Медведєва Марія Олександрівна
к.п.н., доцент кафедри вищої математики
Уманського державного педагогічного університету імені Павла Тичини

ЗАСТОСУВАННЯ ТЕОРІЇ ГРАФІВ У МЕРЕЖЕВОМУ МОДЕЛЮВАННІ ІЗ ВИКОРИСТАННЯМ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В УМОВАХ ОСОБИСТІСНО ОРІЄНТОВАНОГО НАВЧАННЯ

У Національній стратегії розвитку освіти вказано, що пріоритетом розвитку освіти нашої держави є впровадження в освітній процес сучасних інформаційно-комунікаційних технологій, що мають забезпечувати вдосконалення навчально-виховного процесу, ефективність та доступність освіти в цілому, а також підготовку майбутніх фахівців до вступу в інформаційне суспільство [8].

Зміни, що відбуваються в системі освіти, обумовлені організацією навчально-виховного процесу на основі нових технологій, використання яких спрямоване не тільки на забезпечення формування бази фундаментальних професійних знань, умінь та навичок, але й сприяє розвитку особистості майбутнього фахівця, її творчої індивідуальності.

Застосування комп’ютерів в якості засобів навчання дискретної математики створює передумови для вдосконалення традиційних методик навчання. Перевага використання комп’ютера, як засобу навчання, в порівнянні з іншими технічними засобами навчання полягає в тому, що він одночасно є інформаційним, контролюючим і навчальним засобом, що є особливо важливим в умовах особистісно орієнтованої системи навчання дискретної математики.

В процесі дослідження та аналізу наукових положень та практики було встановлено, що для підтримки вивчення дискретної математики можна використовувати відповідні пакети математичних програм [2, 3, 5, 7, 6, 7, 9, 10, 11, 12, 13]. Серед них найбільш відомими є такі як Gran1, Gran-2D, Gran-3D, Mathematica, Maple, MatLab, MathCAD, застосовуються для підтримки процесу вивчення різних математичних дисциплін, зокрема, дискретної математики. Вони можуть бути використані під час лекцій у процесі вивчення математичних моделей з метою більш зрозумілого та доступного їх представлення (у вигляді графіків або ідеальних математичних об'єктів). За їх підтримки можна виконувати загальні перетворення виразів, проводити операції з дійсними та комплексними числами з наперед заданою точністю, виконувати символьні перетворення, графічні операції: побудову двовимірних та тривимірних графічних зображень, створення анімації тощо.

Під час вивчення теми «Мережеве планування» змістового модуля «Графи», для візуалізації понять, надання уявлення про практичне застосування, а інколи і для спрощення обрахунків ми запропонували студентам використати програмний засіб Графоаналізатор.

Графоаналізатор – це візуальне середовище для роботи з графами. Він не лише надає можливість створювати і обробляти графи, але й візуально відображати роботу алгоритмів. Середовище підтримує роботу з орієнтованими і простими графами, із зваженими і незваженими.

Процес створення і зміни графів інтуїтивно зрозумілий. Візуальне представлення є дуже зрозумілою формою представлення графа, також можна побачити результат роботи алгоритму у візуальній формі. Візуальне представлення можна зберегти у файлі зображення. Для більшої наочності можна додавати підписи до елементів графа, змінювати фон, налаштовувати зовнішній вигляд елементів графа.

Для редагування графа можна використовувати різні методи: візуально редагувати граф або редагувати матрицю суміжності графа.

Програма реалізує більшість алгоритмів для обробки графів починаючи із пошуку шляху і закінчуючи перевіркою на планарність. За допомогою графоаналізатора можна знайти мінімальний шлях, ейлерові і гамільтонові маршрути, визначити хроматичне число, перевірити на зв'язність, знайти ексцентриситет, радіус і діаметр графа, перевірити чи являється граф деревом, перевірити на планарність, знайти критичний шлях, цикли, максимальний повний підграф.

Середовище Графоаналізатор 1.3 надає користувачу багато допоміжних функцій для полегшення роботи: можливість збереження і загрузки графа з підтримкою збереження візуального представлення, можливість створення графа з матриці суміжності, швидке перетворення графа, налаштування виду графа, позначення вершин, завантаження підкладки графа, режим створення карти із заданим масштабом, режим конструктора.

Програму Графоаналізатор 1.3 можна використовувати для розв’язування багатьох задач, які можна звести до математичної моделі графів. Типові задачі, які можна розв’язувати:

- пошук мінімального шляху проїзду;
- пошук мінімальних затрат при наймі співробітників;
- пошук мінімальних затрат на прокладку проводки або комп’ютерної мережі;
- розподіл роботи між декількома працівниками;
- розрахунок пропускної здатності комп’ютерної або дорожньої мережі;
- пошук найдешевшого варіанту прокладки проводки;
- пошук найдешевшого варіанту з’єднання доріг;
- перевірка можливості з’єднання електронних елементів на платі;
- пошук методу розфарбування мапи мінімальним числом фарб;
- розв’язування задачі комівояжера.

Мережева модель являє собою графічне зображення плану виконання робіт проекту, що складається з відрізків (процесів роботи) і вузлів (подій), які відображають логічний взаємозв’язок усіх процесів. В основі мережевого моделювання лежить поняття ***графа***.

Теорія графів оперує поняттям шлях, який поєднує послідовність взаємозв’язаних ребер (дуг). Контур означає такий шлях у якого початкова вершина співпадає з кінцевою. *Мережевий графік* – це орієнтований граф без контурів.

