УДК 519.8 doi: 10.15421/321726

В.А. Турчина, Л.Р. Джанашия
Дніпровський національний університет імені Олеся Гончара

АНАЛІЗ ГРАФІВ З ТРАНЗИТИВНИМИ ДУГАМИ ПРИ ПОБУДОВІ ПАРАЛЕЛЬНИХ УПОРЯДКУВАНЬ

При моделюванні виробничих процесів, в яких є відношення часткового порядку, використовуються орієнтовані ациклічні графи. Наявність транзитивних дуг в таких графах не впливає на виробничий процес та може впливати на точність алгоритмів, що розв'язують відповідні оптимізаційні задачі на графах. В роботі розглядаються питання саме такого впливу для задачі в загальній постановці та її часткового випадку.

При моделировании производственных процессов, в которых есть отношение частичного порядка, используются ориентированные ациклические графы. Присутствие транзитивных дуг в таких графах не влияет на производственный процесс, но может влиять на точность алгоритмов, которые решают соответствующие оптимизационные задачи на графах. В работе рассматриваются вопросы именно такого влияния для задачи в общей постановке и ее частного случая.

Oriented acyclical graphs are used for modeling of production processes in which relation of partial order is presented. Transitive arcs in such graphs don't influence on production processes but can influence on accuracy of algorithms which solves optimization problems on graphs. The questions of such impact for problem in general task definition and its particular case are investigated in article.

Ключові слова: задача паралельного упорядкування, транзитивні дуги, лексикографічне порівняння міток вершин.

Вступ. Серед задач дискретної оптимізації одним із цікавих напрямків дослідження є напрям, в якому вивчаються задачі побудови різного типу розкладів і, зокрема, розклади тих завдань, на порядок виконання яких накладаються умови непротирічного слідування. Вони формулюються як задачі упорядкування. В багатьох сферах людської діяльності доводиться приймати рішення по оптимізації, яка буде призводити або до зменшення задіяних ресурсів, або до зменшення часу завершення якихось процесів [2]. На мові теорії упорядкувань ці задачі можна сформулювати як задачі паралельного упорядкування вершин графів. Вони відносяться до класу NP-повних задач і для їх розв'язання виділяють підкласи графів, для яких стає можливим розробити точні алгоритми поліноміальної складності. Виявилось, що один із таких алгоритмів, розроблений для часткового випадку, може бути неточним при наявності в графі транзитивних дуг. Саме такому впливу і присвячена дана робота.

Постановка задачі. Задано граф $G = \{V,U\}$, де $V = \{1, 2, \ldots, n\}$ – множина вершин, а U – множина дуг. Упорядкування S множини V вершин орграфа G

будемо називати паралельним упорядкуванням вершин орграфа G, якщо з того, що $i \in S[p](i,j) \in U$ слідує, що і розміщено в S лівіше j, тобто при $i \in S[p],\ j \in S[q]$ повинно виконуватися $p < q$.

Для існування паралельного упорядкування необхідно і достатньо, щоб граф G не мав петель або орієнтованих циклів, тобто був ациклічним. Існує дві задачі [1]:

1) По заданому графу G, параметру h побудувати допустиме паралельне упорядкування з мінімальним параметром l. Формально будемо записувати цю задачу як S(G,h,l).

2) По заданому графу G, параметру l побудувати допустиме паралельне упорядкування з мінімальним параметром h. Формально будемо записувати цю задачу як S(G,l,h).

Для задачі S(G,2,l) існує точний алгоритм поліноміальної складності [3].

Алгоритм побудови упорядкування на основі лексикографічного порівняння міток.

1-й етап (розстановка поміток). Знаходимо вершини, з яких не виходять дуги, й довільно приписуємо їм помітки, починаючи з 1, так що всі вони отримують різні помітки. Викреслюємо ці вершини й суміжні дуги з графу. Якщо всі вершини помічені, переходимо на другий етап. Знову знаходимо висячі вершини й для кожної такої вершини виписуємо помітки суміжних вершин в порядку спадання. В порядку лексикографічного зростання цих поміток розставляємо нові помітки. Й так, доки не будуть помічені всі вершини.

2-й етап (побудова упорядкування). В порядку спадання поміток виписуємо в список вершини, в які не входять дуги. Викреслюємо їх та суміжні з ними дуги з графу. Ставимо на перше місце в упорядкуванні перші 2 вершини. Якщо кількість вершин менша за 2, то виписуємо одну вершину, залишаючи вільною другу позицію на першому місці в упорядкуванні. Викреслюємо їх зі списку й знову вставляємо в список нові вершини, в які не входять дуги. Описану процедуру повторюємо, доки всі вершини не будуть розставлені. Алгоритм описан.

