УДК 519.854.2

ПАВЛОВ О.А.,
МІСЮРА О.Б.,
ШЕВЧЕНКО К.Ю.

ПОБУДОВА ПДС-АЛГОРИТМУ РОЗВ'ЯЗАННЯ ЗАДАЧІ МІНІМІЗАЦІЇ СУМАРНОГО ЗВАЖЕНОГО ЗАПІЗНЕННЯ ВИКОНАННЯ РОБІТ НА ОДНОМУ ПРИЛАДІ

На основі попередніх досліджень властивостей задачі мінімізації сумарного зваженого запізнення виконання робіт на одному приладі побудований ПДС-алгоритм її розв'язання. Для запропонованого алгоритму виділені поліноміальна та декомпозиційна складові, сформульовані умови реалізації цих складових для окремої індивідуальної задачі. Даний алгоритм дозволяє отримувати точні розв'язки для широкого класу індивідуальних задач великої потужності за прийнятний час, а також близькі до оптимальних розв'язки, якщо обчислення перериваються передчасно. Алгоритм пристосований до паралельних розрахунків.

Based on previous studies concerning properties of a problem to minimize total weighted tardiness of processing jobs on a single machine, constructed here is a PDC-algorithm that solves it. We single out polynomial and decompositional components of a proposed algorithm and formulate conditions under which those components are executed for a separate problem instance. Given algorithm allows for obtaining exact solutions for a wide range of large capacity instances within a reasonable time, as well as for finding close to optimal solutions when calculations are interrupted untimely. Algorithm conforms to parallel computing.

Вступ

В умовах конкуренції на ринку підприємства звертають все більшу увагу на управління пріоритетами рішень в області своєї діяльності. Так, послугами компаній користуються різноманітні клієнти, деякі з яких важливіші за інших. Важливість клієнта для компанії залежить від низки факторів [1], як то тривалість співробітництва клієнта із компанією, як часто вони формують замовлення для компанії, яку частину віддачі компанії припадає на клієнта тощо. Запізнення роботи, якій відповідає певний ваговий коефіцієнт, являє собою приклад критерію, що враховує пріоритетність. Іншим приводом до використання критеріїв, в яких фігурує зважене запізнення – різні пріоритети устаткування (машин), що складають фонди підприємства.

Задача мінімізації сумарного зваженого запізнення виконання робіт на одному приладі має велике теоретичне та практичне значення. Так, найпростіше машинне середовище, що складається з одного приладу, часто має властивості, яких позбавлені складніші середовища. Складне машинне середовище, в якому єдина машина складає вузьке місце відносно критеріїв продуктивності, може розглядатись як єдина машина. Також на практиці задачі операційного планування в більш складних машинних середовищах часто можуть декомпозуватись на підзадачі, кожна з яких моделюється одним приладом. Дуже важливу роль відіграє ця задача в плануванні напівпровідникового виробництва [2].

Про найбільш ефективні методи розв'язання цієї задачі а рамках традиційних підходів можна дізнатись з оглядових статей [3, 4]. За допомогою цих методів можна отримувати точні розв'язки індивідуальних задач розмірності до 50 робіт. Для деяких класів індивідуальних задач отримані розв'язки задач розмірності до 100. Лоулер (Lawler) показав, що розглянута задача є NP-складною в сильному розумінні зведенням до неї задачі про 3-розбиття [5].

В [6] представлений новий підхід до розв'язання окремих задач календарного планування, який розширює межі застосування комбінаторного підходу та дозволяє знаходити точні розв'язки окремих NP-складних задач набагато більших розмірностей, ніж було можливо будь-яким іншим методом. Його суттю є попереднє дослідження властивостей відповідних класів важкорозв'язних задач, доведенні положень, правил, які дозволяють розробити єдиний принцип обчислень, і на їх основі будувати так звані ПДС-алгоритми (алгоритми із поліноміальною та декомпозиційною складовими). Такі алгоритми засновані на виділенні поліноміальної складової, та отриманні умов, коли експоненційна складова може декомпозу-

вати початкову задачу на підзадачі меншого розміру. 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, і часто наперед невідомо, який підалгоритм реалізується для довільної індивідуальної задачі.

