

DOI: <https://doi.org/10.32836/2521-6643-2019-1-57-9>
УДК 004.942:004.55:519.17

Т. В. Січко, кандидат технічних наук,
доцент кафедри прикладної математики
і теорії систем управління Донецького
національного університету
імені Василя Стуса

К. В. Смоктей, кандидат економічних наук,
доцент кафедри прикладної математики
і теорії систем управління Донецького
національного університету
імені Василя Стуса

А. О. Ткачук, студент Донецького
національного університету
імені Василя Стуса

ПРИКЛАДНІ АСПЕКТИ РОЗРАХУНКУ СТРУКТУРНО- ТОПОЛОГІЧНИХ ХАРАКТЕРИСТИК СИСТЕМ

Розроблено односторінковий веб-додаток для обчислення структурно-топологічних характеристик систем, які впливають на різноманітні аспекти розвитку суспільства. Наведено програмну реалізацію алгоритмів обчислень структурно-топологічних характеристик систем, представлено стартовий інтерфейс розробленого додатка. Роботу додатка продемонстровано на прикладі розрахунку структурно-топологічних характеристик організаційної структури компанії, яка розробляє ігри. Окреслено перспективи подальшого вдосконалення розробленого веб-додатка.

Ключові слова: *графові моделі; структурний аналіз; структурно-топологічні характеристики; односторінковий веб-додаток.*

Разработано одностраничное веб-приложение для вычисления структурно-топологических характеристик систем, которые влияют на различные аспекты развития общества. Приведена программная реализация алгоритмов вычисления структурно-топологических характеристик систем, представлено стартовый интерфейс разработанного приложения. Работу приложения продемонстрировано на примере расчета структурно-топологических характеристик организационной структуры компании, ко-

© Т. В. Січко, К. В. Смоктей, А. О. Ткачук, 2019

торая разрабатывает игры. Определены перспективы дальнейшего совершенствования разработанного веб-приложения.

Ключевые слова: графовые модели; структурный анализ; структурно-топологические характеристики; одностраничное веб-приложение.

Often happens that for the analysis of the system structural properties on a graphmetric characteristics are inconsequential, at the same such topological nature parameters as connectivity, compactness and continuity are valuable. These characteristics allow optimizing structural links between elements of any system according to different criteria using structural-topological analysis of systems, which play important role for social and economic development of any country.

In this paper we analyzed the existing applications for graph processing and took into account purchasing costs (that can be licensed use) and concluded that it is advisable to develop a web application that would be an open source code (free to use) and to perform the calculation of the basic characteristics for the structural analysis of the systems that are described by the graph model, for instance: connectivity, continuity, redundancy, compactness, hierarchy, diameter, uneven distribution of bonds.

Applied aspects of the calculation of systems structural-topological characteristics are considered and an approach as for the automation of systems structural-topological characteristics calculations is developed. This approach allows to synthesize the corresponding web application with its publication in the Internet for further use for the decision-making process concerning the optimization of the structural relations of the investigated system.

Single-page format was chosen as one of the most widespread and most convenient to use for the web-application. A single-page application is a web application that uses a single HTML document as a shell for all web pages and organizes user interaction through HTML, CSS, and JavaScript files that are dynamically loaded. Developed application is connected to the version control system Git, downloaded to Git Hub and connected to Git Hub Pages. As an example, the organizational structure of the game developing company is considered and needed calculations of the system characteristics are done. The conclusions from this research and the prospects for further improvement of the developed one-page application are presented.

Key words: graph models; structural analysis; structuraltopological characteristics; one-pageapplication.

Постановка проблеми. Питання управління складними системами обумовлюють дослідження, в яких використовується математична теорія, що дає

можливість поєднати математичні властивості соціально-економічних систем та їх економічний зміст. Такою теорією є теоретико-графове моделювання економічних об'єктів і процесів виробництва до якого останнім часом виявляється більша зацікавленість. Так, теоретико-графові моделі не лише відображають структуру системи та зв'язки між елементами, але й використовуються для прийняття оптимальних управлінських рішень у моделюванні соціально-економічних системи різної складності.

