

УДК 004.942.519.87

О. Г. Додонов, В. О. Додонов, А. І. Кузьмичов

Інститут проблем реєстрації інформації НАН України
вул. М. Шпака, 2, 03113, Київ, Україна

Візуальна підтримка оптимальних рішень у просторових мережах

Практика оптимізаційного моделювання задач про комунікації у просторових мережах стикається з необхідністю оперативного отримання поряд зі знайденим оптимальним планом відповідного зображення. Запропоновано композицію доступних програмних продуктів, що використовуються на звичайних ПК (АРМ); її використання дозволяє отримувати виважені управлінські рішення в режимі динамічної зміни значень наборів даних, користуючись побудованим наочним графом за результатами оптимізаційних обчислень.

Ключові слова: просторова мережа, задача комівояжера, візуалізація графів, мережеве моделювання задач оптимізації, генетичний алгоритм, Excel Solver, yEd Graph Editor.

Вступ

Просторова мережа (*spatial network*) — унікальна форма графа. На відміну від віртуальних мережевих структур (хмари, соціальні мережі), де спосіб розміщення (*layout*) вузлів не суттєвий, і де важливо показати лише характерні зв'язки з'єднаних вузлів, вузли в просторовій мережі мають фіксовані фізичні характеристики, які зберігаються при відображенні вузлів і ліній зв'язку, це, в першу чергу, координати в просторі. Просторові мережі природні та зрозумілі, доволі часто є основним джерелом початкових наборів даних, тож їхнє відображення у вигляді графів надзвичайно корисне і навіть необхідне для кращого розуміння та подальшого об'єктивного аналізу складно організованих мережевих структур зі зв'язаними воедино географічними, фізичними, соціальними, економічними, екологічними та іншими параметрами.

У діловому світі просторові мережі часто прирівнюються до відповідних інфраструктур. Наприклад, логістичні ланцюги поставок явно мають просторовий компонент, що може мати серйозне значення для оптимізації поставок за певним критерієм. Газо-, водо- і нафтопроводи, мережі електропостачання, транспортна інфраструктура та комп'ютерні мережі є характерними прикладами просторових мереж, де графічна візуалізація і аналіз дуже цінні.

Саме наочність, що дозволяє бачити, як елементи мережі пов'язані фізично та логічно, дає повне уявлення про поточний стан і рівень досконалості мережевої структури з метою розуміння її функціонування, оцінювання можливості появи порушень і їхніх впливів на оточення.

Кінцевий результат аналізу просторових мереж — системні маршрути та потоки: конфігурації, перетини, вузькі місця, кількісні оцінки тощо, що отримується розв'язанням певних задач мережевої оптимізації. Наведено два приклади таких задач у матричній постановці: задача комівояжера (*Traveling Salesman Problem, TSP*) та задача побудови мінімального покриваючого дерева (*Minimum Spanning Tree, MST*).

Задача комівояжера

У просторовій мережевій структурі для n пунктів (об'єктів) на географічній карті задані їхні координати (x_i, y_i) , за їхнім значенням за певною метрикою (евклідовою, манхеттенською, спеціальною) можна розрахувати квадратну матрицю відстаней (тривалостей, витрат) між усіма пунктами в обох напрямках.

У ній потрібно знайти *гамільтонів цикл*¹ (маршрут, що включає рівно по одному разу кожен вузол графа) мінімальної довжини, шуканий цикл — маршрут, який відповідає реальній задачі послідовного обходу пунктів з метою, скажімо, регулярного контролю або оновлення налаштувань обладнання або регламенту його обслуговування, проведення консультацій з обслуговуючим персоналом або його зміни. Інші приклади: моніторинг спеціальних розподілених на значній території технічних комплексів типу нафтових вишок або станцій спостереження; поповнення запасів банкоматів або товарів торгових закладів; збирання та розвезення співробітників установи; це маршрути шкільного автобуса, автомобілів для збору сміття, пошти або екскурсійного туру; оптимальна послідовність переналагодження універсального агрегату для переходу від випуску одного продукту до випуску іншого; маршрут лазера при виробництві мікросхеми. У всіх цих наведених прикладах суттєво важлива послідовність переходу від i -го до $(i + 1)$ -го вузла в несиметричному графі.

Задача комівояжера у складі класу задач комбінаторної оптимізації знаменита поєднанням простоти постановки з виключно високою обчислювальною складністю.

Єдиний точний метод — повний перебір маршрутів обходу — вимагає побудови та порівняльного аналізу $(n - 1)!$ варіантів для несиметричної задачі, причому доведено, що при $n > 66$ ця задача стає трансобчислювальною (вимагаючи обробки більше 10^{93} біт інформації²).