У мережевому плануванні є два основних елементи – робота і подія.

Процес – це робота, дія, яка вимагає витрат ресурсів, або очікування, яке призводить до досягнення необхідного результату.

Фіктивний процес – це зв'язок між результатами (подіями), який не вимагає витрат часу та ресурсів.

Подія – це результат (проміжний або кінцевий) виконання одного або декількох попередніх процесів.

Шлях – це будь-яка неперервна послідовність (ланцюг) процесів і подій.

Критичний шлях – це шлях, який немає резервів та містить самі напружені роботи проекту. Процеси, які розміщенні на критичному шляху називають критичними.

Всі інші процеси є некритичними та мають резерви часу, які надають можливість пересувати термін їх виконання, не впливаючи на загальну тривалість виконання проекту.

Отже, критичний шлях має особливе значення у мережевому плануванні, оскільки процеси цього шляху визначають загальний цикл завершення проекту. Для скорочення тривалості проекту необхідно, в першу чергу, скоротити тривалість процесів, які лежать на критичному шляху.

При побудові мережевих моделей необхідно дотримуватися наступних правил:

1. Мережа зображується зліва направо, та кожна подія з більшим порядковим номером зображується, як правило, правіше попереднього. Загальний напрямок стрілок, що зображають процеси, також, в основному, повинні бути розміщенні зліва направо, при цьому кожний процес повинен виходити з події з меншим номером.
2. Дві сусідні події можуть бути об'єднані лише одним процесом. Для зображення паралельних робіт вводиться проміжна подія та фіктивний процес (рис.1).

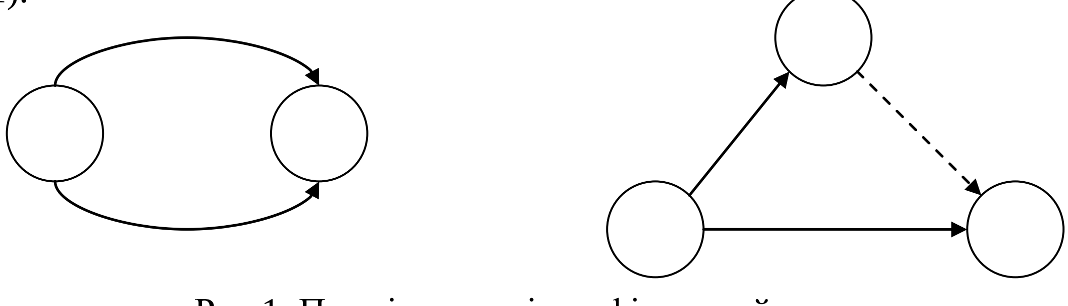


Рис.1. Проміжна подія та фіктивний процес

3. В мережі не повинно бути тупиків, тобто проміжних подій, з яких не виходить жодного процесу (рис 2).

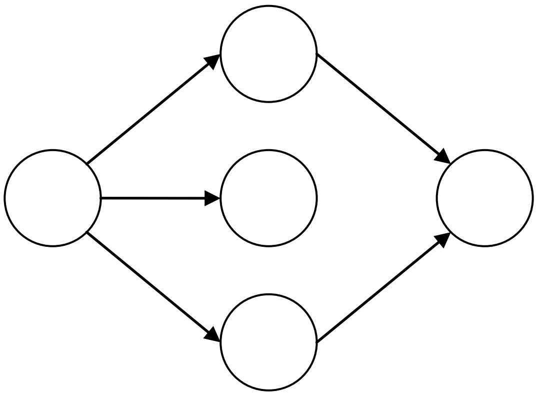


Рис 2. Проміжна подія без виходу

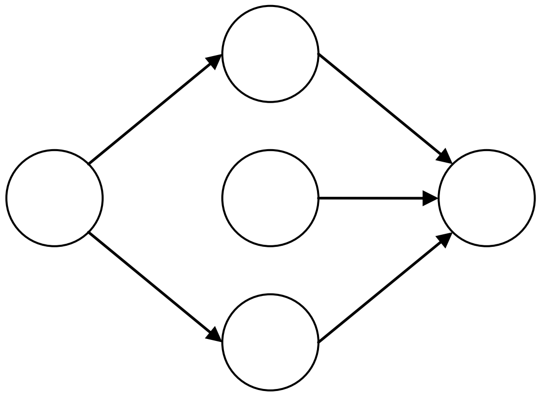


Рис 3. Проміжна подія, якій не передує жодний процес

4. В мережі не повинно бути проміжних подій, яким не передує жодний процес (рис 3).

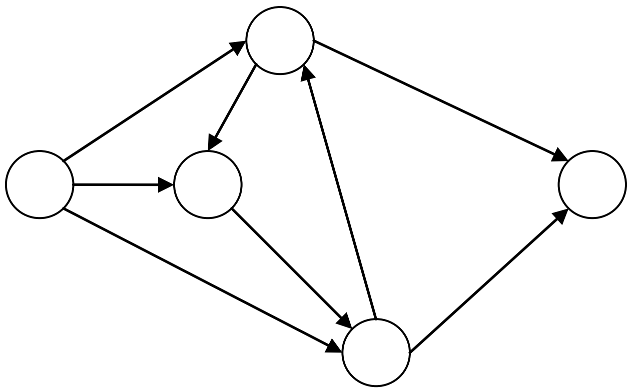


Рис 4. Замкнений ланцюг

5. В мережі не повинно бути замкнених контурів, які складаються з взаємозв’язаних робіт, що створюють замкнений ланцюг (рис. 4).