Аналіз останніх досліджень і публікацій. Графовий підхід до моделювання складних систем розглядається у працях І. М. Мельника, О.М. Парубця, О. Г. Климко, С. П. Кобця, І. І. Скрильник, А. А. Кочкарова, А. М. Штангрета, К. В. Ніколаєва, В. В. Кобійчука та ін.

Теоретико-графові моделі застосовуються для розв'язання великого класу оптимізаційних задач, а саме: задачі розподілу ресурсів, задачі управління запасами, задачі планування та розміщення тощо. Розв'язанню таких задач присвячено дослідження провідних українських учених І. В. Сергієнка, Н. З. Шора, Ю. Г. Стояна, О. О. Ємця, І. М. Ляшенка, С. В. Яковлева.

Алгоритмам розв'язання цих задач та їх програмній реалізації різними мовами програмування (від VBA до C++) присвячено фундаментальні праці Г. А. Черноморова [1, 28–73] та Р. Седжвика [2, 231–304], проте більшість вітчизняних авторів спрямовують свої дослідження на підтримку розв'язання екстремальних задач на графах за допомогою надбудови Solver (Пошук рішення) MS Excel [3–6].

Для розв'язання задач, об'єктами яких є графи, використовують пакети аналітичних обчислень: Mathematica, MATLAB, Mathcad, Maple, що дають можливість виконувати аналітичні символічні перетворення [7].

Існують також готові програмні рішення, які реалізують безліч **алгоритмів** для обробки графів: пошук мінімального шляху різними способами, пошук ейлерових і гамільтонових маршрутів, визначення хроматичного числа, пошук мінімального остовного дерева, визначення максимального потоку, перевірка на зв'язність, пошук радіуса і діаметра графа, перевірка чи є граф деревом, перевірка на планарність, пошук критичного шляху, пошук циклів, пошук максимального повного підграфа [8–10].

Виникають ситуації під час аналізу структурних властивостей системи на графі, коли метричні характеристики несуттєві, а важливими є характеристики зв'язності, компактності та неперервності, тобто характеристики топологічного характеру. Ці характеристики уможливають оптимізацію структурних зв'язків між елементами будь-якої системи за різними критеріями з використанням структурно-топологічного аналізу систем.

Автори проаналізували наявні додатки для обробки графів і дійшли висновку, що з урахуванням вартості покупки (ліцензійного використання)

доцільно здійснити розробку веб-додатка, який буде з відкритим програмним кодом (безкоштовним для використання) і реалізує розрахунок базових характеристик для структурного аналізу систем, що описуються графовою моделлю, а саме: зв'язності, неперервності, надмірності, компактності, ієрархічності, діаметра, нерівномірності розподілу зв'язків.

Мета статті – розглянути прикладні аспекти розрахунку структурно-топологічних характеристик систем і визначити підхід до автоматизації обчислень структурно-топологічних характеристик систем, що, зі свого боку, дає можливість синтезувати відповідний веб-додаток із його публікацією для подальшого використання в мережі Internet та приймати рішення щодо оптимізації структурних зв'язків досліджуваної системи.

Виклад основного матеріалу. Для реалізації веб-додатка було обрано односторінковий формат як один із найпоширеніших та найзручніших у використанні. Односторінковий додаток – це веб-додаток, який використовує єдиний HTML-документ як оболонку для всіх веб-сторінок і організовує взаємодію з користувачем через HTML, CSS, Java Script файли, що динамічно завантажуються.

Перевагами односторінкових веб-додатків є:

- висока швидкість роботи, оскільки додаток не оновлює всю сторінку, а тільки потрібну частину, що істотно підвищує швидкість;
- висока швидкість розробки, оскільки готові бібліотеки і фреймворки дають потужні інструменти для розробки веб-додатків. Над проектом можуть паралельно працювати back-end і front-end розробники. Завдяки чіткому поділу вони не заважатимуть один одному;
- мобільність, оскільки додаток дає змогу легко розробити мобільний додаток на основі готового коду [11].

