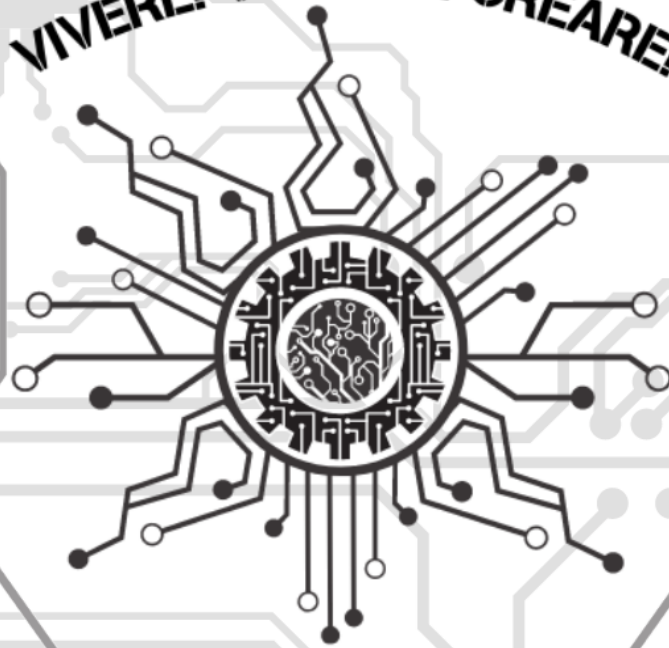


VIVERE! VINCERE! CREARE!



VladosBro



SALESMAN PROBLEM

Зміст

• Опис проекту.....	3
• Демонстраційні матеріали.....	4
• Технології в проекті.....	5
• Примітки щодо реалізації	
-Проблеми, які виникли у ході розробки.....	6
-Особливості проекту.....	7
• Технічний опис проекту	
-Компіляція репозиторія.....	8
- Встановлення.....	8
- Налаштування.....	8
• Керівництво по експлуатації	
-Загальний вигляд програми.....	9
-Елементи інтерфейсу.....	10
- Полотно.....	11
-Панель налаштувань.....	12
-Кнопка запуску алгоритму.....	15
• Інше.....	16



Опис проекту



Salesman Problem

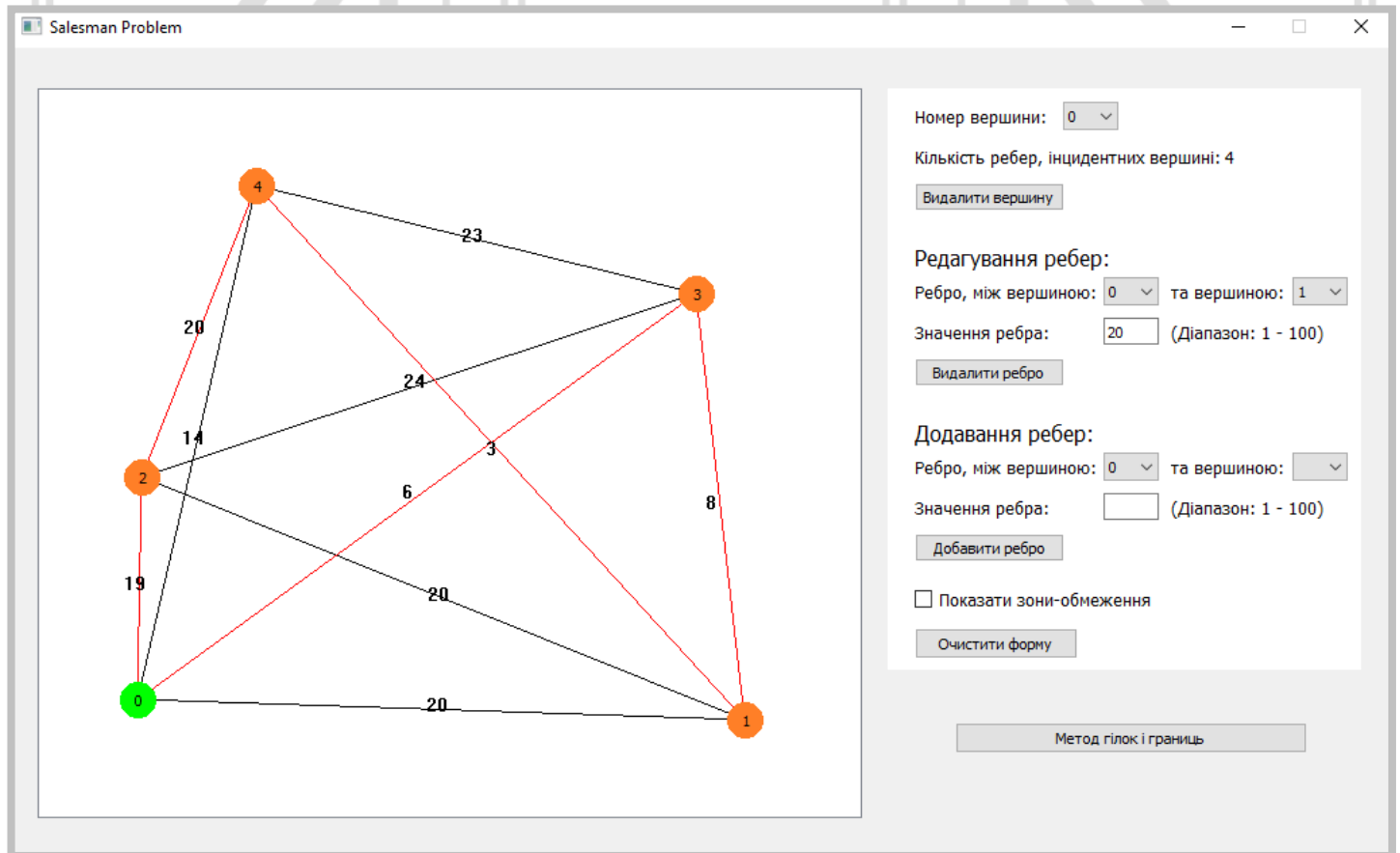
«Wir bezeichnen als Botenproblem (weil diese Frage in der Praxis von jedem Postboten, übrigens auch von vielen Reisenden zu lösen ist) die Aufgabe, für endlich viele Punkte, deren paarweise Abstände bekannt sind, den kürzesten die Punkte verbindenden Weg zu finden.»

