидкий пошук оптимального розв'язку завдяки внутрішньому паралелізму.
- **Недоліки:**
 - проблему формування початкової популяції.

18

Гібридний метод, запропонований магістрантом

- Основні принципи методу:
 - метод мурашиної колонії – основа даного алгоритму;
 - метод деформованого багатогранника – для автоматичного знаходження параметрів методу мурашиних колоній використовується
 - генетичний алгоритм – для зменшення часу роботи алгоритму та збільшення ймовірності попадання в глобальний оптимум.
- Переваги гібридного методу:
 - використання генетичного алгоритму дозволяє зменшити час роботи алгоритму та збільшити ймовірність попадання в глобальний оптимум.

19

Параметри гібридного алгоритму

Привабливість вершини:

- $\eta_{i,j}^{\beta}$ – евристична інформація щодо привабливості розміщення заняття і на місце j в розкладі; обчислюється евристично за одним з правил:

- За загальним правилом:

$$\eta_{i,j} = \frac{1}{s_max} \quad \text{– рівномірний розподіл, коли відсутня інформація о перевагах;}$$

- За правилом суб'єктивних переваг або нечіткої міри.

Параметри впливу:

$$\alpha_1 = \text{random}(0.5, 1), \beta_1 = \text{random}(0, 0.5);$$

$$\alpha_2 = 0.5, \beta_2 = 0.5;$$

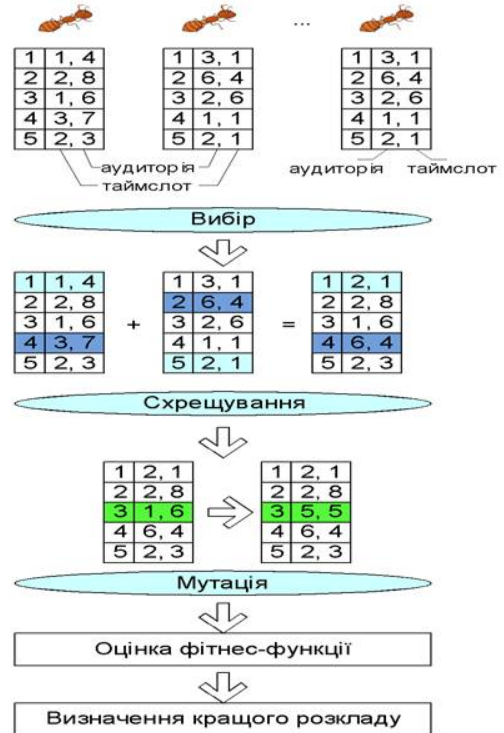
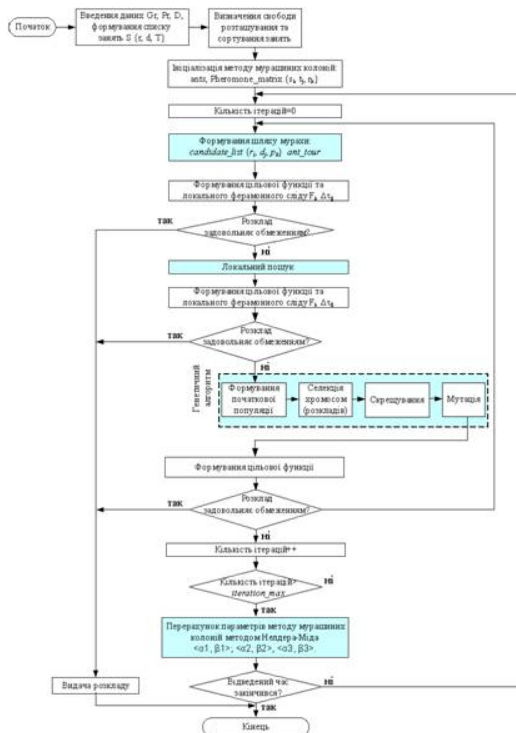
$$\alpha_3 = \text{random}(0, 0.5), \beta_3 = \text{random}(0.5, 1).$$

Ймовірності переходу:

$$P_{ij} = \frac{(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}{\sum (\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}$$

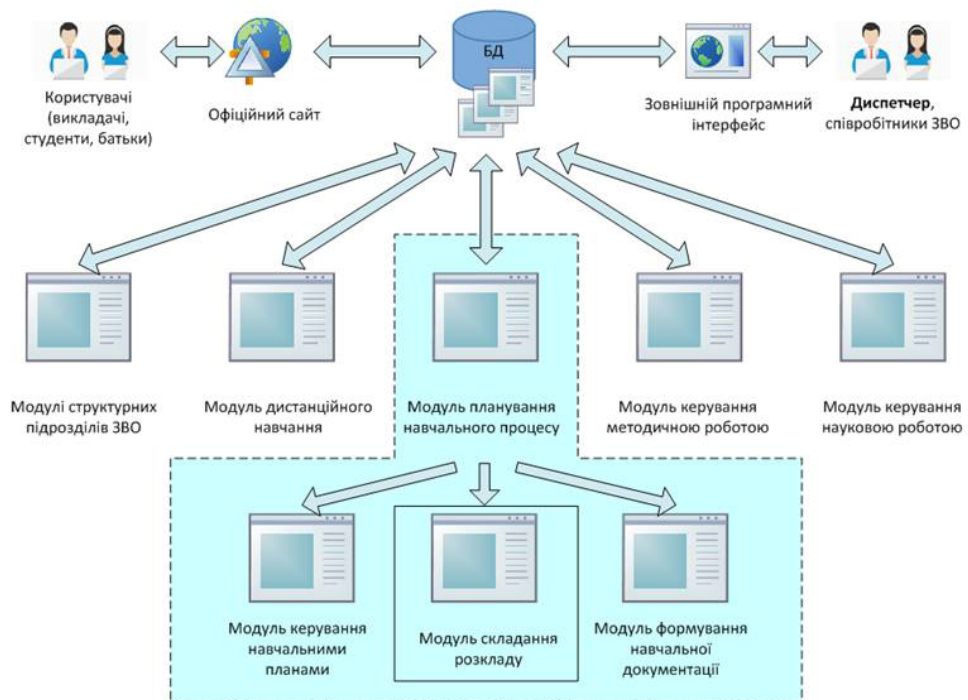
20

Блок-схема роботи гібридного алгоритму



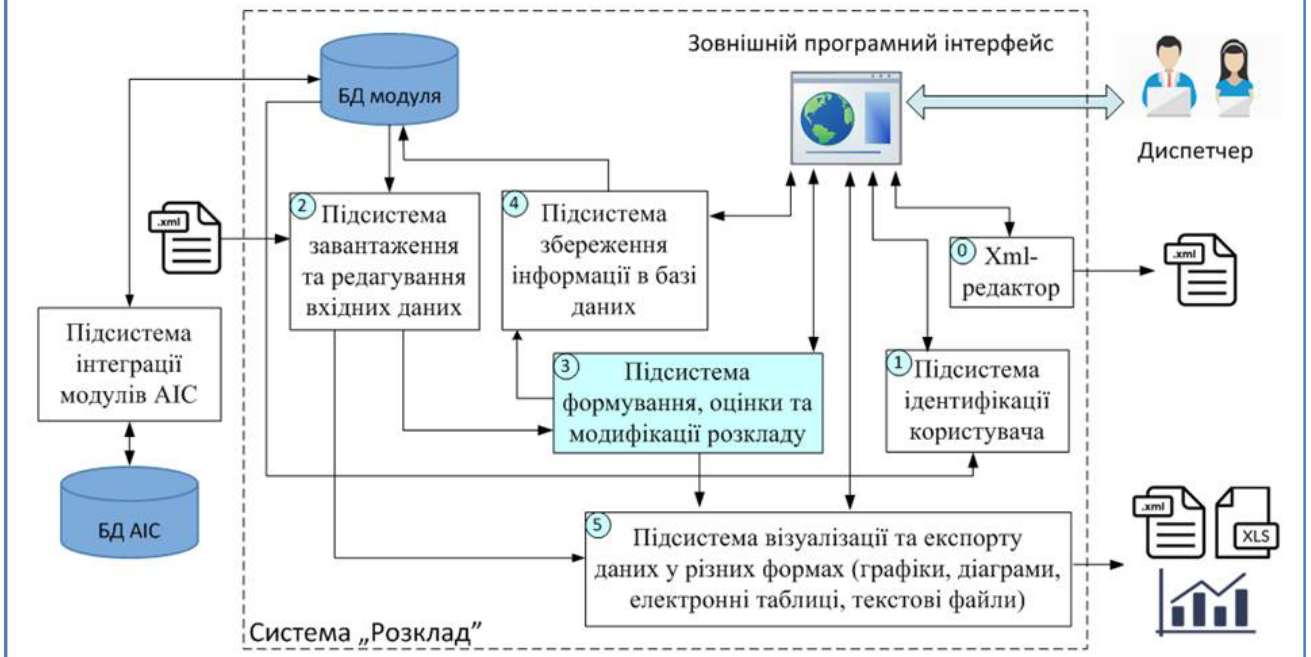
21

Підсистема складання розкладу в загальній структурі автоматизованої інформаційної системи ЗВО



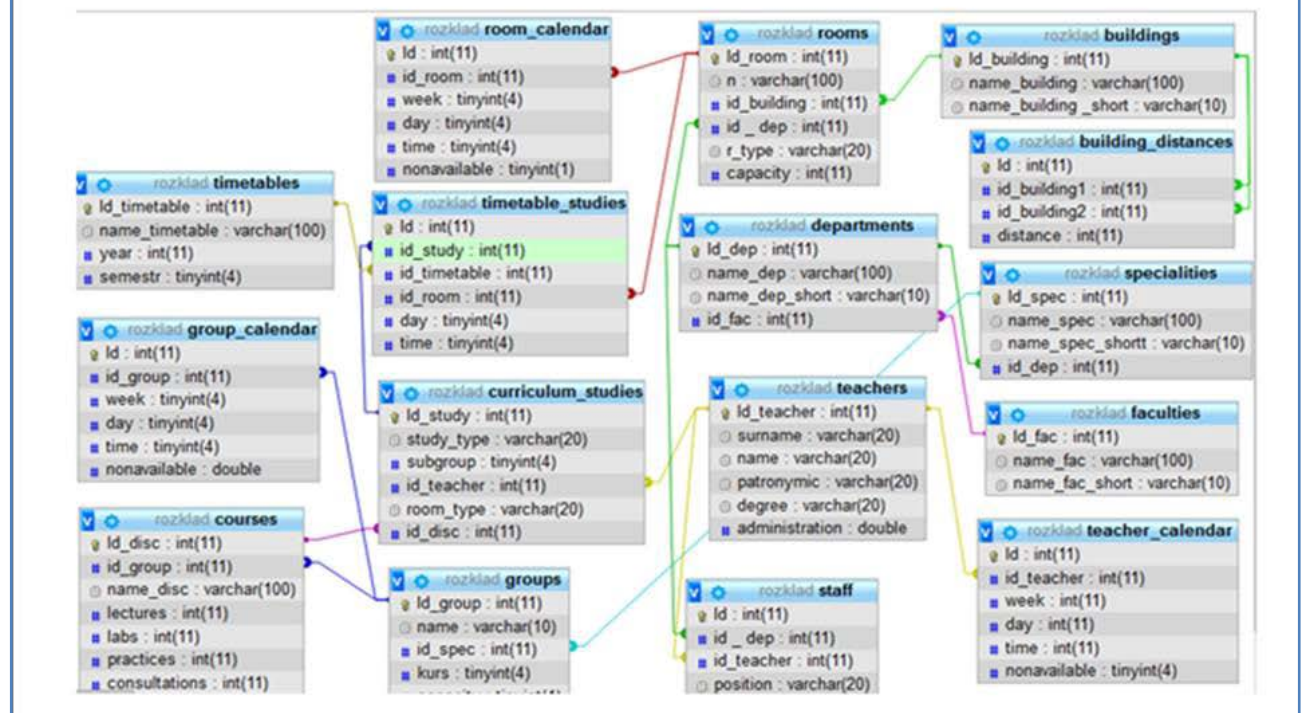
22

Структурна схема системи «Розклад»



23

Схема бази даних системи «Розклад»



24

Можливості системи «Розклад»

Підготовка оперативної семестрової інформації

- Формування даних
- Представлення даних в xml-форматі

Оновлення постійних довідників

- Кафедри
- Викладачі
- Навчальні групи
- Аудиторний фонд

Завантаження оперативної семестрової інформації

- Навчальні плани
- Штатний розклад
- Навантаження

Формування розкладу

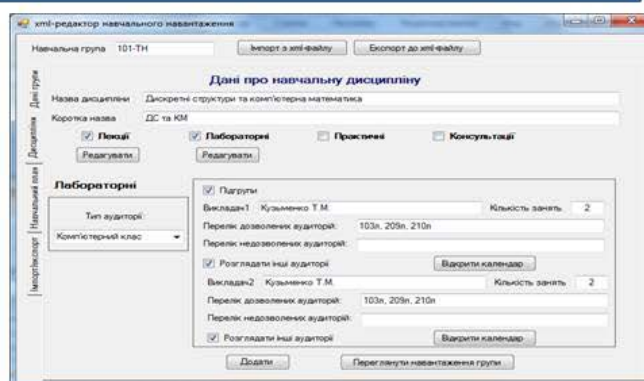
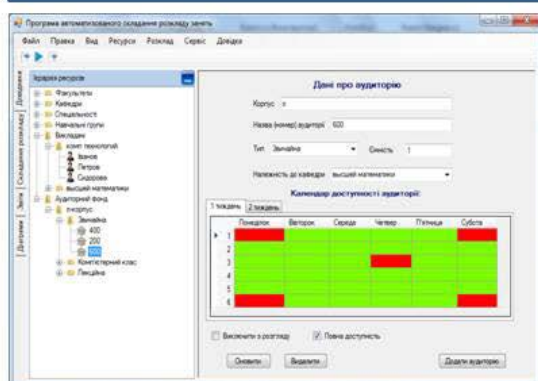
- Налаштування параметрів алгоритму
- Робота алгоритму
- Перевірка та оптимізація розкладу