ПДС-алгоритми побудовано на одноманітних ітераційних процедурах, та таким чином, що на кожній ітерації отримується розклад, який відповідає не гіршому значенню критерію оптимальності. Для кожної задачі намагаються показати, що відхилення отриманого на черговій ітерації допустимого розв'язку від оптимального залежить від самого отриманого розв'язку, і обчислення можна зупинити в будь-який момент, наприклад, при досягненні граничного часу виконання. Також структура побудованих ПДС-алгоритмів зазвичай дозволяє відкидати деякі процедури, що статистично лише незначно впливають на оцінку чергового розв'язку, тим самим виграючи в часі виконання.

Якісна відмінність ПДС-алгоритмів від існуючих – статистична значимість (висока частість отримання) точних розв'язків задач великої розмірності за прийнятний час, що відкриває широкі можливості для розв'язання реальних практичних важкорозв'язних задач.

Один варіант ПДС-алгоритму розв'язання досліджуваної задачі був запропонований в [7]. В даній статті проведений додатковий теоретичний аналіз задачі мінімізації сумарного зваженого запізнення виконання робіт на одному приладі, що доповнює результати з [7], та сформульовані p-умови розв'язання задачі. Реалізації запропонованого алгоритму можуть враховувати паралелізм обчислювального середовища.

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 $P(A) = \sum_{i \in A} p_i$.

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.

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

$$\sum_{i=1}^{n} w_i T_i = \sum_{i=1}^{n} w_i \max\left(0, C_j - d_j\right) \to \min. \qquad (27)$$

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.

Основні теоретичні положення

Наведемо ряд означень.

Означення 1. WSPT-розкладом (weighted shortest processing time – зважена найкоротша тривалість) називається послідовність робіт з J, в якій роботи впорядковані за незростанням пріоритетів. За умови рівності пріоритетів раніше призначається робота із меншим директивним строком, тобто

$$\left(\frac{w_{[i]}}{p_{[i]}} > \frac{w_{[j]}}{p_{[j]}} \vee \frac{w_{[i]}}{p_{[i]}} = \frac{w_{[j]}}{p_{[j]}} \wedge d_{[i]} \le d_{[j]}\right) \Leftrightarrow i \le j \,. \quad (28)$$

Така послідовність позначається σ_{WSPT}.

Подальші означення стосуються довільної послідовності робіт, якщо не зазначено інакше.

Означення 2. Перестановкою називається операція перенесення роботи $[q]$ на позицію $g > q$ разом із роботами $[q+1]$, $[q+2]$,…, $[g-1]$, $[g]$ на позиції q, $q+1$, $q+2$,…, $g-1$.

Означення 3. Інтервалом перестановки роботи $[q]$ на позицію g називається множина робіт

$$F(q,g) = \{[q+1],[q+2],\ldots,[g-1],[g]\}.$$

Нехай виконана перестановка із інтервалом $F = F(q,g)$. Якщо зважене запізнення деякої роботи $i \in F$ з цього інтервалу збільшилось, позначимо величину цього збільшення f_i^+. Означимо

$$f_F^+ = f_{[q]}^+ + \sum_{i \in F} f_i^+ .$$

Подібним чином, якщо зважене запізнення деякої роботи $j \in F$ з інтервалу перестановки зменшилось в результаті перестановки, будемо позначати величину цього зменшення f_j^-, і знову

$$f_F^- = f_{[q]}^- + \sum_{i \in F} f_i^- .$$

Тоді зміна значення цільової функції в результаті перестановки може бути виражена як

$$\Delta_F = f_F^+ - f_F^- . \quad (29)$$

Лема 1. Зміна значення цільової функції в результаті перестановки із інтервалом $F = F(q,g)$ може бути записана наступним чином:

1) збільшення значення цільової функції дорівнює

$$f_F^+ = f_{[q]}^+, \quad (30)$$

де

$$f_{[q]}^+ = \begin{cases} 0, & \text{якщо } S_{[q]} > P(F), \\ w_{[q]}\left(P(F) - S_{[q]}\right), & \text{якщо } P(F) \ge S_{[q]} \ge 0, \\ w_{[q]}P(F), & \text{якщо } S_{[q]} < 0; \end{cases}$$

2) зменшення значення цільової функції записується як

$$f_F^- = \sum_{i \in F} w_i \min\left(T_i, p_{[q]}\right). \quad (31)$$

Доведення. Запізнення робіт на інтервалі перестановки може лише зменшуватись, оскільки після перестановки вони починають виконуватись раніше, і $f_F^+ = \sum_{i \in F} f_i^+$. Натомість для роботи $[q]$ запізнення може лише збільшуватись, тому $f_F^+ = f_{[q]}^+$.

