УДК 519.854.2

ШЕВЧЕНКО К.Ю.

АЛГОРИТМ ГІЛОК ТА МЕЖ ДЛЯ СТАТИСТИЧНИХ ДОСЛІДЖЕНЬ НОВОГО ПДС-АЛГОРИТМУ РОЗВ'ЯЗАННЯ ЗАДАЧІ МІНІМІЗАЦІЇ СУМАРНОГО ЗВАЖЕНОГО ЗАПІЗНЕННЯ ВИКОНАННЯ РОБІТ НА ОДНОМУ ПРИЛАДІ

Зроблений огляд сучасного стану досліджень задачі мінімізації сумарного зваженого запізнення виконання робіт на одному приладі. Пропонується ефективний алгоритм гілок та меж її розв'язання, призначений для використання у статистичних дослідженнях характеристик нового ПДС-алгоритму.

Provided here is a state-of-the-art survey of research results on minimizing total weighted tardiness of processing jobs on a single machine. We suggest an effective branch-and-bound algorithm for solving this problem and which is intended to be used in statistical studies of characteristics of new PDC-algorithm.

Вступ

В багатьох прикладних сферах діяльності задоволення директивних строків виконання завдань та уникання штрафів за запізнення є найважливішою метою планування. Вартість запізнення доставок, яку формують незадоволення клієнту та упущена вигода, різна не тільки для різних замовлень, а й для різних клієнтів. Таким чином повне врахування стратегічних пріоритетів компанії потребує використання інформації про важливість клієнтів під час складання розкладів виконання завдань на виробничих дільницях, що обумовлює оперування зваженими штрафами за запізнення (див. напр. [1]). Іншим приводом до використання критеріїв, в яких фігурує зважене запізнення – різні пріоритети устаткування (машин), що складають фонди підприємства.

В цій статті розглядається задача мінімізації сумарного зваженого запізнення виконання множини робіт на одному приладі в рамках теорії календарного планування. Один прилад являє собою найпростіше машинне середовище. Таке середовище важливе за декількома причинами. По-перше, воно є частковим випадком усіх інших середовищ. Моделі на єдиній машині часто мають властивості, яких не мають ні елементарні середовища, що складаються з паралельних машин, ні багатостадійні конвеєрні середовища. Результати, які отримані для однієї машини (приладу), можуть служити потужними евристиками для побудови розкладів у більш складних машинних середовищах. Також на практиці, задачі операційного планування в більш складних машинних середовищах часто можуть декомпозуватись на підзадачі, кожна з яких моделюється одним приладом. Складне машинне середовище, в якому єдина машина складає вузьке місце відносно критеріїв продуктивності, в моделі календарного планування може бути замінене однією цією машиною.

Лоулер (Lawler) показав, що розглянута задача є NP-складною в сильному розумінні зведенням до 3-розбиття та запропонував псевдополіноміальний алгоритм динамічного програмування для розв'язання її часткового випадку [2].

Найбільш ефективними існуючими точними методами розв'язання загального випадку задачі є методи математичного програмування, що базуються на перегляді області визначення. Ці методи використовують так звані правила переваги (правила домінування), застосування яких полягає в тому, що в ході розв'язання будується відношення часткового порядку на множині робіт, яке відповідає деякому оптимальному розкладу та залежить від конкретних значень параметрів поточного екземпляру задачі.

Найпершим запропонованим і найпростішим правилом домінування є лема Ельмаграбі [3]. Додаткові правила домінування для задачі мінімізації зваженого запізнення були започатковані Ріннуйєм Каном (Rinnooy Kan) та ін. [4]. Актурк та Їлдірім наводять правило домінування для робіт, обробка яких починається після певного моменту часу, який визначається з параметрів цих робіт [5]. Канет (Kanet) зовсім нещодавно навів більш ефективні правила домінування, до яких зводяться більшість правил Ріннуйя Кана та ін., а також запропонував зо-

всім нові правила для пар робіт, ваги яких не узгоджені [6].

В даній статті приділяється увага результатам, які можуть бути використані у алгоритмі, що побудований за схемою гілок та меж. Для того, щоб повністю описати алгоритм гілок та меж, потрібно визначитись у виборі наступних елементарних процедур (що разом складають стратегію алгоритму гілок та меж – для задачі мінімізації):

- *процедури галуження* конструктивного пошуку, тобто способу розбиття множини допустимих розв'язків у вершині дерева пошуку;
- *процедури оцінювання знизу* множин допустимих *розв'язків* у вузлах дерева пошуку.

В деяких випадках виділяють окрему процедуру оцінювання зверху множин допустимих розв'язків. В даній роботі ця процедура стандартна та полягає у фіксуванні рекорду – найкращого значення критерію листового вузла дерева пошуку.

Відомі нижні оцінки Швімера (Shwimer) мінімаксу запізнення [7], Гелдерса (Gelders) та Клайндорфера (Kleindorfer) [8, 9], обчислення якої зводиться до розв'язання транспортної задачі лінійного програмування, Ріннуйя Кана та ін. [4] розв'язання задачі про призначення, Фішера (Fisher) [10] субградієнтного спуску, Поттса (Potts) та Ван Вассенгова (Van Wassenhove) лагранжевої релаксації [11], Актурка (Akturk) та Їлдіріма (Yildirim) на основі їх нового правила домінування [5], Хугевена (Hoogeveen) і Ван де Вельде (Van de Velde) [12] як посилення оцінки Поттса та Ван Вассенгова застосуванням фіктивних змінних, ін. Незалежні дослідження Абдул-Раза (Abdul-Razaq) та ін. [13] показали, що найбільш ефективною є нижня оцінка Поттса та Ван Вассенгова через її дуже просте обчислення за лінійним методом підбору множників, найбільш строгою – оцінка Гелдерса та Клайндорфера.

Нижня оцінка Поттса та Ван Вассенгова потребує використання деякої евристики для побудови наближеного розкладу в кожному вузлі дерева пошуку. Найбільш ефективною за результатами обчислювальних експериментів [5] виявилась евристика очевидного штрафу за запізнення (apparent tardiness cost, ATC), що була запропонована Вепсалайненом (Vepsalainen) та Мортоном (Morton) в [14] та яка дає близькі до оптимальних послідовності робіт.

В [15] представлений новий підхід до розв'язання окремих NP-складних задач календарного планування та дозволяє знаходити точні розв'язки задач набагато більших розмірностей, ніж було можливо до цього будь-яким іншим методом. Його суттю є попереднє дослідження властивостей відповідних класів важкорозв'язних задач, доведенні положень та правил, які дозволяють використати єдиний принцип обчислень при побудові так званих ПДС-алгоритмів (алгоритмів із поліноміальною та декомпозиційною складовими). Для кожного такого алгоритму виводяться логіко-аналітичні умови, які свідчать про можливість розв'язання задачі поліноміальним підалгоритмом («p-умови»), та відповідно «d-умови», при виконанні яких в процесі розв'язання довільної індивідуальної задачі вона строго декомпозується на підзадачі меншої розмірності. Ці умови перевіряються під час розв'язання екземпляру задачі з класу, для якого побудовано ПДС-алгоритм.

Один варіант ПДС-алгоритму розв'язання досліджуваної задачі був запропонований в [16]. Для оцінки ефективності цього, а також будь-якого іншого нового алгоритму, виконують статистичну обробку даних, отриманих в ході обчислювальних експериментів над побудованими реалізаціями алгоритмів. Характеристики ефективності нових алгоритмів важливо порівнювати із аналогічними для відомих алгоритмів, але при цьому останні мають враховувати усі поточні результати вивчення відповідного класу задач. Коректність алгоритмів перевіряється на наборах розв'язаних індивідуальних задач. Ресурс [17] містить базу індивідуальних задач мінімізації сумарного зваженого запізнення та їх оптимальних (найкращих відомих) розв'язків розмірності до 100.