(«Ми називаємо проблему гінця (оскільки це питання може бути вирішене на практиці кожним листоношею, до речі, також багатьма мандрівниками) завданням знайти найкоротший шлях, що з'єднує точки для кінцевої кількості точок, попарні відстані яких відомі.»)

*Карл Менгер
на математичному
колоквіумі 1930 року*

Задача комівояжера - задача, яка полягає у знаходженні найвигіднішого маршруту, що проходить через вказані міста по одному разу. В умовах завдання вказуються критерій вигідності маршруту (найкоротший, найдешевший, сукупний критерій тощо) і відповідні матриці відстаней, вартості тощо. Дана програма пропонує вирішення даної задачі за допомогою алгоритму гілок та границь.

Демонстраційні матеріали



Технології в проекті

Алгоритм гілок та границь

У проекті, задля вирішення задачі комівояжера, реалізований алгоритм гілок та границь.

Метод гілок і меж - один з комбінаторних методів. Його суть полягає в упорядкованому переборі варіантів і розгляді лише тих з них, які виявляються за певними ознаками перспективними, і відкиданні безперспективних варіантів.

Перевірка графа на гамільтоновість

Для того, щоб застосувати алгоритм гілок та границь, необхідно перевірити, чи граф, у якому буде відбуватися пошук найкоротшого шляху, містить гамільтоновий цикл, тобто такий шлях, який проходить через кожную вершину рівно один раз.

Примітки щодо реалізації

Проблеми, що виникли у ході розробки

Під час розробки програми, я зіштовхнувся з певними проблемами, а саме:

- Проблема визначення гамільтонового циклу

Опис: Для реалізації алгоритму гілок та границь, необхідно здійснити перевірку на гамільтоновий цикл. Але умови, які б забезпечували перевірку графа, не завжди працюють вірно (Умова Поша, умова Оре, умова Бонді-Хватала), тому граф вимагав знаходження іншого способу перевірки.

Вирішення: Вищезазначені умови покривають не всі можливі графи (згідно із наукової статі, сама найкраща умова (Бонді-Хватала) покриває лише приблизно 80% всіх можливих графів), тому для перевірки графа на гамільтоновий, не обхідно використати алгоритм, який здійснював би дану перевірку. Через те, що головна ідея програми полягала у реалізації алгоритму гілок і границь та обмежений час, код по перевірці циклу був взятий з просторів мережі Інтернет, і доопрацьований для інтеграції у головний код.