Перегляд та експорт складеного розкладу

- Перегляд розкладу у вигляді таблиць
- Візуалізація навантаження учасників навчального процесу
- Експорт розкладу

25

Екранні форми системи «Розклад»



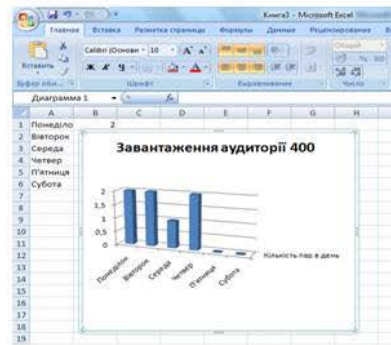
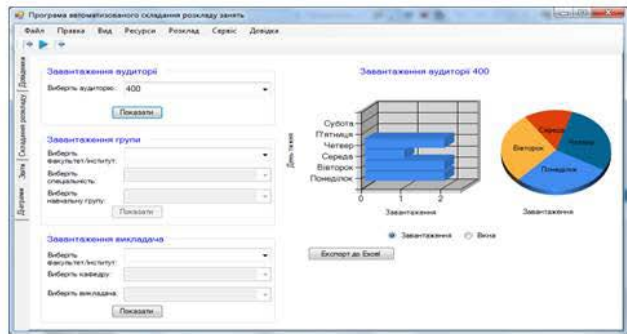
Курс	Спеціальність	Група	Дисципліна	Лекції	Проспекти	Лабораторії	завантаження
1	Комп'ютерні науки	101-ТН	Алгоритми та програмування	Данченко В.Д. (2)	Григорук Е.В. (4)	Григорук Е.В. (4)	2
1	Комп'ютерні науки	102-ТН		Данченко В.Д. (4)	Данченко В.Д. (4)		2
1	Комп'ютерні науки	101-ТН	Введення до спеціальності	Кузьменко Б.П. (2)	Кузьменко Б.П. (2)	Кузьменко Б.П. (2)	2
1	Комп'ютерні науки	102-ТН		Кузьменко Б.П. (2)	Кузьменко Б.П. (2)		2
1	Комп'ютерні науки	101-ТН	Дискретна структура та комп'ютерна математика	Кий Р.Г. (2)	Кий Р.Г. (2)	Кузьменко Б.П. (2)	2
1	Комп'ютерні науки	102-ТН		Кий Р.Г. (2)	Кий Р.Г. (2)	Кузьменко Б.П. (2)	2
1	Комп'ютерні науки	102-ТН	Комп'ютерні мережі та інтернет-технології	Польський Ю.І. (2)	Польський Ю.І. (2)	Польський Ю.І. (2)	2

Курси ресурсів	Діагностика	Час	400	200	600	2-
Понеділок	08:30 - 09:50					
	10:00 - 11:20	101-ТН, 102-ТН, 103-ТН				
	11:50 - 13:10	Віща математика				
	13:20 - 14:40	Лекція				
	14:50 - 16:10					
	16:20 - 17:40					
Вівторок	08:30 - 09:50					
	10:00 - 11:20	101-ТН, 102-ТН, 103-ТН				
	11:40 - 13:00					

26

Екранні форми системи «Розклад»

Д	Час	100-Тч	100-Тч
1	08:30	100-Тч	100-Тч
2	10:00	100-Тч, 100-Тч, 100-Тч	100-Тч, дисципліна математика, Лаб. Дадченко Л.Д.
3	11:30	Вища математика, Лекція, Вулиця І.К.	Вища математика, Лекція, Вулиця І.К.
4	11:30	100-Тч, Дисципліна математика, Лаб. Дадченко Л.Д.	
5	13:30		
6	14:30		
7	16:20		
8	08:30	100-Тч, Дисципліна математика, Лаб. Дадченко Л.Д.	100-Тч, Вища математика, Лаб. Вулиця І.К., 1
9	10:00	100-Тч, 100-Тч, 100-Тч	100-Тч, 100-Тч, 100-Тч
10	11:30	Фізика, Лекція, Чукало Р.В.	Фізика, Лекція, Чукало Р.В.
11	11:30	100-Тч, Програмування, Лаб. Ткаченко Р.В.	100-Тч, Вища математика, Лаб. Сонца Д.А., 2
12	13:30		
13	14:30		
14	16:20		
15	08:30	100-Тч, 100-Тч, 100-Тч	100-Тч, Програмування, Лаб. Ткаченко Р.В.
16	10:00	100-Тч, 100-Тч, 100-Тч	100-Тч, 100-Тч, 100-Тч
17	11:30	Вища математика, Лекція, Вулиця І.К.	Вища математика, Лекція, Вулиця І.К.
18	11:30	100-Тч, Фізкультура, Лаб. Чукало Р.В.	
19	13:30		
20	14:30		
21	16:20		
22	08:30	100-Тч, Дисципліна математика, Лаб. Дадченко Л.Д.	100-Тч, Фізика, Лаб. Радиски Е.
23	10:00	100-Тч, Вища математика, Лаб. Вулиця І.К., 1, 1/Лаб. Сонца Д.А., 2	100-Тч, Дисципліна математика, Лаб. Дадченко Л.Д.
24	11:30		
25	13:30		
26	14:30		
27	16:20		
28	08:30	100-Тч, Фізика, Лаб. Радиски Е.	100-Тч, Дисципліна математика, Лаб. Дадченко Л.Д.
29	10:00	100-Тч, 100-Тч, 100-Тч	100-Тч, 100-Тч, 100-Тч
30	11:30	Програмування, Лекція, Ткаченко Р.В.	Програмування, Лекція, Ткаченко Р.В.
31	13:30		
32	14:30		
33	16:20		



27

Результати тестування системи «Розклад»

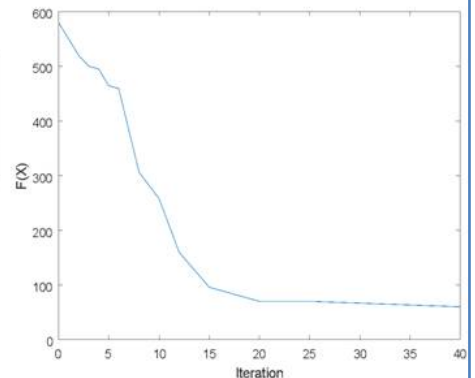
- Тестування розробленої системи на реальних даних НУ «Полтавська політехніка імені Юрія Кондратюка», в Навчально-науковому інституті інформаційних технологій і механотроніки на кафедрі комп'ютерних та інформаційних технологій і систем.

Кількісні показники	Дисципліни	Групи	Підгрупи	Кафедри	Аудиторії	Викладачі	Розмірність розкладу		
							Дні	Пари в день	Максимальне навантаження в день
Кількісні дані	35	13	21	1	25	14	5	6	4

- Результати експериментів

Показник	Максимальне значення	Мінімальне значення	Середнє значення	Середньоквадратичне відхилення
Значення	73	60	64	5.4

- Висновок:** зменшення значення штрафної функції, а отже, підвищення ефективності складання розкладу, на 12,1%.



Висновки

- У роботі розглянуто адаптивні методи складання розкладу закладу вищої освіти.
- Розроблено гібридний підхід до розв'язання даної задачі на основі методу мурашиних колоній, генетичного алгоритму та методу деформованого багатогранника.
- Розроблений гібридний метод є основою програмної реалізації системи складання розкладу навчального процесу в університеті.
- Розроблена система була протестована на реальних даних кафедри комп'ютерних та інформаційних технологій і систем Навчально-наукового інституту інформаційних технологій і механотроніки НУ «Полтавська політехніка імені Юрія Кондратюка». Результати тестування підтвердили ефективність запропонованого метода.
- Очікується, що використання даної системи в університеті дозволить полегшити та підвищити ефективність роботи диспетчерів та якість навчального процесу за рахунок максимального врахування побажань учасників навчального процесу.

ДОДАТОК В
АПРОБАЦІЯ РЕЗУЛЬТАТІВ РОБОТИ

В.1 Стаття до фахового журналу «Системи управління, навігації та зв'язку», том 2 № 60 (2020)

UDC 004.91:378.145

O. Haitan¹, O. Nazarov²

¹*National University «Yuri Kondratyuk Poltava Polytechnic», Poltava, Ukraine*

²*Kharkiv National University of Radio Electronics, Kharkiv, Ukraine*

HYBRID APPROACH TO SOLVING OF THE AUTOMATED TIMETABLING PROBLEM IN HIGHER EDUCATIONAL INSTITUTION

The paper describes a hybrid approach to solving of the automated timetabling problem in higher educational institution based on the ant colony optimization, the genetic algorithm, and the Nelder–Mead method. The ant colony method is the basis of this algorithm, which forms the initial population for the genetic algorithm. The combination of this method with the genetic algorithm and the Nelder–Mead method reduces time of the convergence of an algorithm and eliminates the strong dependence of the results on the initial search parameters, which usually are selected experimentally. The Nelder–Mead method is used to find the parameters of the ant colony optimization method. Use of the genetic algorithm allows for reducing of algorithm running time and increasing of global optimum finding probability. The educational process timetabling in higher school is an important component of the educational process assurance system, since the schedule quality determines the comfort of the educational process participants and its quality and effectiveness. Therefore, the development of methods for computer-aided timetable generation is **an important challenge**. **The subject of study** is adaptive methods of automated university timetabling. **The objective of the work** is development of a hybrid approach to addressing the problem of automated timetabling in university. **The results** are development and research of a hybrid method and software for university timetabling that been implemented this method.

Key words: Multi-Objective Optimization, Ant Colony Optimization, Genetic Algorithm, Nelder–Mead Method, Timetable, Timeslot

Introduction

The educational process timetabling in higher school is an important component of the educational process assurance system, since the schedule quality determines the comfort of the educational process participants and its quality and effectiveness.

The task of the university timetabling is characterized by [1]:

- a significant amount of diverse operational information from various structural units of the university, such as departments, deaneries, the educational office, etc.;

- the difficulty in identifying and formalizing the schedule parameters and limitations; quality of taking them into account completely depended upon the experience and qualification of the dispatcher;

- a conflict of interests of the educational process main participants: students and teachers, departments - owners of rooms and specialized equipment, that imposes restrictions on the freedom of their use in the schedule;

- the complexity of adaptation of universal timetabling algorithms to the needs of a particular educational institution, especially taking into account the new working conditions of educational institutions.

Despite the fact that manual timetabling takes a lot of time, it is still widespread due to the significant cost or lack of appropriate computer programs.

Therefore, the study of appropriate methods and

the development of effective software for automated timetabling is an important challenge.

The relevance of the development of the effective methods of automated university timetabling has been demonstrated by the international conferences on the Practice and Theory of Automated Timetabling (PATAT) [2]. PATAT is held biennially and serves as a forum for an international community of researchers and practitioners on all aspects of computer-aided timetable generation. PATAT-2020 conference will be held on 25 Aug – 28 Aug 2020, Bruges, Belgium. PATAT has been supporting various competitions and challenges, in particular International Timetabling Competition ITC 2019, investing in the timetabling community for the benefit of the field.

This work is concerned with research of the adaptive hybrid method of university timetabling and the development of a software system for the course timetabling. It is envisaged to test the developed system on the real data of National University «Yuri Kondratyuk Poltava Polytechnic», at the Educational and Scientific Institute of Information Technologies and Mechatronics.

Analysis of the problem and problem statement

The problem of computer schedule generation was being considered in many papers. In [3], a survey on the methods for university course timetabling was provided, and their benefits and disadvantages were analyzed. Description of the basic methods used for solving this problem is also presented in [4].

Both classical methods, including linear integer programming [5], graph coloring method [6], simulation modeling [7] etc., and metaheuristic methods, including simulated annealing [8], evolutionary algorithms [9, 10] etc. are widely used for automatic course timetabling. It was noted [11] that most of the classical methods use an iterative technique of continuous optimization, it allows finding only a local optimum, and a global optimum can be found only by chance.

In order to go beyond the local optimum and find a global optimum, let combine the ant colony method with the genetic algorithm which will be run each time after the next iteration of the ant colony method.

So, **the objective of the work** is development of a hybrid approach to addressing the problem of automated university timetabling which will ensure the creation of a schedule during the allowed time with minimal violation of the schedule restrictions.

The subject of study is adaptive methods of automated university course timetabling.

The task of university course timetabling can be presented in a formalized form as follows.

Input data:

According to the curriculum there are d courses S_1, S_2, \dots, S_d to teach in the current semester: for each $i, i = \overline{1..d}$ course S_i consists of lec_i lectures, $pract_i$ practical studies and seminars, and lab_i laboratory studies. Each class s is held by the teacher pr_j . There are gr curricula G_1, G_2, \dots, G_{gr} , which are sets of courses taught by given student group Gr_k . It means that courses in $G_{ik}, k = \overline{1..gr}$ must be scheduled for different periods.