¹ У. Гамільтон (1805–1865), крупний ірландський математик, винайдену задачу популяризував у формі інтелектуальної гри «Навкруги світу», де вимагалось з'єднати 20 вузлів-міст найкоротшим циклом.

² Число 10^{93} («число Бремерманна») — загальна кількість біт, що оброблюються гіпотетичним комп'ютером розміром із Землю протягом тривалості її існування (за сучасними оцінками швидкодії).

Тому для практичних розрахунків і пошуку субоптимального розв'язку використовуються евристичні алгоритми прискореного та скороченого перебору, починаючи від методу відтинань³ і алгоритму «гілок і меж»⁴ для навчально-демонстраційних задач до методів декомпозиції та еволюційного програмування для практичних задач логістики. Більш того, для практики пошук строго оптимального маршруту, що вимагає серйозних ресурсних витрат, потрібен далеко не завжди, адже вхідні дані і реальна ситуація в процесі пошуку оптимального маршруту можуть мати специфічні особливості типу особливого розміщення вузлів (наприклад, уздовж рельєфу гірської місцевості чи берега річки), несподіваного порушення пропускну здатності або навіть неможливості використання ділянки комунікації, про що алгоритм не «знає», а ОПР (особа, що приймає рішення) — знає.

Відзначимо, що ця математична задача, яка вийшла зі сфери інтелектуальних розваг, виконує роль своєрідного «локомотива» комп'ютерної інженерії, ефективного випробувального засобу (*benchmark*) і специфічної інтриги для інженерів-системотехніків, математиків і програмістів, стимулюючи їх на розробку та тестування все більш ефективних обчислювачів, математичних теорій і обчислювальних алгоритмів.

Структурна модель задачі комівояжера — є повнозв'язний орієнтований і асиметричний граф, що утворений n вузлами та n^2 ребрами, де вузли відповідають пунктам, а кожне ребро — зв'язок між i -м та $(i + 1)$ -м пунктами довжиною d_{ij} , $i = 1, \dots, n$.

Задача дискретної оптимізації

I. Знайти матрицю $X = \{x_{ij}\}$ з n^2 елементів, щоб

II. Загальна довжина обходу n вузлів (цільова функція) $D = \sum_{i,j} d_{ij}x_{ij} \rightarrow \min$

III. за обмежень:

$$\sum_{i=1}^n x_{ij} = 1 \text{ в } j\text{-й вузол входить одне ребро,}$$

$$\sum_{j=1}^n x_{ij} = 1 \text{ з } i\text{-го вузла виходить одне ребро,}$$

$$x_{ij} \in \{0, 1\}.$$

Це класична математична модель задачі комівояжера, де відшукується булева матриця досить великого розміру при солідних значеннях n , алгоритми наближених методів, зокрема, досконалий і доступний генетичний алгоритм з обмеженням *alldifferent* (усі різні) для шуканих невідомих, часто генерують у пам'яті віртуальну булеву матрицю, повертаючи результат у вигляді компактного ланцюга n суміжних ребер графа.

³ У 1954 р. відбулась історична подія: на недосконалій ламповій ЕОМ цим методом вдалось розв'язати для того часу досить складну задачу для найкоротшого обходу 49 міст, столиць штатів США (G. Dantzig, D. Fulkerson, S. Johnson).

⁴ J. Little. An algorithm for the traveling salesman problem, OR. 1963.

Візуалізація графів

Проблематика автоматичної побудови графів набула виняткової актуальності з утворенням і використанням комунікацій великого розміру типу інформаційних і соціальних мереж для відображення їхніх конфігурацій і шуканих потоків у мережевих структурах певного призначення. Цінність візуалізації графа або шуканого підграфа — наочного представлення вузлів і ребер — полягає в отриманні нових знань саме шляхом візуального аналізу отриманого зображення, наприклад, у виявленні певних тенденцій або специфічних аномалій при роботі з наборами даними великих розмірів.

Актуальність цієї проблематики викликала появу нових наукових напрямів (*Graph Drawing, Information Visualization*) і спеціальних програмних продуктів — редакторів графів (*Graph Editor*), де для представлення графів великих розмірів застосовуються різноманітні макети, алгоритми їхньої реалізації відображають графи в динамічно змінюваній системі координат (як це робить Google Map).