* Проблема роботи реалізованого алгоритму вершин і границь

Опис: Після того, як основний алгоритм був реалізований, була виявлена проблема помилкової роботи алгоритму при деяких заданих параметрів графа (кількість вершин, кількість ребер, ваги ребер).

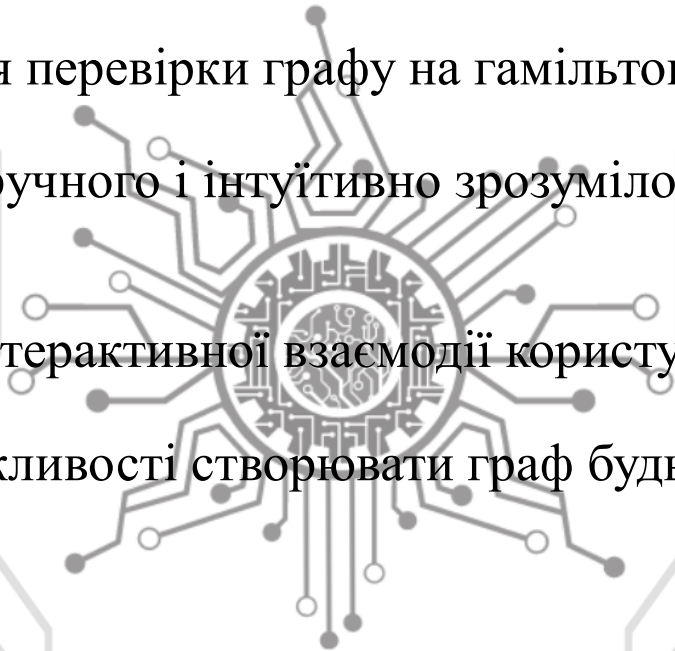
Вирішення: Проблема полягала у дещо неуважному розумінні алгоритму. Через те, що під час роботи алгоритму, необхідно на кожному кроці ітерації вибрати одну з двох дій: включити чи виключити ребро із вибірки шляхів.

У першій реалізації алгоритму, після виключення ребра, на місці нього у матриці ставилась позначка відсутності шляху (-1) і алгоритм продовжував своє виконання, хоча цього не потрібно було робити, а лише перейти до перевірки наступного нуля з найбільшою вагою.

Після усунення цієї неточності алгоритм розпочав вірно визначати мінімальний шлях.

Особливості проекту

- Реалізація алгоритму вершин і границь для вирішення задачі комівояжера
- застосування перевірки графу на гамільтоновість
- створення зручного і інтуїтивно зрозумілого інтерфейсу
- реалізація інтерактивної взаємодії користувача з графом
- надання можливості створювати граф будь-якого розміру



Технічний опис проекту

Компіляція репозиторія

Компіляція репозиторія не вимагає ніяких додаткових дій (налаштування компілятора, підключення сторонніх бібліотек і т. п.)

Встановлення

Проект не вимагає встановлення. Для запуску необхідно лише запустити exe-файл з exe_the_task_of_salesman. Виконавчий файл має бути разом із іншими папками і файлами з exe_the_task_of_salesman.

Для розповсюдження програми поза репозиторієм, необхідно, щоб файл-exe з файлами і папками з exe_the_task_of_salesman мали бути в окремій папці.

Налаштування

Проект не вимагає ніяких окремих налаштувань.



Керівництво по експлуатації

Загальний вигляд програми

Після запуску програми, відкриється вікно з інтерфейсом програми (Рис. 1)