Let ts_max be number of time periods (timeslots), and r_max_k maximum number of classes, that can be scheduled for period t_k (number of rooms available at period $t_k, k = \overline{1..ts_max}$).

It is necessary to schedule classes for all courses within a certain number of rooms and time periods, i.e. find so s_{pr}^{ijk} ($i = \overline{1..d}, j = \overline{1..pr}, k = \overline{1..gr}, t = \overline{1..ts}, r = \overline{1..r_max}$), that

$$\begin{aligned} & \forall i = \overline{1..d} \sum \{s_{pr}^{ijk} | j \in Pr, k \in Gr, t \in Ts, r \in R\} = \\ & = lec_i + pract_i + lab_i - \text{scheduling of all classes;} \\ & \forall t = \overline{1..ts_max} \sum \{s_{pr}^{ijk} | i \in D, j \in Pr, k \in Gr, \\ & t \in Ts, r \in R\} \leq r_max_t - \text{use of limited room quantity;} \\ & \forall t = \overline{1..ts_max} \forall r = \overline{1..r_max} \sum \{s_{pr}^{ijk} | i \in D, \\ & j \in Pr, k \in Gr, t \in Ts\} \leq 1 - \text{absence of room overlaps;} \\ & - \forall t = \overline{1..ts_max} \forall k = \overline{1..gr} \sum \{s_{pr}^{ijk} | i \in D, i \in G_k, \\ & j \in Pr, t \in Ts, r \in R\} \leq 1 - \text{absence of group overlaps;} \\ & \forall t = \overline{1..ts_max} \forall j = \overline{1..pr} \sum \{s_{pr}^{ijk} | i \in D, k \in Gr, \\ & t \in Ts, r \in R\} \leq 1 - \text{absence of teacher overlaps;} \end{aligned}$$

$$\forall i = \overline{1..d} \forall k = \overline{1..ts} s_{ijk} \in \{0,1\}.$$

The schedule restrictions can be structured as follows:

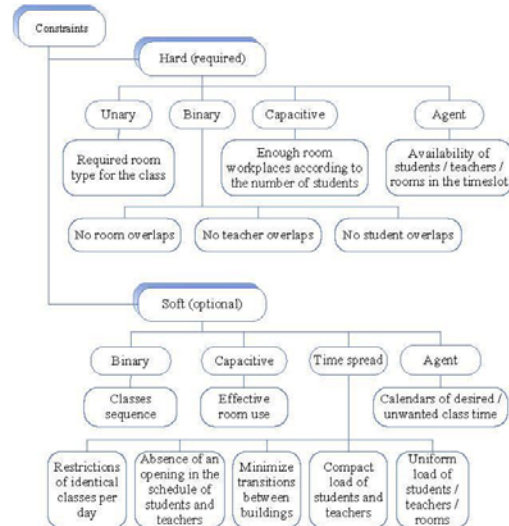


Fig. 1 – Restrictions of the generated schedule

Basic material and results

The teaching process unit and therefore the schedule unit is class (lesson). The class (lesson) is characterized by a triple $\langle \text{Course, group, teacher} \rangle$.

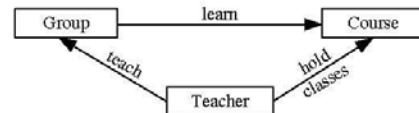


Fig. 2. Triple $\langle \text{Course, group, teacher} \rangle$

Each student can be part of an academic group, subgroup, generalized group (stream), or virtual group.

Lectures on the same course are held by one teacher for all groups of the stream at the same time; therefore, it is necessary to ensure the concept of a generalized group (stream).

Since students can choose courses to teach (from list presented in optional block of a curriculum), it is necessary to form temporary (virtual) groups from various academic groups to teach a specific discipline. In different subgroups of the same academic group classes can be held at the same time, if they are held by different teachers.

Classes in physical education or disciplines of the humanitarian block, such as philosophy, sociology, religion, etc., can be held simultaneously for all students of one level by various teachers of the department, therefore, it is necessary to introduce the concept of a

generalized teacher, which consists of real teachers.

A teacher may not be associated with real teachers if this class is held by an external person, for example, a representative of the business or industry organizations. In this case we introduce the concept of a virtual teacher.

Timetabling for such generalized groups and teachers requires controlling the employment of all academic groups and all real teachers as the part of these unions.

Therefore, the triple $\langle \text{Course, group, teacher} \rangle$ takes the following form:

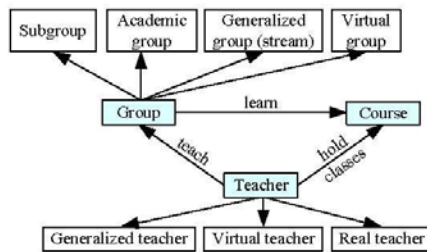


Fig. 3. Relationship among the class entities

The class diagram, which describes the relationship between these entities, is presented in fig. 4.

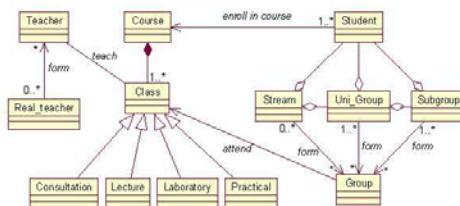


Fig. 4. The class diagram

1. Representation of classes in a schedule

Let use the matrix representation of the schedule in the algorithm. The most optimal schedule presentation is the parallelogram form, where axes are room, day and time, their intersection is class [12]. In this case, only the class number is indicated, by which it is possible to identify uniquely the teacher, group and course.

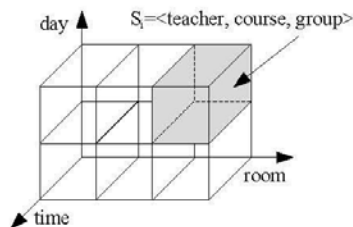


Fig. 5. Representation of the classes in the schedule

Such a presentation will ensure that there are no

room overlaps and the possibility of quick data selection and visualization, for example, using OLAP technologies.

Mathematically we describe this parallelogram in the form of a three-dimensional matrix *candidate_list*.

$$\begin{aligned} \text{candidate_list} \{ \text{room, day, time} \} = \\ = \begin{cases} i, & \text{if } S_i \text{ is taken on } \text{day, time} \text{ in } \text{room} \\ 0, & \text{otherwise} \end{cases} \quad (1) \end{aligned}$$

Relationship among the classes in the overall schedule of the educational institution is represented in the form of *union matrix* and *conflict matrix*, which are created according to the curriculum.

The union matrix is a formalization of the streaming. Classes S_i and S_j , for which $M_{\text{union}_{ij}} = 1$, are held simultaneously in the same room.

The union matrix is formed as follows:

$$\begin{aligned} \forall (s_i, s_j) | s_i \in S, s_j \in S ((s_i^{\text{day}} = s_j^{\text{day}}) \wedge \\ (s_i^{\text{hours}} = s_j^{\text{hours}}) \wedge (s_i^{\text{spec}} = s_j^{\text{spec}}) \wedge \\ (s_i^{\text{lec}} \wedge s_j^{\text{lec}} \wedge (s_i^{\text{modul}} = s_j^{\text{modul}})) \rightarrow M_{\text{union}_{ij}} = 1. \end{aligned} \quad (2)$$

The conflict matrix is a formalization of the requirement that two classes cannot overlap at a time because of presence of one group / subgroup or teacher in these classes. Classes S_i and S_j , for which $M_{\text{conflict}_{ij}} = 1$, cannot be taught at the same time.

The conflict matrix is formed as follows:

$$\begin{aligned} \forall (s_i, s_j) | s_i \in S, s_j \in S ((i \neq j) \wedge (M_{\text{union}_{ij}} = 0) \wedge \\ ((s_i^{\text{pr}} = s_j^{\text{pr}}) \vee ((s_i^{\text{gr}} = s_j^{\text{gr}}) \wedge (s_i^{\text{subgr}} = s_j^{\text{subgr}})) \vee \\ ((s_i^{\text{gr}} = s_j^{\text{gr}}) \wedge (\neg s_i^{\text{subgr}} \vee \neg s_j^{\text{subgr}}))) \rightarrow M_{\text{conflict}_{ij}} = 1. \end{aligned} \quad (3)$$

2. Used methods

2.1 Ant colony optimization method

The ant colony method is a probabilistic technique for solving the computational problems, which can be reduced to finding the best ways with help of graphs using artificial ants that inherit the behavior of natural ant colony.

This iterative algorithm is based on the idea of successive approximations to the optimal solution. Each iteration is the launch of an artificial ant that, according to some rule, tries to choose the best route to "food" (optimum function), using the labels of its predecessors.

Each ant performs a chain of steps:

1. Ant moves from node to node with transition probabilities. In order to determine the next edge of the path, the probability of transition to the j -th node is calculated when the agent is in the i -th node:

$$p_{ij} = \frac{(\tau_{i,j}^\alpha)(\eta_{i,j}^\beta)}{\sum(\tau_{i,j}^\alpha)(\eta_{i,j}^\beta)}, \quad (4)$$

where $\tau_{i,j}^\alpha$ is the amount of pheromones in the node i, j (accumulated statistical information on the quality of choice for position i of requirement j); α is a parameter to control the influence of $\tau_{i,j}^\alpha$; $\eta_{i,j}^\beta$ is node attractiveness (heuristic information about the quality of choice for position i of requirement j); β is a parameter to control the influence of $\eta_{i,j}^\beta$.

2. Recalculation (updating) of the pheromone trace (production and evaporation of the pheromone) is carried out according to the formulas:

$$\tau_{i,j} = (1 - \rho)\tau_{i,j} + \Delta\tau_{i,j}, \quad (5)$$

where $\tau_{i,j}$ is the amount of pheromone deposited on the node i, j (global pheromone trace); ρ is the pheromone evaporation coefficient; $\Delta\tau_{i,j}$ is the amount of pheromone deposited by the current ant (local pheromone trace).

3. The best solution is the one with the highest value of the local pheromone trace.

Several extensions of the ant system method were developed [13], which include: Elitist Ant System; Rank-based Ant System (ASrank); Ant Colony System (ACS); Max-Min Ant System (MAX-MIN AS – MMAS).

In this case, we will use the ant colony system (ACS) method, which accumulates benefits and improves other methods. ACS uses a pseudo-proportional rule to determine the next node to move [13]: with probability q_0 , the agent moves to the point for which the product of the pheromones quantity and heuristic information is maximum; with probability $1 - q_0$ the basic approach that described in the method of ant systems will be applied in determining the next point for the transition.

In this case, a strict elite strategy is used while restoring pheromones on the edges. Only the best agent (globally or locally) adds pheromones after each iteration of the method.

The ant colony method has been developed in detail for the traveling salesman problem (TSP), but the works on course timetabling with this method contain only general theoretical consideration of the possibility of using this method for this task.

The use of the ant colony method for solving optimization problems was considered in [13]. In [14] the timetabling tool based on Best-Worst Ant System and Best-Worst Ant Colony System is considered.

The advantages of the ant colony method are the convergence to the optimal solution, as well as stochasticity, that is, random search, which eliminates

the possibility of looping in a local optimum.

At the same time, the uncertainty of the convergence time is noted in [13], while the convergence is guaranteed; a strong dependence of the method results on the initial search parameters, which are selected experimentally.

Therefore, in our opinion, the use of a hybrid approach is an effective approach. The ACS algorithm provides good results in combination with other methods, for example, with local search or genetic algorithms.

2.1.1 Parameters of the algorithm

The initial algorithm parameters, which affect the speed and quality of the solution, are the parameters of influence α , β , and attractiveness of the node $\eta_{i,j}^\beta$. Let consider formation of these parameters.

The parameters of influence α and β are usually selected experimentally. We use the Nelder–Mead method to determine these parameters.

The Nelder–Mead method (also downhill simplex method) is a simple and effective method for optimizing (finding of minimum or maximum) of an objective function in a multidimensional space; it is a way to optimize functions without using gradients. The method is reliable and, as a rule, shows good results, while there is no convergence theory.

The algorithm is generation of the simplex and its further deformation in the direction of minimum using three operations: reflection, expansion, and contract.

In order to form a simplex we divide the ant colony into three groups, for each of which we set individual parameters of the method:

$\alpha_1 = \text{random}(0.5, 1)$, $\beta_1 = \text{random}(0, 0.5)$ – $\tau_{i,j}^\alpha$ is dominant;

$\alpha_2 = 0.5$, $\beta_2 = 0.5$ – uniform influence of $\tau_{i,j}^\alpha$ and $\eta_{i,j}^\beta$;

$\alpha_3 = \text{random}(0, 0.5)$, $\beta_3 = \text{random}(0.5, 1)$ – $\eta_{i,j}^\beta$ is dominant.

If the necessary solution is not reached during *iteration_max* iterations, it is necessary to change the parameters of the ant colony method, calculating the new ones using the downhill simplex method.

In this case, the current value of the parameters $v_1 = (\alpha_1, \beta_1)$, $v_2 = (\alpha_2, \beta_2)$, $v_3 = (\alpha_3, \beta_3)$ is selected as the initial simplex.

