

УДК 656.2:004.08:004.02

Косолапов А.А., Гриненко Ю.О.

ПОШУК ОПТИМАЛЬНОЇ СТРУКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ. ГЕНЕТИЧНИЙ ПІДХІД

Дніпропетровський національний університет

залізничного транспорту ім. акад. В. Лазаряна

Дніпропетровськ, Лазаряна 2, 49010

У даній роботі запропонований новий підхід до рішення задачі пошуку оптимальної структури інформаційної системи, заснований на модифікованому генетичному алгоритмі, орієнтованому на формування максимального кістякового дерева з використанням чисел Прюфера для представлення деревовидних графових структур і можливостей бібліотек та розроблених програм графічного інтерфейсу в середовищі MatLab.

Ключові слова: структура, інформаційна система, генетичний алгоритм, числа Прюфера, максимальне кістякове дерево, MatLab.

1. Вступ

В роботі [1] поставлена задача пошуку максимального кістякового дерева на мережі інформаційних потоків шляхом реконфігурації каналів, ремаршрутизації потоків і мінімізації підвищення сумарного інформаційного потоку в мережі. Інформаційну мережу або систему (ІС) можна представити у вигляді неорієнтованого графу, в якому вершинами є інформаційні комплекси (ІК), а ребрами – фізичні канали зв'язку між ними (рис.1). Між інформаційними комплексами циркулюють потоки даних, при цьому кожний ІК може виступати приймачем або джерелом інформаційних потоків (ІП) і транзитним вузлом для деяких інформаційних потоків. Кожен потік даних описується парою «джерело-приймач» і об'ємом (біт/с).

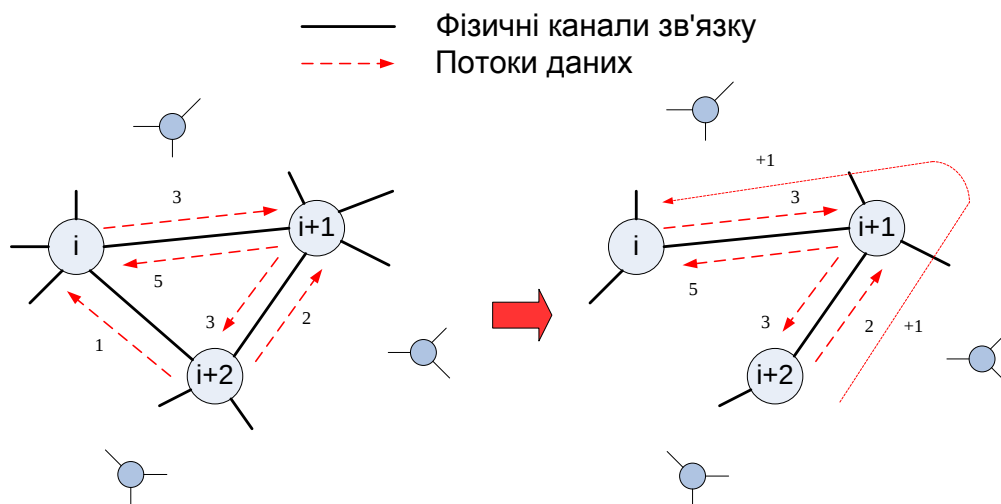


Рис. 1. Реконфігурація графу і ремаршрутизація потоку $(i+2) \rightarrow (i)$

При розв'язанні задачі оптимізації ІС необхідно враховувати обмеження, що накладають інформаційні комплекси на об'єм даних, що передаються через них. Задачею генетичного алгоритму є пошук такого кістякового дерева в графі, для якого приріст сумарного інформаційного потоку в отриманій структурі в порівнянні з повнозв'язною структурою буде мінімальним. В роботі ефективність реалізації генетичних алгоритмів досягається максимальним врахуванням можливостей програмного середовища, в якості якого обрано MATLAB.

2. Алгоритми реалізації розв'язку задачі

2.1 Кодування та декодування розв'язків задачі оптимізації

Для роботи генетичного алгоритму необхідно визначитися з типом особин популяції, тобто з представленням розв'язків завдання оптимізації.