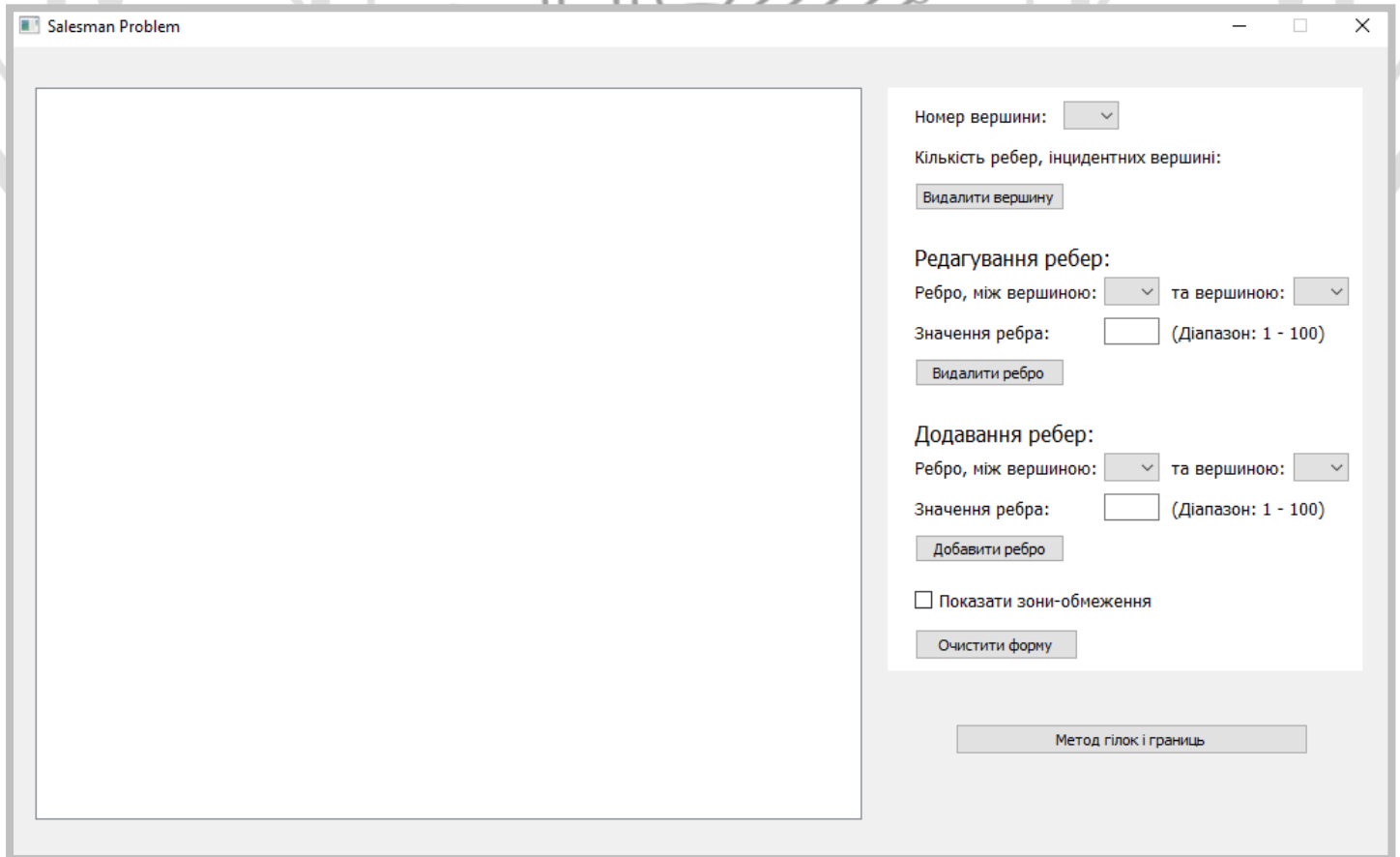


Рис. 1 Інтерфейс програми

Дана програма забезпечує всі необхідні інструменти для виконання необхідних операцій з графами а саме:

- * побудова графу
- * додавання/видалення вершин та ребер графу
- * редагування довжини ребер
- * відображення зони-обмежень, у яких заборонено створювати вершини графу
- * підсвічення вибраних вершин і ребер
- * очищення форми
- * запуск алгоритму пошуку оптимального шляху

Елементи інтерфейсу

Програму можна розбити на кілька умовних «зон» (рис. 2):