Using the Nelder–Mead method requires to calculate the value of the objective function at each point $f_1(\alpha_1, \beta_1)$, $f_2(\alpha_2, \beta_2)$, $f_3(\alpha_3, \beta_3)$, whose equation does not exist, so the maximum local pheromone trace during i iterations in each group of ants is taken as this value.

Sorting the points by the values of the objective function at these points, we obtain a double inequality:

$$f(\text{worst}) \leq f(\text{good}) \leq f(\text{best}). \quad (6)$$

Let use the operation of reflection; it is the projection of *worst* point through the center of gravity:

$$new = mid + \alpha(mid - worst), \quad (7)$$

$$mid = ((\alpha_{good} + \alpha_{best})/2; (\beta_{good} + \beta_{best})/2)$$

where $\alpha=1$.

We get 3 sets of parameters that are selected as the current simplex $v_1=best$, $v_2=good$, $v_3=new$.

The algorithm is repeated until the desired schedule is obtained or during *iteration_max* iterations with a new set of parameters.

If the necessary solution is not reached during *iteration_max* iterations, it is necessary to change the parameters of the ant colony method, calculating the new ones similar to the previous paragraph, but instead of the reflection, use the expansion operation according to the formula:

$$new = mid + \gamma(worst - min), \quad (8)$$

where $\gamma=2$.

The algorithm is repeated until the desired schedule is obtained or during *iteration_max* iterations with a new set of parameters.

If the necessary solution is not reached during *iteration_max* iterations, it is necessary to change the parameters of the ant colony method, calculating the new ones similar to the previous paragraph, but instead of the expansion, use the contract operation according to the formula:

$$new = mid + \beta(worst - min), \quad (9)$$

where $\beta=0,5$.

$\eta_{i,j}^\beta$ is heuristic information about attractiveness of location of class *i* on position *j* in the schedule. This parameter is calculated heuristically by one of the rules:

1. Under the general rule:

$$\eta_{i,j} = \frac{1}{s_{max}} \quad (\text{uniform distribution when there is}$$

no information about the benefits);

2. Under the rule of subjective preferences or fuzzy measure.

With the help of a fuzzy measure, the convenience of placing classes on day *d* in time *p* is defined as a subjective probabilistic measure (fig. 6).

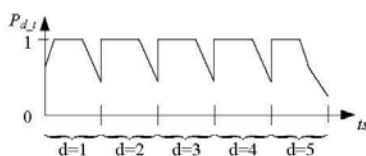


Fig. 6. Subjective probabilistic measure

It is undesirable to hold a lecture on the first lesson on Monday, daily on the fifth or sixth lessons and on the last lessons of the last day of the week, when workability decreases and the least number of students is present. For example, $p_{d,t}=0$ for $d=6$ at 5-day

workweek, $p_{d,t}=0.2$ for $p=6$ (undesirable lecture on the last lesson).

The convenience of taking a class in the *r*-th room is determined by the compliance of the room type to the class type, capacitive room indicators, belonging of room to the department, necessary laboratory equipment, software and hardware, as well as the subjective preferences of the teacher. The simplest definition: $\mu_r = 1$ at full conformity of the room, $\mu_r = 0$ at complete discrepancy of the room, $\mu_r = 0,5$ at undesirable taking of class in this room.

According to $\mu_{d,t}$ and μ_r , parameter $\eta_{i,j}$ is defined:

$$\eta_{i,j} = \max(\mu_{d,t} \cdot \mu_r), \quad (10)$$

where $0 \leq \mu_{d,t} \leq 1$ is the convenience of scheduling classes on day *d* in time *t*, $0 \leq \mu_r \leq 1$ – is the convenience of scheduling classes in *r*-th room.

Such a presentation allows taking into account the wishes of the educational process participants during course timetabling. If $\beta > \alpha$ at the iteration $i = 1, \dots, N$, an unallocated position with the highest value $\eta_{i,j}$ is selected.

The pheromone trace

In the classical ant colony algorithm, implemented, for example, for the traveling salesman problem, the pheromone is applied to the graph edges, since the edge is a transition path from point to point.

In this case, the main sources of information in the graph are the vertices (timeslot / audience combination), therefore, it is proposed to put the pheromone on the vertices, and not on the graph edges.

2.2 Genetic algorithms

Genetic algorithms are the stochastic heuristic optimization methods, the main idea of which is taken from the evolutionary theory [4]. The main mechanism of evolution is natural selection, whose essence is that more adapted individuals are more likely to survive and reproduce and, accordingly, bring more offspring than less adapted individuals. Moreover, due to the transfer of genetic information, descendants inherit their main qualities from their parents.

The procedure of the genetic algorithm:

1. A chromosome structure for storage of solutions is being developed. In this case the schedule as a “chromosome” is considered. The selected structure should take into account all the features and restrictions of the desired solution. An initial chromosomes form population.

2. Each individual in the population is evaluated using the fitness function; the best solutions are copied to the new population unchanged. The fitness function determines the suitability of the chromosome for the formation of the next generation. The principle of elitism allows saving the best decisions and provides increased convergence of the algorithm.

3. The best chromosomes are selected for further reproduction from the set of chromosomes using their fitness function.

4. Selected chromosomes cross-pair with each other to obtain a new population. If the received schedule does not conform to the curriculum, the crossover operation is repeated until the correct schedule is received. In this case, it is better to provide a heuristic mechanism for schedule correcting.

5. The mutation operator is applied to the population.

6. The fitness function is recalculated for the resulting population

7. Go to the step 2.

The genetic algorithm is stable to local minimums, and also provides a relatively quick search.

But application of the evolutionary method for solving the timetabling task causes the problem of chromosome formation. Therefore, it will be relevant using a combination of the considered approaches – the ant colony algorithm for iterative approximation to the optimum and the initial chromosome formation, and the genetic algorithm for search of the global optimum.

The use of genetic algorithms for timetabling is presented in [4, 9, 11, 15].

3. Hybrid course timetabling method

Let's consider the main stages of the hybrid method.

Let's introduce the following variables: s_max is a total number of classes according to the curriculum, d_max is a number of teaching days during the week, p_max is a maximum number of classes per day, $t_max = d_max \times p_max$ is a number of timeslots during the week, r_max is a number of rooms, S is an array of unallocated classes.

Step 1. Initialization of the method.

1.1 Creation of the classes array S with dimension s_max and its sorting by decreasing freedom of location. First of all, classes that have limited opportunities for the time of the classes or require large rooms are located.

1.2 Creation the *Pheromone matrix* $\{s_j, t_j, r_k\}$ with dimension $s_max \times t_max \times r_max$ and initialization it with the initial pheromone value at the vertices *Pheromone_start* for the implementation of the ant colony method.

1.3 Creation of an of artificial ant colony (agents) $ants[]$. For each ant it is determined: *candidate_list* $\{r_i, d_j, p_k\}$ with dimension $r_max \times d_max \times p_max$, ant path *ant_tour* with dimension s_max , which are initialized with zero values.

1.4 Separation of the ant colony of into three groups for the formation of a simplex by the Nelder-Mead method. For each ant group the initial parameters of the method $\langle \alpha_1, \beta_1 \rangle, \langle \alpha_2, \beta_2 \rangle, \langle \alpha_3, \beta_3 \rangle$ are set.

Step 2. Iterative movement of ants

2.1 Setting the initial values of the path, initialization of *candidate_list* and *ant_tour* with zero values.

2.2 Each class i is sequentially scheduled on position j using the transition probabilities. Ant moves from node to node with probability:

$$p_{ij} = \frac{(\tau_{i,j}^\alpha)(\eta_{i,j}^\beta)}{\sum_{j \in \Omega} (\tau_{i,j}^\alpha)(\eta_{i,j}^\beta)}, j \in \Omega \quad (11)$$

where Ω is the set of classes not scheduled yet.

The rule by which class i is located on position j is defined as follows:

$$j = \begin{cases} \arg \max_{h \in \Omega} (\tau_{i,h}^\alpha)(\eta_{i,h}^\beta), & q < q_0 \\ j \text{ is determined randomly according} & \\ \text{to the probability distribution,} & q \geq q_0 \end{cases} \quad (12)$$

where $0 \leq q_0 \leq 1$ is the algorithm parameter ($q_0 = 0.5$), and the value q is calculated randomly at each step.

The result of the selection is fixed in the ant path (*ant_tour* matrix) and *candidate_list* matrix.

An agent moves only to nodes that have not been visited yet and for which *candidate_list* $\{r_i, d_j, p_k\} = 0$.

In this case, the *candidate_list* matrix plays the role of a tabu list and ensures the satisfying of hard restrictions "no room overlaps". Absence of the teacher and group overlaps must be controlled separately.

Step 2.2 is repeated until each agent completes his path.

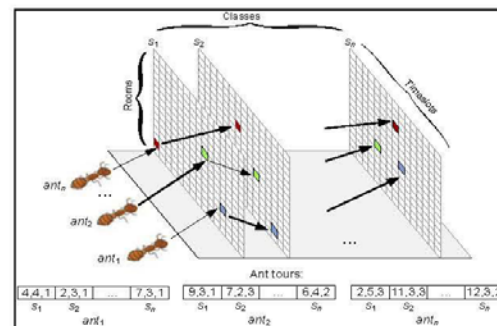


Fig. 7 – Schematic representation of the ant path

2.3 Calculation of the fitness (penalty) function F by the formula:

$$F = \sum_j \omega_j F_j, \quad (13)$$

where F is the schedule quality (the number of violations of soft restrictions); F_j is the number of violations of soft restrictions according to the j -th quality criterion; ω_j is the weight coefficient of the j -th quality criterion.

2.4 Application of the local search method and recalculation of the target (penalty) function F .

Local search methods:

Option 1:

- random class $S_0(r_0, d_0, t_0)$ is selected in the schedule;
- all classes $S_1(r_1, d_1, t_1) \dots S_n(r_n, d_n, t_n)$ with the same timeslot (d_0, t_0) are considered;
- classes i and j alternately change places, while the objective function is recalculated;
- a replacement that gives the best value of the objective function is fixed, other classes return to their positions.

Such change will improve the schedule according to the criteria "effective room use" and "uniform room load". Since the time of the classes does not change, there are no violations of strict restrictions.

Option 2:

- random class $S_i(r_i, d_i, t_i)$ is selected in the schedule;
- class S_i is transferred to all vacant positions in turn, while the fitness function is recalculated;
- a replacement that gives the best value of the objective function is fixed, other classes return to their positions.

Such change will improve the schedule according to the criteria "compact load of teachers and students" and "uniform load of educational process participants".

These options of local search can be considered as a genetic algorithm that is applied to one individual: example 1 is selection operation, example 2 is mutation operation.

2.5. The local trace is determined – the amount of pheromone deposited by the i -th ant on each vertex of the path, according to the formula:

$$\Delta\tau_{i,j} = \frac{1}{1+F_i} \quad (14)$$

where $\Delta\tau_{i,j}$ is the amount of pheromone deposited by the i -th ant; F_i is penalty for violation of soft restrictions in the schedule created by the i -th ant.

For each group of ants, the largest value of the local pheromone trace is calculated and stored.

2.6. The global trace in the pheromone matrix is calculated by the formula:

$$\tau_{i,j} = (1-\rho)\tau_{i,j} + \Delta\tau_{i,j}, \quad (15)$$

if class i is on the position j in the best schedule. Otherwise,

$$\tau_{i,j} = (1-\rho)\tau_{i,j}, \quad (16)$$

where ρ is the rate of the pheromone evaporation, uniformly distributed in the range $[0,1]$.

At the first iteration, each vertex has the opportunity to be selected. To gradually remove the vertices relevant to the worst positions, the pheromone evaporation procedure is applied to all the vertices.

If $|ant_tour^{best} - ant_tour^{worst}| < \varepsilon$, the pheromone

matrix must be restarted.

Schedules obtained by the ant colony method and improved by local search are transferred to the genetic algorithm as chromosomes.

Step 3. Genetic algorithm

3.1 On the basis of fitness functions, the best chromosomes in each ant group are selected for further reproduction and the creation of a new generation with the best indicators.

3.2 Selected chromosomes cross-pair with each other to obtain a new population, exchanging rooms or timeslots. Genes for crossover (classes) are chosen by the roulette-wheel method. The resulting chromosome is checked for validity – whether it satisfies the hard restrictions. If the received schedule does not conform to the curriculum or the crossover worsened the room use indicators, the crossover operation is repeated until the correct schedule is received.

3.3 Mutation operator is applied to a population. It is a random replacement of a room or timeslot.

3.4 The objective function (fitness function), which fixes the satisfying of soft restrictions, and the local pheromone trace are recalculated for the resulting population.

Steps 2 – 3 are performed until the desired schedule is obtained or during *iteration_max* iterations.

The algorithm stops after achievement of the desired result or after *iteration_max* iterations, as well as if during the i iterations the algorithm does not give a significant improvement i.e. $F_{global_best} - F_i < \varepsilon$.

Schematically, the work of the genetic algorithm is presented in fig. 8.

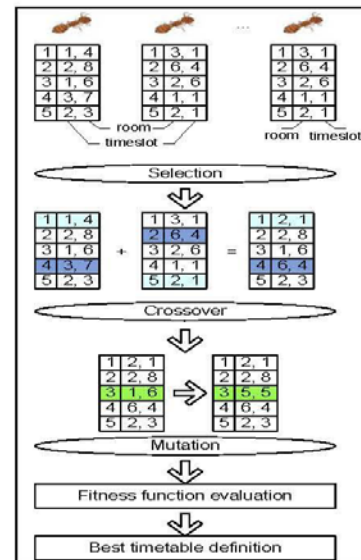


Fig. 8 – Genetic algorithm

The algorithm flowchart is presented in fig. 9.