Такі механізми візуалізації ефективні для абстрактних графів, де важливо виявити характерні тренди, вказуючи зв'язки між певними вузлами, і де розміщення вузлів узгоджується з відповідним макетом: ієрархічним, радіальним, деревом та ін., тоді відображаються динамічні графи, де в різні моменти часу або в різних масштабах отримують різні зображення. Але, на жаль, ці досить досконалі графічні програмні засоби не пристосовані до роботи з просторовими мережами, елементи яких фіксацією просторових координат прив'язані до заданої системи координат: спеціальної, штучної, використовуваної у певному моделюючому комплексі або загальносвітової (широта, довгота). Більш того, якщо все ж існує можливість прив'язки вузлів графа до реальної системи координат, тоді потрібна професійна програмна підтримка на певному мовному рівні для використання визначеного редактора графа, що неможливо здійснити на рівні масової інформаційно-аналітичної діяльності, де переважають оптимізаційні обчислення в електронних таблицях.

Основний результат

Представлений результат стосується зручної і доступної графічної підтримки процесу прийняття управлінських рішень рядовими ОПР, які не мають ані спеціальної підготовки, ані можливості залучати професіоналів-програмістів з машинної графіки. Ними використовуються ліцензійні програмні продукти, в першу чергу, застосунки пакету MS Office, а для цілей візуалізації використовуються доступні версії редакторів графів з базовим набором макетів, спеціальне налаштування яких, якщо це припустимо, скажімо, в уEd Graph Editor, дозволяє будувати просторові графи, як показано у прикладі.

Приклад.

На карті (рис. 1) визначено 71 населений пункт.

У редакторі уEd їхні координати та кінцевий розв'язок показані на рис. 3, за їхніми значеннями розраховані за евклідовою метрикою відстані між усіма парами пунктів.

