

В.А. Турчина, К.Д. Караваєв

Дніпровський національний університет імені Олеся Гончара

ДОСЛІДЖЕННЯ ОЦІНОК ДОВЖИНИ ПАРАЛЕЛЬНОГО УПОРЯДКУВАННЯ ВЕРШИН ГРАФУ

Отримано покращену оцінку знизу часових витрат для задачі пошуку паралельного упорядкування вершин графу з мінімальною довжиною, запропоновано оцінку зверху, обґрунтовано зв'язок задачі із оберненою. На основі обчислювального експерименту досліджено вплив точності оцінки на швидкість знаходження точного розв'язку методом гілок та меж.

Ключові слова: дискретна оптимізація, теорія розкладів, оптимальні упорядкування, метод гілок та меж, оцінювання параметрів упорядкування.

Получена улучшенная оценка снизу временных затрат для задачи поиска параллельного упорядочения вершин графа, имеющего минимальную длину, предложена оценка сверху, обоснована связь задачи с обратной. Посредством вычислительного эксперимента исследовано влияние точности оценки на скорость нахождения точного решения методом ветвей и границ.

Ключові слова: дискретна оптимізація, теорія розкладів, оптимальні упорядкування, метод гілок та меж, оцінювання параметрів упорядкування.

A number of practical tasks require minimizing the human and material resources that are involved in tasks or time expenditures. A special place in this class of problems is occupied by theoretical problems that have a broad practical application, which belong to a class of discrete optimization problems. When minimizing time expenditures in such problems the question of determining the optimal sequencing of execution of a finite set of works (tasks, operations, projects, etc.) is raised. This sequencing can be linear, circular or parallel. The latter is considered by the authors.

This article is devoted to the analysis of one of the problems of discrete optimization, which belongs to the class of problems of the scheduling theory, and, taking into account its specificity, can be considered as an optimization graph problem. Specifically, in terms of the theory of graphs, the problem of finding a parallel sequencing of vertices of a given graph of minimum length, in which at each place there is no more than a given fixed number of vertices, is under consideration. Since this problem is NP-hard, its exact solution can be found by using one of the methods that implements state search scheme. The authors investigated the impact of the accuracy of the estimation of the length of optimal sequencing on the rate of finding the solution by using one of the most common methods, namely the branch and bound method.

As a result, an improved lower-bound estimate of time expenditures was obtained and an upper-bound estimate was proposed. The latter was used to justify the relationship of the problem under consideration with the inverse one. Also, on the basis of the computational experiment results were obtained that refuted the a priori consideration about the impact of the accuracy of the estimation on the rate of finding the exact solution by using the branch and bound method.

Keywords: discrete optimization, scheduling theory, optimal ordering, the branch and bound method, estimating ordering parameters.

Вступ. При розв'язанні оптимізаційних задач розрізняють два випадки: або необхідно знайти лише оптимальний розв'язок, або і розв'язок, і оптимальне значення цільової функції. В задачах, що розглядаються у даній роботі, знаходження оптимального розв'язку є головною задачею, оскільки значення цільової функції при цьому знаходиться елементарно. При пошуку цього розв'язку оцінка значення цільової функції може певним чином впливати на сам пошук розв'язку. Саме такий вплив і досліджується в даній роботі, зокрема у випадку, коли розв'язок шукається за схемою спрямованого перебору.

Постановка задачі. Розглянемо одну з відомих задач упорядкування вершин орієнтованого графу. Нехай задана скінчена множина робіт, на порядок виконання яких накладаються технологічні обмеження [6,7]. В припущенні, що всі роботи мають однаковий час виконання, необхідно визначити мінімальний час, за який, без порушення технологічних обмежень, всі роботи можуть бути виконані заданою кількістю виконавців. Природно задати технологічні обмеження орієнтовним ациклічним графом $G(V,U)$, $|V|=n$, де V – множина вершин, елементам якої поставлені у відповідність роботи. Тоді дуги відповідають технологічним обмеженням. Одна з оптимізаційних задач, що виникає у цьому випадку – це побудова оптимального паралельного упорядкування, під яким розуміють таке розміщення вершин орграфу по місцях, розташованих у лінію, при якому виконуються наступні умови:

- 1) на кожному місці стоїть не більше заданої кількості вершин, яку позначимо через h ;
- 2) якщо пара вершин $(i, j) \in U$, то вершина i розміщується лівіше за j ;
- 3) кількість непорожніх місць (яку ми називаємо довжиною упорядкування і позначаємо l), на якому розміщуються всі вершини орграфу мінімальна.