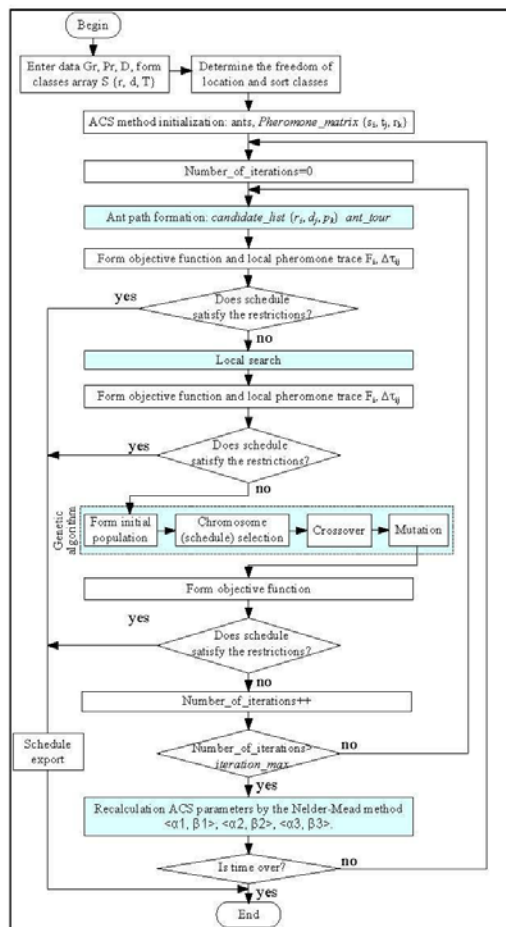


Fig. 9 – The algorithm flowchart

Software for university course timetabling was developed based on this hybrid algorithm. It provides the possibility of manual and automated course timetabling for higher education institution using the discussed methods and drag-and-drop technologies. The permanent data is stored in the sql-database, and operational semester information is imported from xml files.

The screenshots demonstrating room data updating and manual course timetabling are presented in fig. 10-11.

The fragment of the schedule that has generated as a result of automatic timetabling is presented in fig. 12.

The default restriction weights are equal 1, but it is possible to adjust them in manual mode, as well as taking into account individual quality criteria.

The developed system was tested on the following data: 1 faculty, 56 teachers, 54 groups, 20 classrooms. Testing results confirmed the effectiveness of the proposed method.

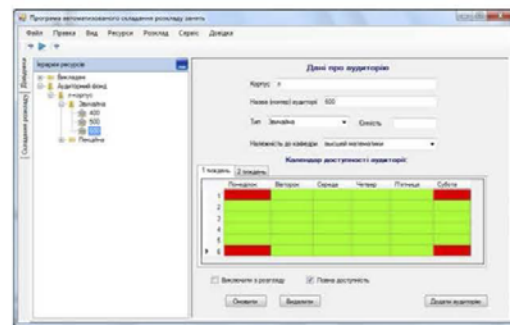


Fig. 10 – Room data updating form

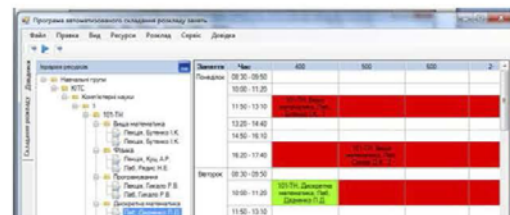


Fig. 11 – Manual course timetabling

	A	B	C	D
1	День	Час	101-TH	102-TH
2	Понедельник	08:30-09:50	Дискретная математика, Лаб.	Дискретная математика, Лаб.
3		10:00-11:20	Высшая математика, Лещицк.	Высшая математика, Лещицк.
4		11:30-13:10	Дискретная математика, Лаб.	Дискретная математика, Лаб.
5		13:20-14:40		
6		14:50-16:10		
7		16:20-17:40		
8	Вторник	08:30-09:50	Дискретная математика, Лаб.	Высшая математика, Лаб.
9		10:00-11:20	Физика, Лещицк.	Физика, Лещицк.
10		11:30-13:10	Куз А.Р., 100л	Куз А.Р., 100л
11		13:20-14:40	Программирование, Лаб.	Высшая математика, Лаб.
12		14:50-16:10	Гимназия Р.В., 101л	Семестр Д.К.(2), 102л
13		16:20-17:40		
14	Среда	08:30-09:50		Программирование, Лаб.
15		10:00-11:20	Высшая математика, Лещицк.	Высшая математика, Лещицк.
16		11:30-13:10	Высшая математика, Лещицк.	Высшая математика, Лещицк.
17		13:20-14:40	Филология	Филология
18		14:50-16:10		
19		16:20-17:40		
20	Четверг	08:30-09:50	Дискретная математика, Лаб.	Физика, Лаб.
21		10:00-11:20	Дискретная математика, Лаб.	Редис Н.Е., 410л
22		11:30-13:10	Высшая математика, Лаб.	Дискретная математика, Лаб.
23		13:20-14:40	Высшая математика, Лаб.	Дискретная математика, Лаб.
24		14:50-16:10		
25		16:20-17:40		
26	Пятница	08:30-09:50	Физика, Лаб.	Дискретная математика, Лаб.
27		10:00-11:20	Программирование, Лещицк.	Программирование, Лещицк.
28		11:30-13:10	Гимназия Р.В., 100л	

Fig. 12 – Results of automatic timetabling

Conclusions

As a result of the work, a hybrid approach to solving of the automated timetabling problem in higher educational institution based on the ant colony optimization, the genetic algorithm, and the Nelder-Mead method has been developed.

Use of this algorithm allows for reducing of algorithm running time and increasing of global

optimum finding probability

The developed hybrid method is the basis of the software for the university course timetabling. Use of this system at the university will facilitate and increase

the efficiency of the dispatchers who are involved in scheduling, and the quality of the educational process by taking into account wish of the educational process participants.

REFERENCES

1. Khasukhadzhiyev A.S., Sibikina IV. Obobshchennyi algoritm sostavleniya raspisaniya v vuze s uchedom novykh trebovaniy federal'nykh gosudarstvennykh obrazovatel'nykh standartov // *Vestnik Astrakhan. gos. tekhn. un-ta. Ser. upravleniye, vychisl. tekhn. inform.*, 2016, № 3. Pp. 78–86.
2. PATAT Conferences. URL: <https://patatconference.org>.
3. Dvoryankin A.M., Chalyshev V.S. Obzor metodov sostavleniya raspisaniya vuzov // *Izv. VolgGTU. Seriya Aktual'nyye problemy upravleniya, vychislitel'noy tekhniki i informatiki v tekhnicheskikh sistemakh: mezhvuz. sb. nauch. st.*, 2011. Vyp. 11, № 9. Pp. 110–113.
4. Tomashevs'kyy V.M., Novikov YUL., Kamins'ka P.A. Skladannya rozkladiv zanyat' u distantsiynykh systemakh navchannya // *Visnyk Natsional'noho tekhnichnoho universytetu Ukrainy "Kyivskyy politekhnichnyy instytut". Ser.: Informatyka, upravlinnya ta obchyslyval'na tekhnika*, 2010. Vyp. 52. Pp. 118–130.
5. Lagosha B.A., Petropavlovskaya A.V. Kompleks modeley i metodov optimizatsii raspisaniya zanyaty v vuze // *Ekonomika i matematicheskiye metody*, 1993. № 4. Pp. 48–56.
6. Bania Kumar Rubul, Duarah, Pinkey. Exam Time Table Scheduling using Graph Coloring Approach // *International Journal of Computer Sciences and Engineering*, 2018. № 6. Pp. 84–93.
7. Bezginov A.N., Tregubov S.YU. Obzor sushchestvuyushchikh metodov sostavleniya raspisaniy // *Informatsionnyye tekhnologii i programmirovaniye. Mezhevuzovskiy sbornik statey, M.*, 2005. Vypusk 2 (14).
8. Leite Nuno, Melicio Fernando, Rosa Agostinho. A fast simulated annealing algorithm for the examination timetabling problem // *Expert Systems with Applications*, 2018. Vol. 122.
9. Astakhova I.F., Firas A.M. Sostavleniye raspisaniya uchebnykh zanyaty na osnove geneticheskogo algoritma // *Vestnik VGU, seriya: Sistemyy analiz i informatsionnyye tekhnologii*, 2013. № 2. Pp. 93 – 99.
10. Zhukova M.YU., Al'-Gabri V.M. Avtomatizatsiya postroyeniya raspisaniya ekzamenov VUZa s ispol'zovaniyem geneticheskogo algoritma // *Inzhenernyy vestnik Dona*, 2017. №3.
11. Boyko O.M. Evolyutsiynna tekhnolohiya rozv'yazuvannya zadachi skladannya rozkladiv navchal'nykh zanyat' // *Shtuchnyy intelekt*, 2006. № 3. Pp. 341 – 348.
12. Mulyava I. YA. Systema formuvannya rozkladu navchal'noho zanyat' z vykorystanniam sub'yektyvnykh perevah // *International scientific journal*, 2016. № 7. Pp. 22–27.
13. Ustenko S.V., Bibko O.O. Vykorystannya metodu murashynoy koloniyi dlya rozv'yazannya optymizatsiynykh zadach. *Naukovyy visnyk NLTU Ukrainy*, vyp. 25, vyp. 3, Berezen', 2015. Pp. 351–359.
14. Thepphakorn T, Pongcharoen P, Hicks C. An ant colony based timetabling tool // *International Journal of Production Economics*, 2014, № 149(3). Pp. 131–144.
15. Yurchak I.YU., Moskovych T.R. Doslidzhennya henetychnykh alhorytmiv ta zastosuvannya yikh v avtomatyzovaniy systemi rozpodilu navantazheniya dlya vykladachiv i studentiv. URL: <http://eom.lp.edu.ua/sntk/doc/ksm2018/moskovyich.pdf>.

Автори:

Гайтан Олена Миколаївна

Національний університет «Полтавська політехніка імені Юрія Кондратюка», старший викладач кафедри комп'ютерних та інформаційних технологій і систем

Назаров Олексій Сергійович

Харківський національний університет радіоелектроніки, доцент кафедри програмної інженерії, заступник декана факультету КН, кандидат технічних наук, доцент

Гібридний підхід до розв'язання задачі автоматизованого складання розкладу вищого навчального закладу

О.М. Гайтан, О.С. Назаров

Анотація. У роботі описано гібридний підхід до розв'язання задачі автоматизованого складання розкладу вищого навчального закладу на основі методу мурашиних колоній, генетичного алгоритму та методу деформованого багатогранника. Метод мурашиної колонії є основою даного алгоритму, що формує початкову популяцію для генетичного алгоритму. Комбінація даного метода з генетичним алгоритмом та методом деформованого багатогранника спрямована на усунення таких недоліків даного метода як невизначеність часу збіжності алгоритму та сильна залежність результатів роботи методу від початкових параметрів пошуку, які зазвичай підбираються експериментально. Метод деформованого багатогранника використовується для знаходження параметрів методу мурашиних колоній. Використання генетичного алгоритму дозволяє зменшити час роботи алгоритму та збільшити ймовірність попадання в глобальний оптимум. Складання розкладу навчального процесу у вищому навчальному закладі (ВНЗ) є важливою складовою системи забезпечення навчального процесу, оскільки від якості розкладу залежить комфорт учасників навчального процесу та його якість і ефективність, отже, розроблення методів генерації комп'ютерного розкладу є актуальною темою. Предметом дослідження є адаптивні методи автоматичного складання розкладу ВНЗ. Мета роботи –

розроблення гібридного підходу до розв'язання задачі автоматизованого складання розкладу вищого навчального закладу. Результати – розробка та дослідження гібридного методу та програмна реалізація системи складання розкладу навчального процесу в університеті з використанням розглянутих методів.

Ключові слова: багатокритеріальна оптимізація, метод мурашиних колоній, генетичний алгоритм, метод деформованого багатогранника, розклад, таймслот.

Гибридный подход к решению задачи автоматизированного составления расписания высшего учебного заведения

Е.Н. Гайтан, А.С. Назаров

Аннотация. В работе описан гибридный подход к решению задачи автоматизированного составления расписания высшего учебного заведения на основе метода муравьиных колоний, генетического алгоритма и метода деформированного многогранника. Метод муравьиной колонии является основой данного алгоритма и формирует начальную популяцию для генетического алгоритма. Комбинация данного метода с генетическим алгоритмом и методом деформированного многогранника направлена на устранение таких недостатков данного метода как неопределенность времени сходимости алгоритма и сильная зависимость результатов работы метода от начальных параметров поиска, которые обычно подбираются экспериментально. Метод деформированного многогранника используется для нахождения параметров метода муравьиных колоний. Использование генетического алгоритма позволяет уменьшить время работы алгоритма и увеличить вероятность попадания в глобальный оптимум. Составление расписания учебного процесса в высшем учебном заведении является важной составляющей системы обеспечения учебного процесса, поскольку от качества расписания зависит комфорт участников учебного процесса и его качество и эффективность, следовательно, разработка методов генерации компьютерного расписания является актуальной темой. Предметом исследования являются адаптивные методы автоматического составления расписания вузов. Цель работы – разработка гибридного подхода к решению задачи автоматизированного составления расписания высшего учебного заведения. Результаты – разработка и исследование гибридного метода и программная реализация системы составления расписания учебного процесса в университете с использованием рассмотренных методов.