До перестановки вклад роботи $[q]$ до критерію

$$f_{[q]} = w_{[q]} \max(0, C_{[q]} - d_{[q]}),$$

а після перестановки

$$f'_{[q]} = w_{[q]} \max(0, C_{[q]} + P(F) - d_{[q]}).$$

Тоді

$$\begin{aligned} f_{[q]}^+ &= f'_{[q]} - f_{[q]} = \\ &= w_{[q]}\left(\max(0, P(F) - S_{[q]}) - \max(0, -S_{[q]})\right) = \\ &= w_{[q]}\left(\max(0, P(F) - S_{[q]}) + \min(0, S_{[q]})\right). \end{aligned}$$

Розглядаючи різні випадки, отримаємо вираз для $f_{[q]}^+$ з (30).

Дослідимо тепер зміну зваженого запізнення для довільної роботи $i \in F$. Її зважене запізнення до перестановки

$$f_i = w_i \max\left(0, C_i - d_i\right),$$

а після перестановки

$$f_i' = w_i \max(0, C_i - p_{[q]} - d_i),$$

причому воно зменшується. Тоді

$$\begin{aligned} f_i^- &= f_i - f_i' = \\ &= w_i\left(T_i - \max(0, C_i - p_{[q]} - d_i)\right) = \\ &= w_i \min\left(T_i, p_{[q]} + T_i - (C_i - d_i)\right) = \\ &= w_i \min\left(T_i, p_{[q]} + \max\left(0, S_i\right)\right). \end{aligned}$$

Якщо $T_i > 0$, $S_i < 0$ і $f_i^- = w_i \min\left(T_i, p_{[q]}\right)$. У випадку $T_i = 0$, $S_i \ge 0$,

$$f_i^- = \min\left(T_i, p_{[q]} + S_i\right) = 0 = \min\left(T_i, p_{[q]}\right).$$

Доведено.

Означення 4. Вставкою називається операція перенесення роботи $[g]$ на позицію $p < g$ разом із роботами $[p]$, $[p+1]$, $[p+2]$, …, $[g-1]$ на позиції $p+1, p+2, \ldots, g-1, g$.

Означення 5. Інтервалом вставки роботи $[g]$ на позицію p називається множина робіт

$$B(p,g) = \{[p],[p+1],\ldots,[g-1]\}.$$

Якщо p визначається з умови

$$0 \le P(B(p,g)) - T_{[g]} \le p_{[p]}, \qquad (32)$$

то такий інтервал називається власним інтервалом вставки роботи $[g]$, і позначається $B(g)$. Відповідна позиція найранішої роботи, що належить цьому інтервалу та визначається із співвідношення (32), позначається як $p_{\min}(g)$. На цій позиції зважене запізнення роботи $[g]$ дорівнює нулю або мінімальне. Якщо такої позиції p не існує, $p_{\min}(g) = 1$.

Як і для операції перестановки, для операції вставки із інтервалом $B = B(p,g)$ означимо зменшення та збільшення зваженого запізнення, що виникають в результаті її проведення. Маємо

$$f_B^+ = f_{[g]}^+ + \sum_{j\in B} f_j^+$$

$$f_B^- = f_{[g]}^- + \sum_{j\in B} f_j^-,$$

та

$$\Delta_B = f_B^+ - f_B^-. \qquad (33)$$

Лема 2. Зміна значення цільової функції в результаті вставки із інтервалом $B = B(p,g)$ записується наступним чином:

1) збільшення значення цільової функції дорівнює

$$f_B^+ = \sum_{i\in B} f_i^+, \qquad (34)$$

де для $i \in B$

$$f_i^+ = \begin{cases} 0, & \text{якщо } S_i > p_{[g]}, \\ w_i\left(p_{[g]} - S_i\right), & \text{якщо } p_{[g]} \ge S_i \ge 0, \\ w_i p_{[g]}, & \text{якщо } S_i < 0; \end{cases}$$

2) зменшення значення цільової функції

$$f_B^- = f_{[g]}^- = w_{[g]} \min\left(T_{[g]}, P(B)\right). \qquad (35)$$

Доведення. В результаті вставки запізнення роботи $[g]$ може лише зменшуватись, оскільки вона буде починатись раніше, і тому $f_B^- = f_{[g]}^-$. Роботи на інтервалі вставки натомість зміщуються вперед, їх запізнення може лише збільшуватись, і отже $f_B^+ = \sum_{i\in B} f_i^+$.