Архітектурно розроблений односторінковий додаток є сторінкою html з одним контейнером div, логікою відображення контенту, в якому керує скрипт мовою javascript:

```
<body>
  <div id="root">
    <script src="Topological_characteristics.js"></script>
  </div>
</body>
```

Рис. 1. Місце підключення керуючого скрипту до html сторінки

Це дає можливість установлювати розміри всіх елементів сторінки відповідно до розміру вікна.

Скрипт може змінювати зміст html-сторінки завдяки використанню об'єктної моделі документа (DOM). Об'єктна модель документа – специфікація прикладного програмного інтерфейсу для роботи зі структурованими документами (як правило, документами XML). Визначається ця специфікація консорціумом W3C.

Разом із поширенням та розвитком веб-технологій і веб-переглядачів почали з'являтися різні, часто несумісні інтерфейси роботи із HTML-документами в інтерпретаторах Java Script, вбудованих у веб-переглядачі. Це спонукало World Wide Web Consortium (W3C) узгодити та визначити низку стандартів, які отримали назву W3C Document Object Model (W3C DOM). Специфікації W3C не залежать від платформи або мови програмування [12].

Логіка обчислень структури реалізована у класі MyGraph (рис. 2). Структура зберігається у вигляді незваженого неорієнтованого графа.

```
class MyGraph{
|   constructor(n) {
|       this.n=n;//Кількість вершин
|       this.adjacency=create2DimArray([this.n,this.n]);//Матриця суміжності
|       this.distance=create2DimArray([this.n,this.n]);//Матриця відстаней
|       this.connection=create2DimArray([this.n,this.n]);//Матриця зв'язності
|       for(var i=0;i<n;i++){
|           for(var j=0;j<n;j++){
|               this.adjacency[i][j]=0;
|               this.distance[i][j]=0;
|               this.connection[i][j]=0;
|           }
|       }
|       this.redundancy=0;//Збитковість
|       this.redundancy2=0;//Квадратне відхилення заданого розподілу вершин від рівномірного
|       this.Q=0;//Абсолютна компактність
|       this.compactness=0;//Відносна компактність
|       this.d=0;//Діаметр системи
|       this.sigma=0;//Ступінь централізації
|   }
}
```

Рис. 2. Структура класу, що зберігає граф

Відповідні змінні зберігають показники графа. Кількість вершин і матриця суміжності вводяться користувачем, а всі інші показники обчислюються у процесі роботи. Наведемо назви й деякі реалізації методів для обчислення структурно-топологічних характеристик систем.

Обчислення структурної надлишковості (надмірності, збитковості), яка показує перевищення загальної кількості зв'язків над мінімально необхідною, використовується для непрямой оцінки економічності й надійності досліджуваних систем. Реалізацію зображено на рис. 3 у методі calc_redundancy.

Нерівномірність розподілу зв'язків, що характеризує недовикористання можливостей заданої структури в досягненні максимальної зв'язності, реалізовано в методі `calc_redundadncy2`.

```
calc_redundadncy() {
    var R=0;
    for (var i = 0; i <this.n; i++)
        for (var j = 0; j < this.n; j++)
            R += this.adjacency[i][j];
    R=R*0.5/(this.n-1)-1;
    this.redundancy=R;
}
```

Рис. 3. Фрагмент обчислення структурної надлишковості

Обчислення матриці зв'язності реалізовано в методі `calc_connection` (рис. 4).

```
calc_connection(){
    if (this.n > 0){
        var Ak=createNDimArray([this.n,this.n,this.n]);
        for (var i = 0; i < this.n; i++)
            for (var j = 0; j < this.n; j++)
                Ak[0][i][j] = this.adjacency[i][j];
        for (var k = 1; k < this.n; k++){
            for (var i = 0; i < this.n; i++)
                for (var j = 0; j < this.n; j++){
                    Ak[k][i][j] = 0;
                    for (var m = 0; m < this.n; m++)
                        Ak[k][i][j] += Ak[k - 1][i][m]*this.adjacency[m][j];
                }
        }
        for (var i = 0; i < this.n; i++)
            for (var j = 0; j < this.n; j++)
                this.connection[i][j] = 0;
        for (var k = 0; k < this.n; k++){
            for (var i = 0; i < this.n; i++)
                for (var j = 0; j < this.n; j++)
                    this.connection[i][j] += Ak[k][i][j];
        }
        for (var i = 0; i < this.n; i++)
            for (var j = 0; j < this.n; j++)
                if (this.connection[i][j] >= 1)this.connection[i][j]=1;
    }
}
```

Рис. 4. Фрагмент обчислення матриці зв'язності

Обчислення абсолютної та відносної структурної компактності реалізовано у методі `calc_compactness`. Наприклад, структурна компактність транспортних систем дає можливість отримати інформацію про ступінь використання дорожніх зв'язків за повного навантаження на систему (одночасне перевезення вантажу з усіх елементів системи в усі напрямки).