6. Для правильної нумерації подій роблять наступним чином: нумерація подій починається з початкової події, якій надається номер 1. З вихідної події 1 викреслюють всі процеси які з неї виходять. В мережі, що заломилась знаходять подію, в яку не входить жодний процес. Цієї події надається номер 2. Потім викреслюють процеси, які виходять з події 2, та знову знаходять на мережі, що залишилась, подію, в яку не входить жодний процес та йому надається номер 3 і т.д. (рис 5).

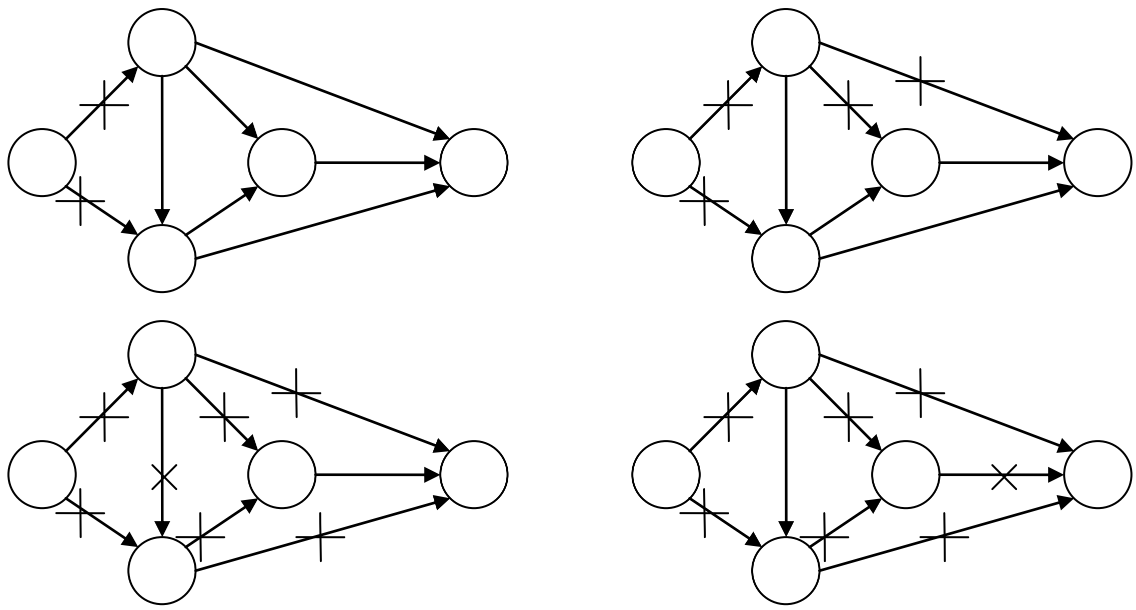


Рис 5. Нумерація подій

7. Тривалість процесів встановлюється на підставі діючих нормативів або за експертними оцінками фахівців. У першому випадку часові оцінки є детермінованими (однозначними), у другому – стохастичними (ймовірнісними).

Основними часовим параметром мережевого графіка є тривалість критичного шляху. Кінцевим результатом застосування методу критичного шляху буде побудова часового графіка виконання проекту. Для цього проводяться спеціальні обчислення, в результаті яких отримують такі відомості:

- загальну тривалість проекту;
- розподіл множини процесів, що складають проект, на критичні та некритичні.

Розрахунок критичного шляху містить два етапи. Перший називається *прямим проходом*. Обчислення починається з початкової події та продовжується до тих пір, доки не буде досягнута кінцева подія. Для кожної події визначається найраніший термін її настання.

На другому етапі, який називається зворотнім проходом, обчислення починаються з кінцевої події та закінчуються початковою подією. Для кожної події обчислюється найпізніший термін її настання.

Введемо позначення:

t_j – найраніший термін настання події j,

T_j – найпізніший термін настання події j,

d_{ij} – тривалість процесу (i, j).

Прямий прохід

Початковий крок. Припускаємо $t_1 = 0$; це вказує на те, що проект починається в нульовий момент часу.

Основний крок. Для вершини *j* визначаємо вершини з *p*, *q*,…, *v*, що безпосередньо зв'язані з вершиною *j* процесами (*p*, *j*), (*q*, *j*),…, (*v*, *j*), для яких вже обчислені найраніші терміни настання відповідних подій. Найраніший термін настання події *j* обчислюється за формулою:

$$t_j = \max\{t_p + d_{pj}, t_q + d_{qj}, ..., t_v + d_{vj}\}.$$

Зворотній прохід

Початковий крок. Припускаємо $T_n = t_n$ (*n*-кінцева подія); це вказує на те, що найраніший та найпізніший терміни завершення проекту співпадають.

Основний крок. Для вершини визначено *p*, *q*, *v*, які безпосередньо зв'язані з вершиною *j* процесами (*j*, *p*), (*j*, *q*), (*j*, *v*) для яких вже обчислені найпізніші терміни настання відповідних подій. Найпізніший термін настання події j обчислюється за формулою:

$$T_j = \min\{T_p - d_{jp}, T_q - d_{jq}, ..., T_v - d_{jv}\}.$$

Процес (i, j) буде *критичним*, якщо виконуються три умови:

1. $T_i = t_i$,

2. $T_j = t_j$,

3. $t_j - t_i = T_j - T_i = d_{ij}$.

Якщо ці умови не виконуються, то процес *некритичний*.

Критичні процеси повинні утворювати неперервний шлях через всю мережу від початкової події до кінцевої.