Розглянемо зміну зваженого запізнення для роботи $[g]$. До вставки воно складає

$$f_{[g]} = w_{[g]} \max\left(0, C_{[g]} - d_{[g]}\right),$$

а після вставки

$$f'_{[g]} = w_{[g]} \max\left(0, C_{[g]} - P(B) - d_{[g]}\right),$$

і оскільки запізнення цієї роботи зменшується

$$f_{[g]}^- = f_{[g]} - f'_{[g]}.$$

Викладками, які аналогічні тим, що використовувались в доведенні другого пункту леми 1 отримаємо (35).

Доведення другого пункту аналогічне доведенню першого пункту леми 1, для цього для довільної роботи в інтервалі вставки $i \in B$ потрібно взяти

$$f_i = w_i \max\left(0, C_i - d_i\right),$$

$$f_i' = w_i \max\left(0, C_i + p_{[g]} - d_i\right),$$

$$f_i^+ = f_i' - f_i.$$

Отже (34) також справедливе.

Доведено.

Далі, якщо не важливо, яка операція була виконана при зміні роботою j своєї позиції, то будемо узагальнено казати, що виконано перенесення цієї роботи. Якщо перенесення (перестановка чи вставка) призводить до зменшення цільової функції, будемо казати, що має місце продуктивне перенесення (перестановка чи вставка).

Лема 3. Вставка роботи $[g]$, яка запізнюється, на позицію $f < p_{\min}(g)$ не може виявитись більш продуктивною, ніж вставка цієї роботи із її власним інтервалом.

Доведення. Зміна значення критерію за вставки роботи $[g]$ із її власним інтервалом, враховуючи (32), записується як

$$\Delta_{B(g)} = \sum_{i\in B(g)} f_i^+ - w_{[g]} T_{[g]},$$

а зміна за її вставки на інтервалі $B(f,g)$, $f < p_{\min}(g)$, для якого

$$P(B(f,g)) > P(B(g)) \ge T_{[g]},$$

$$\Delta_{B(f,g)} = \sum_{i\in B(f,g)} f_i^+ - w_{[g]} T_{[g]} =$$

$$= \sum_{i\in B(g)} f_i^+ + \sum_{j=f}^{p_{\min}(g)-1} f_{[j]}^+ - w_{[g]} T_{[g]}.$$

Маємо

$$\Delta_{B(f,g)} - \Delta_{B(g)} = \sum_{j=f}^{p_{\min}(g)-1} f_{[j]}^+ \ge 0.$$

Доведено.

Означення 6. Процедурою вільної перестановки називається операція перестановки роботи $[q]$ на позицію $g > q$ за умови, що $C_{[g]} \le d_{[q]} < C_{[g+1]}$, та $\exists i \in F(q,g): S_i < 0$.

Примітка (до означення 6). При виконанні декількох вільних перестановок для уникання зациклення починають із робіт, що мають найбільший директивний строк.

Означення 7. Послідовність, отриману із σ_{WSPT} виконанням усіх вільних перестановок, позначимо через $\sigma_{ВП}$. В послідовності $\sigma_{ВП}$ порядок WSPT (28) зберігається лише для робіт, які запізнюються. Роботи із резервами часу (для яких $S_i > 0$) можуть порушувати цей порядок.

Примітка (до означення 7). За означенням операція вставки на інтервалі $B(p,g)$ в послідовності $\sigma_{ВП}$ супроводжується сортуванням усіх робіт, що знаходяться на цьому інтервалі, у порядку WSPT (28).

Лема 4. Якщо в послідовності σ_{WSPT} роботам, які запізнюються, не передують завдання із резервом часу, то не існує переносів робіт, що можуть зменшити цільову функцію, і така послідовність оптимальна.

Доведення. Нехай в σ_{WSPT} для $i = \overline{1,s}$ $S_{[i]} \le 0$, а для $j = \overline{s+1,n}$ $S_{[j]} \ge 0$. Зрозуміло, що продуктивні перенесення можливі лише для робіт на позиціях $i = \overline{1,s}$.