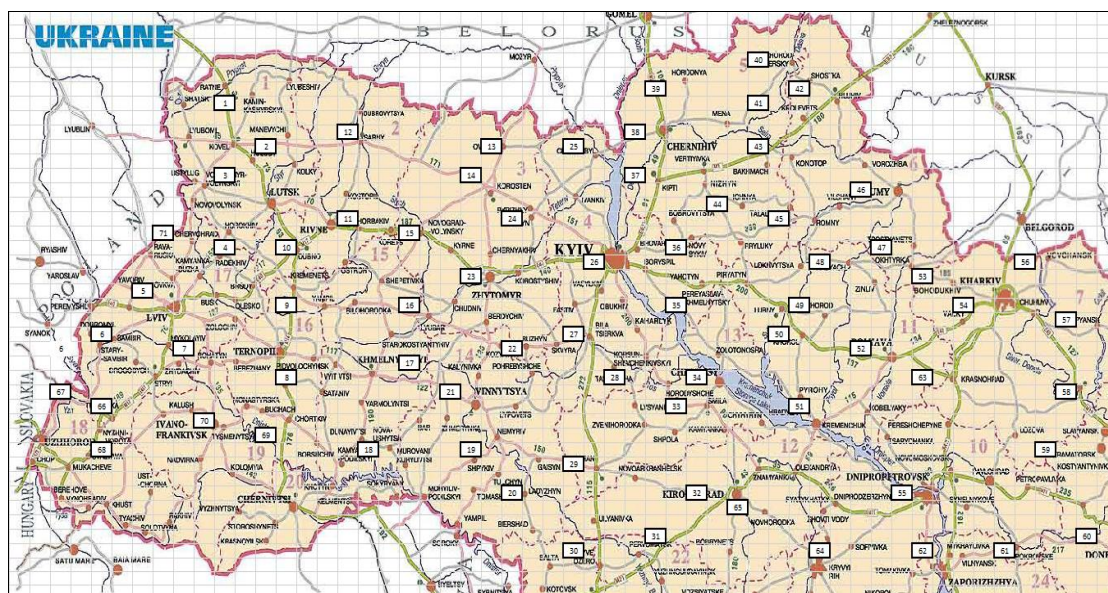


Рис. 1

Генетичним алгоритмом в Excel Solver за обмеженням *alldifferent* для шуканих змінних знайдено наближений розв'язок задачі комівояжера (X). На рис. 2 показано фрагмент матриці відстаней розміром 71×71 і результат X — ланцюг суміжних ребер шуканого циклу.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
60	51.6	48.3	48.9	46.5	49.0	50.0	45.9	40.5	42.2	43.8	42.2	45.6	39.6	39.1	39.1	36.7	35.1	35.5	30.6	28.2	32.6	30.9	35.0	35.6	36.8	30.6	28.7	25.5	25.5	25.0	21.0	19.2	21.9	22.0	25.6
61	49.0	45.6	46.0	43.4	45.7	46.5	42.4	37.0	38.9	40.8	39.4	43.2	37.5	36.8	36.4	33.6	31.8	26.9	24.3	29.2	27.8	32.2	33.2	35.0	28.3	25.8	22.5	21.8	21.0	17.0	15.5	18.9	19.2	23.4	
62	46.0	42.5	42.8	40.0	42.1	42.7	38.6	33.2	35.4	37.4	36.2	40.3	35.0	34.1	33.3	30.2	28.2	27.9	23.1	20.4	25.5	24.4	29.1	30.5	32.8	25.6	22.7	19.2	18.0	17.0	13.0	11.7	15.6	16.3	20.8
63	37.1	34.0	35.1	33.7	37.2	39.0	35.0	30.1	30.2	30.8	28.5	30.9	24.4	24.2	25.3	24.2	24.0	26.9	22.1	21.5	22.2	19.0	21.6	21.0	21.3	16.2	16.0	14.1	17.9	21.3	17.7	14.1	11.7	10.2	11.4
64	42.5	38.9	39.0	35.8	37.6	38.1	34.0	28.6	31.1	33.4	32.5	37.0	32.3	31.1	29.7	26.3	23.9	23.1	18.4	15.5	21.1	20.5	25.5	27.5	30.5	22.8	19.2	15.6	13.4	12.0	8.1	7.2	12.2	13.4	18.4
65	37.5	34.0	34.0	30.8	32.7	33.2	29.2	23.8	26.1	28.4	27.6	32.2	27.7	26.4	24.8	21.3	18.9	18.4	13.6	11.1	16.1	15.6	20.6	22.8	26.3	18.4	14.4	10.8	8.5	8.5	4.5	2.2	7.6	9.2	14.3
66	21.8	19.7	17.1	12.5	8.3	5.0	5.7	9.2	11.4	14.2	17.7	22.5	26.2	24.1	19.2	16.6	15.3	13.3	18.3	20.9	17.0	20.4	20.1	23.9	29.2	26.0	23.5	25.1	23.4	25.1	28.5	29.6	28.0	29.1	28.9
67	26.3	24.2	21.5	17.0	12.7	9.2	10.0	12.5	15.6	18.6	22.0	26.9	30.4	28.3	23.4	20.3	18.4	15.0	20.0	22.1	19.7	23.4	23.9	27.8	33.3	29.5	26.6	27.7	25.0	25.7	29.4	31.1	30.3	31.6	32.0
68	26.3	23.8	21.4	16.5	13.0	10.2	9.2	9.9	13.9	17.5	20.6	26.0	28.6	26.4	21.4	17.7	15.3	11.2	16.1	18.0	16.2	20.1	21.3	25.5	31.1	26.6	23.3	24.0	21.0	21.6	25.3	27.0	26.5	27.9	28.6
69	30.0	27.1	25.0	20.0	17.5	15.2	13.2	11.4	16.3	20.2	22.8	28.6	30.0	27.7	22.9	18.4	15.0	9.2	13.4	14.3	14.9	19.1	21.6	26.1	31.9	26.2	22.0	22.0	17.7	17.0	21.0	23.2	23.8	25.5	27.2
70	22.0	19.2	17.0	12.0	9.5	7.8	5.1	5.0	8.9	12.7	15.7	21.2	23.6	21.4	16.4	12.8	10.8	8.3	13.2	15.8	12.2	15.8	16.4	20.5	26.2	22.0	19.0	20.2	18.3	20.1	23.4	24.5	23.0	24.2	24.4
71	9.5	7.8	5.0	5.2	4.1	7.6	8.1	11.7	7.8	6.1	9.1	11.4	17.1	15.5	12.0	13.0	15.0	18.0	21.2	24.8	17.8	18.8	15.3	17.0	20.9	21.1	21.2	24.2	25.6	29.7	31.9	31.6	27.7	27.9	25.5
X	40	39	38	43	44	45	49	52	63	55	51	50	35	36	37	25	13	14	10	11	15	23	16	17	18	69	68	70	8	9	7	6	5	1	
	5.39	3.16	6.08	4.47	3.16	6.08	4.24	2	2.24	6.32	5.1	5.39	4	5.39	3.61	4	2.24	10.3	3.61	3.16	4.24	3.61	4	6.32	9.22	5.66	4.12	4.47	5.1	5	5	5.83	4.12	3.61	

Рис. 2

Для візуалізації знайденого циклу ($40 \rightarrow 39 \rightarrow 38 \dots 42 \rightarrow 41 \rightarrow 40$) довжиною 332,25 од. використана булева матриця того ж розміру, яка представляє візуальні властивості ребер: її значення (1, 0) визначають 71 видиму і біля 5000 невидимих ліній графа. Доповнивши цю матрицю властивостями вузлів — координатами (X, Y), у середовищі редактора графів будується просторовий граф із знайденим гамільтоновим циклом обходу 71 вузла, де, головне, позиціям розміщення вузлів відповідають задані координати (рис. 3).