В загальному випадку, тобто для довільних графів G і ширини h , для знаходження точного розв'язку застосовують схеми спрямованого перебору, зокрема метод гілок та меж [8-10]. При реалізації схем цього методу одним із важливих питань є питання, як оцінити значення цільової функції як на всій допустимій множині, так і на відповідних підмножинах [1,2,4]. Зупинимося на деяких підходах до таких оцінок.

Оцінки, що використовують спеціальні упорядкування. При оцінюванні довжини упорядкування часто використовують два спеціальних упорядкування вершин орграфів \underline{S} та \bar{S} [5]. В цих упорядкуваннях кожна вершина орграфу займає відповідно крайнє ліве (упорядкування \underline{S}) та крайнє праве (\bar{S}) допустиме місце в упорядкуванні, довжина якого дорівнює довжині критичного шляху у графі. Тоді базовою оцінкою будемо вважати

$l^* \geq \max\left(l, \left\lceil \frac{n}{h} \right\rceil\right)$. При більш детальному аналізі графу було отримано

уточнену оцінку $l^* \geq \max_k \left(l, \left\lceil \frac{1}{h} \sum_{i=k}^l |\bar{S}[l-i+1]| \right\rceil + k - 1 \right), k = 1, \dots, l$ [3].

Розглянемо поведінку уточненої оцінки для довжини упорядкування, в якій \bar{S} замінимо на \underline{S} , тобто

$$l = \max_k \left(l, \left\lceil \frac{1}{h} \sum_{i=k}^l |\underline{S}[l-i+1]| \right\rceil + k - 1 \right), k = 1, \dots, l.$$

Порівняння уточненої та зміненої оцінок проведитимемо при $h=3$ для графів з рис. 1-3.



Рис. 1. Граф G1

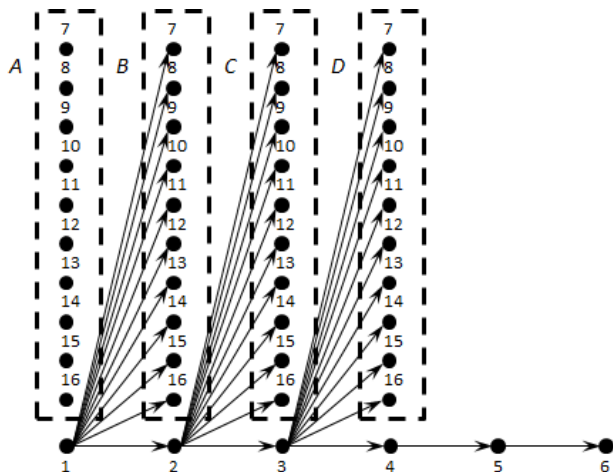


Рис. 2. Графи G2 (A, B, C, D)

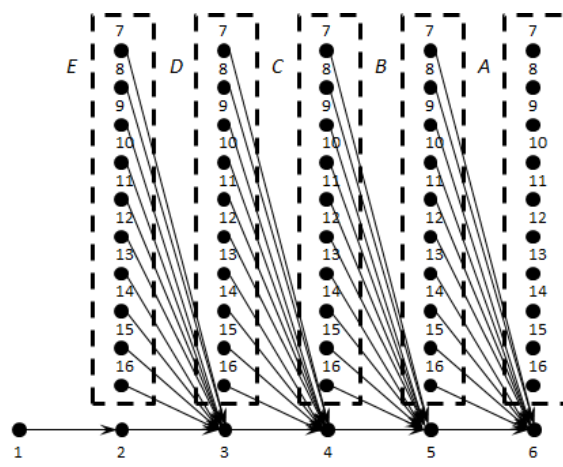


Рис. 3. Графи G3 (A-E)