Виконаємо перестановку із інтервалом $F(q,g)$, $q < s, g \le s$. Згідно із (29) з урахуванням того, що $S_{[q]} < 0$ отримаємо

$$\Delta_F = w_{[q]}P(F) - \sum_{i\in F} w_i \min\left(T_i, p_{[q]}\right) \ge$$

$$\ge \sum_{i=q+1}^{g} \left(w_{[q]}p_{[i]} - w_{[i]}p_{[q]}\right) \ge 0,$$

оскільки $w_i / p_i \le w_{[q]} / p_{[q]}$, $i \in F(q,g)$ відповідно до (28).

Тепер виконаємо вставку із інтервалом $B(p,g)$, $p < s, g \le s$. Використавши (33) та врахувавши, що $S_i < 0$, $i \in B(p,g)$, можна отримати

$$\Delta_B = \sum_{i\in B} w_i p_{[g]} - w_{[g]} \min\left(T_{[g]}, P(B)\right) \ge$$

$$\ge \sum_{i=p}^{g-1} w_i p_{[g]} - w_{[g]}p_{[i]} \ge 0,$$

оскільки $w_{[g]} / p_{[g]} \le w_i / p_i$, $i \in B(p,g)$.

Доведено.

Примітка (до леми 4). Оскільки в доведенні розглядаються лише роботи, які запізнюються, а в послідовності $\sigma_{ВП}$ порядок (28) для таких робіт зберігається, то ця лема справедлива також і для послідовності $\sigma_{ВП}$.

Лема 5. Вставка роботи $[g]$ в послідовності σ_{WSPT} із інтервалом $B = B(p,g)$ може виявитись продуктивною лише якщо $\exists i \in B: S_i > 0$.

Доведення. Нехай $\forall i \in B: S_i \le 0$. Результат отримується аналогічно другій частині леми 4.

Доведено.

Примітка (до леми 5). Як і для леми 4, цей результат справедливий для послідовності $\sigma_{ВП}$.

Будемо позначати

$$T(\sigma) = \left\{ j \in J : T_j(\sigma) > 0 \right\}$$

та

$$S(\sigma) = \left\{ j \in J : S_j(\sigma) > 0 \right\}.$$

Якщо зрозуміло, яка послідовність мається на увазі, далі вона може не вказуватись.

Введемо подібні позначення також для робіт на позиціях з заданого інтервалу, а саме

$$T(p,g) = \left\{ [j] \in J,\ j = \overline{p,g} : T_{[j]} > 0 \right\},$$

$$S(p,g) = \left\{ [j] \in J,\ j = \overline{p,g} : S_{[j]} > 0 \right\}.$$

При цьому конкретна послідовність не вказується та розуміється з контексту.

Лема 6. Якщо в послідовності σ_{WSPT} жодній роботі $[g]$, яка запізнюється, не передує робота $[i]$, $S_{[i]} > 0$, $i = \overline{1, g-1}$, для якої виконується

$$d_{[i]} > d_{[g]} - p_{[g]}, \qquad (36)$$

то не існує продуктивних перенесень робіт, які запізнюються.

Доведення. Спочатку покажемо, що робота із резервом часу, для якої не виконується (36) не може безпосередньо передувати роботі, яка запізнюється. Нехай робота із резервом часу призначена на позиції i, а наступна за нею робота запізнюється, тобто $d_{[i]} > C_{[i]}$, $d_{[i+1]} < C_{[i+1]}$, тоді

$$d_{[i]} \le d_{[i+1]} - p_{[i+1]} < C_{[i+1]} - p_{[i+1]} = C_{[i]}.$$

Отримали протиріччя, що і доводить вихідне положення.

Розглянемо можливі вставки. З огляду на попереднє положення умови леми можуть виконуватись лише тоді, коли на інтервалі вставки роботи $[g]$, яка запізнюється, відсутні роботи з резервами. Тоді результат випливає з леми 5.

Тепер будемо розглядати можливі перестановки. Знову, з огляду на попереднє положення на інтервалі перестановки у всіх робіт відсутні резерви, і результат отримується аналогічно доведенню першої частини леми 4.