Редагування результату

Доведено, що гамільтонів цикл, який зв'язує n точок на площині, отриманий точним методом, це замкнута ламана, ділянки якої (ребра графа) не перетинаються. Зате у циклах, що отримані наближеними методами, можуть бути перетини, які, лише за необхідності, виключають заміною одних ребер іншими, бо для обхо-

ду це не має значення. І саме можливість побудувати граф і бачити отримане рішення в мережі дозволяє зафіксувати ці перетини або інші ділянки, що вимагають внесення змін — чи у конфігурацію отриманого циклу або ж у вхідні дані — щоб результат відповідав реальності.

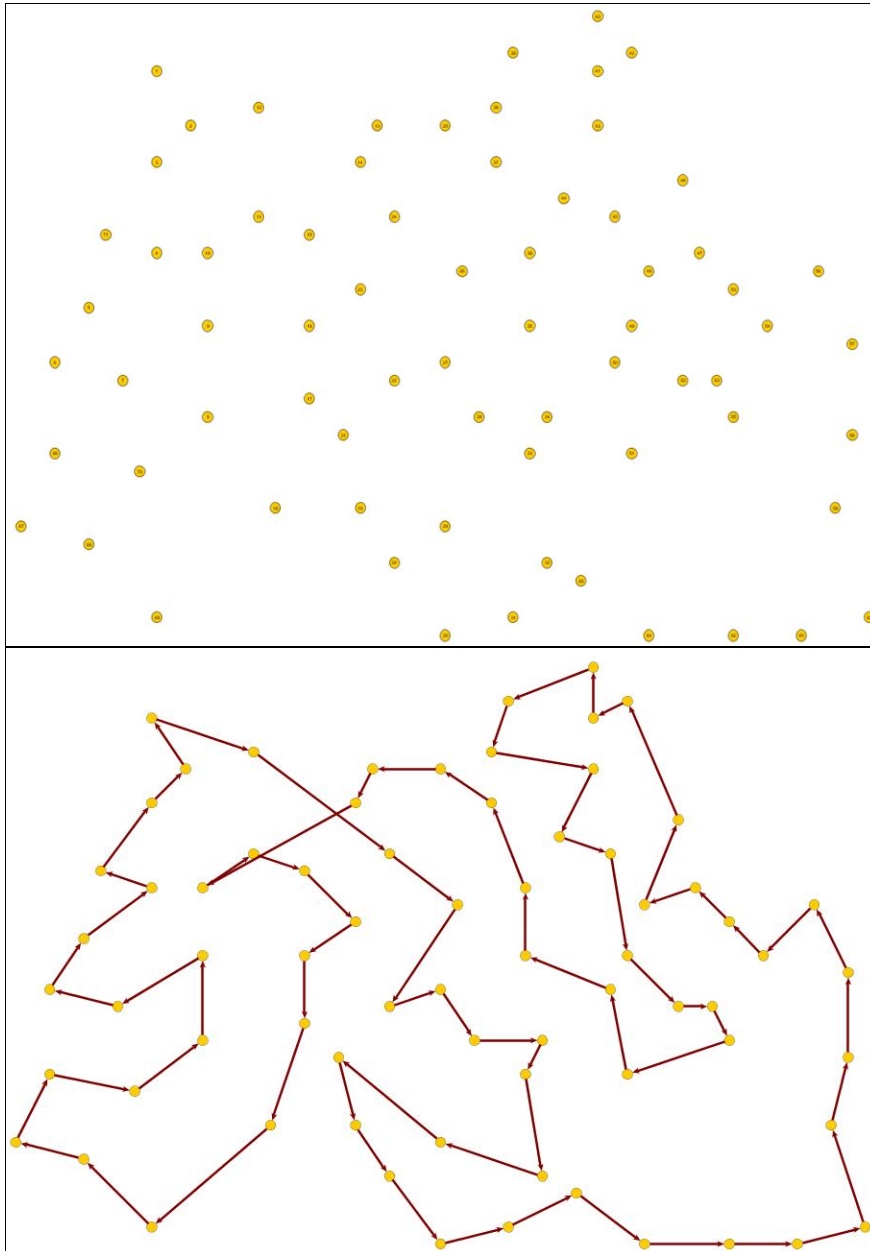


Рис. 3

Приклад перетину, його виключення заміною ребер і наступне відображення на карті показано на рис. 4.