- Полотно
- Панель налаштувань
- Кнопка запуску алгоритму

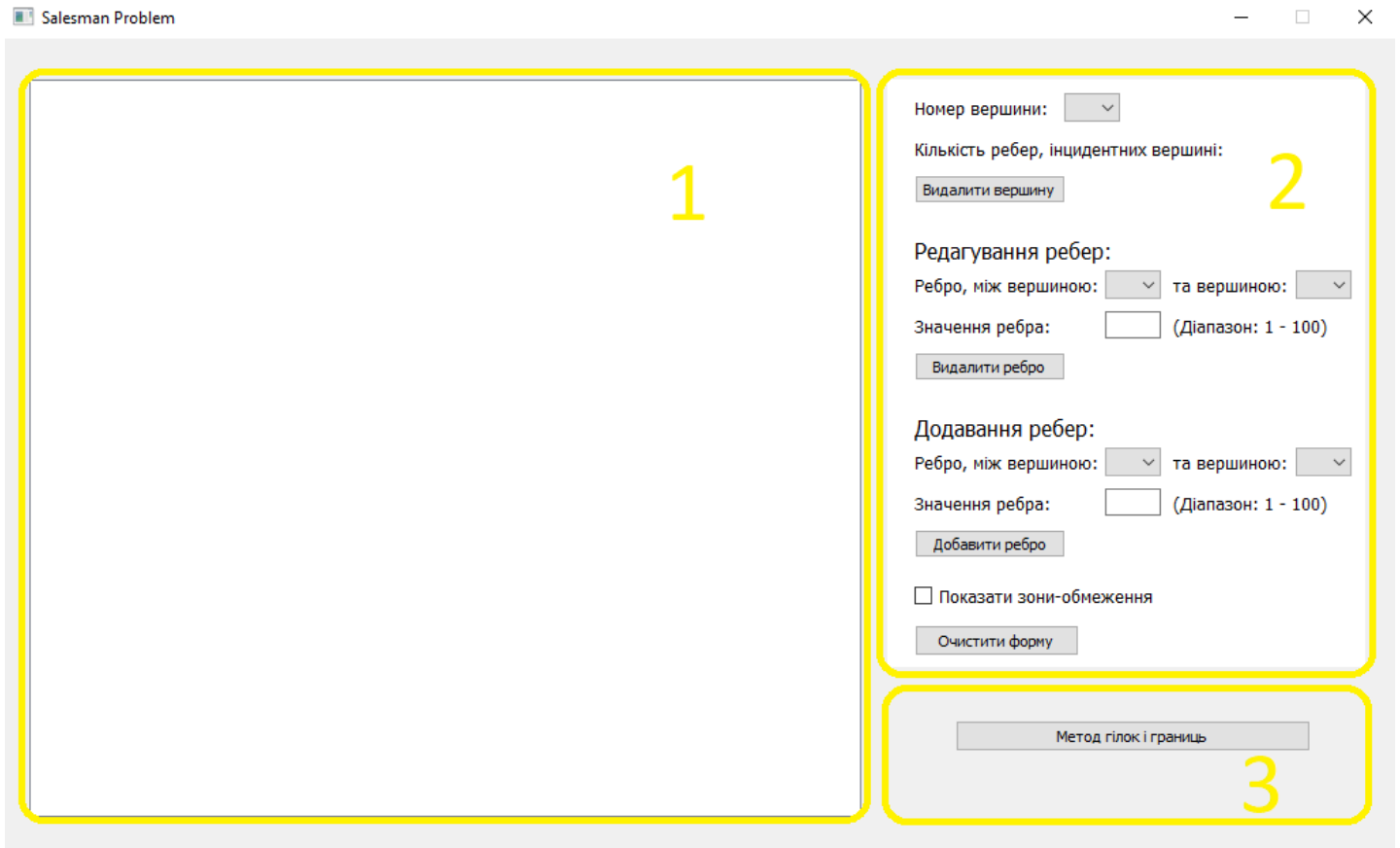


Рис. 2 Умовні зони програми



Полотно

Полотно:

Дана область забезпечує можливість створення графу. Для того, щоб додати вершину графу, необхідно натиснути по довільному порожньому місці на полотні, після чого з'явиться вершина графу (і якщо раніше вже були створені вершини, то від поточної вершини і до існуючих будуть створені ребра із випадково заданими величинами від 1 до 25).