Запас часу некритичного процесу – це частина максимального інтервалу часу виконання цього процесу (який більше реальної тривалості процесу). Розрізняють *загальний запас часу* на *вільний запас часу*.

Загальний запас часу процесу (i, j) визначається як перевищення над тривалістю виконання цього процесу інтервалу часу від найранішого моменту здійснення події i до найпізнішого часу здійснення події j:

$$\tau_{заг} = (T_j - t_i) - d_{ij}.$$

Вільний запас часу процесу (i, j) визначається як перевищення над тривалістю виконання цього процесу інтервалу часу від найранішого моменту здійснення події i до найранішого здійснення події j:

$$\tau_{віл} = (t_j - t_i) - d_{ij}.$$

За визначенням $\tau_{віл} \le \tau_{заг}$.

Правило “червоного прапорця”. Для некритичного процесу (i, j):

а) якщо $\tau_{віл} = \tau_{заг}$, тоді даний процес може виконуватись у будь-який час всередині максимального інтервалу (t_i, T_j) без порушення відношень передування;

б) якщо $\tau_{віл} < \tau_{заг}$, тоді без порушення відношень передування даний процес може початись із зсувом, який не перевищує $\tau_{віл}$, відносно найранішого моменту початку процесу t_i. Зсув початку процесу на величину, що перевищує $\tau_{віл}$ (але не більше $\tau_{заг}$), повинно супроводжуватись таким же зсувом відносно всіх процесів, які починаються з події j.

Приклад 1. Розглянемо програму створення нового побутового приладу, який користується попитом у населення. Побудуємо мережевий графік та обчислимо його часові параметри. Необхідні данні показані у таблиці 1.

Таблиця 1

Вихідні дані до прикладу 1

<table>
<tr><th>Процеси</th><th>Зміст процесу</th><th>Безпосередньо попередні процеси</th><th>Тривалість, тижні</th></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td>A</td><td>Розробка технічної документації (ТД) на прилад</td><td rowspan="2">–</td><td rowspan="2">A – 3, B – 2</td></tr>
<tr><td>B</td><td>Розробка технічної документації (ТД) на його електронну частину</td></tr>
</table>

1	2	3	4
C	Розробка технологічної документації на електронну частину приладу	A, B	C – 2, D – 2
D	Розробка технологічної документації на прилад		
E	Передача ТД на прилад	A	3
F	Виготовлення приладів	C	7
G	Виготовлення електронної частини приладів	D, E	3
H, I	Розробка ТД на експлуатацію приладу та електронну частину приладу	C, D	H – 5, I – 2
J	Зборка та випробування приладу	F, G	6

Розв'язання:

1. На основі даних таблиці 1 будуємо мережний графік створення приладу з урахуванням рекомендацій 1 – 7 (рис. 6).

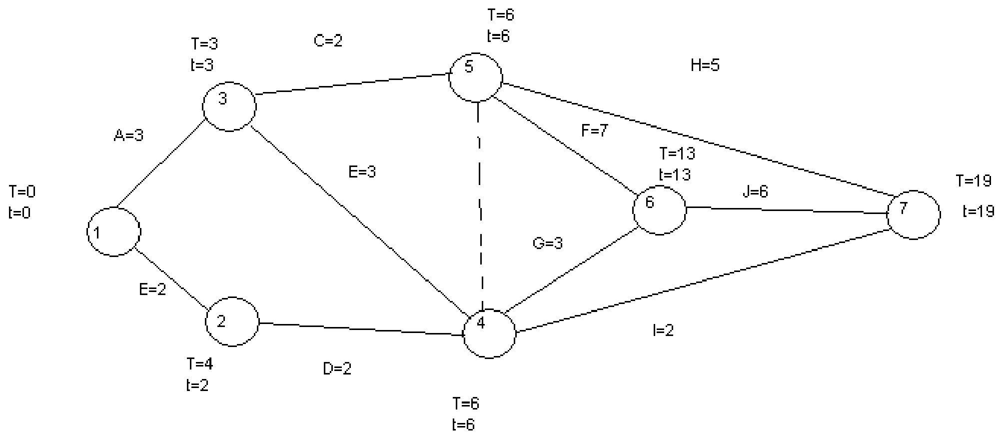


Рис 6. Мережевий графік створення приладу та результати обчислень

2. Знайдемо критичний шлях до мережі проекту.

Прямий прохід:

Вершина 1. Припускаємо $t_1 = 0$.

Вершина 2. $t_2 = t_1 + d_{12} = 0 + 2 = 2$.

Вершина 3. $t_3 = t_1 + d_{13} = 0 + 3 = 3$.

Вершина 4. $t_4 = \max\{t_2 + d_{24}, t_3 + d_{34}\} = \max\{2 + 2, 3 + 3\} = 6$.

Вершина 5. $t_5 = \max\{t_3 + d_{35}, t_4 + d_{45}\} = \max\{3 + 2, 6 + 0\} = 6$.

Вершина 6. $t_6 = \max\{t_4 + d_{46}, t_5 + d_{56}\} = \max\{6+3,\ 6+7\} = 13$.

Вершина 7. $t_7 = \max\{t_4 + d_{47}, t_5 + d_{57}, t_6 + d_{67}\} = \max\{6+2, 6+5, 13+6\} = 19$.

Прямий підхід завершений та розрахунки показують, що проект можна виконати за 19 тижнів.

Зворотній прохід:

Вершина 7. Припускаємо $T_7 = 19$.

Вершина 6. $T_6 = T_7 - d_{57} = 19 - 6 = 13$.