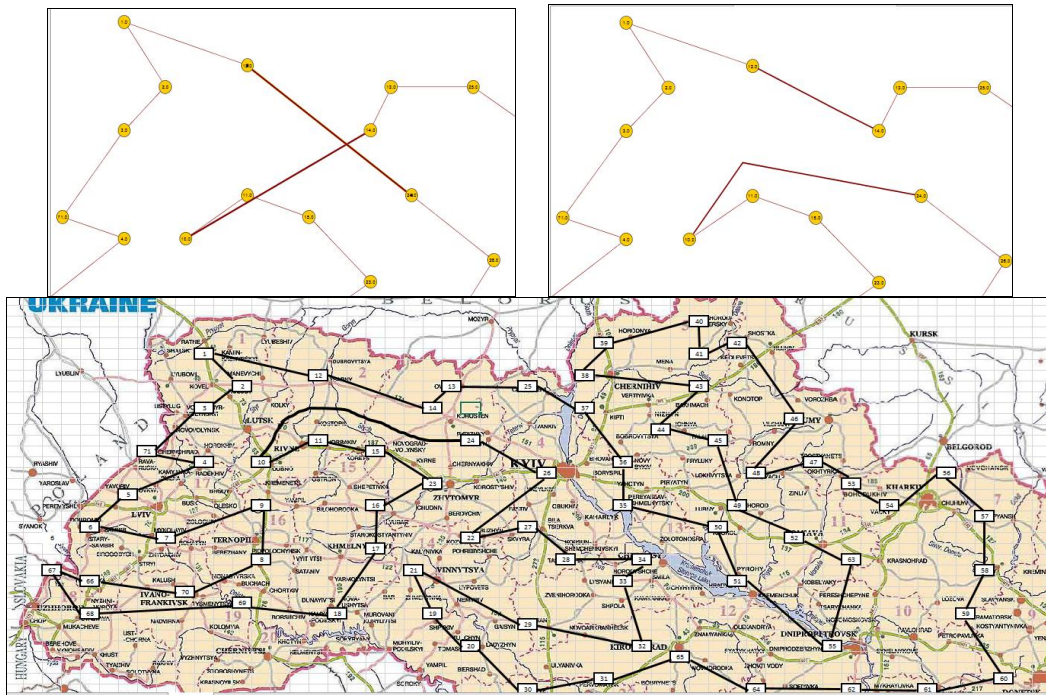


Рис. 4

Ясно, що будь-яка внесена зміна до оптимального плану призводить до погіршення значення цільової функції, робить її «вагомішою», але це справедливо при використанні точного методу, де зроблено порівняння всіх циклів, що зробити для цього прикладу в принципі неможливо.

І оскільки наближений метод перевіряє не всі цикли, у чому полягає його практична цінність, можлива «ручна оптимізація», коли, лише поглянувши на отриманий цикл, ОПР бачить, що рішення можна поліпшити заміною певної групи ребер. «Вагу» зміни, якщо вона гідна розгляду на етапі обговорення результату, легко оцінити, користуючись своєрідним «калькулятором», де використовується функція ИНДЕКС() для роботи з матрицею відстаней.

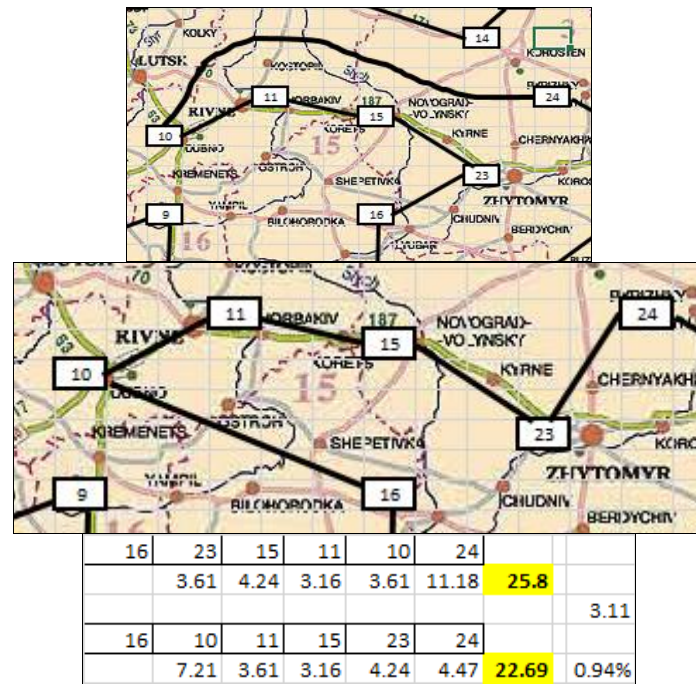
Наприклад, оцінка зробленої вище заміни перетину:

— заміна ребра (12–24) на ребро (12–14) → 10,0 на 6,71, економія — 3,29;

— заміна ребра (14–10) на ребро (10–24) → 10,3 на 11,18, перевищення + 0,88.