Приклад графу, створеного на полотні, можна переглянути на рисунку 3

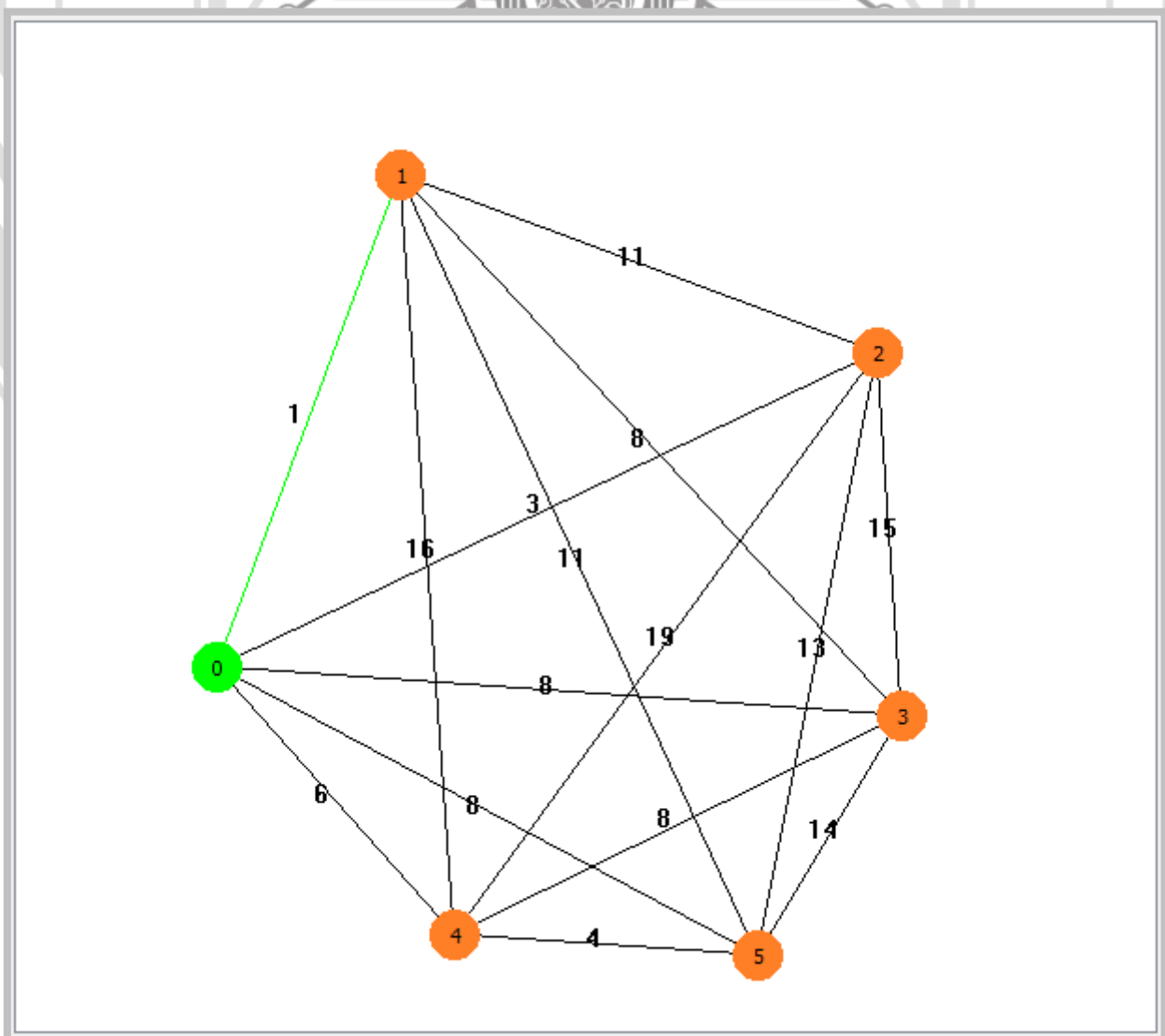


Рис. 3 Вершини і ребра графу

Також введено обмеження на розміщення вершин, у місцях безпосередній близькості до інших вершин, на інших вершинах і на цифрах.

Панель налаштувань

Номер вершини:

Кількість ребер, інцидентних вершині: 7

Редагування ребер:

Ребро, між вершиною: та вершиною:

Значення ребра: (Діапазон: 1 - 100)

Додавання ребер:

Ребро, між вершиною: та вершиною:

Значення ребра: (Діапазон: 1 - 100)

☐ Показати зони-обмеження

Рис. 4 Елементи налаштувань графу програми

Програма містить різноманітні інструменти на панелі налаштувань, що забезпечують роботу з графом.

Випадаючий список «Номер вершини»

Дозволяє вибрати вершину зі списку, задля подальших операцій над нею. При додаванні вершин на полотно, список автоматично оновлюється

Поле «Кількість ребер, інцидентних вершині:»

Дане поле показує кількість ребер, які прилягають до вибраної вершини.