Обчислення діаметра структури реалізовано в методі `calc_d`. Діаметр структури визначає найкоротшу відстань між найвіддаленішими вершинами.

```
calc_compactness() {  
    var Q=0;  
    var Qmin=this.n*(this.n-1);  
    for (var i=0; i<this.n; i++)  
        for (var j=0; j<this.n; j++) {  
            Q=Q+this.distance[i][j];  
        }  
    this.Q=Q;  
    var Qrel=(Q/Qmin) -1;  
    this.compactness=Qrel;  
}
```

Рис. 5. Фрагмент обчислення абсолютної та відносної структурної компактності

Значення діаметра структури може характеризувати, скажімо, пропускну здатність системи. Економічний зміст даної характеристики виявляється під час оптимізації інформаційних, матеріальних, фінансових потоків.

```
calc_d() {  
    var max=this.distance[0][0];  
    var size = this.n;  
    for (var i = 0; i < size; i += 1)  
        for (var j = 0; j < size; j += 1)  
            if (max<this.distance[i][j]) max=this.distance[i][j];  
    this.d=max;  
}
```

Рис. 6. Фрагмент обчислення діаметра структури

Обчислення ступеня централізації структури реалізовано в методі `calc_sigma`. Цей показник змінюється в діапазоні від 0 до 1. Нуль відповідає абсолютно децентралізованій системі, одиниця – абсолютно централизованій.

Матриця відстаней обчислюється за допомогою алгоритму Флойда–Воршела, який являє собою динамічний алгоритм для знаходження найкоротших відстаней між усіма вершинами графа [13]. Він реалізований у методі класу `calc_distancematr` (рис. 8).

У додатку передбачено дві можливості введення матриці суміжності графа: з клавіатури та з файлу. Можливість уведення матриці суміжності з файлу реалізована з допомогою `FileAPI`. `FileAPI` – це набір інструментів `JavaScript` для роботи з файлами, він підтримується більшістю сучасних браузерів [14].

```
calc_sigma() {
    var size = this.n;
    var Z=new Array(size);
    for(var i=0;i<size;i++){
        var tmp=0;
        for(var j=0;j<size;j++){
            if(isFinite(this.distance[i][j])==false){
                tmp=Infinity;
                break;
            }
            tmp=tmp+this.distance[i][j];
        }
        Z[i]=this.Q/(2*tmp);
    }
    var Zmax=Z[0];
    for(var i=1;i<size;i++)
        if(Z[i]>Zmax) Zmax=Z[i];
    this.sigma=((size-1)*(2*Zmax-size))/(Zmax*(size-2))
}
```

Рис. 7. Фрагмент обчислення ступеня централізації структури

Можливість уведення з клавіатури реалізовано в 2 етапи: введенням кількості вершин та введенням матриці суміжності. Поле для введення матриці реалізовано як таблиця `select`-елементів, у кожному з яких є тільки два варіанти 0 або 1 (оскільки граф незважений). За замовчуванням усі елементи заповнюються нулями. За зміною будь-якого елемента змінюється елемент, симетричний йому відносно головної діагоналі (оскільки граф є неорієнтованим). Реалізовано можливість зберегти введену матрицю у файл, що дає можливість її змінити і знову завантажити для подальших розрахунків.


```

calc_distancematr(){
var size = this.n;
for (var i = 0; i < size; i += 1) {
for (var j = 0; j < size; j += 1) {
if (i == j) {
this.distance[i][j] = 0;
} else if (this.adjacency[i][j]==0) {
this.distance[i][j] = Infinity;
} else {
this.distance[i][j] = this.adjacency[i][j];
}
}
}
for (var k = 0; k < size; k += 1) {
for (var i = 0; i < size; i += 1) {
for (var j = 0; j < size; j += 1) {
if (this.distance[i][j] > this.distance[i][k] + this.distance[k][j])
this.distance[i][j] = this.distance[i][k] + this.distance[k][j];
}
}
}
}
}

```

Рис. 8. Фрагмент обчислення матриці відстаней

Розроблений додаток підключено до системи контролю версій git, завантажено на Git Hub [15] та підключено до Git Hub Pages.