Для задачі мінімізації сумарного зваженого запізнення подібні експерименти щодо алгоритмів гілок та меж проводились в роботі [13] та не враховували отримані пізніше результати, такі як [4], [18], [5]. В [6] автор наголошує на тому, що із запровадженням нових правил домінування оцінка Гелдерса та Клайндорфера може конкурувати із оцінкою Поттса та Ван Вассенгова через те, що остання є доволі слабкою навіть після посилення за рахунок введення фіктивних змінних за зразком [12].

Дана стаття носить методологічний характер, та описує алгоритм гілок та меж, що використовує нові правила домінування та модифіковані процедури оцінювання. В якості нижніх оцінок оптимального значення критерію множин допустимих розв'язків використовуються два варіанти процедур. Процедура Поттса та Ван Васенгова будується на локально оптимальній послідовності відносно попарних перестановок робіт. Для процедури Гелдерса та Клайндорфера визначений найефективніший алгоритм обчислення (результати обчислювальних експериментів до статті не увійшли). Нарешті, наведено опис ходу розв'язання екземпляру задачі потужністю n = 15 робіт за допомогою сформульованого алгоритму.

Формальна постановка задачі

Тут і далі будуть використовуватись наступні позначення (велика літера в позначенні параметру робіт означає, що значення цього параметру залежить від позиції роботи в розкладі, інакше параметр роботи не залежить від конкретного розкладу; без обмеження загальності розглядаємо лише цілі часові параметри).

J – множина усіх робіт, які призначаються на прилад.

σ – деякий допустимий розклад, тобто послідовність робіт із J.

n – кількість робіт у розкладі, нумерація не залежить від їх розміщення у послідовності, $|J| = n$.

[j] – номер роботи, що стоїть в допустимому розкладі на позиції j.

p_j – тривалість роботи із номером j, усі роботи складаються із однієї операції.

d_j – директивний строк виконання роботи із номером j.

w_j – вага роботи із номером j, $\forall j \in J\ w_j > 0$; величина w_i / p_i називається пріоритетом роботи.

C_j – момент завершення обробки роботи j.

$S_j = d_j - C_j$ – резерв часу роботи j; залежність резерву часу від моменту початку обробки роботи j записується як $S_j(t) = d_j - p_j - t$.

$T_j = \max\left(0, C_j - d_j\right)$ – запізнення роботи j.

$P(A)$, де $A \subseteq J$ – сумарна тривалість обробки робіт з множини A, тобто $P(A) = \sum_{i \in A} p_i$.

$\overline{A} = J$, A, $A \subseteq J$.

В стандартній нотації теорії розкладів досліджувана задача позначається як $1 \| \sum w_i T_i$. Роботи поступають на прилад одночасно в момент $t = 0$, переривання робіт не дозволяються, час на переналадку приладу не залежить від послідовності обробки робіт, є постійним та вважається, що він включений в тривалості робіт. Усі роботи складаються з однієї операції. Якщо в тексті зустрічається посилання «робота j», мається на увазі робота із номером j.

При складанні розкладів на одному приладі широко застосовують відношення передування однієї роботи іншою. Якщо робота j передує роботі k, це записується як $j \to k$ (при цьому роботи не обов'язково суміжні). Множина робіт, щодо яких відомо, що вони передують роботі $j \in J$ буде позначатись $B_j \subseteq J$, а множина робіт, для яких з'ясовано, що вони слідують за роботою j, позначатиметься $A_j \subseteq J$. Слід зазначити, що $j \notin A_j,\ j \notin B_j$. A_j, B_j називаються *множинами домінування*.

Досліджується задача мінімізації сумарного зваженого запізнення, тобто задача пошуку такого розкладу n робіт, для якого

$$\sum_{i=1}^{n} w_i T_i = \sum_{i=1}^{n} w_i \max\left(0, C_j - d_j\right) \to \min. \quad (1)$$

Цей критерій є регулярним (це неспадна функція від моментів завершення обробки робіт $C_1, \ldots, C_n$), а отже, як відомо з елементарної теорії розкладів, оптимальний розклад відносно регулярного критерію не містить інтервалів очікування, і всі роботи виконуються одна за одною. З цього випливає, що областю допустимих розв'язків задачі $1 \| \sum w_i T_i$ є усі можливі перестановки потужності n.

Характеристики складності екземплярів задач

Розглядаючи випадкові величини ваги роботи w, тривалості p та директивного строку d для певного класу екземплярів задач мінімізації сумарного зваженого запізнення для багатьох застосувань можна вважати, що w та p незалежні так само, як незалежні w та d. Натомість p та d можуть бути залежними, розподіленими за де-

яким сумісним розподілом із щільністю $f_{pd}(x,y) = f_p(x) f_{d/p}(y/x)$ [4].

Позначимо через $\lambda_d(p)$ довжину інтервалу, на якому $f_{d/p}(y/p) > 0$. Введемо наступні величини. Відносним діапазоном директивних строків класу екземплярів задач називається випадкова величина

$$R(p) = \frac{\lambda_d(p)}{np}. \qquad (2)$$

Далі, фактором запізнення називається випадкова величина

$$t(p) = 1 - \frac{M(d/p)}{np}. \qquad (3)$$

Якщо $f_{d/p}(y/p)$ – щільність рівномірного розподілу, то R та t стають детермінованими. Як показали численні обчислювальні експерименти [13, 18, 5, 4] в цьому випадку середні характеристики ефективності алгоритмів гілок та меж залежать від значень величин R та t. Особливо складними виявляються екземпляри задач із значеннями цих величин, близькими до $R = t = 0{,}6$ або $R = 0{,}4$ та $t = 0{,}8$. Через це R та t називаються характеристиками складності екземплярів задач.

Одним із способів отримання оцінок характеристик складності для конкретного екземпляру задачі (за гіпотези, що умовний розподіл d для усіх значень p рівномірний) може бути використання наступних співвідношень:

$$\hat{R} = \frac{d_{\max} - d_{\min}}{P(J)}, \qquad (4)$$

$$\hat{t} = 1 - \frac{\sum_{i=1}^{n} d_j}{nP(J)}, \qquad (5)$$

де $d_{\min} = \min_{i\in\{1,\ldots,n\}}(d_i)$, $d_{\max} = \max_{i\in\{1,\ldots,n\}}(d_i)$.

Перестановки суміжних робіт

Розрізняють декілька типів відношень передування. Введемо означення.

Означення 1. Відношенням глобального передування роботою i роботи j $i \to j$ називається транзитивне відношення на множині J, в якому для кожної пари робіт існує оптимальний розклад призначення робіт в заданому порядку.

Означення 2. Відношенням безумовного передування роботою i роботи j $i \prec\prec j$ називається відношення на множині J, що задає порядок слідування суміжних робіт, при перестановці місцями яких значення критерію завжди збільшується. Для цього відношення властивість транзитивності не виконується.

Означення 3. Відношенням передування роботою i роботи j, залежним від часу (відношенням умовного передування) $i \prec j$, називається відношення на множині J, яке задається так само, як і відношення безумовного передування, але залежить від моментів початку обробки робіт.

Актурк та Їлдірім [5, 18] запропонували алгоритм, який для будь-якої початкової послідовності робіт знаходить локально оптимальну послідовність відносно перестановок суміжних робіт. В такій послідовності перестановка місцями суміжних робіт може призводити лише до збільшення значення критерію оптимізації.

Означення 4. Критичною точкою для пари суміжних робіт i та j називається момент часу t_{ij} такий, що якщо $t = \min\{C_i - p_i, C_j - p_j\} \ge t_{ij}$, то $i \prec j$ $(j \prec i)$, інакше $j \prec i$ $(i \prec j)$.