При виборі способу кодування кістякових дерев необхідно враховувати, який вигляд приблизно може мати оптимальна структура ІС. Якщо не враховувати обмеження на об'єм трафіку, що накладають ІК, оптимальною структурою інформаційної системи є структура «зірка» (рис. 2), при якій один ІК знаходиться в центрі «зірки», інші ІК «променями» з'єднуються з центром.

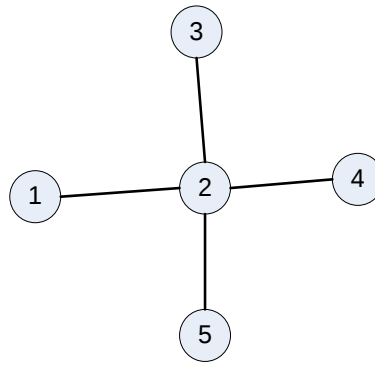


Рис. 2. – Структура ІС «зіврка»

Оптимальність такої структури в рамках даної роботи забезпечується мінімальним приростом сумарного інформаційного потоку через те, що дублювання інформаційних потоків у системі – мінімальне. Якщо враховувати обмеження на об'єми трафіку, то, ймовірноше всього, оптимальною структурою буде деяка зіркоподібна структура, так званий розвиток «зіврки» (рис.3).

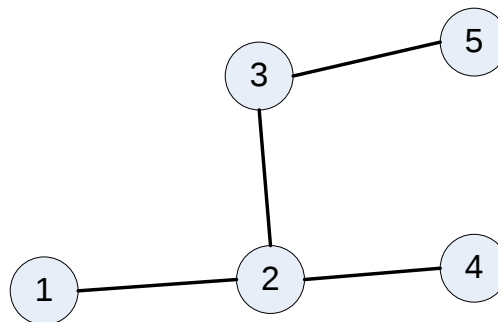


Рис. 3. Зіркоподібна структура ІС

Для кодування кістякових дерев обрано код Прюфера, який представляє собою спосіб однозначного кодування дерева за допомогою послідовності чисел. Згідно теореми Келі [2], на N вершинах, що пронумеровані натуральними числами від 1 до N , існує рівно N^{N-2} різних кістякових дерев. Прюфер довів, що існує однозначне перетворення між кістяковими деревами з N вершинами і векторами довжиною $N-2$, в яких кожен елемент є натуральне число від 1 до N [3].

Алгоритм кодування структури кістякового дерева наведено на рис. 4. В масиві code в кінці роботи алгоритму знаходиться код Прюфера відповідного дерева.

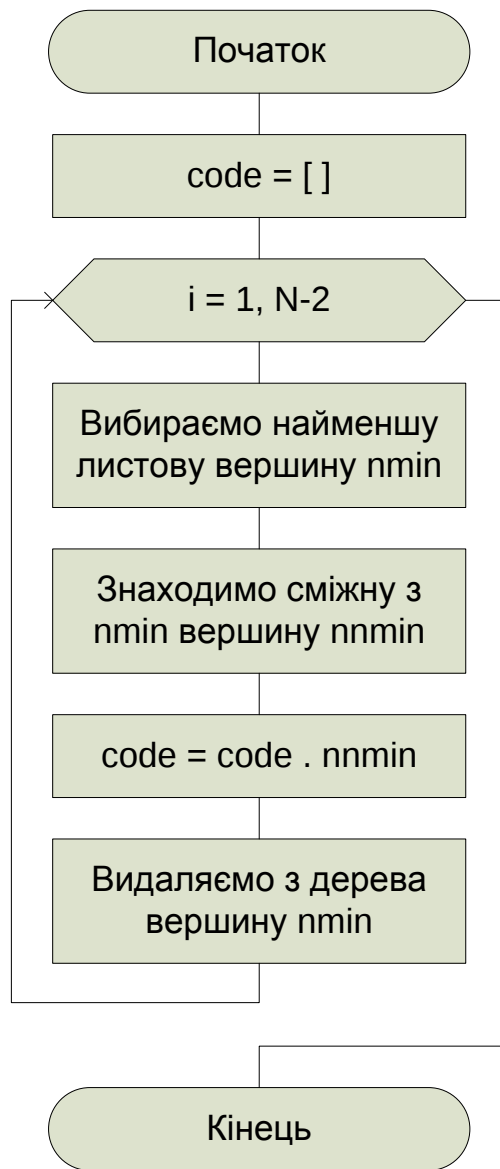


Рис. 4. Алгоритм кодування дерев кодом Прюфера

Розглянемо приклад кодування (рис. 5).