Почнемо з графу на рис. 1. Очевидно, що його оптимальне упорядкування має вигляд $S^* = \{1;2;3;4;5;6\} = \underline{S} = \bar{S}$, тобто шукана довжина $l^* = 6$. Використовуючи уточнену оцінку отримаємо:

1) для \bar{S} : $l_1 = \max \left\{ \left\lceil \frac{6}{3} \right\rceil; \left\lceil \frac{5}{3} \right\rceil + 1; \left\lceil \frac{4}{3} \right\rceil + 2; \left\lceil \frac{3}{3} \right\rceil + 3; \left\lceil \frac{2}{3} \right\rceil + 4; \left\lceil \frac{1}{3} \right\rceil + 5 \right\} = 6;$

2) для \underline{S} : $l_2 = \max \left\{ \left\lceil \frac{6}{3} \right\rceil; \left\lceil \frac{5}{3} \right\rceil + 1; \left\lceil \frac{4}{3} \right\rceil + 2; \left\lceil \frac{3}{3} \right\rceil + 3; \left\lceil \frac{2}{3} \right\rceil + 4; \left\lceil \frac{1}{3} \right\rceil + 5 \right\} = 6.$

Бачимо, що у цьому випадку $l_1 = l_2 = l^*$.

Розглянемо тепер графи з рис. 2. Випадки A-D відрізняються розташуванням вершин 7-16 та вершиною, з якої йдуть дуги до кожної з зазначених вершин. Помітимо, що, незважаючи на зміну структури, упорядкування \bar{S} для усіх графів буде однаковим, воно буде відрізнятись від випадку графу з рис.1 тим, що на останньому місці будуть також

знаходиться вершини 7-16. А отже, l_1 для всіх випадків буде однаковим і дорівнюватиме: $l_1 = 6$. l_2 та l^* для різних випадків будуть різними, тому розглянемо далі кожний випадок окремо.

Випадок А. Оптимальним упорядкуванням графу є наприклад таке $S^* = \left\{ \begin{array}{c} 1 \ 2 \ 3 \ 4 \ 5 \\ 7, 9, 11, 13, 15, 6 \\ 8 \ 10 \ 12 \ 14 \ 16 \end{array} \right\}$, отже $l^* = 6$. Упорядкування \underline{S} буде відрізнятися від

упорядкування у випадку графу з рис. 1 тим, що на першому місці будуть знаходитись вершини 1 та 7-16, тоді $l_2 = 9$.

Випадок В. Отримаємо $l^* = 6$. На відміну від випадку А, вершини 7-16 будуть знаходитись у \underline{S} на другій позиції, тоді для l_2 отримаємо: $l_2 = 8$.

Випадок С. У цьому випадку $l^* = 7$. \underline{S} , аналогічно до випадків А і В, буде містити вершини 7-16 на третьому місці. Для l_2 маємо: $l_2 = 8$.

Випадок Д. Довжина оптимального упорядкування $l^* = 8$, у \underline{S} вершини 7-16 стоять на четвертій позиції. Для l_2 отримаємо: $l_2 = 7$.

Графи з рис. 3, аналогічно до графів з рис. 2, відрізняються лише розташуванням вершин 7-16 та вершиною, до якої йдуть дуги з усіх зазначених вершин. На відміну від графів з рис. 2, для цих графів упорядкування \underline{S} буде однаковим і відрізнятиметься від випадку графу з рис. 1 тим, що міститиме вершини 7-16 на першому місці. Тоді l_2 для таких графів приймає сталі значення: $l_2 = 9$. Як і для графів з рис. 2, l_1 та l^* матимуть різні значення, тому дослідимо кожний граф окремо.