Вплив транзитивних дуг при розв'язанні задачі S(G,2,l). Якщо в графі відсутні транзитивні дуги, то вибір міток для вершин, з яких не виходять дуги, не впливає на оптимальність упорядкування. Але якщо граф містить транзитивні дуги, то цей порядок може впливати. Розглянемо один із таких прикладів.

Порівнюючи два упорядкування на рис. 1,2, ми бачимо, що в випадку, коли серед вершин, в які не входять дуги, вершина 6 буде мати помітку більшу, ніж вершини 7 та 8, ми отримаємо оптимальне упорядкування. Це відбувається в результаті того, що при вилученні вершини 6 не пізніше 7 та 8 потужність множини, з якої вибираються вершини на наступному кроці, більше ніж 2. Якщо ж на першому кроці помітки вершин 7 та 8 більше за помітку вершини 6, то на наступному кроці потужність множини, з якої проводиться вибір, менша за 2. Звідси можна зробити висновок, що, якщо вершини графа занумеровані так, що на кожному кроці (за вийнятком можливо останнього) потужність множини, з якої буде вестись вибір не менша двух, то транзитивні дуги не вплинуть на оптимальність алгоритму. Тому це треба враховувати, коли розставляємо помітки для вершин останнього рівня.

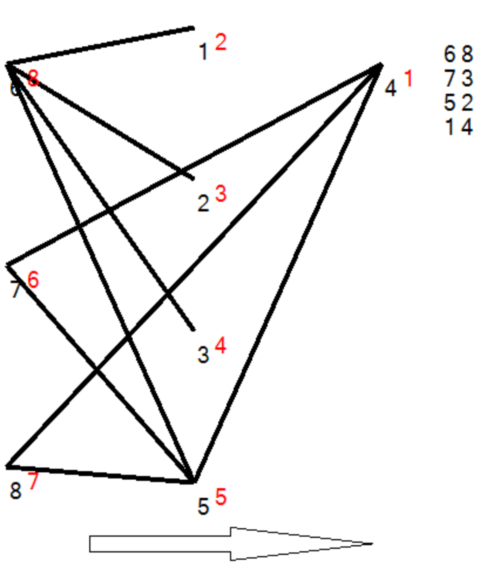

Рис. 1.
Оптимальне упорядкування

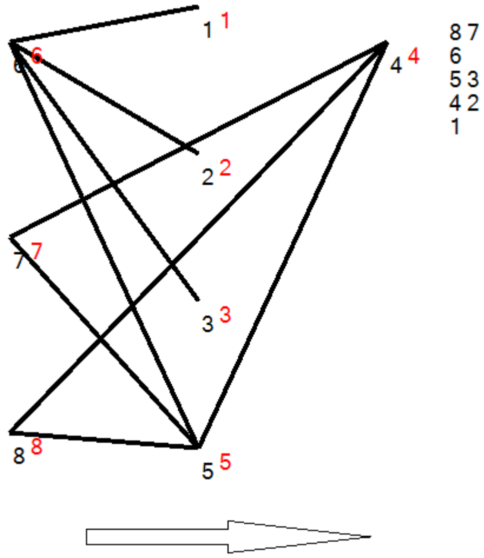

Рис. 2.
Неоптимальне упорядкування

А тепер розглянемо випадок, коли наявність транзитивної дуги не впливає на оптимальність.