Залишилось показати, що перестановка роботи $[q]$ із резервами, для якої не справедливе (36), на позицію роботи $[g]$, що запізнюється, не є продуктивною. При цьому на інтервалі перестановки $F = F(q,g)$ можуть бути присутні інші роботи, що запізнюються. Позначимо позицію роботи, що запізнюється на інтервалі F та призначена найраніше, через g^*. Маємо

$$S_{[q]} = d_{[q]} - C_{[q]} \le d_{[g^*]} - C_{[q]} - p_{[g^*]} <$$

$$< C_{[g^*]} - C_{[q]} - p_{[g^*]} = \sum_{i=q+1}^{g^*-1} p_{[i]}.$$

$$\Delta_F = w_{[q]}\left(P(F) - S_{[q]}\right) - \sum_{i\in F} w_i \min\left(T_i, p_{[q]}\right) \ge$$

$$\ge w_{[q]}\left(P(F) - S_{[q]}\right) - \sum_{i\in T(q+1,g)} w_i p_{[q]} >$$

$$> w_{[q]}\left(\sum_{i=q+1}^{g} p_{[i]} - \sum_{i=q+1}^{g^*-1} p_{[i]}\right) - \sum_{i\in T(q+1,g)} w_i p_{[q]} =$$

$$= w_{[q]} \sum_{i=g^*}^{g} p_{[i]} - \sum_{i\in T(q+1,g)} w_i p_{[q]} \ge 0,$$

оскільки в σ_{WSPT} виконується (28).

Доведено.

Примітка (до леми 6). Висновок леми зроблений з використанням WSPT-порядку (28) лише тих робіт, які запізнюються, тому лема справедлива також і для послідовності $\sigma_{ВП}$.

Лема 7. В послідовності σ_{WSPT} додаткові резерви часу на інтервалі вставки $B(p,g)$ роботи $[g]$, яка запізнюється, можуть утворювати лише роботи $[k]$, $k = \overline{1, p-1}$, для яких $S_{[k]} > 0$, $d_{[k]} > d_{[g]} - p_{[g]}$ та

$$d_{[k]} > C_{[p]} - p_{[p]} = C_{[p-1]}. \qquad (37)$$

Доведення. Зрозуміло, що роботи, які вносять додаткові резерви часу на інтервал вставки, мають відповідати умовам, за яких вставка буде продуктивною (лема 6), тому $d_{[k]} > d_{[g]} - p_{[g]}$. Додаткові резерви на інтервалі вставки утворюють лише роботи, для яких виконується (32), звідки

$$T_{[g]} \le \sum_{i=p+1}^{g} p_{[i]},$$

$$d_{[k]} + p_{[g]} > d_{[g]} \ge \sum_{i=1}^{p} p_{[i]},$$

$$d_{[k]} > \sum_{i=1}^{p-1} p_{[i]} = C_{[p]} - p_{[p]}.$$

Це автоматично означає, що $S_{[k]} > 0$.

Доведено.

Примітка (до леми 7). В доведення цієї леми використовується лема 6, справедлива для послідовності $\sigma_{ВП}$, отже і сама ця лема також справедлива для $\sigma_{ВП}$.

Теорема 1. Нехай в послідовності $\sigma_{ВП}$ $[g]$ – робота, що запізнюється. Тоді зменшення значення цільової функції при переміщенні $[g]$ на більш ранні позиції в результаті перенесень робіт можливі лише за виконання наступних умов:

1) $\exists i \in B(g) : S_i > 0, d_i > d_{[g]} - p_{[g]}$;
2) $\exists q : q < g, d_{[q]} > C_{[g]}$;
3) $\exists q : q < g, C_{[g]} \ge d_{[q]} > C_{[g]} - p_{[g]}$,
   $\min\left(C_{[g]} - d_{[g]}, p_{[q]}\right) w_{[g]} - \left(C_{[g]} - d_{[q]}\right) w_{[q]} > 0$.
4) $\forall i \in B(g,p), S_i \le 0\ \exists k : k < p$,
   $S_{[k]} > 0, d_{[k]} > C_{[p-1]}, d_{[k]} > d_{[g]} - p_{[g]}$.

Доведення. Усі твердження випливають з попередньо викладеного матеріалу.

Твердження 1 випливає з леми 6, якщо додатково використати лему 3.