Ключевые слова: многокритериальная оптимизация, метод муравьиных колоний, генетический алгоритм, метод деформированного многогранника, расписание, таймслот.

**В.2 Стаття до фахового журналу «Вчені записки
Таврійського національного університету імені В.І. Вернадського,
серія «Технічні науки», том 31 (70) № 2, 2020**

Інформатика, обчислювальна техніка та автоматизація

УДК 004.91:378.145

Гайтан О.М.¹,

¹ старший викладач кафедри комп'ютерних та інформаційних технологій і систем
Національного університету «Полтавська політехніка імені Юрія Кондратюка»

**АВТОМАТИЗАЦІЯ ГЕНЕРАЦІЇ РОЗКЛАДУ НАВЧАЛЬНОГО ПРОЦЕСУ
УНІВЕРСИТЕТУ**

У статті розглядається питання автоматизації генерації розкладу навчального процесу університету, досліджуються наявні системи складання розкладу, виявлені їх функціональні особливості, проаналізовані переваги та недоліки. Описано у формалізованому вигляді постановку задачі побудови розкладу, виявлено та систематизовано основні жорсткі та м'які обмеження, що накладаються на створений розклад. Запропоновано гібридний підхід до комп'ютерної генерації розкладу навчального процесу, який ґрунтується на методах мурашиної колонії (Ant colony optimization) та генетичних алгоритмах (genetic algorithms). Поєднання даних методів дозволить зменшити час збіжності алгоритму та збільшити ймовірність знаходження глобального оптимуму. Також представлено програмну систему складання розкладу навчального процесу вищого навчального закладу, яка на основі постійних довідників (дані про кафедри, спеціальності, навчальні групи, викладачів, навчальні корпуси, аудиторний фонд), що зберігаються в sql-базі даних, та оперативної семестрової інформації (навчальні плани, штатне навантаження викладачів, календарі доступності та пріоритети учасників навчального процесу), імпортованої з xml-довідників, забезпечує можливість ручного, автоматичного та автоматизованого складання розкладу навчального процесу в університеті з використанням розглянутого методу та технології drag&drop. Передбачена можливість ручного коректування створеного розкладу та його експорт у xml та xlsx файли. Використання даної системи у вищих навчальних закладах дозволить полегшити та підвищити ефективність роботи диспетчерів, які займаються складанням розкладу, та якість навчального процесу за рахунок контролю відсутності конфліктів учасників навчального процесу та максимального врахування побажань учасників навчального процесу при складанні розкладу.

Ключові слова: генетичний алгоритм, комбінаторна оптимізація, метаевристика, метод мурашиних колоній, розклад, таймслот.

Постановка проблеми. Генерація розкладу навчального процесу університету є важливою складовою системи забезпечення навчального процесу, оскільки від якості складеного розкладу залежить комфорт учасників навчального процесу, його якість та ефективність. Незважаючи на широке використання інформаційних технологій, у багатьох вищих навчальних закладах розклад до цього часу складається вручну і комп'ютер використовується лише для ві-

зуалізації та розповсюдження складеного таким чином розкладу. Тому задача автоматичної генерації комп'ютерного розкладу навчального процесу в університеті є актуальною.

Проблема складання розкладу навчального процесу в університеті полягає у розподілі набору занять згідно навчальних планів в межах заданої кількості аудиторій та періодів часу (пар). Основна відмінність розкладу в університеті від середньої

школи полягає в тому, що університетські курси можуть мати спільних студентів, тоді як шкільні класи – це множини учнів, що не перетинаються. Якщо два заняття мають спільних студентів, вони конфліктують, і тому не можуть бути заплановані на той самий період. Крім того, необхідно передбачити можливість поділу академічних груп на підгрупи та об'єднання лекційних занять у потоки.

В проблемі складання розкладу університету важливу роль відіграє наявність аудиторій (їх розмір, тип і обладнання, належність до відповідної кафедри), тоді як у середній школі цим нехтують, оскільки в більшості випадків можна припустити, що кожен клас має свій кабінет і окремі вимоги висуваються лише до занять з іноземної мови, інформатики, хімії та фізкультури.

Основна мета статті полягає у розробленні гібридного метода автоматичної генерації розкладу навчального процесу та побудові програмної системи на її основі.

Об'єкт дослідження – методи та системи автоматичного складання розкладу.

Аналіз останніх досліджень і публікацій. Проблеми автоматизованої генерації розкладу навчального процесу протягом останніх десятиліть приділяється значна увага.

Публікації з даного питання можна поділити на декілька категорій:

– *формалізація предметної області:*

Так, в [1] розглянуто математичну постановку задачі складання розкладу занять у ВНЗ в умовах розбіжності вимог і побажань викладачів і студентів, введено поняття віртуальних та узагальнених груп і підгруп, проаналізовано жорсткі і нежорсткі обмеження в задачі складання розкладу.

Робота [2] присвячена стандартизації даних для складання розкладу у навчальних закладах. Автори описують формат представлення даних у частині опису навчального плану і розкладу занять на основі XML-технологій.

– *аналіз існуючих та розробка нових алгоритмів та методів складання розкладу:*

У роботі [3] проведено огляд методів складання розкладу ВНЗ, проаналізовані їхні переваги та недоліки. Опис основних методів, що використовуються для розв'язання даної задачі, також представлено у [4]. Для автоматичного складання широко використовуються як класичні методи такі, як лінійне цілочисельне програмування, метод розфарбовування графу, метод імітаційного моделювання, так і метаевристичні методи.

Оскільки дану задачу можна віднести до класу NP-важких задач зі значною кількістю обмежень і складністю побудови математичної моделі, використання класичних методів обмежене. Тому на даний час для розв'язання задачі складання розкладу ВНЗ дуже поширене застосування метаевристичних методів. Застосовують такі метаевристичні методи як метод імітації відпаду, генетичні алгоритми, метод мурашиних колоній тощо.

Метаевристичні методи зазвичай мають дві важливі особливості:

– у результаті їх роботи послідовно будуються кілька розв'язків;

– побудова кожного нового рішення ґрунтується на накопичених знаннях про якість попередніх отриманих рішень.

Класифікація основних метаевристичних методів, що застосовуються при автоматичному складанні розкладів, наведена на рис. 1.

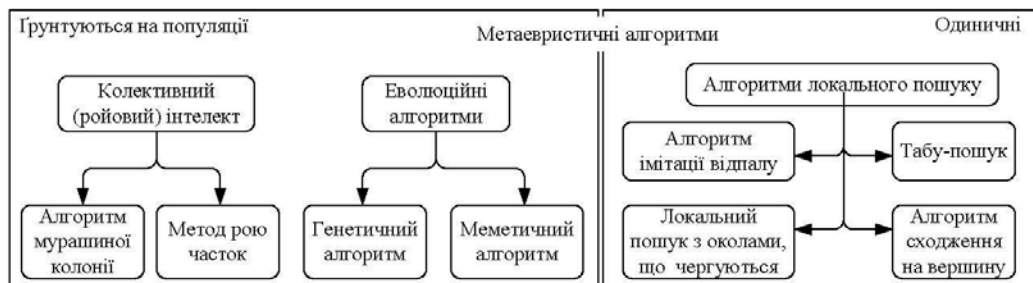


Рис. 1 – Класифікація основних метаевристичних методів



Рис. 3 – Структура системи UniTime

Систему можна використовувати окремо для створення і ведення розкладу занять і / або іспитів або для взаємодії з існуючою інформаційною системою ВНЗ. Дані для розкладу можуть вводитися безпосередньо в системі або імпортуватися із xml-файлів.

Алгоритм пошук в UniTime ґрунтується на ітеративному алгоритмі прямого пошуку (iterative forward search algorithm). Цей алгоритм схожий з локальними методами пошуку; однак, на відміну від класичних локальних методів пошуку, він працює з допустимими, але не обов'язково повними рішеннями. У цих рішеннях деякі заняття можуть залишатися не розміщеними. Однак усі жорсткі обмеження щодо призначених занять повинні виконуватися. Такі рішення

простіші для візуалізації та більш значущі для користувачів, ніж повні, але нездійсненні рішення. Через ітеративний характер алгоритму алгоритм може запускати, зупиняти або продовжувати обробляти будь-яке допустиме рішення – повне або неповне.

Дане програмне забезпечення поширюється безкоштовно за ліцензією з відкритим вхідним кодом для забезпечення можливості використання іншими коледжами та університетами, а також для сторонніх дослідників, які захочуть внести вклад в поточні дослідження в цій області. Недоліком цієї системи є орієнтація на англomовні країни.

2. Система Галактика: Розклад навчальних занять [8,9].

Система Галактика: Розклад навчальних занять призначена для автоматизації процесу складання розкладів навчальних занять в освітніх установах вищої та середньої професійної освіти. Система може використовуватися як самостійний додаток, так і в комплексі з іншими системами, наприклад, з Галактикою ERP (рішення Галактика Управління Вузом). Дані для розкладу можуть вводитися безпосередньо в системі або імпортуватися із зовнішніх систем у xml та excel-файлах.



Рис. 4 – Структура системи Галактика: Розклад навчальних занять

"Галактика" автоматизує процес складання розкладу, з огляду на множину обов'язкових, умовно обов'язкових і бажаних умов, зокрема, облаштування аудиторій під окремі дисципліни, приналежність до одного або різних навчальних корпусів, завантаженість викладачів, форму-

вання зведених груп студентів.

Важливим інструментом системи є можливість планування порядку вивчення академічних предметів. Ця опція включає графік вивчення дисциплін протягом тижня і технологічну карту з послідовністю їх проходження.

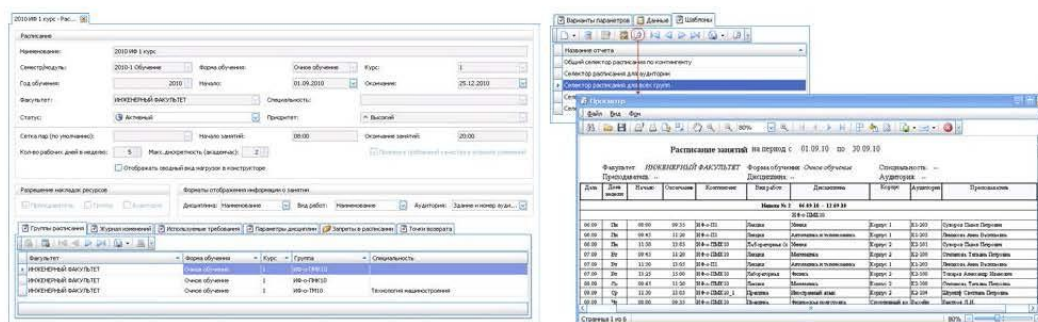


Рис. 5 – Экранні форми системи Галактика: Розклад навчальних занять

Система розкладу навчальних занять дозволяє: паралельно вести кілька розкладів по різних корпусах, днях тижня, групам; контролювати зв'язок вільних аудиторій, зокрема, спеціально обладнаних, з предметами, видами робіт, кафедрами, факультетами; налаштовувати пріоритети використання ресурсів; враховувати віддаленість корпусів, їх пріоритетність; підтримувати різні групи студентів (потік, підгрупа, зведена група), контролювати їх переміщення при формуванні розкладу, щоб заняття не дублювалися і не перетиналися; використовувати чотири десятки показників ефективності розкладу.

На відміну від попереднього рішення головна мета системи Галактика – системність рішень, спроба уникнути клаптикової автоматизації, коли для кожної критичної області, в тому числі і для складання роз-

кладу, застосовується окрема система.

Дана система орієнтована на російськомовну аудиторію і використовується переважно в російських ВНЗ. Вартість ліцензії – індивідуальна для кожного ВНЗ.

3. Програма «1С:Автоматизированное составление расписания. Университет» [10].

Продукт призначений для автоматизованого складання розкладів, управління ними під час навчального процесу і оперативного управління приміщеннями. З його допомогою складати розклад можна в автоматичному, ручному і змішаному режимах, а також в режимах вибірки по приміщеннях, по групах, по викладачах з урахуванням обмежень і умов. При цьому можна побудувати як допустимий розклад, так і оптимізований, в якому зменшено кількість вікон або кількість використовуваних приміщень.