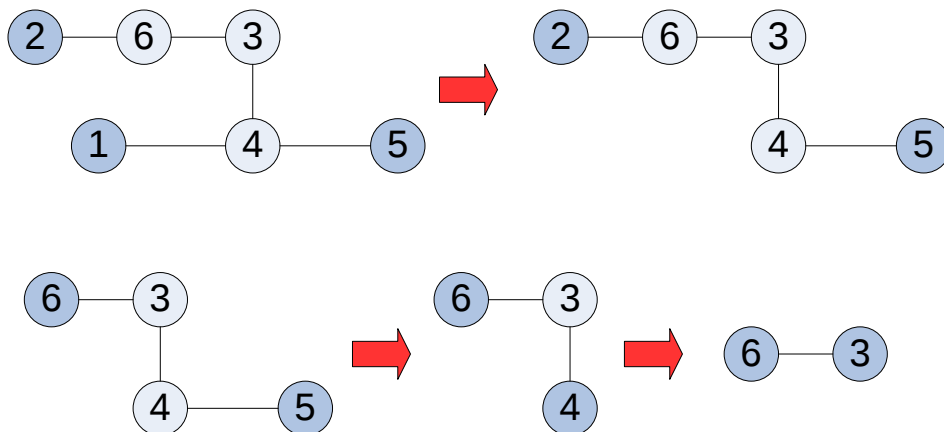


Рис. 5. Процес кодування дерева

Вершина 1 є найменшою листовою вершиною графу. Вершина 4 є суміжною з вершиною 1. Тобто, цифра 4 буде першою в коді Прюфера. Видаляємо вершину 1 і ребро (1,4). Повторюємо процес, доки не залишиться ребро (3,6). Отриманий код Прюфера для прикладу з рис.5 – [4 6 4 3].

Алгоритм декодування дерева з коду Прюфера наведена на рис.6.

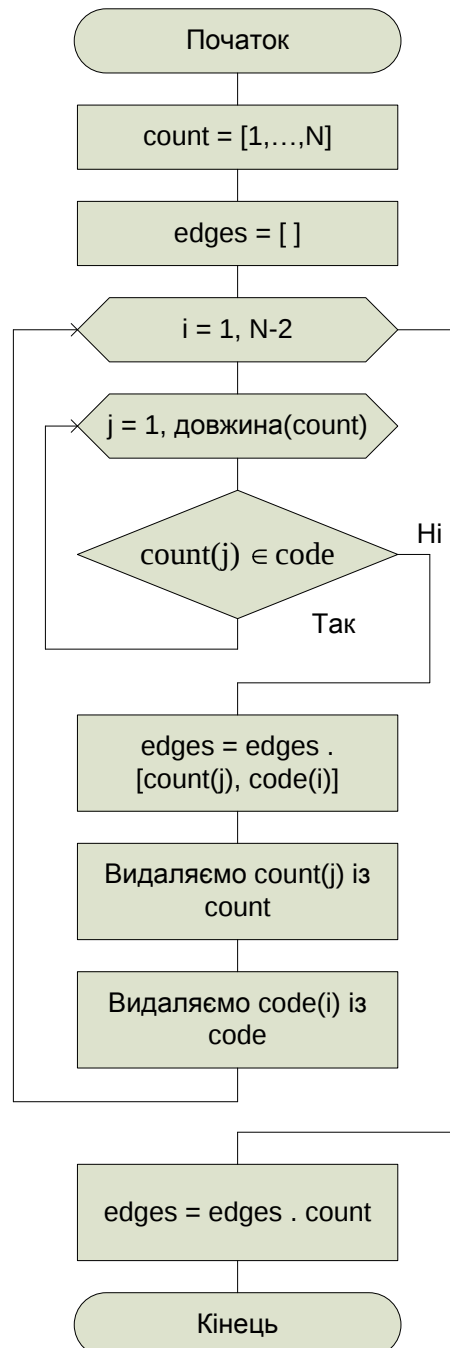


Рис. 6. Алгоритм декодування дерев числами Прюфера

Розглянемо приклад декодування коду Прюфера [3 2 2 1] (див. рис.7).