Випадок А. Довжина оптимального упорядкування $l^* = 6$. Упорядкування \bar{S} буде відрізнятися від упорядкування для графу з рис. 1 тим, що вершини 7-16 у ньому стоять на останньому місці, тому $l_1 = 6$.

Випадок В. У цьому випадку $l^* = 6$. У \bar{S} вершини 7-16 знаходяться на передостанній (п'ятій) позиції, а $l_1 = 6$.

Випадок С. Маємо $l^* = 7$. В упорядкуванні \bar{S} вершини 7-16 знаходяться на четвертому місці, аналогічно до випадків А та В. Тоді $l_1 = 7$.

Випадок Д. Отримаємо $l^* = 8$. Аналогічно до попередніх випадків, у цьому випадку вершини 7-16 будуть знаходитись у \bar{S} на третій позиції і $l_1 = 8$.

Випадок Е. Для цього графу $l^* = 8$. Аналогічно, у \bar{S} вершини 7-16 розташовані на другій позиції, $l_1 = 8$.

З аналізу поведінки значень l_1 та l_2 для графів з рис. 1-3 можна зробити такі висновки:

1) оцінку у модифікованому вигляді не можна використовувати окремо ані для оцінювання довжини оптимального упорядкування знизу, ані зверху, оскільки як продемонстровано для графів з рис. 2, вона може бути як більшою, так і меншою за оптимальне значення.

2) уточнена оцінка не завжди має необхідну властивість адаптації під зміну структури графу, як це можна побачити для графів з рис. 2. Це, очевидно, вповільнює пошук розв'язку методом гілок та меж, що говорить про можливість та необхідність її покращення.

3) можна припустити, що для більшості графів хоча б одна з оцінок зміниться при тій чи іншій модифікації множини дуг графу.

Це нашо вухує на думку, що можливо потрібно якось об'єднати упорядкування \bar{S} та \underline{S} у одній оцінці.

Уточнення оцінок для довжини упорядкування. Розглянемо уточнену оцінку знизу. Під графом (позначимо його $\hat{G}(V, \hat{U})$), розгорнутим до графу $G(V, U)$, будемо розуміти граф, у якому змінена орієнтація всіх дуг.

Розгорнуте упорядкування до упорядкування S позначатимемо \hat{S} (для них виконується $S[i] = \hat{S}[l - i + 1]$, де l – довжина упорядкування S).

Твердження 1. Якщо S допустиме упорядкування для графу G при заданому h , то \hat{S} – допустиме упорядкування для \hat{G} при тому ж h .

Доведення. Від супротивного. Нехай S – допустиме для G , а \hat{S} не є допустимим для \hat{G} . Тоді знайдеться дуга $(c, d) \in \hat{U}$ така, що d знаходиться у \hat{S} раніше, аніж вершина c , але тоді у S вершина d знаходиться пізніше, ніж вершина c , що неможливо, оскільки $(d, c) \in U$ за визначенням \hat{G} та S – допустиме упорядкування. Отже, отримали протиріччя, що впливає з неправильного припущення. ■

Наслідок 1. Якщо S^* – оптимальне упорядкування для G , тоді \hat{S}^* – оптимальне упорядкування для графу \hat{G} при заданому h .

Доведення. Від супротивного. Нехай S^* – оптимальне упорядкування для G при заданому h , проте \hat{S}^* не є оптимальним для графу \hat{G} . Тоді існує таке упорядкування \hat{S}' , що $l(\hat{S}') < l(\hat{S}^*)$. Тоді з твердження 1 випливає, що S' – допустиме для G і $l(S') < l(S^*)$, значить S^* не є оптимальним. Отже, отримали протиріччя, що впливає з неправильного припущення. ■

Наслідок 2. Довжини оптимальних упорядкувань для графів G та \hat{G} співпадають при заданому h .