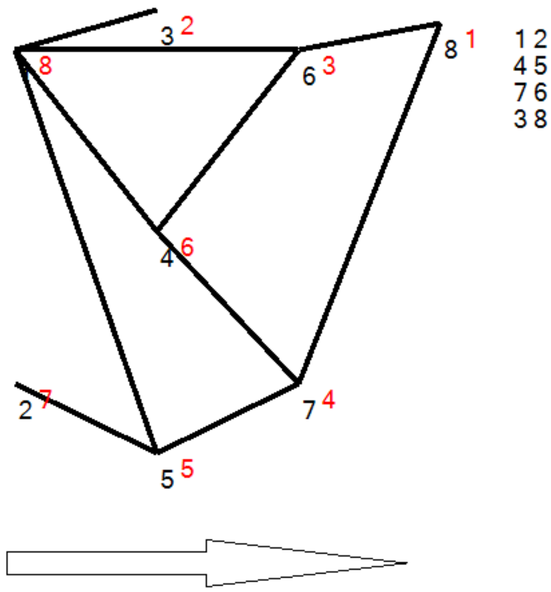

Рис. 3. Оптимальне упорядкування

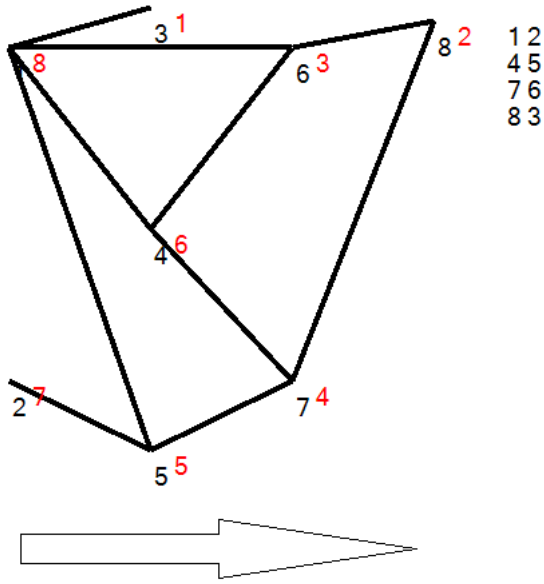

Рис. 4. Оптимальне упорядкування для другого варіанту помічування

В даному випадку наявність транзитивної дуги (1,6) не впливає, оскільки як би ми не помітили вершини, що не мають вхідних дуг, на кожному кроці побудови упорядкування (за вийнятком можливо останнього) потужність множини, з якої буде вестись вибір, не менша двох.

Вплив транзитивних дуг при розв'язанні задачі S(G,h,l). Застосовуючи схему алгоритму, що наведена вище, для розв'язання задачі S(G,h,l), відмітимо, що перший етап залишається без змін, а на другому етапі на кожному кроці ми обираємо не більше ніж h вершин.

Проілюструємо роботу алгоритму на прикладі. Нехай задано орієнтований граф, зображений на рис. 5.

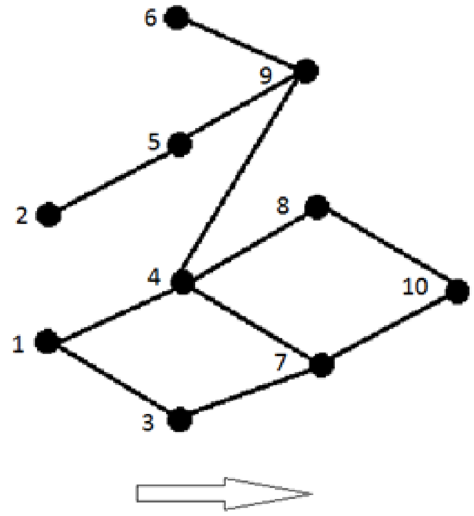

Рис. 5. Заданий орієнтований граф

Стрілка вказує на напрямок дуг.

Необхідно побудувати упорядкування для h = 3.

Розстановка поміток.

Знаходимо вершини, з яких не виходять дуги. В даному випадку це 9 і 10. Помічаємо їх в довільному порядку мітками 1, 2.

Викреслюємо помічені вершини та вхідні в них дуги. Знову шукаємо вершини, з яких не виходять дуги. Відсортуємо їх в порядку лексикографічного неспадання поміток суміжних з ними вершин (тих, які ми видалили на попередньому етапі). Відсортовані вершини помічаємо, продовжуючи нумерацію. Повторюємо цей пункт, доки в графі ще залишилися непомічені вершини. Видаливши вершини 10 та 9, маємо таке упорядкування: 7 – 1, 8 – 1, 5 – 2, 6 – 2, де червоним позначені помітки вершин, що суміжні з відповідними вершинами. Тепер помічаємо їх, продовжуючи нумерацію від 3.

Видаливши дуги (2, 5), (4, 8), (4, 7) та (3, 7), вільними стають вершини 2, 4 та 3. Відсортуємо їх: 3 – 3, 4 – 4, 3, 2 – 5.