У результаті виграш — довжина циклу менше на 2,41 од.

Припустимо, що тепер виконавець вважає, що внесене заміною ребро (10–24) довжиною 11,18 од. приводить до неприпустимо тривалого переходу між пунктами (припустимо обмежено значенням 8 од.), і пропонує інший варіант для цього перетину: замість ланцюжка: 16 → 23 → 15 → 11 → 10 → 24 пропонується ланцюжок: 16 → 10 → 11 → 15 → 23 → 24; оцінка: цей варіант гірше попереднього на 3,11 од., зате максимальна довжина переходу тепер не більше 7,21, а з урахуванням попереднього виграшу маємо невеличке зменшення цільової функції: $-2,41 + 3,11 = -0,7$ од.



Аналогічне редагування зображеного графа може стосуватися переходів між вузлами, які є прямими лініями, без урахування місцевих умов типу сполучення між пунктами гірськими перевалами або відсутності мостових переходів — адже на карті ці ребра є короткими, але реально довгі або тривалі при реалізації. Тож до матриці відстаней вводять реальні значення, виконують перерахунок і повторюють графічне відображення циклу обходу вузлів. Крім того, саме на графі є можливість сформуванати план, який має відповідати новим умовам, скажімо, треба знайдений цикл розділити на регіональні цикли із єдиним центром (рис. 5).

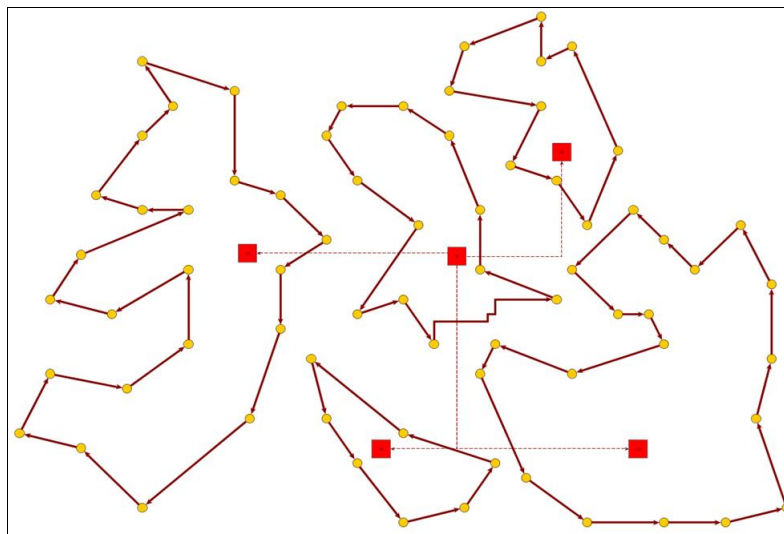


Рис. 5

Цим підходом можна скористатися з метою кластеризації простору — розділення території на специфічні зони за їхніми певними властивостями.

Приклад. Побудова дерева мінімального покриття.

Покриваюче або остовне дерево (spanning tree) — підграф типу дерева, що містить усі n вузлів та $n - 1$ дугу, що з'єднують ці вузли.

Дерево мінімального покриття (ДМП) — покриваюче дерево, сума $n - 1$ вагових коефіцієнтів (довжин, вартостей) дуг якого мінімальна.

Теорема Келі про ці дерева (Cauley, 1889) — у мережі з n вузлів існує n^{n-2} дерев.

Застосування. Якщо населені пункти в регіоні, свердловини з видобутку нафти чи газу, елементи електронної схеми на платі, верстати у цеху, телефони-автомати чи банкомати у місті тощо вважати вузлами відповідної просторової мережі, тоді ДМП — сукупність дуг (ділянок доріг, трубопроводів, кабелів чи провідників), що з'єднують вузли, мінімальної загальної довжини (чи вартості).

Пошук ДМП здійснено за точним n -кроковим алгоритмом Пріма (R. Prim) — рис. 6.