Доведення. Впливає безпосередньо з наслідку 1. ■

Твердження 2. $l = \max_k \left(l, \left\lceil \frac{1}{h} \sum_{i=k}^l |S[i]| \right\rceil + k - 1 \right), k = 1, \dots, l$ є оцінкою знизу

для довжини оптимального упорядкування l^* для графу G та заданого h .

Доведення. Розглянемо граф G . З означення графу \hat{G} випливає, що вершинам без вхідних дуг з G відповідають вершини без вихідних дуг у \hat{G} і навпаки, тоді \bar{S} для графу \hat{G} співпадає з розгорнутим \underline{S} для графу G . Розглянемо оцінку знизу для довжини оптимального упорядкування для

графу \hat{G} : $l^* \geq \max_k \left(l, \left\lceil \frac{1}{h} \sum_{i=k}^l |\bar{S}[l-i+1]| \right\rceil + k - 1 \right), k = 1, \dots, l$. Тоді, оскільки \bar{S} для \hat{G} співпадає із \hat{S} , де упорядкування \underline{S} відповідає графу G , і $\hat{S}[l-i+1] = \underline{S}[i]$, отримаємо для графу \hat{G} : $l^* \geq \max_k \left(l, \left\lceil \frac{1}{h} \sum_{i=k}^l |\underline{S}[i]| \right\rceil + k - 1 \right), k = 1, \dots, l$. Але, тоді за наслідком 2, це справедливо і для графу G . ■

Твердження 3. $l = \max \left(l, \max_k \left(\left\lceil \frac{1}{h} \sum_{i=k}^l |\bar{S}[l-i+1]| \right\rceil + k - 1 \right), \max_k \left(\left\lceil \frac{1}{h} \sum_{i=k}^l |\underline{S}[i]| \right\rceil + k - 1 \right) \right), k = 1, \dots, l$ – оцінка знизу для довжини оптимального упорядкування l^* для графу G та заданого h .

Доведення. Впливає з того, що кожен з аргументів зовнішнього максимуму є оцінкою знизу, та властивості максимуму: якщо $c \geq a$ і $c \geq b$, то $c \geq \max(a, b)$. ■

Перейдемо до оцінки зверху. Використовуючи, наприклад, упорядкування \underline{S} побудуємо нове упорядкування S' за наступним правилом: якщо $\underline{S}[i]$ містить не більше h вершин, то залишаємо її без змін, інакше розділимо її на $\lceil |\underline{S}[i]|/h \rceil$ множин, у кожній з яких буде не більше h вершин. Отримане упорядкування є допустимим, оскільки отримане з \underline{S} , а отже відношення порядку не порушується, і на кожній позиції стоїть не більше h вершин. Тоді його довжина буде не меншою аніж довжина оптимального, оскільки воно є допустимим. Отже, можемо використати її як оцінку зверху для довжини оптимального упорядкування, тобто: $l^* \leq \sum_{i=1}^l \lceil |\underline{S}[i]|/h \rceil$.

Нескладно побачити, що аналогічні міркування можна застосувати і до \bar{S} , і до будь-якого іншого упорядкування, у якому не порушується відношення порядку. Тоді можна дещо уточнити оцінку зверху, використовуючи \bar{S} : $l^* \leq \min \left(\sum_{i=1}^l \lceil |\underline{S}[i]|/h \rceil, \sum_{i=1}^l \lceil |\bar{S}[i]|/h \rceil \right)$.

Можна припустити, що чим менше вершин знаходиться на кожному місці в упорядкуванні, до якого можна застосувати наведені міркування, тим ближче отримана оцінка буде до істинного значення. Поставимо мету серед усіх таких упорядкувань із заданою довжиною l знайти таке, у якого на кожному місці буде мінімальна кількість вершин. Розв'язок задачі $S(G, l, h)$ якраз і буде цим шуканим упорядкуванням. У якості l можна використати будь-яку оцінку знизу для довжини оптимального упорядкування. Зрозуміло, що чим точнішою буде оцінка знизу, тим менше буде h^* , а отже оцінка зверху буде точнішою. Отже, оцінка зверху набуває вигляду:

$l^* \leq \sum_{i=1}^l \lceil |S^*[i]|/h \rceil$, де S^* – оптимальне упорядкування для задачі $S(G, l, h)$, де $l \leq l^*$.