Для прикладу розглянемо організаційну структуру компанії, яка розробляє ігри.

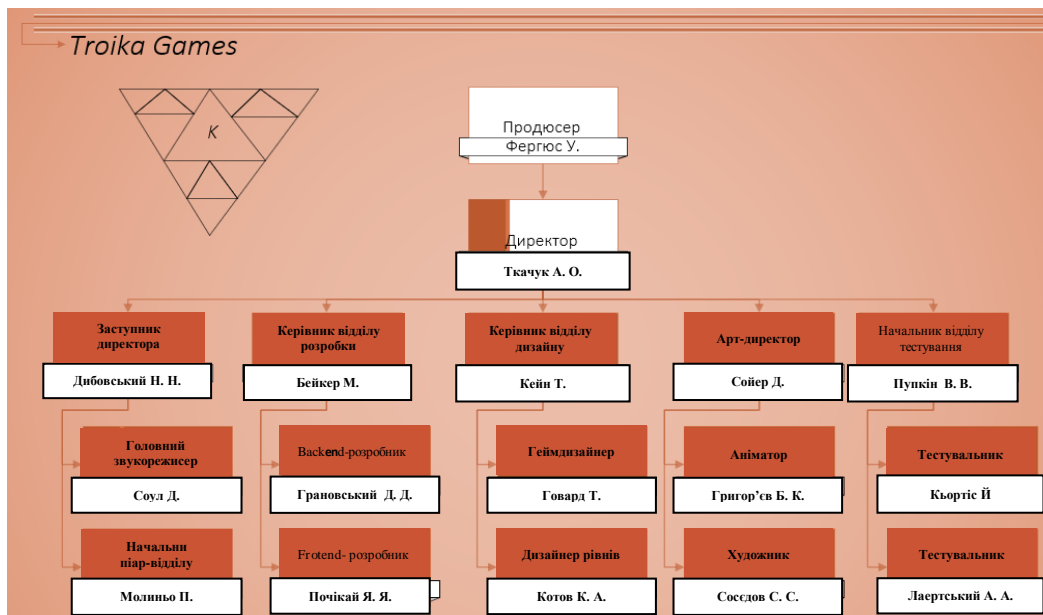


Рис 9. Організаційна структура управління компанії, яка розробляє ігри

Створимо текстовий файл, у першому рядку якого буде кількість вершин структури, а в наступних – матриця суміжності структури.

```
17
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0
0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
```

Рис. 10. Вміст файлу з матрицею суміжності структури

Запустимо на виконання розроблений односторінковий додаток. Стартовий інтерфейс додатка зображено на рис. 12.

Оберіть спосіб введення структури

Увести матрицю суміжності структури з клавіатури
Завантажити матрицю суміжності структури з файлу

Рис. 12. Стартовий інтерфейс розробленого односторінкового додатка

Оберемо варіант завантаження матриці суміжності структури з файлу.

Формат файлу:
Кількість вершин
Матриця суміжності

Приклад файлу:
5
0 1 1 1 0
1 0 1 1 1
1 1 0 1 0
1 1 1 0 1
0 1 0 1 0

Рис. 13. Інтерфейс для завантаження файлу

Натиснемо кнопку “Завантажити файл”. У наступному вікні оберемо файл, який було створено раніше, й натиснемо “відкрити”. Отримали розраховані топологічні характеристики організаційної структури компанії, яка розробляє ігри. Наведемо нижче деякі з них:

Надлишковість структури $(R)=0$

Квадратне відхилення заданого розподілу вершин від рівномірного $(\epsilon^2)\approx 31.765$

Абсолютна компактність структури $(Q)=772$

Відносна компактність структури $(Q_{\text{відн}})\approx 1.838$

Діаметр структури $(d)=4$

Рис. 14. Результати роботи розробленого односторінкового додатка

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі. Запропонований підхід дає можливість автоматизувати проведення аналізу організаційної структури управління та розрахунку структурно-топологічних характеристиках, що, зі свого боку, дає змогу приймати рішення з оптимізації структурних зв’язків досліджуваної системи.