Для коду Прюфера $P = [3\ 2\ 2\ 1]$ набір вершин, що відсутні у коді – $\bar{P} = [4\ 5\ 6]$. Вершина 4 є найменшою з $\bar{P}$, а вершина 3 є першою у P . Додаємо ребро (3,4) до дерева, видаляємо вершину 3 з P і вершину 4 з $\bar{P}$. Оскільки вершина 3 тепер відсутня у P , додаємо її в $\bar{P}$. На наступному кроці маємо $P = [2\ 2\ 1]$, $\bar{P} = [3\ 5\ 6]$. Повторюємо зазначені вище дії, поки в P не залишиться елементів, а $\bar{P} = [1\ 6]$. Додаємо до дерева останнє ребро (1,6). На цьому процес декодування завершується.

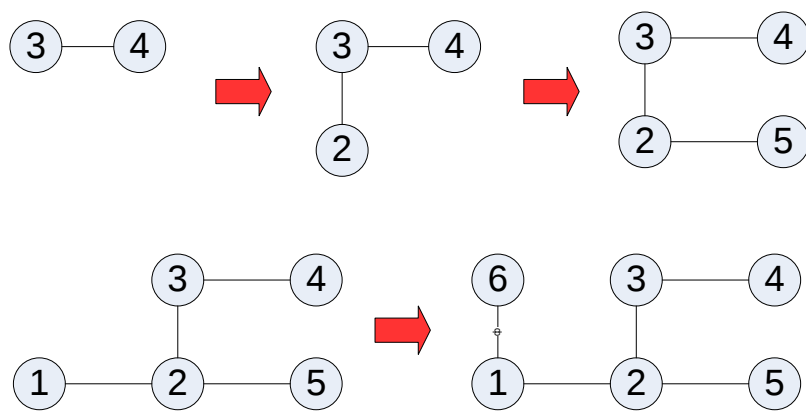


Рис. 7. Процес декодування дерева

Код Прюфера широко використовується дослідниками при розв'язку MST-завдань [4]. Даний код демонструє свою ефективність при пошуці саме зіркоподібних кістякових дерев. «Зірка» з N вершин в коді Прюфера представляє собою послідовність довжиною $N - 2$, в якій всі елементи рівні. Наприклад, послідовність $[3\ 3\ 3]$ описує «зірку» із 5 вершин з центральною вершиною 3. Таку властивість коду можна використати для прискорення пошуку оптимальної структури при формуванні першого покоління алгоритму.

Через те, що тип особин генетичного алгоритму при використанні коду Прюфера – цілочисельні рядки, в MATLAB необхідно самостійно реалізувати процедури ініціалізації популяції, оператор кросинговеру та мутації.

2.2 Ініціалізація популяції

Процес ініціалізації популяції проходить на початку роботи генетичного алгоритму. Реалізація даної процедури може значно прискорити пошук оптимальної структури, чим підвищується ефективність усього алгоритму.

Зважаючи на те, що оптимальним розв'язком задачі оптимізації ІС найчастіше є «зірка» чи її розвиток, є сенс в ініціалізації частини початкового покоління усіх можливих «зірок» для заданої кількості ІК. Інша частина популяції є повністю випадковою для внесення більшої генетичної різноманітності. Таким чином, якщо за умовами задачі, ІК не накладають обмеження на трафік, а також розмір популяції є достатнім для ініціалізації усіх можливих «зірок», оптимальний розв'язок буде знайдений вже на першій ітерації генетичного алгоритму.

Алгоритм ініціалізації популяції наведений на рис.8.