Виберемо пару робіт $i, j \in J$ таку, що виконується

$$\begin{aligned} &(d_i < d_j) \vee \\ &(d_i = d_j \wedge p_i < p_j) \vee \\ &(d_i = d_j \wedge p_i = p_j \wedge w_i \ge w_j). \end{aligned} \qquad (6)$$

За таких позначень Актурк та Їлдірім показали [18], що критичні точки для робіт i та j обмежуються наступними:

$$t_{ij}^1 = \frac{w_i d_i - w_j d_j}{w_i - w_j} - (p_i + p_j) \qquad (7)$$

при $d_j - (p_i + p_j) \le t_{ij}^1 \le d_i - p_i$,

$$t_{ij}^2 = d_j - p_i - p_j\left(1 - \frac{w_i}{w_j}\right) \qquad (8)$$

при $\max\{d_i - p_i, d_j - (p_i + p_j)\} \le t_{ij}^2 < d_j - p_j$,

$$t_{ij}^3 = d_j - p_j - p_i\left(1 - \frac{w_j}{w_i}\right) \qquad (9)$$

при $d_j - p_j \le t_{ij}^3 < d_i - p_i$.

Якщо відповідна критична точка лежить в означених границях, то кажуть, що ця критична точка справджується (інакше – не справджується). Справедлива наступна теорема, яка дозволяє визначати критичні точки, які справджуються для пари суміжних робіт замість того, щоб перевіряти це безпосередньо.

Теорема 1 [5]**.** Для пари суміжних робіт $i, j \in J$ такої, що виконується (6), позначивши $t = \min\{C_i - p_i, C_j - p_j\}$, справедливі наступі твердження:

1) якщо $d_i < d_j$, $p_i w_j < p_j w_i$ та $p_i(w_j - w_i) > (d_j - d_i)w_i$, тоді справджуються критичні точки t_{ij}^1, t_{ij}^3, та $j \prec i$ при $t_{ij}^1 \le t \le t_{ij}^3$, та $i \prec j$ інакше;

2) якщо $d_i = d_j$, $w_i < w_j$, $p_i w_j < p_j w_i$, то справджується лише критична точка t_{ij}^3, та $j \prec i$ при $t \le t_{ij}^3$, та $i \prec j$ інакше (замість умови $w_i < w_j$ можна використовувати $p_i \le p_j$);

3) якщо $d_i < d_j$, $p_i w_j \ge p_j w_i$ та $p_i(w_j - w_i) > (d_j - d_i)w_i$, то справджується лише критична точка t_{ij}^1, причому $i \prec j$ при $t \le t_{ij}^1$, та $j \prec i$ інакше;

4) якщо $d_i \le d_j$, $p_i w_j \le p_j w_i$ та $p_i(w_j - w_i) \le (d_j - d_i)w_i$, тоді $i \prec\prec j$;

5) якщо $d_i = d_j$ та $p_i w_j \ge p_j w_i$, тоді робота $j \prec\prec i$;

6) якщо $d_i < d_j$, $p_i w_j > p_j w_i$ та $p_j(w_j - w_i) \le (d_j - d_i)w_j$, тоді справджується лише критична точка t_{ij}^2 та при $t \le t_{ij}^2$ $j \prec i$, і $i \prec j$ після критичної точки.

Означимо матрицю T критичних точок, елементами (i, j) якої можуть бути наступні символи:

- «⟶» якщо $i \to j$;
- «$\prec\prec$» якщо $i \prec\prec j$;
- «індекс критичної точки», якщо для пари робіт *i,j* виконується умова (6) та справджується єдина критична точка із цим індексом;
- «13», якщо для пари робіт *i,j* реалізується випадок 1 теореми 1;
- «-» у всіх інших випадках.

З теореми 1 можна отримати універсальну процедуру обчислення елементу (i, j) матриці критичних точок (при цьому обчислюється також симетричний відносно головної діагоналі елемент (j, i)). Вона є наступною:

1. Якщо $d_i = d_j$, то:
   1.1) якщо $p_i w_j \ge p_j w_i$, то $T_{ji} :=$ «$\prec\prec$», $T_{ij} :=$ «–»;
   1.2) інакше якщо $w_i \ge w_j$, то $T_{ij} :=$ «$\prec\prec$», $T_{ji} :=$ «–»;
   1.3) в противному випадку $T_{ij} :=$ «3», $T_{ji} :=$ «–».
2. Інакше якщо $p_i w_j \le p_j w_i$, то:
   2.1) якщо $p_i(w_j - w_i) \le (d_j - d_i)w_i$, то $T_{ij} :=$ «$\prec\prec$», $T_{ij} :=$ «–»;
   2.2) інакше $T_{ij} :=$ «13», $T_{ij} :=$ «–».
3. Інакше:
   3.1) якщо $p_j(w_j - w_i) \le (d_j - d_i)w_j$, тоді $T_{ij} :=$ «3», $T_{ji} :=$ «–»;
   3.2) інакше $T_{ij} :=$ «1», $T_{ji} :=$ «–».

Елементи «⟶» заповнюються окремо та дана процедура для них не виконується, на діагоналі ставляться символи «-».

Алгоритм знаходження локально оптимальної послідовності відносно перестановок суміжних робіт може бути сформульованим наступним чином (задано початкову послідовність σ):

1. Для $i = \overline{1, n-1}$, $j = \overline{i+1, n}$ якщо $[i] \to [j]$ та $i > j$ або $[j] \to [i]$, та $i < j$, тоді переставити місцями роботи $[i]$ та $[j]$.
2. $k := 1, t := 0$. Поки $k \le n-1$ виконувати:
   2.1) $i := [k]$, $j = [k+1]$, *swap* := false;
   2.2) якщо $T_{ji} =$ «$\prec\prec$», поміняти місцями роботи *i* та *j, swap* := true;
   2.3) інакше якщо умова (6) виконується, $T_{ij} =$ «13», $d_j - p_i - p_j < t$ і $t_{ij}^1 < t \le t_{ij}^3$, переставити місцями роботи *i* та *j*, *swap* := true;
   2.4) інакше якщо умова (6) виконується, $T_{ij} =$ «1» («2») та $t > t_{ij}^1 (t > t_{ij}^2)$, переставити місцями роботи *i* та *j*, *swap* := true;
   2.5) інакше якщо умова (6) не виконується, $T_{ji} =$ «13», $d_j - p_i - p_j < t$ і $t < t_{ij}^1 \vee t_{ij}^3 < t$, переставити місцями роботи *i* та *j*, *swap* := true;
   2.6) інакше якщо умова (6) не виконується, $T_{ji} =$ «1» («2») та $t \le t_{ij}^1 (t \le t_{ij}^2)$, переставити місцями роботи *i* та *j*, *swap* := true;
   2.7) якщо *swap* = true та $k > 1$, $t := t - p_{k-1}$, $k := k - 1$;
   2.8) інакше $t := t + p_k$, $k := k + 1$.

3. Кінець, отримана локально оптимальна послідовність.

Цей алгоритм має часову складність $O(n^3)$ подібно до алгоритму із [5] і відрізняється тим, що може знаходити локальний оптимум на частковій послідовності робіт з *J*.

Правила домінування

Часто кажуть, що важкорозв'язні задачі бракують структури, тобто при проведені конструктивного пошуку в області допустимих рішень відсутні співвідношення між параметрами задачі, які б дозволяли достатньо скорочувати цю область по мірі конструювання часткової послідовності. Правила домінування якраз і містять такі співвідношення, і в окремих випадках їх застосування дозволяє відносно легко отримати оптимальну послідовність. Твердження про те, що при виконанні певного співвідношення між параметрами задачі одна робота обов'язково передує іншій хоча б в одному оптимальному розкладі, називається правилом домінування.

Окрім правил домінування відомі елементарні факти, що стосуються часткових випадків індивідуальних задач. Далі наведемо їх у вигляді переліку лем.

Лема 1 [5]. Якщо в задачі $1||\sum w_iT_i$ роботи упорядкувати за неспаданням директивних строків (побудувати EDD-розклад), і щонайбільше одна робота при цьому буде запізнюватись, то така послідовність оптимальна.

