ВПЛИВ ТРАНЗИТИВНИХ ДУГ ГРАФА НА ОПТИМАЛЬНІСТЬ В УЗАГАЛЬНЕНІЙ ЗАДАЧІ ПАРАЛЕЛЬНОГО УПОРЯДКУВАННЯ

Поліна Василькович, Валентина Турчина
Дніпропетровський національний університет імені Олеся Гончара

INFLUENCE OF TRANSITIVE GRAPH ARCS ON OPTIMALITY IN THE GENERALIZED PARALLEL ARRANGEMENT PROBLEM

Abstract. The work considers one of the well-known problems of parallel ordering of the vertices of a digraph. These tasks emerge with determining the optimal order of execution of a finite set of tasks, when certain technological restrictions are imposed on the order. Some cases are considered for the well-known distribution algorithm when the presence of transitive arcs affects the accuracy of the obtained result. Also, the rule for adding such arcs to the graph is proposed.

В даній роботі розглядється одна з відомих задач, що виникають при призначанні працівників на скінчену множину робіт за наявності технологічних обмежень на порядок їх виконання так, щоб загальний час виконання був мінімальним. Зручними інструментами, що задають частковий порядок, можуть бути орієнтовані ациклічні графи.

При розв'язанні задач в класичній постановці виходять з припущення, що будь-яке завдання може виконувати будь-який виконавець, а час виконання кожного завдання однаковий [1].

Розглядається наступне узагальнення однієї із класичних задач побудови оптимальних упорядкувань [2].

По заданому графу $G = \{V, U\}$ і вектору $h = (h_1, ..., h_r)$, $r \leq n$, побудувати паралельне упорядкування S мінімальної довжини таке, що $|S[i]| \leq h_i$, $i = 1,r$, і не порушувались технологічні обмеження.

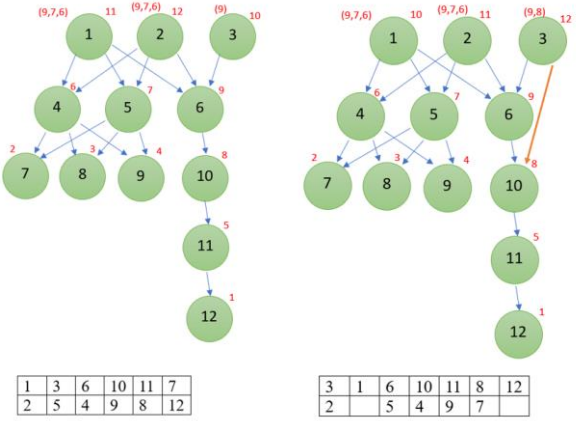
Ця задача, як і класичні задачі, відноситься до класу NP-повних [3].

Серед точних методів, що мають поліномальну складність, для класичної задачі, що відповідає випадку, коли всі значення $h_i = 2$ ефективним є алгоритм заснований на лексикографічному помічуванні вершин [4]. Але в цьому випадку при наявності транзитивних дуг в графі розв'язок може бути наближеним. Це проілюстровано на наступному прикладі з рис. 1.

Тобто ми бачимо, що вилучення транзитивної дуги, яка для даної постановки задачі не є інформативною, оскільки технологічні обмеження на порядок не порушуються, впливає на точність розв'язку.

Виникає питання чи будуть транзитивні дуги впливати на оптимальність розв'язку і для узагальненої задачі, яку ми розглядаємо.

Наявність транзитивних дуг не впливає на порядок, але може впливати на мітки, що отримують вершини графу.



1	3	6	10	11	7
2	5	4	9	8	12

3	1	6	10	11	8	12
2		5	4	9	7	

Рис. 1 – *Приклад 1. Неоптимальне упорядкування після додаванні однієї транзитивної дуги при h = 2*

Іноді у графах на етапі розставлення поміток, виникає неоднозначність вибору для декількох вершин одного рівня. Як виявилося, для узагальненої задачі додавання транзитивних дуг дозволяє позбутися цієї неоднозначності та отримати оптимальне упорядкування при тій самій стратегії помічування вершин.

Розглянемо граф із рис. 2 при $h = (1,3,1,1,1)$.