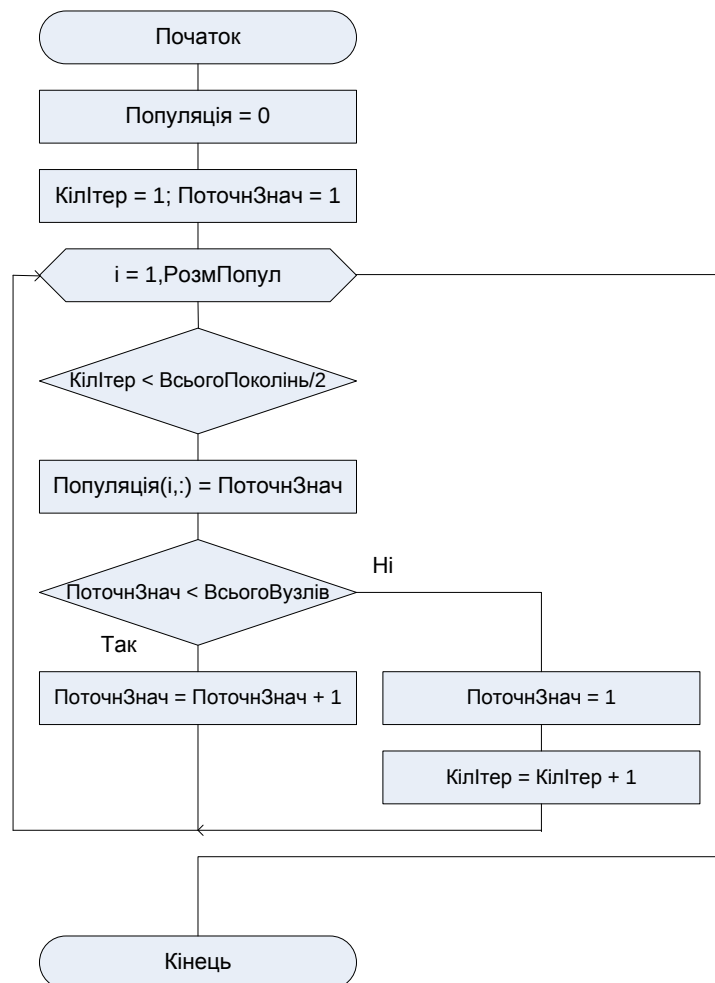


Рис. 8. Алгоритм ініціалізації популяції

2.3 Оператор кросинговеру

Через специфічний тип особин генетичного алгоритму авторами був розроблений спеціальний оператор кросинговеру. В [3] для розв'язку задач MST (Minimum Spanning Tree) рекомендується використовувати двоточковий кросинговер, який є ефективніше одноточкового. Особливістю запропонованої реалізації кросинговеру є утворення лише одного нащадка, при чому порядок обміну батьків своїми фрагментами є випадковим.

Алгоритм реалізації кросинговеру наведений на рис.9.

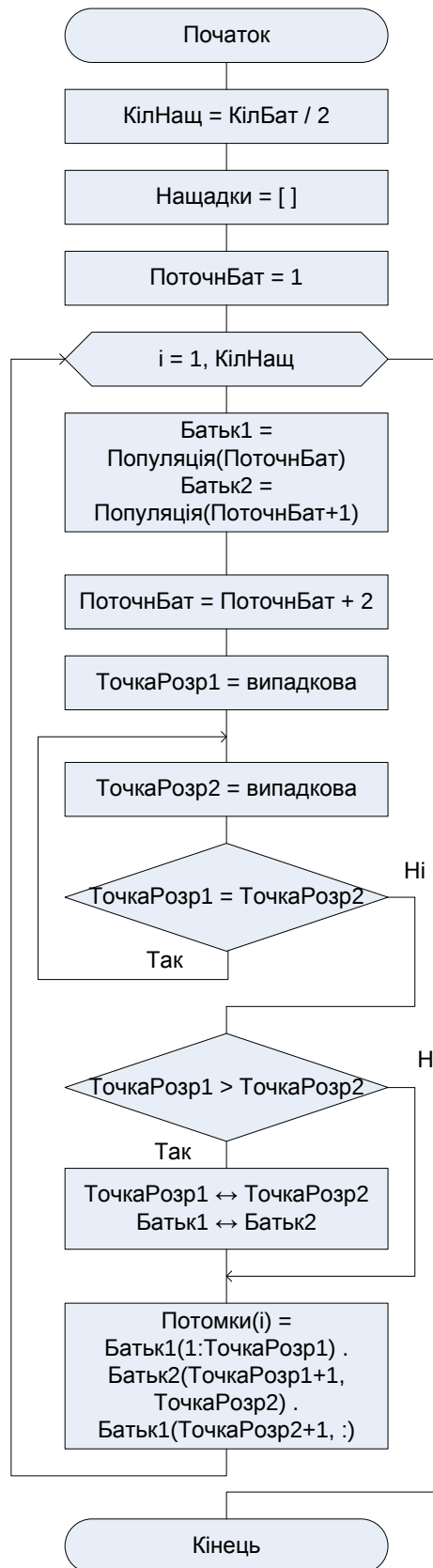


Рис. 9. Алгоритм оператора кросинговеру

2.4 Оператор мутації

Для більш ефективної боротьби з передчасної збіжністю, а також для внесення більшої різноманітності в популяцію використовуємо двоточкову мутацію. За основу взята стандартна двоточкова мутація для бітових особин, що пропонується утилітою GATool в пакеті MATLAB, модифікована для цілочисельних рядків. До того ж, якщо протягом п'ятих поколінь алгоритм не покращує розв'язок, випадкова особина в популяції мутує повністю.