Вершина 5. $T_5 = \min\{T_7 - d_{57}, T_6 - d_{56}\} = \min\{19 - 5,\ 13 - 7\} = 6$.

Вершина 4. $T_4 = \min\{T_7 - d_{47}, T_6 - d_{46}, T_5 - d_{45}\} = \min\{19 - 2, 13 - 3, 6 - 0\} = 6$.

Вершина 3. $T_3 = \min\{T_5 - d_{35}, T_4 - d_{34}\} = \min\{6 - 2,\ 6 - 3\} = 3$.

Вершина 2. $T_2 = T_4 - d_{24} = 6 - 2 = 4$.

Вершина 1. $T_1 = \min\{T_3 - d_{13}, T_2 - d_{12}\} = \min\{3 - 3,\ 4 - 2\} = 0$.

Результати обчислень показані на рис 6. Правила визначення критичних процесів показують, що критичний шлях складають процеси $1 \to 3 \to 4 \to 5 \to 6 \to 7$.

3. Побудуємо часовий графік проекту (рис 7).

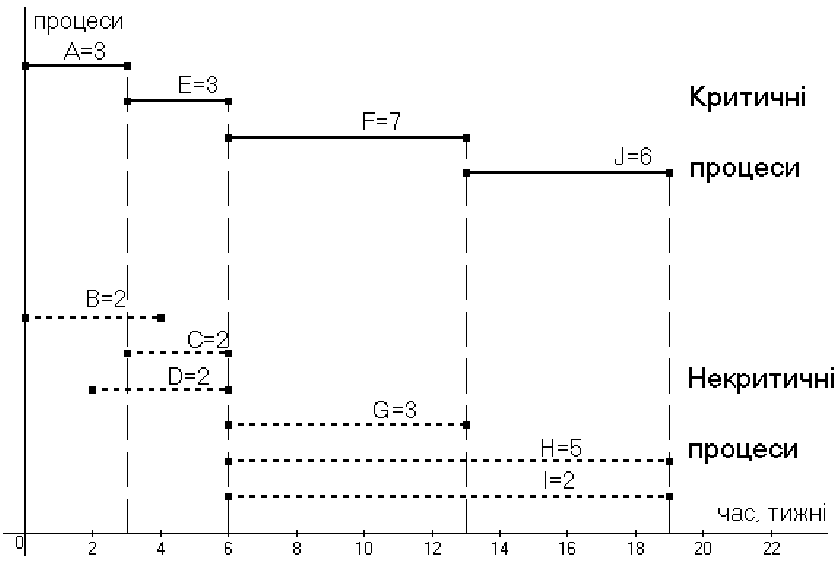


Рис 7. Часовий графік проекту

4. Визначимо запаси часу для некритичних процесів.

Загальні та вільні запаси часу некритичних процесів подані у таблиці 2.

Таблиця 2

Загальні та вільні запаси часу некритичних процесів

Некритичний процес	Тривалість процесу	Загальний запас часу $\tau_{заг}$	Вільний запас часу $\tau_{віл}$	"Червоний прапорець"
B (1, 2)	2	4-0-2=2	2-0-2=0	+
C (3, 5)	2	6-3-2=1	6-3-2=1	
D (2, 4)	2	6-2-2=2	6-2-2=2	
G (4, 6)	3	13-6-3=4	13-6-3=4	
H (5, 7)	5	19-6-5=8	19-6-5=8	
I (4, 7)	2	19-6-2=11	19-6-2=11	

Правило "червоного прапорця" слід застосовувати до процесу B, оскільки для нього $\tau_{віл} < \tau_{заг}$. Всі інші процеси (C, D, G, H, I) мають $\tau_{віл} = \tau_{заг}$, тому вони можуть виконуватись у будь-який час всередині своїх максимальних інтервалів часу виконання.

Розглянемо процес B, який відмічений "червоний прапорець". Оскільки $\tau_{заг} = 2$ тижні, то він може починатися у будь-який тиждень з інтервалу 0-2 тижні від початку виконання всього проекту. Але, оскільки $\tau_{віл} = 0$, то початок процесу B в 1-й та 2-й тижні від початку проекту потребує зсуву процесу D на 1, 2 тижні відповідно від свого найранішого початку. А це в свою чергу вплине на початок процесів G та I.

Мінімізація мережі

Мінімізація мережі або алгоритм побудови мінімального остову дерева передбачає з'єднання усіх вершин мережі за допомогою ребер найменшої довжини. Типовою задачею, для розв'язання якої необхідний такий алгоритм, є проектування мережі доріг з твердим покриттям, які з'єднують населені пункти у сільській місцевості, де дороги, що з'єднують два яких-небудь пункти, можуть проходити через інші населені пункти. Найбільш економний проект дорожньої системи повинен мінімізувати загальну довжину доріг з твердим покриттям.

Алгоритм мінімізації мережі. Починають з будь-якої вершини та з'єднують її з найближчою вершиною мережі. З'єднані дві вершини утворюють зв'язану множину, а інші незв'язану. Далі в незв'язаній множині вибирають вершину, яка розміщена ближче інших до будь-якої вершини зв'язаної множини. До зв'язаної множини додається, а з незв'язної множини вибуває відповідна вершина. Процес повторюють до тих пір, доки у зв'язану множину не попадуть усі вершини мережі. У випадку однаково віддалених вершин обирають будь-яку з них, що вказує на альтернативність (неоднозначність) мінімального кістяка дерева.