Помітимо, що останнє співвідношення не лише показує зв'язок між оцінками зверху та знизу для довжини оптимального упорядкування, а ще й зв'язок між їх точністю. Окрім цього, воно показує деякий зв'язок між задачами $S(G, l', h)$ та $S(G, h', l)$ при деяких l' та h' , а у деяких випадках і між їх алгоритмічними складностями. Так, наприклад, якщо оптимальне упорядкування для графу G є щільним, то оцінка $l^* \geq \lceil n/h \rceil$ є точною, а тоді, якщо взяти $l = \lceil n/h \rceil$ у задачі $S(G, l, h)$, то отримаємо оцінку зверху $l^* \leq \lceil n/h \rceil$, оскільки щільне упорядкування існує. А тоді задачі пошуку щільних упорядкувань для задач $S(G, l', h)$ та $S(G, h', l)$ мають однакову алгоритмічну складність.

Порівняльний аналіз оцінок. Проаналізуємо поведінку відомих оцінок знизу довжини упорядкування

$$l^* \geq \max \left(l, \left\lceil \frac{n}{h} \right\rceil \right), \quad l^* \geq \max_k \left(l, \left\lceil \frac{1}{h} \sum_{i=k}^l |S[l-i+1]| \right\rceil + k - 1 \right), \quad k = 1, \dots, l$$

та запропонованої оцінки

$$l^* \geq \max \left(l, \max_{1 \leq k \leq l} \left(\left\lceil \frac{1}{h} \sum_{i=k}^l |S[l-i+1]| \right\rceil + k - 1 \right), \max_{1 \leq k \leq l} \left(\left\lceil \frac{1}{h} \sum_{i=k}^l |S[i]| \right\rceil + k - 1 \right) \right).$$

Незважаючи на різну «точність» оцінок, в одних випадках вони можуть давати один і той самий результат, а у деяких уточнені оцінки можуть зменшити кількість операцій, необхідних для знаходження оптимального упорядкування.

Можна припустити, що чим точнішою є оцінка, тим менше знадобиться розгалужень аби знайти оптимальний розв'язок.

Перевіримо цю гіпотезу, провівши наступний обчислювальний експеримент за допомогою розробленої програми. Будемо генерувати випадковий довільний граф або граф, де усі вершини знаходяться на критичних шляхах, із заданою кількістю вершин. Для пошуку його оптимального упорядкування застосуємо алгоритм гілок та меж з різними оцінками знизу для довжини. І порівняємо кількість вершин у отриманому дереві варіантів. Проведемо попарне порівняння. Розглядаємо графи з кількістю вершин від 10 до 100, причому розділимо цей проміжок на 4: [10, 20], [21, 40], [41, 60] та [61, 100]. Для кожного проміжку ширина упорядкування обирається довільно, але не перевищує половини верхньої грані кількості вершин у проміжку. Для кожного проміжку проведено по 10000 експериментів. Виконання алгоритму зупиняється, якщо кількість вершин у дереві перевищує у 30 разів верхню межу кількості вершин у проміжку в перших трьох випадках та у 10 разів для останнього випадку (зменшення пов'язано з тим, що таке обмеження помітно не впливає на відсоток графів, для яких за цих умов не встигаємо отримати оптимальне

упорядкування, проте дуже впливає на час виконання програми). Усі попередні умови підсумовані у вигляді таблиці 1.

Результати усіх порівнянь наведені у таблиці 2. Результати містять кількість тестових випадків, в яких алгоритм при використанні першої оцінки завершився швидше (потребував менше вершин у дереві варіантів аби отримати оптимальне упорядкування), в яких завершився швидше, при використанні другої оцінки, та в яких алгоритм потребував однакової кількості розгалужень. Окрім цього, результати містять частку випадків, для яких алгоритм не встиг завершитись за задану максимальну кількість розгалужень.