Алгоритм оператора двоточної мутації наведений на рис.10.

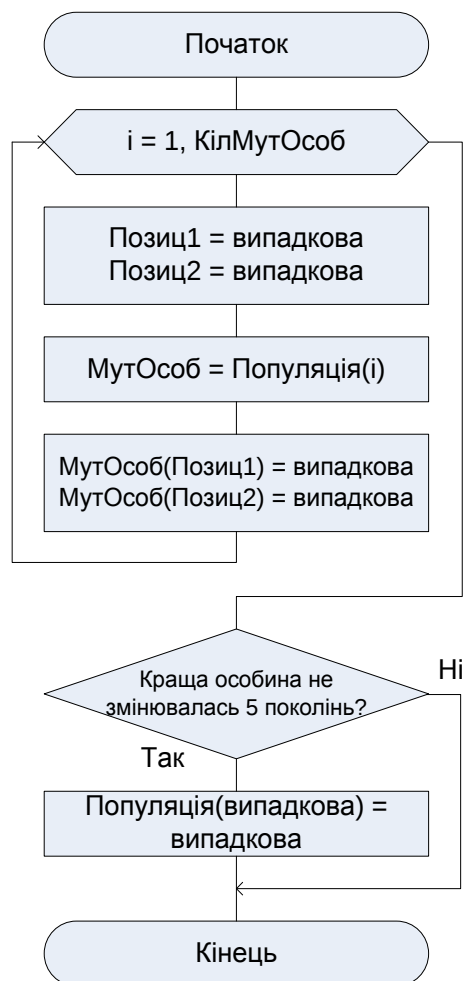


Рис. 10. Алгоритм оператора мутації

2.5 Функція пристосованості

Розробка функції пристосованості є невід'ємною частиною проектування генетичного алгоритму. Функція пристосованості має давати об'єктивну оцінку ефективності кожної особини, враховуючи всі обмеження задачі.

У випадку оптимізації структури ІС функція пристосованості має обчислювати приріст сумарного інформаційного потоку в структурі, що закодована в особині популяції, в порівнянні з повнозв'язною структурою системи. Функція має декодувати код Прюфера (тобто особину популяції), розраховувати маршрути для всіх інформаційних потоків в ІС, на основі яких розраховується приріст сумарного інформаційного потоку, а також враховувати, чи виконуються обмеження на трафік ІК. Значення функції складається із значення приросту сумарного інформаційного потоку, а також з сумарного перевищення встановлених обмежень на трафік всіх ІК системи. Таким чином, чим менше значення функції пристосованості, тим кращий розв'язок пропонує особина популяції.

Алгоритм розрахунку функції пристосованості наведений на рис. 11.