Приклад 2. Планується газифікувати п'ять невеликих сіл (рис. 8). Числа на ребрах вказують довжину газових труб (в км). Вершина 1 вже газифікована. Відсутність ребра між двома вершинами означає, що з'єднання відповідних сіл або пов'язане з великими рядками, або неможливе. Знайти таке з'єднання трубами сіл, при якому довжина їх була мінімальною.

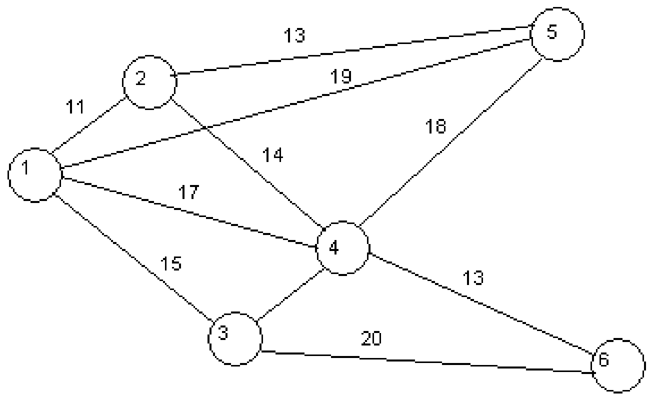


Рис. 8. План газифікації

Розв'язання. Мінімальна довжина газових труб 11+13+14+13+15=66 км (рис 9).

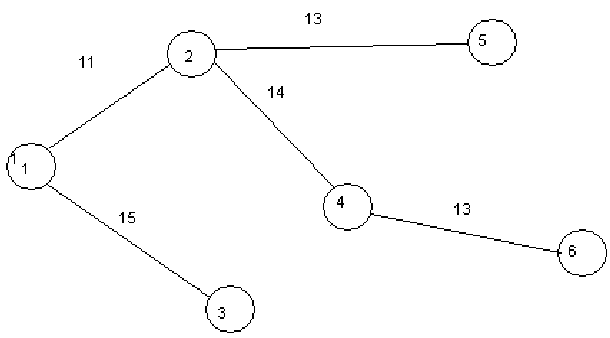


Рис. 9. Мінімальний кістяк дерева

Використовуючи середовище Графоаналізатор 1.3. дану задачу можна розв'язати так:

1. Для створення графу спочатку потрібно обрати його тип (рис. 10).

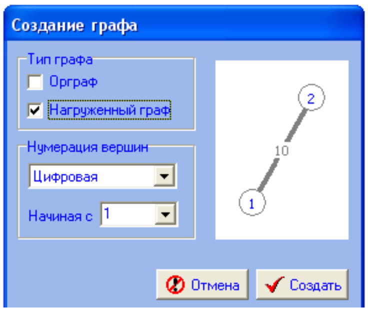


Рис. 10. Обрання типу графу в середовищі Графоаналізатор

2. Потім потрібно додати вершину. Це можна зробити декількома способами:

- використавши гарячу клавішу «F3»;

- кнопку на панелі інструментів ;

- використавши пункт із головного меню програми (рис.11).

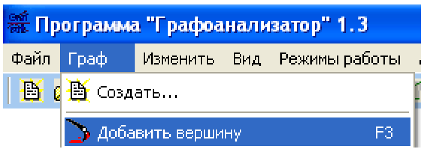
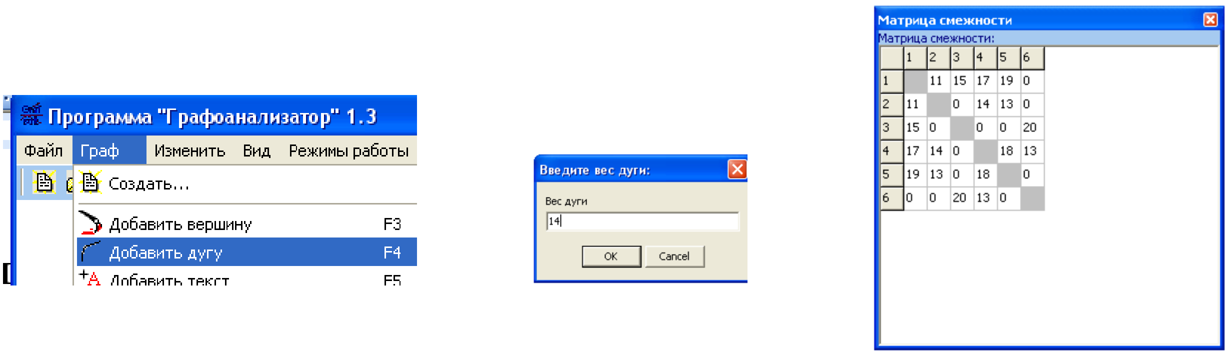


Рис. 11. Пункт із меню програми, що надає можливість додати вершину

3. Додати ребра. Це також можна зробити декількома способами:

- використавши гарячу клавішу «F4» або пункт меню (рис.12,а). Далі потрібно ввести номер вершини з якої буде йти дуга і в яку, при цьому вказавши вагу дуги (рис.12,б);

- редагувати матрицю суміжності, вводячи значення у відповідну клітинку (рис.12,в).



а) б) в)

Рис. 12. Додавання ребер використовуючи пункт меню або редагуючи матрицю суміжності

4. В пункті меню **Алгоритми** обрати **Пошук мінімального кістяку дерева** (рис.13).