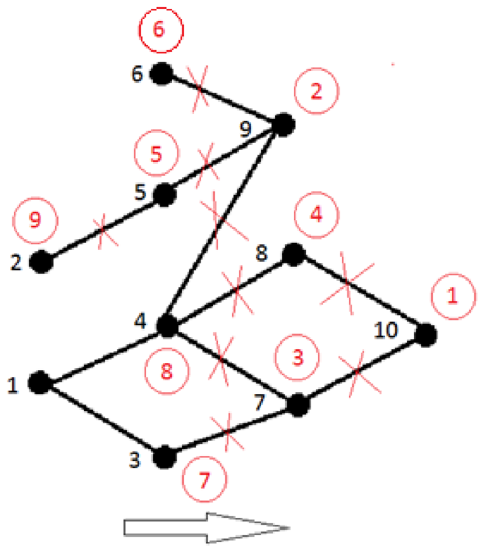

Рис. 6. Граф G на поточному етапі

І нарешті помічаємо останню вершину.

Етап розстановки поміток завершений.

Побудова упорядкування.

Знаходимо вершини, в які не входять дуги. Складаємо із них список в порядку спадання їх поміток. Зі списку беремо перші h вершин та записуємо на перше місце в упорядкуванні. Якщо в списку кількість вершин менша h, то виписуємо їх в упорядкування на перше вільне місце. В іншому випадку на це місце беремо перші h вершин. Видаляємо із списку записані в упорядкування вершини та суміжні з ними дуги. Повторюємо етап 2, доки ще залишилися вершини в графі. На першому етапі маємо такий список: 1 (10), 2(9), 6 (6).

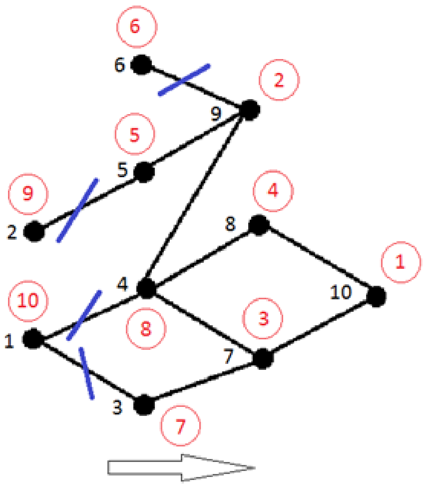

Рис. 7. Поточне помічення вершин

Додаємо вершини в упорядкування й отримуємо порожній список.

Упорядкування: 1, 2, 6.

Наступний список: 4 (8), 3 (7), 5 (5)

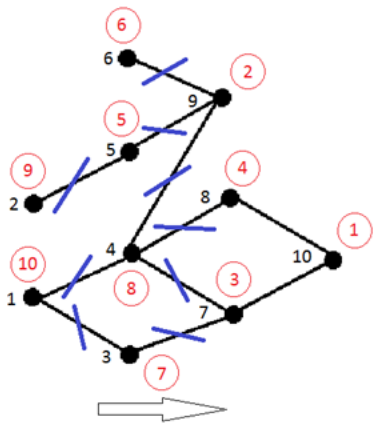

Рис. 8. Поточне помічення вершин

Упорядкування: 1, 2, 6;
4, 3, 5.

Наступний список: 8 (4), 7 (3), 9 (2)

Упорядкування: 1, 2, 6;
4, 3, 5;
8, 7, 9.

І нарешті залишається одна вершина.

Побудоване упорядкування: 1, 2, 6;
4, 3, 5;
8, 7, 9;
10.

Як бачимо, воно є оптимальним. Вплив транзитивних дуг на оптимальність упорядкувань.

Для прикладу розглянемо два випадки:

1. Додавання транзитивної дуги до орієнтованого графу не впливає на оптимальність упорядкування.

2. Додавання транзитивної дуги до орієнтованого графу впливає на оптимальність упорядкування.

1-й випадок. Оптимальне упорядкування для початкового графу без транзитивних дуг:

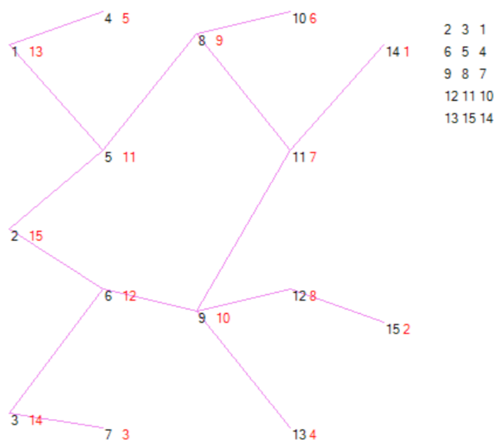

Рис. 9. Початковий граф без транзитивних дуг та його упорядкування

Упорядкування, що побудоване для початкового графа, але з додаванням транзитивної дуги:

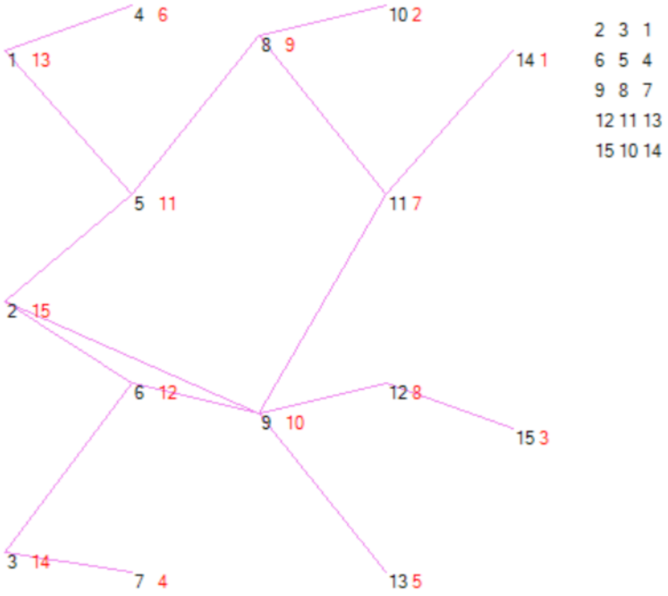

Рис. 10. Граф з транзитивною дугою та його упорядкування

Як бачимо, додавання дуги не вплинуло на оптимальне упорядкування.

2-й випадок. Початковий граф без транзитивних дуг:

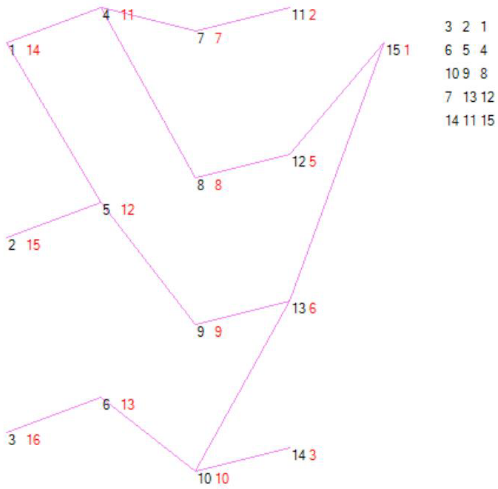

Рис. 11. Початковий граф без транзитивних дуг та його упорядкування

При додаванні транзитивної дуги маємо неоптимальне упорядкування:

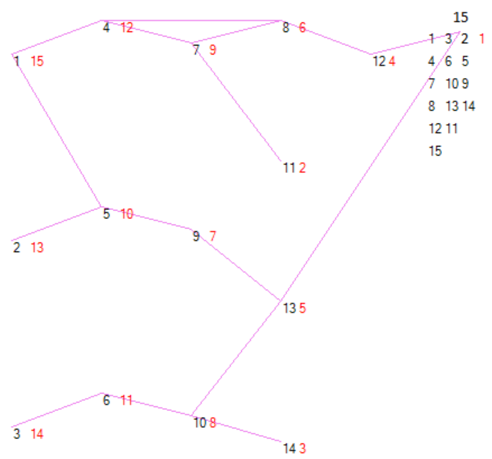

Рис. 12. Початковий граф без транзитивних дуг та його упорядкування

Причому неоптимальне упорядкування отримане для всіх варіантів розстановки поміток для першого рівня вершин.

Висновки. Аналіз графів вказує на необхідність отримання умов, при яких наявність транзитивних дуг не впливає на оптимальність упорядкувань та умов, при яких ця наявність може суттєво впливати на упорядкування.

Бібліографічні посилання

1. **Бурдюк, В.Я.** Алгоритмы параллельного упорядочения: [Текст] / В.Я. Бурдюк, В.А. Турчина– Днепропетровск: ДГУ, 1985. – 84 с.
2. **Воеводин, В.В.** Параллельные вычисления [Текст] / В.В. Воеводин, Вл.В. Воеводин – Санкт-Петербург: БХВ-Петербург, 2002. – 608 с.
3. **Танаев, В. С.** Теория расписаний. Одностадийные системы. [Текст] / В.С. Танаев, В.С. Гордон, Я.М. Шафранский - Москва: Наука, 1984. – 381 с.

Надійшла до редколегії 02.04.2017