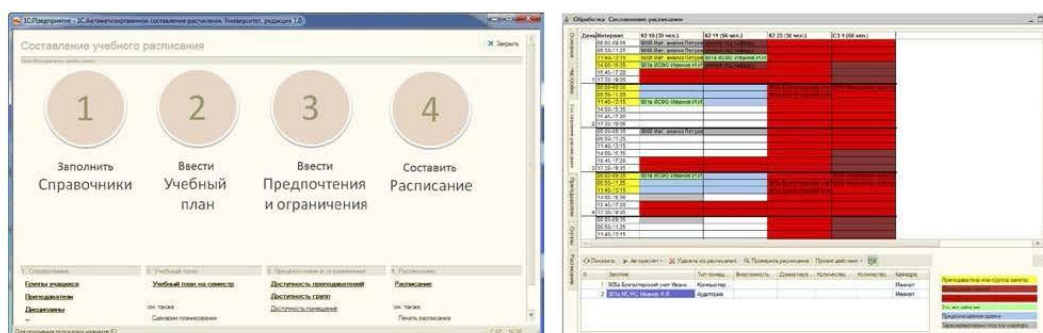


Рис. 5 – Экранні форми системи

Система розкладу навчальних занять дозволяє: ручну модифікацію розкладу перетягуванням drag & drop; складання кількох розкладів і вибір кращого; консолідацію розкладів і оптимізацію за одним з критеріїв; враховувати побажання і можли-

вості викладачів, груп студентів, приміщень; перевіряти на допустимість при складанні розкладу (за показниками: тип приміщення / тип заняття, ємність приміщення / кількість студентів в групі); вибирати довільну періодичність розкладу; вести

облік паралельних занять, розбиття на підгрупи і потоки лекцій; регулювати максимально допустиму кількість занять в день для групи студентів або викладача при складанні розкладу; вести облік змін; оперативно резервувати приміщення.

Для автоматичної генерації розкладу навчального процесу дана програмна система використовує алгоритм розв'язання задачі, запропонований співробітниками лабораторії. № 68 «Теорії розкладів і дискретної оптимізації» Інституту проблем управління ім. В.А Трапезникова РАН.

Система орієнтована на російськомовну аудиторію і використовується переважно в російських ВНЗ. Крім того, необхідно відмітити, що компанія-постачальник ERP-систем "1С" підпадає під дію економічних санкцій згідно Указу Президента України №133/2017. Вартість ліцензії – 70000 руб.

4. Програма "Ректор-ВУЗ" [11].

Розклад занять можна складати в автоматичному, ручному або змішаному режимі. При складанні розкладу в автоматичному режимі програма враховує всі сформульовані вимоги до розкладу. При складанні розкладу в ручному режимі програма підказує можливі варіанти розстановки занять обраного викладача, можливі варіанти заповнення порожніх клітин в розкладі групи, стежить за кількістю місць в аудиторіях. Розклад занять груп і викладачів можна зберегти у форматах Microsoft Word, Excel або HTML.

Дана система використовується переважно в російських ВНЗ, але має переклад на українську мову.

Серед розглянутих систем програма "Ректор-ВУЗ" має найменший функціонал, оскільки не забезпечує врахування паралельних занять, поділу на підгрупи при складанні розкладу, оперативного резервування приміщень, врахування побажань викладачів та студентських груп тощо. Вартість ліцензії – 12000 руб на рік (оренда на 3 комп'ютера).

Порівняльний аналіз функціональних можливостей систем «1С: Автоматизоване складання розкладів. Університет», «Ректор-ВНЗ» та «Галактика» наведено у [16].

Постановка завдання. Задачу генерації розкладу навчального процесу у форма-

лізованому вигляді можна представити наступним чином:

У семестрі необхідно прочитати n навчальних дисциплін S_1, S_2, \dots, S_n ; для кожного $i, i = \overline{1..n}$ дисципліна S_i складається з lec_i лекцій, $pract_i$ практичних занять і семінарів й lab_i лабораторних робіт. Заняття s проводиться визначеним викладачем pr_j . Є gr навчальних планів G_1, G_2, \dots, G_{gr} , які являють собою набір навчальних дисциплін, що читаються групі студентів G_{rk} . Це означає, що дисципліни в $G_{jk}, k = \overline{1..gr}$ повинні бути рознесені у часі. ts_max – кількість навчальних періодів (пар), r_max_k – максимальна кількість занять, яку можна запланувати в період t_k (кількість аудиторій, доступних у період t_k).

Необхідно розподілити заняття по всім курсам у межах певної кількості аудиторій і періодів часу, тобто

знайти таке $s_{ir}^{ijk} (i = \overline{1..d}, j = \overline{1..pr}, k = \overline{1..gr}, t = \overline{1..ts}, r = \overline{1..r_max})$, що

$$\forall i = \overline{1..d} \sum_{t \in Ts, r \in R} \{s_{ir}^{ijk} | j \in Pr, k \in Gr, t \in Ts, r \in R\} = lec_i + pract_i + lab_i$$

розподіл усіх занять;

$$\forall t = \overline{1..ts_max} \sum_{j \in Pr, k \in Gr, t \in Ts, r \in R} \{s_{ir}^{ijk} | i \in D, j \in Pr, k \in Gr, t \in Ts, r \in R\} \leq r_max_t$$

використання обмеженого аудиторного фонду;

$$\forall t = \overline{1..ts_max} \forall r = \overline{1..r_max} \sum_{i \in D, j \in Pr, k \in Gr, t \in Ts} \{s_{ir}^{ijk} | i \in D, j \in Pr, k \in Gr, t \in Ts\} \leq 1$$

– запобігання накладок по аудиторії;

$$\forall t = \overline{1..ts_max} \forall k = \overline{1..gr}$$

$$\sum_{i \in D, i \in G_k, j \in Pr, t \in Ts, r \in R} \{s_{ir}^{ijk} | i \in D, i \in G_k, j \in Pr, t \in Ts, r \in R\} \leq 1$$

– запобігання накладок по групі;

$$\forall t = \overline{1..ts_max} \forall j = \overline{1..pr}$$

$$\sum_{i \in D, k \in Gr, t \in Ts, r \in R} \{s_{ir}^{ijk} | i \in D, k \in Gr, t \in Ts, r \in R\} \leq 1$$

– запобігання накладок по викладачу;

$$\forall i = \overline{1..d} \forall k = \overline{1..ts} s_{ijk} \in \{0, 1\}$$

$$s_{ir}^{ijk} = \begin{cases} 1, & \text{якщо } d_i \text{ призначено на пару } t, \\ 0, & \text{в іншому випадку} \end{cases}$$

Загальна класифікація вимог до розкладу в університеті наведена на рис. 7.

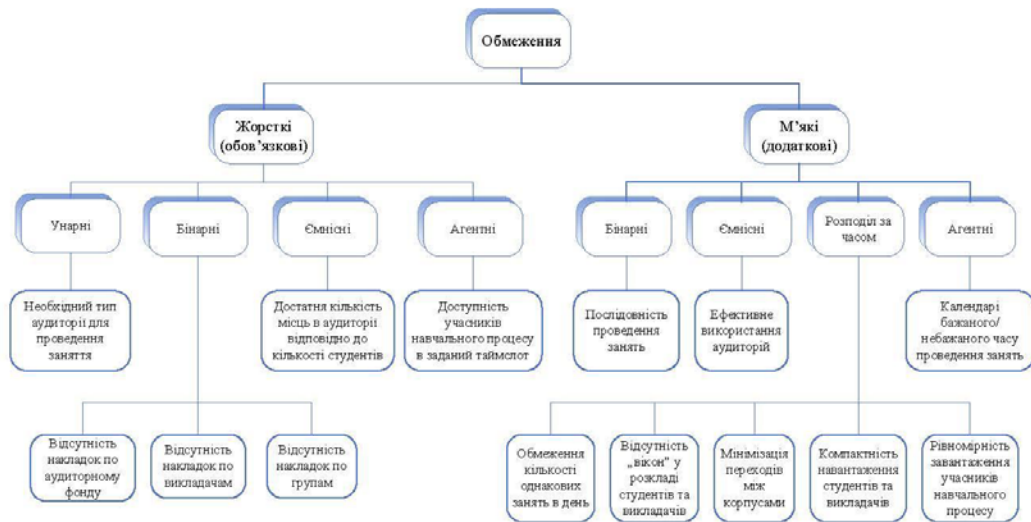


Рис. 7 – Загальна класифікація обмежень до розкладу навчального процесу

Виклад основного матеріалу дослідження.

Методи, що використовуються.

Один із ефективних метаевристичних методів, що застосовують для розв'язання даної задачі, – метод мурашиної колонії.

Метод мурашиної колонії – це ймовір-

нісна техніка розв'язання обчислювальних задач, яка може бути зведена до пошуку кращих шляхів за допомогою графів з використанням штучних мурах, які наслідують поведінку колонії природних мурах.

Поеднаємо даний метод з генетичним алгоритмом.

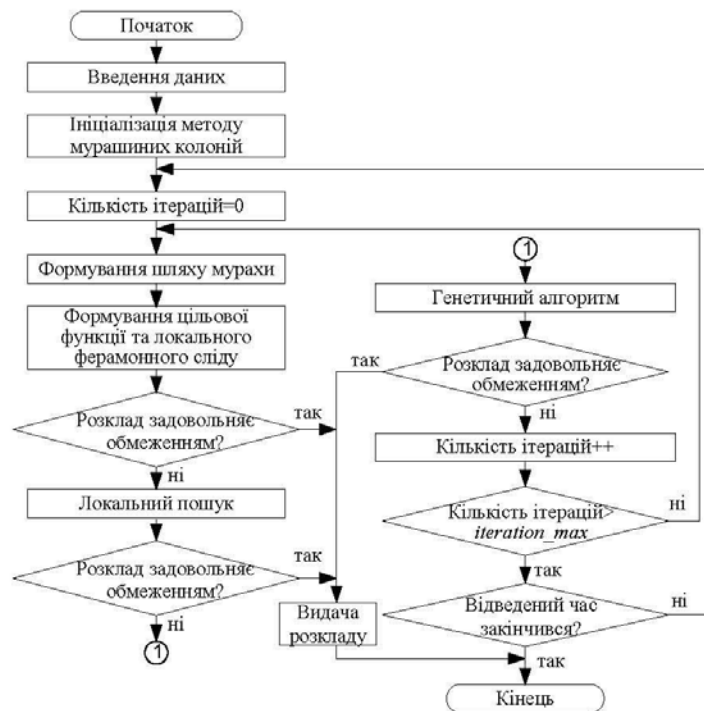


Рис. 8 – Схема роботи алгоритма

Використання генетичного алгоритму дозволяє зменшити час роботи алгоритму та збільшити ймовірність попадання в глобальний оптимум. Схема роботи алгоритму зображена на рис. 8.

Робота системи.

Основні користувачі системи автоматизованої генерації розкладу – посадові особи

навчально-методичних підрозділів вищих навчальних закладів, відповідальні за формування розкладу. Для доступу до інформації, що зберігається в системі, та для розмежування прав доступу до функціональності необхідно пройти авторизацію.

Робота програмної системи автоматизованого складання розкладу навчального процесу складається з наступних етапів:



Рис. 9 – Загальна схема автоматизованого складання розкладу навчального процесу

На початку роботи з системою необхідно ввести довідкову інформацію, яка необхідна при створенні розкладу. Існує два типи довідників: постійна інформація, яка може потребувати незначного редагування, та оперативна семестрова інформація.

Постійні дані в системі зберігаються в sql-базі даних, доступ до якої реалізується за допомогою sql-запитів. Схема бази даних системи наведена на рис. 10.

Створення та редагування постійних довідників може здійснюватися вручну або за допомогою функції імпорту з зовнішніх xml-файлів. Оперативна семестрова інформація завантажуються лише за допомогою імпорту з зовнішніх xml-файлів.

Можливі такі опції імпорту: імпорт – в цьому випадку відбувається видалення і повторне створення довідника з додаванням імпортованих даних; імпорт з оновленням – відбувається оновлення вже наявних і додавання відсутніх в довідниках даних;

Складання розкладу проходить в автоматичному, ручному або змішаному режимах.

При ручному режимі складання розкладу вибирається відповідне заняття із дерева занять та методом Drag & Drop переноситься у сітку розкладу. При цьому система підсвічує доступні та заборонені таймслоти. Розмістити конфліктуючі заняття в один час або одній аудиторії неможливо. Червоним кольором виділяються заборонені таймслоти, зеленим – такі самі заняття, уже розміщені в сітці розкладу, сірим – не рекомендовані, блакитним – рекомендовані варіанти. Рекомендовані та не рекомендовані варіанти розміщення визначаються типом занять, календарями завантаження учасників навчального процесу та даними про заняття, що містяться в імпортованих файлах.

Автоматична генерація виконується відповідно до розглянутого вище методу.