Лема 2 [19]. Якщо в задачі $1||\sum w_iT_i$ при сортуванні робіт за незростанням пріоритетів (WSPT-розклад) всі роботи запізнюються, або жодна робота не запізнюється, то отримано оптимальний розклад.

З лем 1 та 2 випливає наступний наслідок: якщо для задачі $1||\sum w_iT_i$ EDD-розклад збігається із WSPT-розкладом, то ці розклади оптимальні.

Лема 3 (Ельмаграбі [3]). Якщо для задачі $1||\sum w_iT_i$ $\exists k: d_k \ge P(\sigma)$, то робота із номером *k* в оптимальній послідовності призначається останньою.

Лема 4 [2, 4]. Якщо в задачі $1||\sum w_iT_i$ знайдуться дві роботи $j,k \in J$ із $d_j \le d_k$, $p_j \le p_k$ та $w_j \ge w_k$, то існує оптимальна послідовність, в якій $j \to k$.

Правила домінування додатково враховують множини домінування. Виберемо дві роботи $j,k \in J, j \ne k$: відомі A_j, A_k, B_j, B_k, причому $j \notin A_k, B_k$, $k \notin A_j, B_j$. Відомі правила домінування сформулюємо у вигляді теорем.

Теорема 2 [4] Для задачі $1||\sum w_iT_i$ та пари робіт $j,k \in J, j \ne k$, для яких відомі A_j, A_k, B_j, B_k, $j \notin A_k, B_k$, $k \notin A_j, B_j$ якщо $d_k \ge P\left(\overline{A_j}\right)$, то $j \to k$.

Можна бачити, що з цієї теореми випливає лема Ельмаграбі.

Теорема 3 [6]. Для задачі $1||\sum w_iT_i$ та пари робіт $j,k \in J, j \ne k$, для яких відомі A_j, A_k, B_j, B_k, $j \notin A_k, B_k$, $k \notin A_j, B_j$ виконуються наступні твердження:

1) якщо $p_j \le p_k, w_j \ge w_k$ та

$$d_j \le \max\left\{d_k, \frac{w_j - w_k}{w_j}\left(P(B_j \cup B_k) + p_j + p_k\right) + \frac{w_k}{w_j}d_k\right\},$$

або

$$d_j \le \frac{w_j - w_k}{w_j}\left(P(B_j \cup B_k) + p_j + p_k\right) + \frac{w_k}{w_j}\left(P(B_k) + p_k\right),$$

тоді $j \to k$;

2) якщо $p_j \le p_k$, $w_j < w_k$, та

$$d_k \ge \frac{w_k - w_j}{w_k}P(\overline{A_j}) + \frac{w_j}{w_k}d_j$$

і одночасно

$$d_k \ge \frac{w_k - w_j}{w_k}P(\overline{A_j}) + \frac{w_j}{w_k}\left(P(\overline{A_j \cup A_k}) - p_k\right),$$

тоді $j \to k$;

3) якщо $p_j \le p_k$, $w_j < w_k$, та

$$d_j \le \frac{w_j - w_k}{w_j}P(\overline{A_j}) + \frac{w_k}{w_j}\left(P(B_k) + p_k\right)$$

і одночасно

$$p_j \le \frac{w_j - w_k}{w_j}\left(P(\overline{A_j}) - P(B_k)\right) + \frac{w_k}{w_j}p_k,$$

тоді $j \to k$;

4) якщо $w_j \ge w_k$, та

$$d_k \le \min\Big\{d_j, \frac{w_k - w_j}{w_k}\big(P(B_j \cup B_k) + p_j + p_k\big) + \frac{w_j}{w_k} d_j\Big\},$$

і одночасно

$$d_k \ge P(\overline{A_j}) - \frac{w_j}{w_k} p_k,$$

тоді $j \to k$;

5) якщо $w_i \ge w_k$, та якщо

$$d_j \le \frac{w_j - w_k}{w_j}\big(P(B_j \cup B_k) + p_j + p_k\big) + \frac{w_k}{w_j}\big(P(B_k) + p_k\big)$$

і одночасно

$$p_k \ge \frac{w_k}{w_j}\big(P(\overline{A_j}) - P(B_k) - p_k\big),$$

тоді $j \to k$;

6) якщо $w_i < w_k$, та якщо

$$d_k \ge \frac{w_k - w_j}{w_k} P(\overline{A_j}) + \frac{w_j}{w_k} d_j$$

і одночасно

$$d_k \ge P(\overline{A_j}) - \frac{w_j}{w_k} p_k,$$

тоді $j \to k$.

Можна бачити, що лема 4 є наслідком першого випадку теореми 3.

Теорема 4 [5]. Для задачі $1||\sum w_i T_i$ якщо для пари робіт $i, j \in J$ справедлива умова (6), $d_i \le d_j$, $p_i > p_j$, $w_i < w_j$, то при

$$t = \min\big\{C_i - p_i, C_j - p_j\big\} > t_{ij},$$

де t_{ij} – одна з критичних точок, що можуть справджуватись в даному випадку (t_{ij}^1 або t_{ij}^2), $j \to i$.

Теореми 2-4 стверджують відомий на даний час набір глобальних правил домінування (передування) для задачі $1||\sum w_i T_i$.

Процедура галуження

В методі гілок та меж здійснюється пошук в глибину. Кожний вузол дерева пошуку відповідає частковому розкладу σ_η, який складається із робіт, призначених на останні $\eta \le n$ позицій. Позначимо множину робіт, що належать σ_η через $S(\eta)$, та $U(\eta) = J$, $S(\eta)$ (конкретні часткові розклади однієї потужності для описання процедури галуження розрізняти не має необхідності). Галуження відбувається для $\eta < n$ шляхом призначення деякої роботи $k \in U(\eta)$ безпосередньо перед усіма роботами з $S(\eta)$, таким чином отримуючи частковий розклад $\sigma_{\eta+1}$. Щодо роботи *k* приймаються наступні рішення:

1) $\exists j \in U(\eta) : k \in B_j \Rightarrow$ галуження не виконується;

2) для пари робіт *k*, *j*, $j \in S(\eta)$ справджуються умови теореми 4 $\Rightarrow$ галуження не виконується;

3) $k \in A_j \ \forall j \in U(\eta) \Rightarrow$ робота k безумовно фіксується на позиції $n - \eta$;

4) після призначення $\forall j \in U(\eta)$ $A_j := A_j \cup \{k\}$, та множини домінування оновлюються для усіх робіт з $U(\eta)$ – будується транзитивне замикання відношення передування наступним чином:

$$A_i := A_i \cup \{k\} \cup A_k \quad \forall i \in B_j \cup \{j\},$$
$$B_i := B_i \cup \{j\} \cup B_j \quad \forall i \in A_k \cup \{k\}.$$

Таким чином уникаються циклічні перевірки, і будується антирефлексивне відношення.

Для того, щоб враховувати усі зміни множин домінування, після додавання дуги в граф глобального передування, заданий матрицею суміжності *M,* виконуються наступні дії.

1. $i := 1$, $j := 1$.
2. Якщо $i \ne j$ та $M_{ij} = 0 \vee M_{ji} = 0$, перевірити послідовно усі правила домінування для пари робіт *i* та *j*.
3. Якщо виконується хоча б одне з них, задати $M_{ij} := 1$, знайти транзитивне замикання *M*, перейти на крок 1.
4. Інакше якщо $j = n, i := i + 1, j := 1$, інакше $j := j + 1$.
5. Якщо $i > n$, кінець, інакше перейти на крок 2.

Аналогічна логіка будується для будь-якого іншого представлення графу відношення глобального передування.