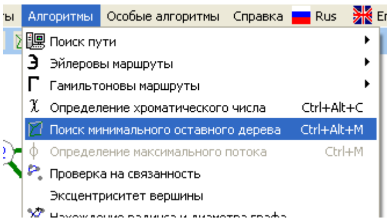


Рис. 13. Пункт меню «Алгоритми»

5. Результат обчислень представлений на рисунку 14.

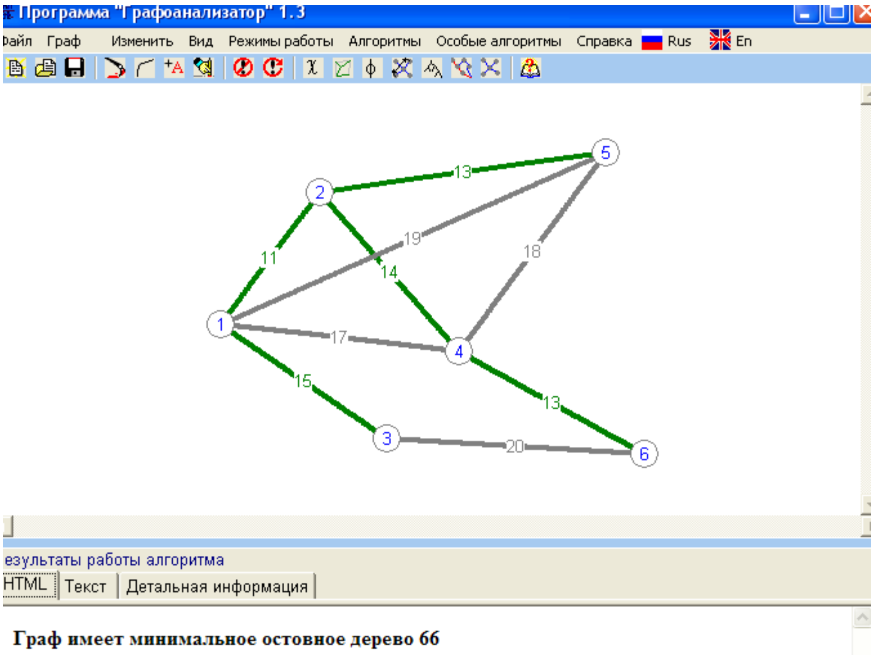


Рис. 14. Мінімальний кістяк дерева засобами Графоаналізатора 1.3

Задача знаходження найкоротшого шляху

Задача полягає у знаходженні зв'язаних між собою доріг на транспортній мережі, які у сукупності мають мінімальну довжину від вихідного пункту до пункту призначення.

Введемо позначення:

l_{ij} – відстань на мережі між сусідніми вершинами *i* та *j* довжина ребра (i, j);

M_j – найкоротша відстань від вихідної вершини 1 до вершини *j*, $M_1 = 0$.

Формула для обчислення M_j:

$$M_j = \min_i \begin{Bmatrix} \textit{Найкоротша відстань до попередньої вершини} \\ \textit{плюс відстань між поточною вершиною j} \\ \textit{та поперередньою i} \end{Bmatrix} = \min_i \{M_i + d_{ij}\}.$$

З формули видно, що найкоротшу відстань M_j до вершини *j* можна обчислити лише після того, як визначена найкоротша відстань до кожної попередньої вершини i, яка з'єднана ребром з вершиною *j*. Процедура завершується, коли отримано M_i останньої ланки.

Приклад 3. (Задача заміни автомобільного парку). Фірма, що займається прокатом автомобілів, планує заміну автопарку на наступні п'ять років. Автомобіль повинен пропрацювати не менше одного року, перш ніж фірма поставить питання про його заміну. На рис 15 наведені вартості заміни автомобілів в умовних грошових одиницях, які залежать від часу заміни і кількості років, на протязі яких автомобіль знаходиться в експлуатації.

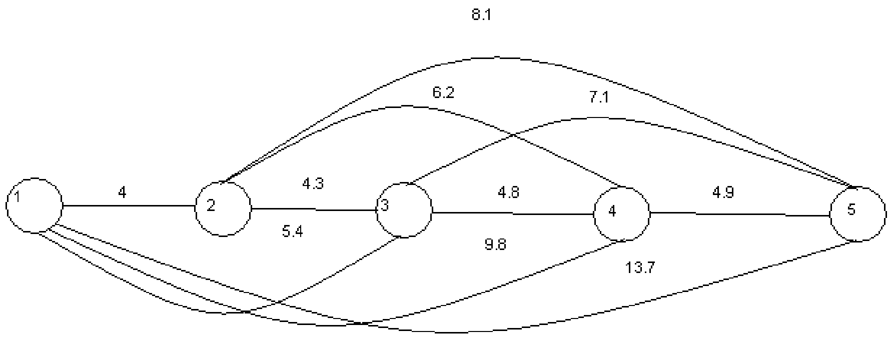


Рис 15. Графічне представлення вартості заміни автомобілів

Визначити план заміни автомобілів, який забезпечує при цьому мінімальні витрати.

Розв'язання. Знайдемо мінімальні відстані:

$M_1 = 0;$

$M_2 = M_1 + d_{12} = 0 + 4 = 4;$

$M_3 = \min\{M_1 + d_{13};\ M_2 + d_{23}\} = \min\{0 + 5{,}4;\ 4 + 4{,}3\} = 5{,}4;$

$M_4 = \min\{M_1 + d_{14};\ M_2 + d_{24};\ M_3 + d_{34}\} = \min\{0 + 9{,}8;\ 4 + 6{,}2\ 5{,}4 + 4{,}8\} = 9{,}8;$

$M_5 = \min\{M_1 + d_{15};\ M_2 + d_{25};\ M_3 + d_{35};\ M_4 + d_{45}\} =$
$= \min\{0 + 13{,}7;\ 4 + 8{,}1;\ 5{,}4 + 7{,}1;\ 9{,}8 + 4{,}9\} = 12{,}1.$

Найкоротший шлях $1 \to 2 \to 5$, вартість якого 12,1 ум. гр. од. Це означає, що кожен автомобіль замінюється через два роки, а через п’ять років списується.