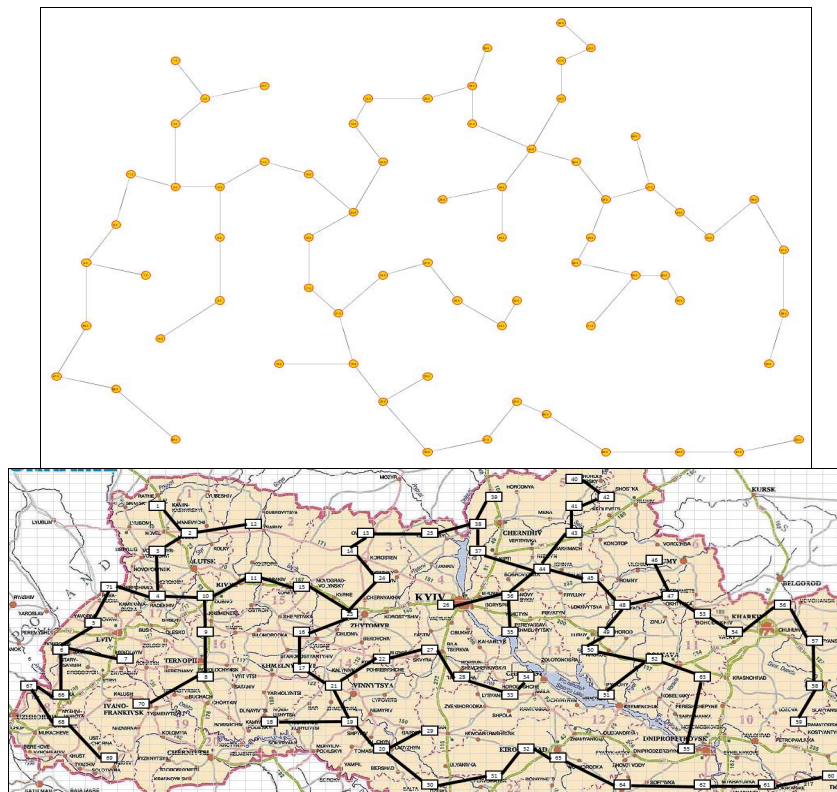


Рис. 6

Висновок

Доведено та показано на прикладах розв'язання непростих оптимізаційних задач про пошук і зображення підграфів (гамільтонова циклу, ДМП) можливість за допомогою доступних програм на звичайному ПК розв'язок задач мережевої оптимізації, отриманий у вигляді шуканого підграфа заданого графа, супровод-

жувати побудовою зображення цього підграфа відповідно до просторових параметрів мережових структур, що моделюються у середовищі електронних таблиць із застосуванням відповідних обчислювальних засобів (програм-надбудов, матричних функцій і процедур). Це зображення — підстава сформуванню зваженого управлінського рішення, де з-за наочності отриманого зображення враховано специфічні властивості мережової структури як об'єкта дослідження.

1. Кузьмичов А. І. Аналітика мережових структур. Київ: Ліра-К, 2018. 208 с.
2. Powell S., Baker K. Business Analytics: The Art of Optimization Modeling with Spreadsheets. 5-ed. Wiley, 2017. 555 p.
3. Додонов О.Г., Кузьмичов А.І. Оптимізаційне електронно-табличне моделювання багатоцільових системних задач. *Ресстрація, зберігання і оброб. даних*. 2016. Т. 18. № 3. С. 12–19.
4. Brath R., Jonker D. Graph Analysis and Visualization: Discovering Business Opportunity in Linked Data. Wiley, 2015. 538 p.
5. Rouse W. Modeling and Visualization of Complex Systems and Enterprises: Explorations of Physical, Human, Economic, and Social Phenomena. Wiley, 2015. 294 p.
6. Tamassia R. Handbook of Graph Drawing and Visualization. CRC Press, 2014. 860 p.
7. Ware C. Information Visualization: Perceptron for Design. 3-ed. Elsevier, 2013. 537 p.
8. Кузьмичов А.І. Оптимізаційні методи і моделі: Практикум в Excel. Київ: АМУ, 2013. 438 с.
9. Cook W. In Pursuit of Traveling Salesman: Mathematics at the Limit of Computation. Princeton Univ. Press., 2012. 272 p.
10. Kaufman M., Wagner D. Drawing Graphs: Methods and Models. Springer, 2001. 325 p.
11. Додонов В.А. Методология системного проектирования и разработки автоматизированных систем обработки информации и управления. *Ресстрація, зберігання і оброб. даних*. 2007. Т. 9. № 4. С. 91–106.

Надійшла до редакції 11.12.2017