Нижні оцінки мінімуму критерію для множини робіт

В загальному випадку задача оцінювання має наступний вигляд. У деякому вузлі дерева конструктивного пошуку отримано частковий розклад σ_η, який складається із η призначених робіт. Нижня оцінка цього вузла *LB* буде складатись з двох компонент, а саме

$$LB = \sum_{i\in\sigma_\eta} w_i T_i + LB^*, \qquad (10)$$

де $LB^* \le \sum_{\{i:i\notin\sigma_\eta\}} w_i T_i$. В цьому розділі розглянемо дві процедури оцінювання: лагранжевої релаксації Поттса та Ван Вассенгова, а також транспортну оцінку Гелдерса і Клайндорфера.

Введемо оцінку на основі підходу лагранжевої релаксації. Задача $1||\sum w_i T_i$ може бути записана наступним чином:

$$\sum_{i=1}^{n} w_i T_i \to \min$$

при обмеженнях

$$T_i \ge 0, \quad i = \overline{1,n}, \qquad (11)$$

$$T_i \ge C_i - d_i, \quad i = \overline{1,n}, \qquad (12)$$

де мінімізація проводиться по всім наборам моментів закінчення обробки робіт. Відкинемо обмеження (12). Перетворена відповідно до принципу лагранжевої релаксації задача тоді буде мати наступний вигляд:

$$\min_{(T_i,C_i)} \sum_{i=1}^{n} \left((w_i-\mu_i)T_i + \mu_i(C_i-d_i)\right) \to \max \qquad (13)$$

при обмеженнях

$$T_i \ge 0, \quad i = \overline{1,n},$$

де $\mu \ge 0$ – вектор множників Лагранжа. Позначимо функцію Лагранжа як $L(\mu)$, виключення з області допустимих розв'язків точок, в яких $L(\mu) = -\infty$, еквівалентно накладанню додаткових обмежень

$$\mu_i \le w_i, i = \overline{1,n}. \qquad (14)$$

Нехай задана деяка послідовність робіт σ^*, причому робота *i* в цій послідовності завершується в момент C_i^*. Після спрощення перетворення задача отримує наступний вигляд [11]:

$$L(\mu) = \sum_{i=1}^{n} \mu_i \left(C_i^* - d_i\right) \to \max \qquad (15)$$

при обмеженнях

$$\frac{\mu_i}{p_i} \ge \frac{\mu_{i+1}}{p_{i+1}}, i = \overline{1,n-1} \qquad (16)$$

$$0 \le \mu_i \le \hat{w}_i, \ , i = \overline{1,n}, \qquad (17)$$

де величини

$$\hat{w}_i = p_i \min_{h\in\{1,\ldots,i\}} \left\{\frac{w_h}{p_h}\right\} \qquad (18)$$

називаються *зведеними вагами робіт*. Вибір послідовності σ^* є довільним, обчислювальні експерименти показали, що дана оцінка буде тим більш строгою, чим ближче до оптимальної є σ^*.

Розклад σ^* пропонується будувати наступним чином: спочатку знаходиться послідовність із застосуванням евристики очевидного штрафу за запізнення (АТС), в якій роботи послідовно призначаються на прилад у порядку спадання залежних від часу пріоритетів, що обчислюються для $\forall i \in J$ за формулою:

$$I_j(t) = \frac{w_j}{p_j} \exp\left(-\frac{|U| \cdot \max(0, d_j - p_j - t)}{2P(U)}\right) \qquad (19)$$

де U – множина ще не призначених робіт. Після цього на знайденій послідовності шукається локальний оптимум відносно перестановок суміжних робіт.

Для розв'язання задачі (15), (17) запропоновано так званий метод підбору множників, який будує оптимальний вектор множників Лагранжа за лінійний час, що обумовлене простою структурою розв'язку. Наведемо остаточну процедуру обчислення, її обґрунтування можна знайти в [11]. Спочатку сформулюємо теорему.

Теорема 5 [11]. Нехай

$$\Lambda_0 = 0, \Lambda_j = \sum_{i=1}^{j} p_i L_i, \ j = \overline{1,n}.$$

Задамо множину цілих чисел $V = \{v_1, \ldots, v_r\}$ таку, що $v_0 = 0$, а v_k – це найменше число таке, що

$$v_k > v_{k-1} \Rightarrow \Lambda_{v_k} > \Lambda_{v_{k-1}}.$$

Тоді оптимальним розв'язком задачі (15), (17) буде вектор множників Лагранжа μ^*, компоненти якого обчислюються наступним чином:

$$\mu_i^* = \mu_{i+1}^* \frac{p_i}{p_{i+1}}, \quad i \notin V, i \ne n, \qquad (20)$$

$$\mu_i^* = \hat{w}_i, \quad i \in V, \qquad (21)$$

$$\mu_n^* = 0, \quad \text{якщо n} \notin V. \qquad (22)$$

Теорема 5 стверджує, що якщо пронумерувати роботи в порядку їх слідування відповідно до використаної евристики σ^*, вклад робіт із номерами $v_{i-1}+1, \ldots, v_i$ до нижньої оцінки буде складати $\hat{w}_{v_i}(\Lambda_{v_i} - \Lambda_{v_{i-1}})/p_{v_i}$. Таким чином для кожного елементу $v_i \in V$ до нижньої оцінки додається означений член, а отже вона обчислюється за лінійний час від кількості робіт в *J*.

Ідея транспортної нижньої оцінки полягає в тому, що відкидається вимога про заборону переривань робіт. Замість простору Π усіх можливих перестановок довжини n розглядається простір S усіх розкладів, в яких роботи можуть перериватись (але заборона інтервалів очікування все ще має місце). Таким чином маємо

$$\min_{\Pi} \sum_{j\in J} w_j \max\left(0, C_j - d_j\right) \geq \geq \min_{\mathsf{S}} \sum_{j\in J} w_j \max\left(0, C_j - d_j\right). \quad (23)$$

Розіб'ємо часовий інтервал $\left[0; P(J)\right]$ на v відрізків, вибравши з цією метою v моментів часу

$$\mathrm{T} = \left(\tau_1, \tau_2, \ldots, \tau_v \mid 0 = \tau_1 \leq \tau_2 \leq \ldots \leq \tau_v \leq P(J)\right).$$

Довжину відрізка i позначимо через s_i. Введемо невідомі x_{ij} – це обсяг часу, який робота j обробляється в часовому інтервалі i. Розглянемо наступну транспортну задачу лінійного програмування (ТЗЛП):

$$L(x, \mathrm{T}) = \sum_{i=1}^{v}\sum_{j=1}^{n} c_{ij} x_{ij} \to \min \quad (24)$$

при обмеженнях

$$\sum_{j=1}^{n} x_{ij} = s_i,\ i = \overline{1, v}, \quad (25)$$

$$\sum_{i=1}^{v} x_{ij} = p_j,\ j = \overline{1, n}, \quad (26)$$

$$x_{ij} \geq 0 \quad i = \overline{1, v},\ j = \overline{1, n}. \quad (27)$$

Задача (24) – (27) моделює задачу $1 \mid pmpt \mid \sum w_i T_i$, де поле *pmpt* вказує про можливі переривання робіт, що надалі буде називатись *послабленою задачею*. Обмеження (25) відповідають умові, що загальний обсяг часового інтервалу, що розподіляється по роботах, точно рівний його довжині. Подібним чином обмеження (26) стверджують, що загальний обсяг часу обробки, який виділяється кожній роботі, повинен бути рівним її тривалості. Обмеження (27) також обов'язкові, оскільки компоненти невідомого вектору x позначають час, і тому цей вектор має бути напівдодатним.

Єдиним невизначеним питанням залишається вибір ваг c_{ij} цільової функції. Обґрунтуємо такий вибір у вигляді теореми.