В середовищі Графоаналізатор 1.3 розв’язок задачі має такий вигляд (рис. 16).

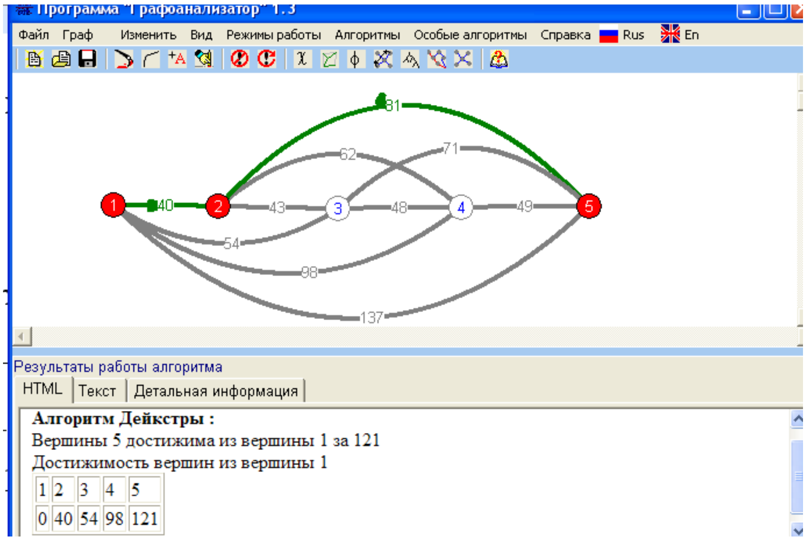


Рис. 16. Розв’язок задачі в середовищі Графоаналізатор 1.3

Для використання даного середовища можна дібрати відповідні задачі для розв’язування студентами і при вивченні інших тем змістового модуля «Графи».

Отже, педагогічно доцільне і виважене впровадження в навчальний процес підготовки студентів у ВНЗ особистісно орієнтованого навчання дискретної математики дає змогу забезпечити розвиток їхніх інтелектуальних умінь, глибоке розуміння завдань, що розв’язуються, формувати вміння застосовувати теорію на практиці, поглиблювати знання і вміння з інформатичних та математичних дисциплін, удосконалювати навички роботи в різних програмних середовищах.

Література

1. Березовський В.Є. Дослідження операцій. Практичний курс: Навч. посіб. / В.Є. Березовський, М.М. Гузій, В.М. Дякон, Л.Є. Ковальов, М.О. Медведєва. – Умань: Видавець “Сочінський”. – 2011. – 238 с.
2. Бидасюк Ю. М. Mathsoft MathCad 12 / Ю. М. Бидасюк. – М. : Диалектика, 2005. – 224 с.
3. Высшая математика на Mathcad 14 : видео-курс. – М. : Интернет-университет информационных технологий, 2009. – [CD. 957 Мб].

4. Дьяконов В. П. Mathcad 11/12/13 в математике : справочное пособие / В. П. Дьяконов. – М. : Горячая линия-Телеком, 2007. – 958 с.
5. Дьяченко С. А. Использование интегрированной символьной системы Mathematica при изучении курса высшей математики в вузе : автореф. ... дис. канд. пед. наук : 13.00.02 “Теория и методика обучения и воспитания (по областям и уровням образования)” / Дьяченко С. А. ; Орловский государственный университет. – Орел, 2000. – 17 с.
6. Жалдак М.І. Математика з комп’ютером. Посібник для вчителів. / М.І. Жалдак, Ю.В. Горошко, Є.Ф. Вінниченко. – 2-ге вид., – К: НПУ імені М.П. Драгоманова, 2009. – 282 с.
7. Кузьмін А. В. Символьні та наближені обчислення в системі Maple : навч. посіб. / А. В. Кузьмін, Н. М. Кузьміна, А. Б. Телейко. – К. : Персонал, 2008. – Ч. 2. – 128 с.: іл.
8. Національна стратегія розвитку освіти в Україні на 2012-2021 роки [Електронний ресурс]. – Режим доступу: <htpp://www.mon.gov.ua/images/files/news/12/05/4455.pdf.> – Загол. з екрану. – Мова укр.
9. Охорзин В. А. Прикладная математика в системе MathCad : учебное пособие для ВУЗов / В. А. Охорзин. – М. : Лань, 2008. – 352 с.
10. Очков В. Ф. MathCad 14 для студентов и инженеров : русская версия / В. Ф. Очков. – М. : BHV, 2009. – 512 с.
11. Семененко М. Г. Математическое моделирование в MathCad / М. Г. Семененко. – М. : Альтекс, 2003. – 208 с.
12. Солонина А. И. Цифровая обработка сигналов. Моделирование в MATLAB : учебное пособие для ВУЗов / А. И. Солонина, С. М. Арбузова. – М. : BHV, 2008. – 816 с.
13. Триус Ю. В. Комп’ютерно-орієнтовані методичні системи навчання математики : монографія / Ю. В. Триус. – Черкаси : Брама-Україна, 2005. – 400 с.