Таблиця 1

Значення параметрів, що використовувались при тестуванні

n	h	Кількість тестів	Кількість вершин у дереві
[10, 20]	[2, 10]	10000	600
[21, 40]	[2, 20]	10000	1200
[41, 60]	[2, 30]	10000	1800
[61, 100]	[2, 50]	10000	1000

Таблиця 2

Результати тестування для усіх пар оцінок

Перемогла оцінка (1)	Нічия	Перемогла оцінка (2)	Частка незавершених
Базова(1) оцінка та уточнена(2) оцінка (випадкові)			
0	9683	317	0,0217
2	9661	337	0,0703
1	9718	281	0,0947
0	9888	112	0,1033
Базова(1) оцінка та уточнена(2) оцінка (критичні)			
0	9551	449	0,0325
1	9619	380	0,0918
0	9731	269	0,1349
0	9876	124	0,1293
Базова(1) оцінка та запропонована(2) оцінка (випадкові)			
2	9565	433	0,0227
4	9558	438	0,0701
2	9639	359	0,0875
0	9842	158	0,1002
Базова(1) оцінка та запропонована(2) оцінка (критичні)			
0	9410	590	0,0256
2	9426	572	0,0903
2	9576	422	0,1282
1	9852	147	0,1167

Продовження таблиці

Уточнена(1) оцінка та запропонована(2) оцінка (випадкові)			
0	9922	78	0,0239
0	9893	107	0,0614
0	9892	108	0,0904
0	9949	51	0,1005
Уточнена(1) оцінка та запропонована(2) оцінка (критичні)			
0	9743	257	0,0254
3	9780	217	0,0907
3	9867	130	0,1274
0	9975	25	0,1252

З результатів тестування можна помітити наступне:

1) у переважній більшості випадків (94%) для всіх пар оцінок методу гілок та меж знадобилася однакова кількість розгалужень, тобто суто статистично покращення у якості оцінювання є незначним.

2) зі збільшенням кількості вершин графу кількість випадків, у яких точніші оцінки перемагають, зменшується. Для графів, де кількість вершин змінюється у проміжку [60;100], кількість таких випадків не перевищує 2%.

3) існують графи, для яких оптимальне упорядкування можна отримати за меншу кількість кроків, якщо використовувати менш точну оцінку. Наприклад, для графу з рис. 4 можна отримати оптимальне упорядкування при $h=5$ у 3 рази швидше, якщо замість уточненої оцінки використовувати базову (879 проти 290).

Можемо зробити наступні висновки щодо результатів тестування: базова оцінка майже завжди не поступається більш точним, а іноді й перевершує їх; якість оцінок спадає із збільшенням числа вершин графу, а отже і з ускладненням їх структури, тому використання простіших оцінок є доцільнішим, оскільки їх знаходження потребує меншого числа операцій; існують такі графи, які суперечать принципу «точніше – краще», що говорить про те, що, можливо, навіть для абсолютно точної оцінки знайдуться приклади, для яких метод гілок та меж знаходить оптимальне упорядкування повільніше, аніж при використанні гірших оцінок.

Такий ефект пов'язаний із тим, що обираються вершини і відповідні упорядкування, які спочатку мали менші оцінки через неточність оцінки, але на шляху до оптимального упорядкування потребували менше розгалужень. З останнього випливає, що, можливо, краще розглядати не просто оцінки знизу, а деякі метрики, які не обов'язково є оцінками довжини оптимального упорядкування, проте для яких виконується, що кращі (описані вище) упорядкування мають менші значення метрики.