Теорема 6 [8]. Якщо в задачі (24) – (27) компоненти вектора ваг c обрати наступним чином:

$$c_{ij} = \begin{cases} \left(1 + \left\lfloor \dfrac{\tau_i - d_j}{p_j} \right\rfloor w_j\right) \text{при } \tau_i \geq d_j, \\ 0, \text{ інакше,} \end{cases} \quad (28)$$

то оптимальний розв'язок послабленої задачі з такими вагами буде нижньою оцінкою оптимального розв'язку відповідної задачі $1 \| \sum w_i T_i$ для будь-якого розбиття T інтервалу $\left[0; P(J)\right]$. Тут $\lfloor a \rfloor$ позначає найбільше додатне ціле число, що $\leq a$.

Наведемо ще один результат, який вказує на те, що вибір розбиття T напряму впливає на ефективність обговорюваної оцінки.

Теорема 7 [8]. Для задачі (24) – (27), в якій ваги обираються відповідно до виразу (28), розглянемо два розбиття часового інтервалу $\left[0; P(J)\right]$

$$\mathrm{T} = \left(\tau_1, \tau_2, \ldots, \tau_n, \tau_{n+1}, \ldots, \tau_v\right)$$

та

$$\mathrm{T}' = \left(\tau'_1, \tau'_2, \ldots, \tau'_n, \ldots, \tau'_{v+p}\right)$$

такі, що

$$\tau_i = \tau'_i,\ i = \overline{1, n},$$
$$\tau_n = \tau'_n \leq \tau'_{n+1} \leq \tau'_{n+2} \ldots \leq \tau'_{n+p+1} = \tau_{n+1},$$
$$\tau_i = \tau'_{i+p},\ i = \overline{n+1, v},$$

тобто в розбитті T′ між точками τ_n та τ_{n+1} знаходяться додаткові проміжні точки, і тому T′ продукує більше інтервалів, ніж T. Тоді $L(x, \mathrm{T}) \leq L(x', T')$.

Чим більше інтервалів утворює розбиття, тим більш строгою є дана нижня оцінка. В той же час із збільшенням кількості інтервалів, які утворюються в результаті розбиття, збільшується складність розв'язання послабленої задачі. Саме у виборі конкретного розбиття відображається компроміс між строгістю оцінки та складністю її обчислення.

Для того, щоб алгоритм гілок та меж, що використовує транспортну оцінку Гелдерса та Клайндорфера, давав найкращі результати, послаблена задача у кожному вузлі дерева пошуку повинна розв'язуватись якомога більш ефективним з точки зору кількості обчислень методом. До того ж бажано, щоб складність алгоритму, що розв'язує послаблену задачу, не залежала від характеру вхідних даних, а лише від розмірності задачі, тобто щоб цей алгоритм був строго поліноміальним. Більше того, в послабленій задачі x_{ij} – не обов'язково цілі числа, тому і з

цього положення вибір строго поліноміального алгоритму є більш привабливим (інакше для обчислень слід використовувати представлення чисел з фіксованою точністю).

Послаблена задача є ТЗЛП (24) – (27), і в сучасній практиці математичного програмування її розв'язують, зводячи до задачі знаходження потоку мінімальної вартості [20]. Слід відмітити, що стандартні методи розв'язання ТЗЛП як задачі лінійного програмування загального вигляду є набагато менш ефективними, ніж спеціалізовані.

Згідно із [20], строго поліноміальні алгоритми розв'язання ТЗЛП, найкращі на сьогодні за часовою складністю – це запропоновані в [21], [22], [23]. Але обчислювальна практика показала, що великі приховані константи виконання для алгоритмів [21] та [23] обмежують вибір алгоритмом Орліна масштабування пропускних здатностей без стягування вершин [22].

Відповідно до (10) під час конструктивного пошуку задачу оцінювання потрібно розв'язувати кожного разу як частковий розклад змінюється. Хоча при русі до листових вузлів дерева пошуку розмірність транспортної задачі зменшується, в деяких випадках раціонально знаходити розв'язок не у всіх, а лише в певних ключових вузлах дерева пошуку. Для інших часткових розкладів можна отримувати дещо слабшу оцінку простішим методом, опишемо його.

При розв'язанні задачі (24) – (27) можна отримати набори двоїстих змінних u_i^* та v_j^*, $i=\overline{1,v}$, $j=\overline{1,n}$, які являють собою потенціали вершин транспортної мережі, та відповідають оптимальному потоку в ній (див напр. [23]). При цьому виконується

$$\sum_{i=1}^{v}\sum_{j=1}^{n} c_{ij}x_{ij}^* = \sum_{i=1}^{v} s_i u_i^* + \sum_{j=1}^{n} p_j v_j^*, \qquad (29)$$

де x_{ij}^* позначає оптимальний розв'язок транспортної задачі. Можна легко показати, використовуючи теорему 7, що

$$L(x,T,\eta) = \sum_{i=1}^{I-1} s_i' u_i^* + \sum_{j=1}^{\eta-1} p_j v_j^* \qquad (30)$$

– це нижня оцінка оптимального значення критерію на підмножині розкладів, які починаються із робіт $J\setminus\{\sigma_\eta\}$. Якщо нижня оцінка (30) не призводить до виключення із розгляду вузла дерева пошуку, для цього вузла все одно слід розв'язати задачу (24) – (27).

Приклад розв'язання екземпляру задачі

Згенеруємо екземпляр задачі із характеристиками складності $R=t=0,6$ потужністю $n=15$. Для цього візьмемо n реалізацій $w_i \in_U [1;10]$, $p_i \in_U [1;100]$,

$$d_i \in_U \left[P(J)\cdot(1-t-R/2);\, P(J)\cdot(1-t+R/2)\right].$$

Тоді $Md = P(J)\cdot(1-t)$, $\lambda_d = R\cdot P(J)$. Беремо цілу частину від отриманих значень. Дані екземпляру наведені в табл. 1.

Табл. 1. Згенерований екземпляр задачі

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
w_i	1	2	6	4	9	2	8	4	1	2	10	2	1	6	7
p_i	17	46	6	79	52	88	96	54	47	87	78	100	62	27	85
d_i	301	468	98	246	419	557	362	505	347	506	425	500	410	177	329

Значення критерію для поданого порядку виконання робіт $f=11735$, горизонт планування $C_{\max}=924$. Складемо деякі базові розклади: розклад EDD (сортування за неспаданням директивних строків), розклад WSPT (сортування за незростанням пріоритетів), розклад ATC та на основі нього локально оптимальний відносно перестановок суміжних робіт (LM-розклад). Матриця критичних точок T наведена в табл. 2. Отримані наступні послідовності:

$$\sigma_{EDD} = (3,14,4,1,15,9,7,13,5,11,2,12,8,10),$$
$$f_{EDD} = 4731;$$
$$\sigma_{WSPT} = (3,14,5,11,7,15,8,1,4,2,10,6,9,12,13),$$
$$f_{WSPT} = 3566;$$
$$\sigma_{ATC} = (3,14,4,15,7,5,11,8,1,2,10,6,9,12,13),$$
$$f_{ATC} = 2548;$$
$$\sigma_{LM} = (3,14,4,15,7,5,11,1,8,2,10,6,9,12,13),$$
$$f_{LM} = 2494.$$

Табл. 2. Матриця критичних точок

Роботи	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	-	≺≺	-	-	2	≺≺	1	2	≺≺	≺≺	2	≺≺	≺≺	-	1
2	-	-	-	-	-	≺≺	-	2	-	≺≺	-	≺≺	-	-	-
3	≺≺	≺≺	-	≺≺	≺≺	≺≺	≺≺	≺≺	≺≺	≺≺	≺≺	≺≺	≺≺	≺≺	≺≺
4	2	≺≺	-	-	2	≺≺	2	2	≺≺	≺≺	2	≺≺	≺≺	-	2
5	-	≺≺	-	-	-	≺≺	-	≺≺	-	≺≺	≺≺	≺≺	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	≺≺	-	-	2	≺≺	-	≺≺	-	≺≺	2	≺≺	≺≺	-	-
8	-	-	-	-	-	≺≺	-	-	-	≺≺	-	-	-	-	-
9	-	2	-	-	2	2	1	2	-	2	2	≺≺	≺≺	-	-
10	-	-	-	-	-	≺≺	-	-	-	-	-	-	-	-	-
11	-	≺≺	-	-	-	≺≺	-	≺≺	-	≺≺	-	≺≺	-	-	-
12	-	-	-	-	-	2	-	1	-	2	-	-	-	-	-
13	-	2	-	-	1	2	-	2	-	2	1	2	-	-	-
14	≺≺	≺≺	-	≺≺	≺≺	≺≺	≺≺	≺≺	≺≺	≺≺	≺≺	≺≺	≺≺	-	≺≺
15	-	≺≺	-	-	2	≺≺	2	≺≺	≺≺	≺≺	2	≺≺	≺≺	-	-