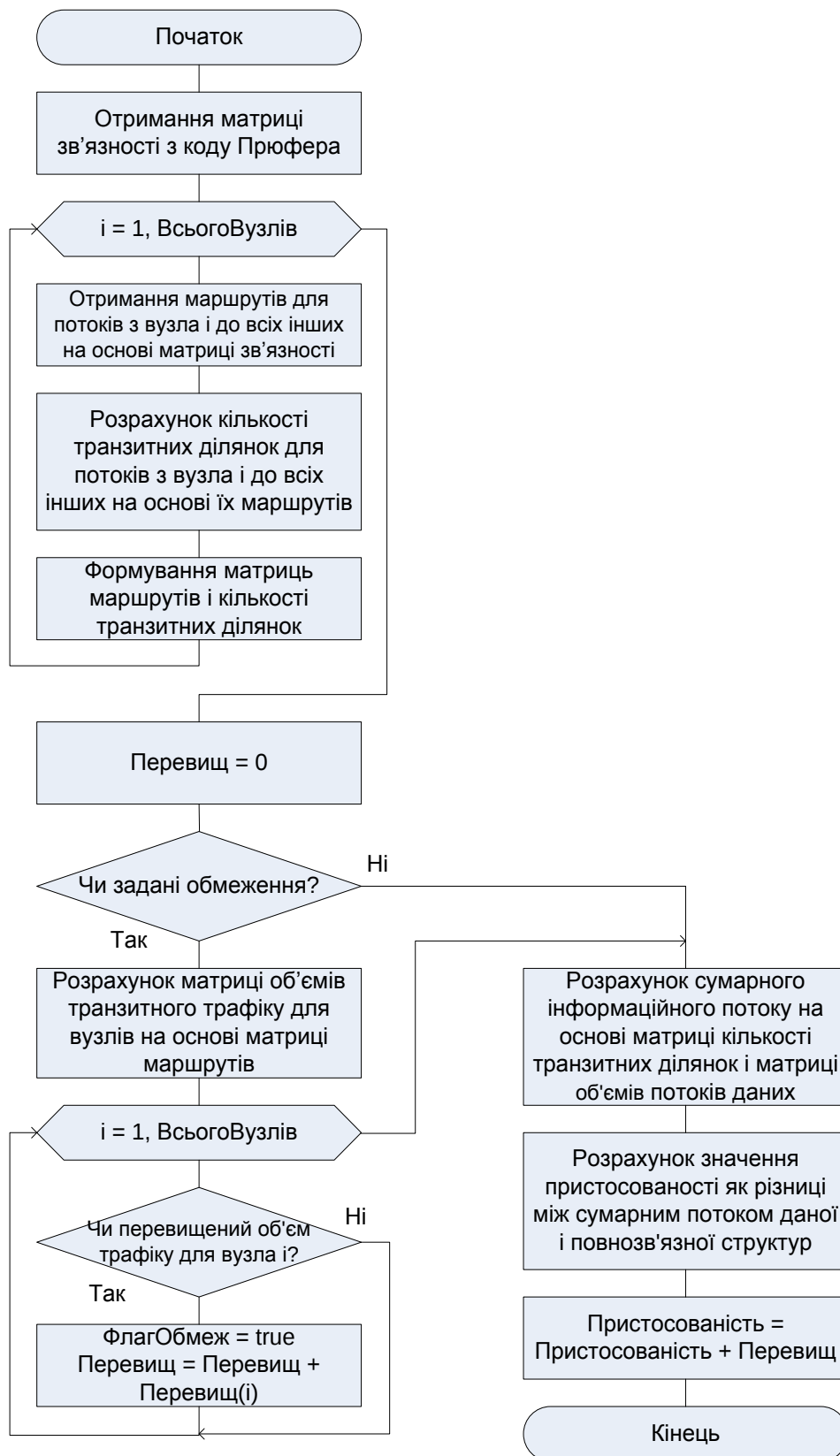


Рис. 11. Алгоритм функції пристосованості

3. Розробка програм реалізації розв'язку задачі оптимізації

На основі алгоритмів, описаних вище, на мові програмування MATLAB була реалізована програма, що виконує оптимізацію структури ІС. Вхідними даними для неї є матриця потоків даних та матриця обмежень трафіку. Результатом роботи програми є набір ребер графу.

Література:

1. Косолапов А.А. Оптимизация распределения информационных потоков при проектировании автоматизированных систем [Текст] // А.А. Косолапов / Оптимизация производственных процессов. - Сб. научн. трудов. - Вып. № 3, Севастополь: изд. СевГТУ, 2000. - С. 143-146.

2. Теорема Кэли о числе деревьев — Википедия. URL: http://ru.wikipedia.org/wiki/Теорема_Кэли_о_числе_деревьев (дата звертання: 4.02.12).

3. Lixia Hanr. A Novel Genetic Algorithm for Degree-Constrained Minimum Spanning Tree Problem/ Lixia Hanr, Yuping Wang. – IJCSNS International Journal of Computer Science and Network Security, Vol. 6 No.7A, July 2006.

4. Jens Gottlieb. Prufer Numbers: A Poor Representation of Spanning Trees for Evolutionary Search / Jens Gottlieb , Bryant A. Julstrom, Günther R. Raidl, Franz Rothlauf. - Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001): 343–350.