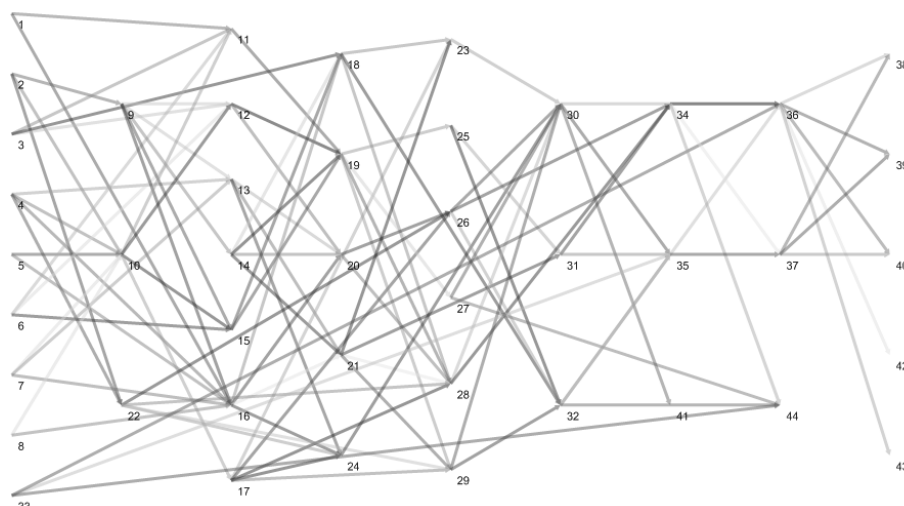


Рис. 4. Приклад графу, для якого базова оцінка знаходить оптимальне упорядкування швидше ніж уточнена

Висновки. Отримані результати дозволяють зробити висновок, що не завжди більш точна оцінка довжини упорядкування приводить до зменшення кількості вершин у дереві варіантів, що ілюструє роботу методу гілок та меж. Тому невирішеним питанням залишається апріорне отримання точного значення оптимальної довжини упорядкування.

Бібліографічні посилання

1. **Fernandez E.B.** Bounds on the number of processors and time for multiprocessor optimal schedules [Text] / E.B. Fernandez, B. Busspell // IEEE Trans. Comput. – 1973. – С–22. – Р. 745–751.
2. **Graham R.L.** Bounds on multiprocessing timing anomalies [Text] / R.L. Graham // SIAM J. Appl. Math. – 1964. – № 2. – Р. 416–429.
3. **Hu T.C.** Parallel sequencing and assembly line problems [Text] / T.C. Hu // Oper. Res. – 1961. – Р. 841–849.
4. **Бекишев Г.А.** Обоснование алгоритма Т.С. Ху [Текст] / Г.А. Бекишев // Матем. анализ и его приложения. – Ростов–н/Д. – 1974. – Т.5. – С. 120–126.
5. **Бурдюк В.Я.** Алгоритмы параллельного упорядочения [Текст] : учебное пособие / В.Я. Бурдюк, В.А. Турчина. – Днепропетровск: ДГУ, 1985. – 84 с.
6. **Танаев В.С.** Введение в теорию расписаний [Текст] / В.С. Танаев, В.В. Шкурба. – М.: Наука, 1975. – 256 с.
7. **Танаев В.С.** Теория расписаний. Групповые технологии [Текст] / В.С. Танаев, М.Я. Ковалев, Я.М. Шафранский. – Мн.: Университетское, 1998. – 290 с.
8. **Тушкина Т.А.** Методы и программы генерирования случайных графов [Текст] / Т.А. Тушкина. – Зап. науч. семинаров ЛОМИ АН СССР. – 1975. – № 54. – С. 259–264.
9. **Шахбазян К.В.** Метод ветвей и границ для задачи параллельного упорядочивания [Текст] / К.В. Шахбазян, Т.А. Тушкина. – Зап. науч. семинаров ЛОМИ АН СССР. – 1973. – № 35. – С. 146–155.
10. **Шахбазян К.В.** Обзор методов составления расписаний для многопроцессорных систем [Текст] / К.В. Шахбазян, Т.А. Тушкина. – Зап. науч. семинаров ЛОМИ АН СССР. – 1975. – № 54. – С. 229–258.

Надійшла до редколегії 16.06.2018.