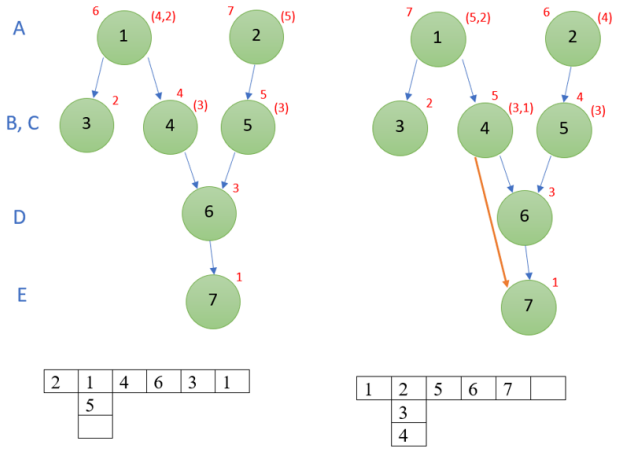


Рис. 2 – *Приклад 2. Побудова оптимального упорядкування при додаванні однієї транзитивної дуги*

Спробуємо прослідкувати залежність у таких випадках.

На прикладі 2 (рис. 2) $h_2 = 3$. Призначити три вершини (2, 3, 4) на друге місце можна тільки якщо на перше місце встановити вершину під номером 1, але у графі без транзитивних дуг на це місце стновиться вершина 2, оскільки має більшу помітку. Така ситуація виникає через те, що вершини 4 та 5 мають однакові лексикографічні послідовності. Додавання однієї транзитивної дуги з вершини 4 до вершини 7 дозволяє уникнути неоднозначності вибору поміток.

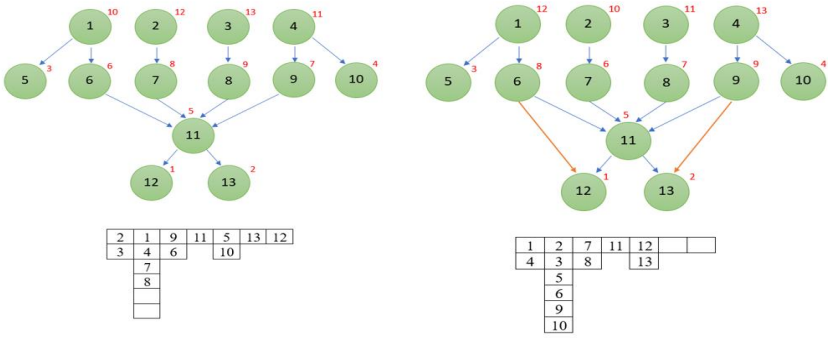


Рис. 3 - *Приклад 3. Додавання двох транзитивних дуг приводить до оптимального упорядкування*

Розглянемо ще один приклад такого впливу транзитивних дуг. На рис. 3 можна побачити, що додавання однієї транзитивної дуги було б недостатнім для того, щоб позбутися неоднозначності. Тут $h_1 = 2$ і, щоб побудувати оптимальне упорядкування, необхідно поставити вершини 2 і 3 на перше місце, тому необхідно додати дві транзитивні дуги, що виходять з вершин 6 і 9.

Тепер, виходячи з цих прикладів, наведемо деякі спостереження та узагальнемо їх у вигляді правила.

Під рівнем будемо вважати умовну лінію, що визначає положення множини вершин, на ній розташованих, відносно критичних шляхів графу.

У графі повинно бути не менше 4 рівнів. Перший з них будемо позначати літерою *A,* а V_1 – множина вершин на ньому. На ньому розташовані вершини, порядок призначення на місця яких буде суттєво впливати на оптимальність упорядкування. Другий рівень *B* іде одразу за першим, V_2 – множина вершин рівня. На рівні *B* окрім вершин, що лежать на критичних шляхах, повинно бути $|V_2| - |V_1|$ вершин, які не мають вихідних дуг і в які входять дуги з вершин рівня *A*. Третій – *C*, це рівень на якому розташовуються вершини, для яких виникає неоднозначність вибору поміток. Зауважимо, що рівні *B* і *C* можуть співпадати. На четвертому рівні *D* повинна бути вершина, у яку входять дуги з вершин рівня *C*, для яких існує неоднозначність вибору. На останньому рівні *E* стоять вершини, у які входять транзитивні дуги. Також зауважимо, що між *B* і *C* та *D* і *E* можуть стояти ще декілька інших рівнів. Із цього випливає наступне правило.

Правило 1.

Нехай в упорядкуванні на місце *i* потрібно поставити h_i вершин з рівня *A* ($h_i < |V_1|$, $h_i < h_{i+1}$), а на *i+1* місце $|V_1| - h_i$ вершин рівня *A*, $|V_2| - |V_1|$ вершин рівня *B*, що не мають вихідних дуг, та h_i вершин рівня *B*, що стоять на критичних шляхах, ($h_{i+1} = |V_1| - h_i + |V_2| - |V_1| + h_i = |V_2|$). За таких умов, може виникнути ситуація, коли h_i вершин рівня *A* повинні мати пріоритетність над іншими вершинами цього рівня, щоб відкрити шляхи для $|V_2| - |V_1| + h_i$ вершин рівня *B*. Цю пріоритетність їм забезпечить додавання h_i транзитивних дуг з вершин рівня *C* до вершин рівня *E*, що стоять на тих самих шляхах, що і вершини, порядок призначення на місця яких буде суттєво впливати на оптимальність упорядкування.