Локально оптимальний розклад був отриманий перестановкою робіт 8 та 1, що починаються в розкладі АТС в момент часу $t = 423$. Як видно з таблиці, для робіт справджується критична точка t_{18}^2, значення якої $t_{18}^2 = 447,5$, що і обумовлює перестановку. Сумарне зважене запізнення робіт в LM-розкладі є попередньою верхньою оцінкою оптимального значення критерію.

Далі, побудуємо відношення глобального передування для робіт, починаючи із порожніх множин домінування. Результат обчислення наведений в табл. 3. Для кожної пари робіт, що знаходяться у відношенні, вказана теорема та номер твердження, з якого це випливає. Пара може знаходитись у відношенні також за транзитивністю, в цьому випадку над стрілкою відношення позначки не ставляться.

Табл. 3. Матриця відношення глобального передування

Роботи	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1						$\rightarrow^{T3.3}$			$\rightarrow^{T3.1}$	$\rightarrow^{T3.3}$		$\rightarrow^{T3.3}$	$\rightarrow^{T3.1}$		
2						$\rightarrow^{T3.1}$				$\rightarrow^{T3.1}$		$\rightarrow^{T3.1}$	$\rightarrow^{T3.1}$		
3	$\rightarrow^{T3.1}$	$\rightarrow^{T3.1}$		$\rightarrow^{T3.1}$	$\rightarrow^{T3.2}$	$\rightarrow$	$\rightarrow^{T3.2}$	$\rightarrow^{T3.1}$	$\rightarrow$	$\rightarrow$	$\rightarrow^{T3.2}$	$\rightarrow$	$\rightarrow$	$\rightarrow^{T3.1}$	$\rightarrow^{T3.2}$
4						$\rightarrow^{T3.1}$				$\rightarrow^{T3.1}$		$\rightarrow^{T3.1}$	$\rightarrow^{T3.4}$		
5		$\rightarrow^{T3.4}$				$\rightarrow^{T3.1}$		$\rightarrow^{T3.1}$		$\rightarrow^{T3.1}$		$\rightarrow^{T3.1}$	$\rightarrow$		
6												$\rightarrow^{T3.1}$	$\rightarrow^{T3.5}$		
7		$\rightarrow^{T3.4}$			$\rightarrow^{T3.6}$	$\rightarrow^{T3.4}$		$\rightarrow^{T3.4}$		$\rightarrow^{T3.4}$	$\rightarrow^{T3.6}$	$\rightarrow^{T3.1}$	$\rightarrow^{T3.4}$		
8						$\rightarrow^{T3.1}$				$\rightarrow^{T3.1}$		$\rightarrow^{T3.1}$	$\rightarrow^{T3.1}$		
9												$\rightarrow^{T3.3}$	$\rightarrow^{T3.1}$		
10						$\rightarrow^{T3.1}$						$\rightarrow^{T3.1}$	$\rightarrow^{T3.5}$		
11		$\rightarrow^{T3.4}$				$\rightarrow^{T3.1}$		$\rightarrow^{T3.4}$		$\rightarrow^{T3.1}$		$\rightarrow^{T3.1}$	$\rightarrow$		
12													$\rightarrow^{T3.5}$		
13															
14	$\rightarrow^{T3.4}$	$\rightarrow^{T3.1}$		$\rightarrow^{T3.1}$	$\rightarrow^{T3.2}$	$\rightarrow$	$\rightarrow^{T3.2}$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$		$\rightarrow^{T3.2}$
15	$\rightarrow^{T2}$	$\rightarrow^{T3.4}$		$\rightarrow^{T2}$	$\rightarrow$	$\rightarrow^{T3.1}$	$\rightarrow^{T3.2}$	$\rightarrow$	$\rightarrow^{T3.4}$	$\rightarrow^{T3.1}$	$\rightarrow$	$\rightarrow$	$\rightarrow$		

Можна легко бачити, що роботи 12 та 13 призначаються в оптимальному розкладі останніми і тому виключаються з розгляду, оскільки їм передують усі інші роботи. Подібним чином підпослідовність робіт (3,14,15) призначається в тому самому оптимальному розкладі на початку.

Таким чином, в корені дерева пошуку $\eta = 2$, $\sigma(2) = (12,13)$. Щоб остаточно визначитись із структурою кореневого вузла, сформуємо множину кандидатів, які можуть призначатись безпосередньо перед $\sigma(2)$.

Роботи 3, 14, 15 вже були виключені. Можна не включати до кандидатів роботи 1, 2, 4, 5, 7, 8, 10, 11, оскільки вони належать B_6. Таким чином кандидатами на галуження з кореня є роботи 6 та 9. Хід подальших обчислень та дерево пошуку зображене на рис. 1.