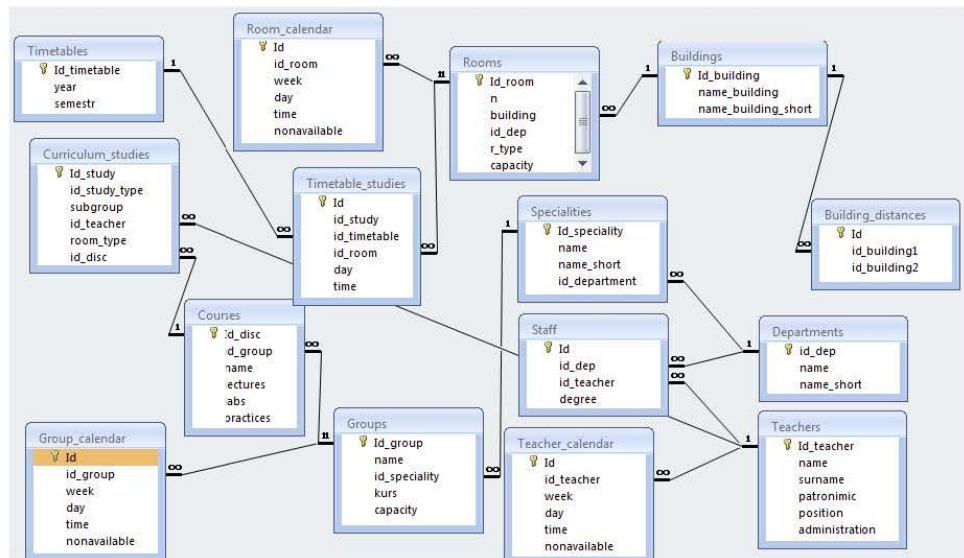


Рис. 10 – Схема бази даних

а)

б)

Дня	Час	101-ТН	102-ТН
Понеділок	08:30 - 09:50		Дискретна математика, Лаб., Дадченко П.Д., 303п
	10:00 - 11:20	Вища математика, Лекція, Бутенко І.К., 301п	Вища математика, Лекція, Бутенко І.К., 301п
	11:50 - 13:10	Дискретна математика, Лаб., Дадченко П.Д., 303п	
	13:20 - 14:40		
	14:50 - 16:10		
	16:20 - 17:40		
Вівторок	08:30 - 09:50	Дискретна математика, Лаб., Дадченко П.Д., 304п	Вища математика, Лаб., Бутенко І.К.(1), 102к
	10:00 - 11:20	Фізика, Лекція, Куш А.Р., 100п	Фізика, Лекція, Куш А.Р., 100п
	11:50 - 13:10	Програмування, Лаб., Гикало Р.В., 101п	Вища математика, Лаб., Сошов Д.К.(2), 102к
	13:20 - 14:40		
	14:50 - 16:10		
	16:20 - 17:40		
Середа	08:30 - 09:50		Програмування, Лаб., Гикало Р.В., 101п
	10:00 - 11:20	Вища математика, Лекція, Бутенко І.К., 301п	Вища математика, Лекція, Бутенко І.К., 301п
	11:50 - 13:10	Фізкультура	Фізкультура
	13:20 - 14:40		
	14:50 - 16:10		
	16:20 - 17:40		
Четвер	08:30 - 09:50	Дискретна математика, Лаб., Дадченко П.Д., 304п	Фізика, Лаб., Радис Н.Є., 410Ф
	10:00 - 11:20	Вища математика, Лаб., Бутенко І.К.(1), 101п/Вища математика, Лаб., Сошов Д.К.(2), 102к	Дискретна математика, Лаб., Дадченко П.Д., 303п
	11:50 - 13:10		
	13:20 - 14:40		
	14:50 - 16:10		
	16:20 - 17:40		
П'ятниця	08:30 - 09:50	Фізика, Лаб., Радис Н.Є., 410Ф	Дискретна математика, Лаб., Дадченко П.Д., 303п
	10:00 - 11:20	Програмування, Лекція, Гикало Р.В., 100п	Програмування, Лекція, Гикало Р.В., 100п
	11:50 - 13:10		

в)

Рисунок 11 – Екранні форми системи: а) редагування даних про аудиторії; б) генерація розкладу в ручному режимі; в) результат автоматичної генерації розкладу

Висновки

У роботі розглядається побудова програмної системи для автоматизації складання розкладу навчального процесу в університеті. Використання даної системи в університеті дозволить полегшити та підвищити ефективність роботи диспетчерів,

які займаються складанням розкладу, та якість навчального процесу за рахунок контролю відсутності конфліктів учасників навчального процесу та максимального врахування побажань учасників навчального процесу при складанні розкладу.

Список літератури:

1. Бурнасов П.В. Математическая постановка задачи составления расписания занятий. *Вестник Иркутского государственного технического университета*. 2014. № 4(87). С. 12–18.
2. Бульонков М.А., Емельянов П.Г., Пак Е.В. К стандартизации данных для составления расписания в учебных заведениях. *Открытое образование*. 2010. № 3. С. 45–57.
3. Дворянкин А.М. Чалышев В.С. Обзор методов составления расписания вузов. *Изв. ВолгГТУ. Серия Актуальные проблемы управления, вычислительной техники и информатики в технических системах*. 2011. Вып. 11. № 9. С. 110-113.
4. Томашевський В.М., Новіков Ю.Л., Камінська П.А. Складання розкладів занять у дистанційних системах навчання. *Вісник НТУУ КПІ. Серія Інформатика, управління та обчислювальна техніка*. 2011. Вип. 52. С. 118-130.
5. Leite Nuno, Melicio Fernando, Rosa Agostinho. A fast simulated annealing algorithm for the examination timetabling problem. *Expert Systems with Applications*. 2018. № 122.
6. Гусейн А. Разработка механизма интеллектуального управления отношениями «студент-преподаватель» в пространстве виртуального образования с применением нейронных сетей. *Open education*. V. 22. № 5. 2018. P. 94-103.
7. UniTime | University Timetabling. URL: www.unitime.org.
8. Система для составления расписаний в вузах. URL: <https://ru.osvita.ua/vnz/53319/>
9. Галактика: Расписание учебных занятий. URL: http://galaktika.ua/manuals/ruz/index.html?ruz_raspisanie.htm (дата звернення: 01.03.2020).
10. IC: Автоматизированное составление расписания. Университет. URL: https://solutions.ic.ru/catalog/asp_univer (дата звернення: 05.03.2020).
11. Расписание занятий: «Ректор-ВУЗ». URL: <http://rector.spb.ru/raspisanie-vuz-4u.php> (дата звернення: 01.03.2020).
12. Blum C. Ant colony optimization: Introduction and recent trends // *Physics of Life Reviews* 2, 2005. С. 353–373.
13. Бойко О.М. Еволюційна технологія розв'язування задачі складання розкладів навчальних занять. *Штучний інтелект*. 2006. № 3. С. 341 – 348.
14. Астахова И.Ф., Фирас А.М. Составление расписания учебных занятий на основе генетического алгоритма. *Вестник ВГУ, серия: Системный анализ и информационные технологии*. 2013. № 2. С. 93 – 99.
15. Thepphakorn T, Pongcharoen P, Hicks C. An ant colony based timetabling tool. *International Journal of Production Economics*. 2014, № 149(3). С. 131-144.
16. Юрчак І.Ю., Москович Т.Р. Дослідження генетичних алгоритмів та застосування їх в автоматизованій системі розподілу навантаження для викладачів і студентів. URL: <http://eom.lp.edu.ua/sntk/doc/ksm2018/moskovytech.pdf>.

**В.3 Тези доповіді на 72-й науковій конференції професорів, викладачів,
наукових працівників, аспірантів та студентів Національного
університету «Полтавська політехніка імені Юрія Кондратюка»**

Міністерство освіти і науки України
Північно-Східний науковий центр НАН України та МОН України
Національний університет
«Полтавська політехніка імені Юрія Кондратюка»

Тези

**72-ої наукової конференції професорів,
викладачів, наукових працівників,
аспірантів та студентів університету,
присвяченої 90-річчю
Національного університету
«Полтавська політехніка імені Юрія Кондратюка»**

Том 1

21 квітня – 15 травня 2020 р.

Полтава 2020

УДК 004.91:378.145

*Гайтан О.М., старший викладач,
Національний університет
«Полтавська політехніка імені Юрія Кондратюка»*

ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ МЕТОДУ МУРАШИНИХ КОЛОНІЙ В ЗАДАЧІ КОМП'ЮТЕРНОЇ ГЕНЕРАЦІЇ РОЗКЛАДУ НАВЧАЛЬНОГО ПРОЦЕСУ ВНЗ

Задача комп'ютерної генерації розкладу відноситься до задач автоматизації діяльності вищого навчального закладу. Незважаючи на широке використання інформаційних технологій, у багатьох вищих навчальних закладах розклад до сих пір складається вручну і комп'ютер використовується лише для візуалізації та розповсюдження згенерованого вручну розкладу. Тому задача автоматичної генерації комп'ютерного розкладу навчального процесу ВНЗ є актуальною.

Проблема складання розкладу навчального процесу в університеті полягає у розподілі набору занять згідно навчальних планів в межах заданої кількості аудиторій та періодів часу (пар). Основна відмінність розкладу університету від середньої школи полягає в тому, що університетські курси можуть мати спільних студентів, тоді як шкільні класи – це множини учнів, що не перетинаються. Якщо два заняття мають спільних студентів, вони конфліктують, і тому не можуть бути заплановані на той самий період. Крім того, необхідно передбачити можливість поділу академічних груп на підгрупи та об'єднання лекційних занять у потоки.

В проблемі складання розкладу університету важливу роль відіграє наявність аудиторій (їх розмір, тип і обладнання, належність до відповідної кафедри), тоді як у середній школі цим нехтують, оскільки в більшості випадків можна припустити, що кожен клас має свій кабінет і окремі вимоги висуваються лише до занять з іноземної мови, інформатики та фізкультури.

Проблема складання розкладу іспитів полягає в розподілі певної кількості іспитів протягом заданої кількості часу. Складання розкладу іспитів аналогічно складанню розкладу навчального процесу, ці два процеси відрізняються м'якими вимогами, що висуваються до розкладу.

Розклади навчальних занять складаються, як правило, на кожен семестр відповідним відділом в організаційній структурі університету та мають періодичність 2 тижні. Розклади екзаменів складаються на одну заліко-екзаменаційну сесію без повторень.

Завдання складання розкладу відноситься до NP-повних задач багатокритеріальної комбінаторної оптимізації. Якість складання розкладу навчального процесу визначається кількістю порушень обмежень, що висуваються до такого розкладу. Цільова функція розраховується на основі набору штрафів (штрафних функцій), які виникають при невиконанні заданих обмежень в розкладі.

На даний час для розв'язання задачі складання розкладу в університеті дуже поширене застосування метаевристичних методів таких як метод імітації відпалу, генетичні алгоритми, метод мурашиних колоній тощо.

Один із ефективних метаевристичних методів, що застосовують для розв'язання даної задачі, – метод мурашиної колонії.

Метод мурашиної колонії – це ймовірнісна техніка розв'язання обчислювальних задач, яка може бути зведена до пошуку кращих шляхів за допомогою графів з використанням штучних мурах, які наслідують поведінку колонії природних мурах. Даний метод належить до класу ітераційних алгоритмів і ґрунтується на ідеї послідовного наближення до оптимального розв'язку.

Загальна схема алгоритму:

1. Для кожного заняття i послідовно вибирається місце j в розкладі, використовуючи ймовірності переходу. Мураха рухатиметься від вузла до вузла з ймовірністю:

$$P_{ij} = \frac{(\tau_{i,j}^\alpha)(\eta_{i,j}^\beta)}{\sum (\tau_{i,j}^\alpha)(\eta_{i,j}^\beta)}, \quad (1)$$

де $\tau_{i,j}^\alpha$ – кількість феромонів у вершині i, j (накопичена статистична інформація про якість вибору для заняття i позиції j);

α – параметр, який контролює вплив $\tau_{i,j}^\alpha$;

$\eta_{i,j}^\beta$ – привабливість вершини (евристична інформація про зручність розміщення заняття i позиції j в розкладі);

β – параметр, який контролює вплив $\eta_{i,j}^\beta$.

2. Здійснюється перерахунок (оновлення) феромонного сліду (вироблення і випаровування феромону) за формулами:

$$\tau_{i,j} = (1 - \rho)\tau_{i,j} + \Delta\tau_{i,j}, \quad (2)$$

де $\tau_{i,j}$ – кількість феромону на вершині i, j (глобальний феромонний слід);

ρ – швидкість випаровування феромону;

$\Delta\tau_{i,j} = 1/(1+F_i)$ – кількість відкладеного феромону (локальний феромонний слід);

F_i – функція штрафів згенерованого розкладу.

Найкращим вважається рішення з найбільшим значенням локального феромонного сліду.

Перевагами метода мурашиних колоній є гарантування збіжності до оптимального рішення, а також стохастичність, тобто випадковість пошуку, за рахунок чого виключається можливість зациклення в локальному оптимумі.

Разом з тим у [1] відмічається невизначеність часу збіжності при тому,

що збіжність гарантується; сильна залежність результатів роботи методу від початкових параметрів пошуку, які підбираються експериментально.

Для зменшення часу збіжності та знаходження глобального оптимуму поєднаємо метод мурашиних колоній з локальним пошуком та генетичним алгоритмом, які будуть запускатися кожного разу після чергової ітерації методу мурашиних колоній і використовувати згенеровані мурахами розклади у якості вхідних даних. Для знаходження параметрів методу мурашиних колоній використаємо метод деформованого багатогранника.

Інтеграція локального пошуку та генетичних алгоритмів в метод мурашиних колоній забезпечить підвищення ефективності даного методу за рахунок зменшення часу роботи алгоритму та підвищення ймовірності знаходження глобального оптимуму, тобто розкладу, що буде задовольняти усім жорстким та максимальній кількості м'яких обмежень.

Література

1. Устенко С.В., Бірко О.О. Використання методу мурашиної колонії для розв'язання оптимізаційних задач // Науковий вісник НЛТУ України, 2015. Вип. 25, вип. 3. С. 351-359.
2. Thepphakorn T, Pongcharoen P, Hicks C. An ant colony based timetabling tool // *International Journal of Production Economics*, 2014, № 149(3). С. 131-144.
3. Титов Ю.П. Модификации метода муравьиных колоний для разработки программного обеспечения решения задач многокритериального управления поставками // *Современные информационные технологии и ИТ-образование*, 2017. Том 13. № 2. С. 64-74.
4. Chernigovskiy A., Kapulin D., Noskova E., etc. Production scheduling with ant colony optimization // *IOP Conference Series: Earth and Environmental Science*, 2017. Vol. 87.