Повернемось до прикладу 2. На рівні *A* стоять вершини 1 і 2, $i = 1$, $h_i = 1$, $|V_1| = 2$, $|V_2| = 3$, $h_{i+1} = 3$. Рівні *B* та *C* співпадають. Для того, щоб відкрити шлях до вершини 3, що не має вихідних дуг, та вершини 4 рівня *B*, що лежить на критичному шляху, треба забезпечити пріоритетність

вершини 1, забезпечивши їй перше місце. Це можна зробити додавши транзитивну дугу з вершини рівня C, що стоїть на одному шляху з вершиною 1, до рівня E. Такою вершиною є вершина 4.

У прикладі 3 аналогічно пріоритетність вершин 1 і 4 встановлюється додаванням дуг з вершин 6 і 9, що стоять на тих самих шляхах.

В наступному прикладі ситуація є менш прозорою.

На прикладі 4 (рис. 4) можна спостерігати, як для встановлення пріоритетності вершин довелося застосувати правило 1 два рази. Маємо наступний вектор h = (2, 5, 1, 4, 1, 1, 1, 1). Так при i = 1, h_1 = 2, встановлюється пріоритетність двох вершин цього рівня, однією з яких обов'язково є вершина 3. Для i = 3, h_3 = 1, встановлюється пріоритетність вершини 5.

Зауважимо, що для кожної транзитивної дуги рівні визначаються окремо. Головне, на що треба звернути увагу, є те, що кількість доданих транзитивних дуг у графі буде рівною сумі h_i, які обираються при кожному застосуванні правила 1 ($h_1 + h_3 = 3$).

Таким чином, справедливе наступне твердження:

Якщо в упорядкуванні на місце i повинно стати h_i вершин з рівня A ($h_i < |V_1|$, $h_i < h_{i+1}$), а на $i+1$ місце $|V_1| - h_i$ вершин рівня A, $|V_2| - |V_1|$ вершин рівня B, що не мають вихідних дуг, та h_i вершин рівня B, що стоять на критичних шляхах, то існує таке додавання транзитивних дуг між відповідними рівнями, яке вплине на пріоритетність вершин попередніх рівнів та відповідно на оптимальність упорядкування.

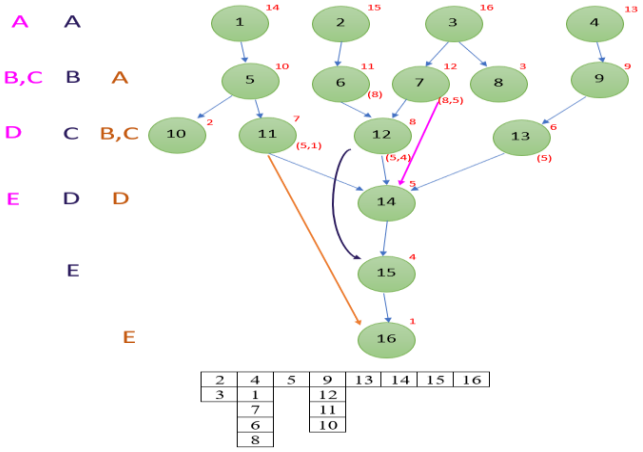


Рис. 4 - *Приклад 4. Застосування правила 1 забезпечує побудову оптимального упорядкування*

У подальшому потребує досліждення питання впливу властивостей обмежуючої послідовності h на кількість транзитивних дуг, які покращують наближені розв'язки.

Література.

1. Василькович П. О., Турчина В. А. Алгоритми побудови паралельних упорядкувань, засновані на аналізі структури графів. *International scientific journal «Grail of Science»*. 2022. №12-13. С. 300–306.
2. Бурдюк, В. Я., Турчина В. А. Алгоритмы параллельного упорядочения: Учебное пособие. Днепропетровск: ДГУ, 1985. 85 с.
3. Ullman J. D. NP-complete scheduling problems. *J. Comput. Systems Sci*., 1975. P. 384– 393.
4. Computer and job-shop scheduling theory / ed. by E.G. Coffman. John Wiley & Sons, 1976. 336 p.