Перспективи подальшого вдосконалення розробленого односторінкового додатка автори вбачають у розробці більш лаконічного дизайну та розширенні підтримки мобільних платформ, чого можна досить швидко досягти завдяки застосуванню мови дизайну Material Design [16]. Material Design використовує більше макетів на основі сітки, адекватних анімацій і переходів, доповнення та ефектів глибини, таких як освітлення й тіні.

Доцільно було б реалізувати можливість уведення структури шляхом її зображення у вигляді графа й можливість автоматичного зображення структури відповідно до її матриці суміжності. Цього можна швидко досягти завдяки бібліотеці з відкритим вихідним кодом, написаній на JS – cytoscape.js [17]. Cytoscape.js дає можливість легко відображати та маніпулювати детальними інтерактивними графами. Cytoscape.js легко інтегрується та підтримує мобільні версії браузерів.

Список використаних джерел:

1. *Черноморов Г. А.* Теория принятия решений. Новочеркасск: Известия вузов: Электромеханика, 2002. 276 с.
2. *Седжвик Р.* Фундаментальные алгоритмы на C++. Алгоритмы на графах. Санкт-Петербург: ООО “Диа Софт ЮП”, 2002. 496 с.
3. *Гліненко Л. К., Яковенко Є. І.* Розв’язання задач на графах за допомогою надбудови Solver MS Excel // Вісник Хмельницького національного університету. Технічні науки. 2013. № 5. С. 176–184.

4. Гліненко Л. К., Фаст В. М. Розв'язання задач комбінаторної оптимізації радіоелектронних систем у середовищі MS Excel Solver. URL: <http://science.lpnu.ua/sites/default/files/journal-paper/2017/jun/5120/25glinenkofast.pdf> (дата звернення: 8.04.2019).

5. Кузьмичов А. І. Математичне програмування в Excel: навч. посіб. Київ: Вид-во Європ. Ун-ту, 2005. 320 с.

6. Гліненко Л. К., Фаст В. М. Автоматизація розв'язання екстремальних задач на графах у конструкторському проектуванні РЕА // Вісник НТУУ "КПІ". Серія: "Радіотехніка", Радіоапаратобудування. 2013. № 54. С. 90–101.

7. Курсанов М. Н. Графы в Maple. Задачи, алгоритмы, программы. Москва: ФИЗМАТЛИТ, 2007. 168 с.

8. Сайт программы для визуализации графов "Графоанализатор". URL: <http://grafoanalizator.unick-soft.ru> (дата звернення: 8.04.2019).

9. Cytoscape: An Open Source Platform for Complex Network Analysis. URL: <https://cytoscape.org> (дата звернення: 10.04.2019).

10. Gephi – The Open Graph Viz Platform. URL: <https://gephi.org> (дата звернення: 10.04.2019).

11. Single Page Application (SPA) и Multi Page Application (MPA): преимущества и недостатки. URL: <https://merehead.com/ru/blog/single-page-application-vs-multi-page-application/> (дата звернення: 10.04.2019).

12. Об'єктна модель документа. URL: https://uk.wikipedia.org/wiki/Об%27єктна_модель_документа (дата звернення: 10.04.2019).

13. Floyd–Warshall algorithm. URL: https://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm (дата звернення: 10.04.2019).

14. FileAPI. URL: <https://javascript.ru/blog/brmaley-ee/fileapi> (дата звернення: 10.04.2019).

15. Репозиторій розробленого додатка на GitHub. URL: https://github.com/ArtiomTkachuk1/Topological_characteristics (дата звернення: 12.04.2019).