Редагування ребер

Випадаючі списки «Ребро між вершиною:» та «та вершиною:»

Дані списки дозволяють вибрати ребро для редагування, між вибраними зі списку, вершинами.

Поле введення «Значення ребра:»

Надає можливість змінити значення ребра. Значення перезаписується автоматично, відразу після введення\стирання цифри.

Діапазон: 1 - 100.

Кнопка «Видалити ребро»

При натисненні - видаляє ребро, проте при умові, що у разі видалення даного ребра, граф залишиться гамільтоновим (залишиться можливість обітнути всі вершини графу один раз і вернутися до початкової точки).

Додавання ребер

Випадаючі списки «Ребро між вершиною:» та «та вершиною:»

Дані списки дозволяють вибрати вершини, між якими можна додати нове ребро

Поле введення «Значення ребра:»

Надає можливість задати значення для нового ребра

Діапазон: 1 - 100.

Кнопка «Додати ребро»

При натисненні - додає ребро між указаними вершинами.

Чекбокс «Показати зони-обмеження»

При поставлені галочки, відобразяться зони-обмеження, які вказують область, в якій неможливо створити нову вершину. На рисунку 5 можна переглянути граф з видимими зонами обмеження:

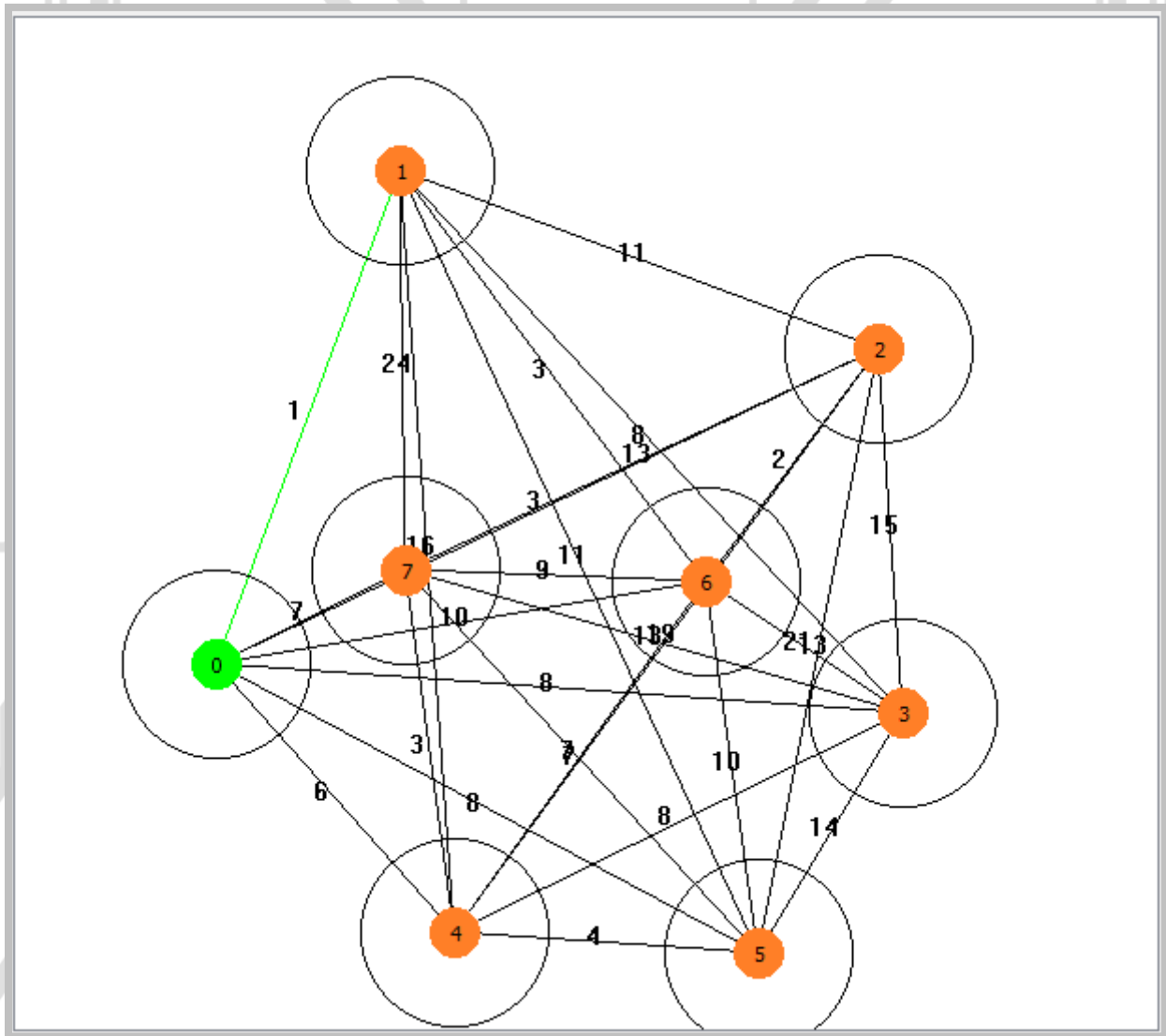


Рис. 5 Граф з зонами-обмеженнями

Кнопка «Очистити форму»

Після натиснення на цю кнопку, полотно очищається разом з панеллю керування програми.

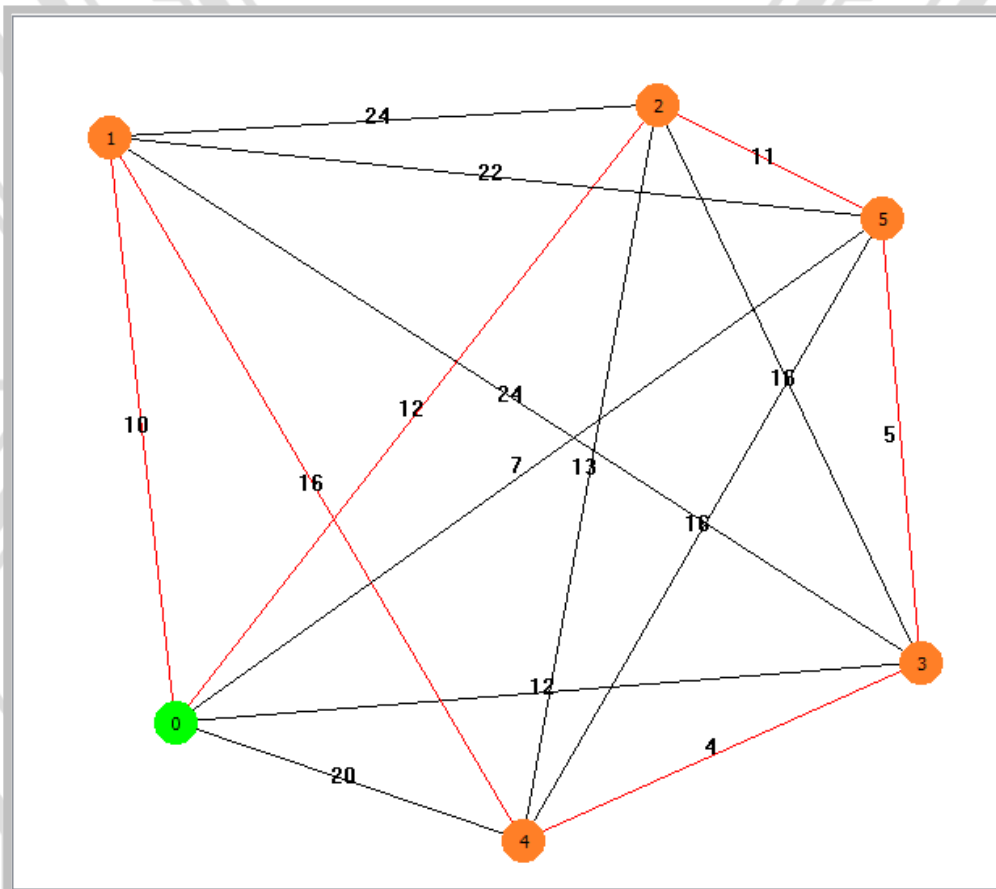
Кнопка запуску алгоритму

Нарешті, після створення графу, і налаштуванню його вершин, можна запустити алгоритм пошуку найоптимальнішого шляху. Дана дія здійснюється натисненням кнопки «Метод гілок і границь» (рис. 6)

Метод гілок і границь

Рис. 6 Кнопка активації алгоритму гілок і границь

Після натисненню кнопки, програма знайде і відобразить знайдений шлях, перефарбувавши ребра, що входять у даний шлях, у червоний колір (рис. 7)



Вітаємо, шлях знайдений!

Інше

Моя пошта для зв'язку: levchuk.ua.101@gmail.com