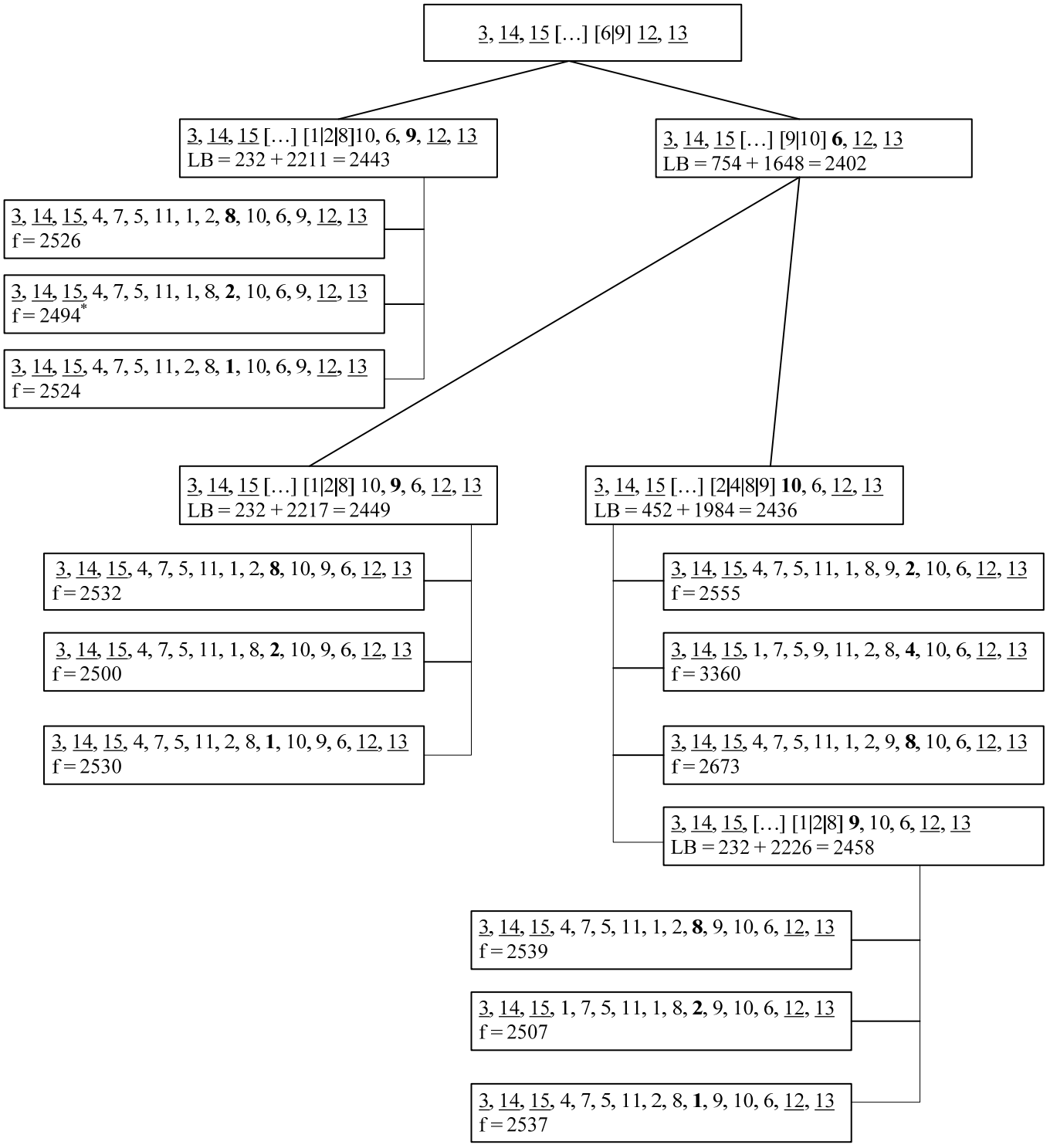

Рис. 1. Дерево пошуку

Підкреслення номерів робіт означає їх безумовну фіксацію на початку або у кінці розкладу. В квадратних дужках зазначені кандидати на галуження в кожному вузлі.

Для оцінки вузлів була використана процедура Гелдерса та Клайндорфера, оскільки на невеликих за потужністю екземплярах оцінка Поттса та Ван Вассенгова часто нульова. Це компенсується швидкодією її обчислення для задач більшої розмірності. Після кожного галуження оновлюється матриця відношення глобального передування, в кожному вузлі зберігаються лише пари робіт, які були додані до відношення, обчисленого на початку. Це дозволило вже на 3-4 рівні заглиблення отримати конкретні розклади у листових вузлах. Таким чином послідовність σ_{LM} є оптимальною.

Висновки

Зроблений огляд останніх досягнень в області дослідження задачі мінімізації сумарного зваженого запізнення виконання робіт одним приладом, та був сформульований на їх основі ефективний алгоритм гілок та меж, який призначений для статистичних досліджень характеристик нових алгоритмів, таких, як запропонований ПДС-алгоритм для цієї задачі.

Даний алгоритм широко використовує 10 відомих глобальних правил домінування, більшість з яких були встановлені нещодавно та не згадувались в емпіричних дослідженнях, дві процедури нижньої оцінки оптимального значення критерію, одна з яких є найбільш строгою серед відомих із регульованою складністю обчислень, а інша – обчислюється за лінійний час та дає приріст ефективності на задачах великої розмірності, процедуру отримання локально оптимальної послідовності з метою одержання строгої верхньої оцінки оптимального значення критерію та удосконалення нижньої оцінки на основі лагранжевої релаксації.

Подальші дослідження включатимуть реалізацію запропонованого алгоритму та проведення обчислювальних експериментів для порівняння продуктивності згаданого ПДС-алгоритму із продуктивністю описаного в цій статті.

Список літератури

1. Jensen J.B., Philipoom P.R., Malhotra M.K. Evaluation of scheduling rules with commensurate customer priorities in job shops // Journal of Operations Management. – 1995, Vol.13. – P.213-228.
2. Lawler E.L. A "Pseudopolynomial" Algorithm for Sequencing Jobs to Minimize Total Tardiness // Annals of Discrete Mathematics. – 1977, Vol. 1. – P.331-342.
3. Elmaghraby S.E., The One Machine Sequencing Problem with Delay Costs // Journal of Industrial Engineering. – 1967, Vol. 19. – P.105-108.
4. Rinnooy Kan, A.H.G., Lageweg B.J., Lenstra J.K. Minimizing Total Costs in One-machine Scheduling // Operations Research. – 1975, Vol. 23. – P.908-927.
5. Akturk M.S., Yildirim M.B. A New Lower Bounding Scheme for the Total Weighted Tardiness Problem // Computers Operations Research. – 1998, Vol. 24 (4). – P.265-278.
6. Kanet, J.J. New precedence theorems for one-machine weighted tardiness // Mathematics of Operations Research. – 2007, Vol. 32. – P.579–588.
7. Shwimer J. On the N-Job, One-Machine, Sequence-Independent Scheduling Problem with Tardiness Penalties: a Branch-and-Bound Solution // Management Science. – 1972, Vol. 18. – P.301-313.
8. Gelders L., Kleindorfer P.R. Coordinating Aggregate and Detailed Scheduling Decisions in the One-Machine Job Shop I – Theory // Operations Research. – 1974, Vol. 22. – P.46-60.
9. Gelders L., Kleindorfer P.R. Coordinating Aggregate and Detailed Scheduling Decisions in the One-Machine Job Shop II – Computation and Structure // Operations Research. – 1975, Vol. 23. – P.312-324.
10. Fisher M.L. A Dual Algorithm for the One-machine Scheduling Problem // Mathematical Programming. – 1976,Vol. 11. – P.229-251.
11. Potts C.N., Van Wassenhove L.N. A Branch and Bound Algorithm for Total Weighted Tardiness Problem // Operations Research. – 1985, Vol. 33. – P.363-377.
12. Hoogeveen J.A., Van de Velde S.L. Stronger Lagrangian Bounds by Use of Slack Variables: Applications to Machine Scheduling Problems // Mathematical Programming. – 1995, Vol. 70. – P.173-190.
13. Abdul-Razaq T. S., Potts C. N., Van Wassenhove L. N. A Survey of Algorithms for Single Machine Total Weighted Tardiness Scheduling Problems // Discrete Applied Mathematics. – 1990, Vol. 26(2–3). – P.235–253.

14. Vepsalainen A.P.J., Morton T.E. Priority Rules for Job Shops with Weighted Tardiness Costs // Management Science. – 1987, Vol. 39. – P.626-632.
15. Згуровский М.З., Павлов А.А. Принятие решений в сетевых системах с ограниченными ресурсами: Монография.– К.: Наукова думка. – 2010. – 573 с.
16. Павлов А.А., Мисюра Е.Б. Новый подход к решению задачи «Минимизация суммарного взвешенного опоздания при выполнении независимых заданий с директивными сроками одним прибором» // Системні дослідження та інформаційні технології. – 2002, № 2. – С.7-32.
17. OR-Library: Weighted Tardiness [Електронний ресурс] // Режим доступу: http:// people.brunel.ac.uk/~mastjjb/jeb/orlib/wtinfo.html.
18. Akturk M.S., Yildirim M.B. A New Dominance Rule for the Total Weighted Tardiness Problem // Production Planning & Control: The Management of Operations – 1999, Vol.10 (2). – P.138-149.
19. Rachamadugu R.M.V. A Note on Weighted Tardiness Problem // Operations Research – 1987, Vol.35. – P.450-452.
20. Schrijver A. Combinatorial Optimization: Polyhedra and Efficiency, vol. A – Springer, Berlin. – 2004. – 647 p.
21. Kleinschmidt P., Schannath H. A strongly polynomial algorithm for the transportation problem, Mathematical Programming. – 1995, Vol. 68. – P.1-13.
22. Orlin J.B. A Faster Strongly Polynomial Minimum Cost Flow Algorithm // Operations Research – 1993. – Vol 41 (2). – P.338-350.
23. Tokuyama T., Nakano J. Efficient algorithms for the Hitchcock transportation problem, SIAM Journal on Computing. – 1995, Vol. 24. – P.563-578.