16. Material_Design. URL: https://en.wikipedia.org/wiki/Material_Design (дата звернення: 12.04.2019).

17. Cytoscape.js. URL: <http://js.cytoscape.org> (дата звернення: 12.04.2019).

References:

1. Chernomorov H. A. (2002), *Teoriya pryniatyia resheniy* [Decision Theory], Press University news: Electromechanics, Novocherkassk, 276 p. [Russia].

2. Sedzhvyk R. (2002), *Fundamental'nyye algoritmy na S++*. *Algoritmy na grafakh* [Fundamental Algorithms in C ++. Algorithms on graphs], Press Dia Soft YuP, Ltd, St. Petersburg, 496 p. [Russia].

3. Hlinenko L. K. and Yakovenko Ye. I. (2013), " *Rozv'yazannya zadach na hrafakh za dopomohoyu nadbudovy Solver MS Excel*" [Solving problems in graphs using the add-in Solver MS Excel], Collection of scientific works *Visnyk Khmel'nyts'koho natsional'noho universytetu. Tekhnichni nauky* [Bulletin of the Khmelnytsky National the university. Technical sciences, vol. 5, pp. 176–184 [Ukraine].

4. Hlinenko L. K. and Fast V. M. (2017), *Rozviazannia zadach kombinatornoi optymizatsii radioelektronnykh system u seredovyshechi MSEXcelSolver* [Solving problems of combinatorial optimization of radio electronic systems in the MS Excel Solver environment], available at: <http://science.lpnu.ua/sites/default/files/journal-paper/2017/jun/5120/25/glinenkofast.pdf> (date of the application: 8.04.2019) [Ukraine].

5. Kuz'mychov A. I. (2005), *Matematychni prohramuvannia v Excel* [Mathematical programming in Excel], Tutorial, Press European University, Kyiv, 320 p. [Ukraine].

6. Hlinenko L. K. and Fast V. M. (2013), "Avtomatyzatsiia rozviazannia ekstremalnykh zadach na hrafakh u konstruktorskomu proektuvanni REA" ["Automation of solving extreme problems in graphs in the design of REE"], Collection of scientific works *Visnyk NTUU "KPI"*, Series: "Radiotechnics", Radio Apparatus Building, vol. 54 pp. 90–101 [Ukraine].

7. Kirsanov M. N. (2007), *Grafy v Maple. Zadachi, algoritmy, programy* [Graphs in Maple. Tasks, algorithms, programs], Press FIZMATLIT, Moskva: FYZMATLYT, Moscow, 168 p. [Russia].

8. *Sayt programy dlya vizualizatsii grafov "Grafoanalizator"* [Website graph visualization program "Graph Analyzer"], available at: <http://grafoanalizator.unick-soft.ru> (date of the application: 8.04.2019).

9. Cytoscape: An Open Source Platform for Complex Network Analysis, available at: <https://cytoscape.org> (date of the application: 10.04.2019).

10. Gephi – The Open Graph Viz Platform, available at: <https://gephi.org> (date of the application: 10.04.2019).

11. *Single Page Application (SPA) i Multi Page Application (MPA): preimuschestva i nedostatki* [Single Page Application (SPA) and Multi Page Application (MPA): advantages and disadvantages], available at: <https://merehead.com/ru/blog/single-page-application-vs-multi-page-application> (date of the application: 10.04.2019).

12. *Obiektna model dokumenta* [Object model document], available at: https://uk.wikipedia.org/wiki/Объектна_модель_документа (date of the application: 10.04.2019).

13. Floyd–Warshall algorithm, available at: https://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm (date of the application: 10.04.2019).

14. FileAPI, available at: <https://javascript.ru/blog/brmaley-ee/fileapi> (date of the application: 10.04.2019).

15. *Repozytoriy rozroblenoho dodatka na GitHub* [Repository of the developed application on GitHub], available at: https://github.com/ArtiomTkachuk1/Topological_characteristics (date of the application: 12.04.2019).

16. Material_Design, available at: https://en.wikipedia.org/wiki/Material_Design (date of the application: 12.04.2019).

17. Cytoscape.js, available at: <http://js.cytoscape.org> (date of the application: 12.04.2019).