Твердження 2 та 3 випливають з (29) та леми 1. Мається на увазі перестановка з інтервалом $F(q,g)$, в результаті якої зміна цільової функції буде мати вигляд

$$\Delta_F = f^+_{[q]} - \sum_{i\in F} w_i \min\left(T_i, p_{[q]}\right).$$

Якщо $d_{[q]} > C_{[g]}$, $S_{[q]} > P(F)$, і тому $\Delta_F < 0$ (твердження 2). Якщо ж

$C_{[g]} \ge d_{[q]} > C_{[g]} - p_{[g]}$,

$P(F) \ge S_{[q]} \ge 0$, і

$$\Delta_F \le w_{[q]}\left(P(F) - S_{[q]}\right) - w_{[g]} \min\left(C_{[g]} - d_{[g]}, p_{[q]}\right).$$

Щоб $\Delta_F < 0$, повинна виконуватись умова твердження 3.

Твердження 4 є повторним ствердженням леми 7.

Доведено.

Наслідок 1 (до теореми 1). Якщо в послідовності $\sigma_{ВП}$ $|T| > 1$ та для жодної з робіт $[g]$, які запізнюються, не виконуються твердження 1-4

теореми 1, то $\sigma_{ВП}$ оптимальна. Це саме стосується і послідовності σ_{WSPT}.

Нехай в послідовності σ_{WSPT} робота $[g]$, що запізнюється, в результаті вставки із власним інтервалом зайняла позицію $m = p_{\min}(g)$, але запізнюватись не перестала. Позначимо отриману послідовність $\sigma([g])$, помітимо роботу $[m]$ в цій послідовності символом «*».

Лема 8. В послідовності $\sigma([g])$ робота $[k]$, де $k = \overline{m+1, g}$, може бути переміщена на більш ранню позицію в результаті продуктивної вставки (якщо на інтервалі вставки резерви відсутні, див. лему 5) лише в результаті перестановки $[m]^*$ після роботи $[k]$.

Доведення. Виразимо зміну значення цільової функції для випадку перестановки $[m]^*$ на позицію k. Будемо розглядати випадок, коли $S_i < 0, i \in F$, і

$$\Delta_F = w_{[m]}P(F) - \sum_{i\in F} w_i \min\left(T_i, p_{[m]}\right) =$$

$$= \sum_{i=m+1}^{k}\left[w_{[m]}p_{[i]} - w_{[i]}\min\left(C_i - d_i, p_{[m]}\right)\right].$$

Розіб'ємо інтервал перестановки F на дві підмножини: $F = P \cup Q, P \cup Q = \varnothing$,

$$P = \left\{i \in F : p_{[m]} \ge C_i - d_i\right\},$$

$$Q = \left\{i \in F : p_{[m]} < C_i - d_i\right\},$$

$$\Delta_F = \sum_{i\in P}\left[w_{[m]}p_i - w_i\left(C_i - d_i\right)\right] +$$

$$+\sum_{i\in Q}\left[w_{[m]}p_i - w_i p_{[m]}\right].$$

Винесемо з кожного члену суми вираз $p_i p_{[m]}$, позначимо

$$\xi_i = \frac{C_i - d_i}{p_{[m]}}, i \in P,$$

тоді $\xi_i \in [0,1], i \in P$. Маємо

$$\Delta_F = \sum_{i\in P} p_i p_{[m]}\left[\frac{w_{[m]}}{p_{[m]}} - \frac{w_i}{p_i}\xi_i\right] +$$

$$+\sum_{i\in Q} p_i p_{[m]}\left[\frac{w_{[m]}}{p_{[m]}} - \frac{w_i}{p_i}\right].$$

Якщо $|P| < |Q|$ або ξ_i близькі до 1, Δ_F може бути менше нуля, оскільки $w_{[m]}/p_{[m]} \le w_i/p_i$, $i \in F(m,g)$.

Доведено.

Означення 8. Робота $[g] \in T(\sigma_{ВП})$, для якої виконується

$$\exists k < g : S_{[k]} > 0, d_{[k]} > d_{[g]} - p_{[g]},$$

називається конкуруючою.

Означення 9. Робота із резервом часу, що належала інтервалу вставки $B(p,g)$ роботи $[g]$, яка запізнюється, та після вставки роботи $[g]$ на позицію p починає запізнюватись, називається породженою.

Кількість ітерацій запропонованого алгоритму визначається числом конкуруючих робіт у послідовності $\sigma_{ВП}$. На k-й ітерації шляхом направлених продуктивних перенесень будується оптимальна підпослідовність на інтервалі $\overline{1, \eta_k}$, де η_k – позиція k-ї конкуруючої роботи у $\sigma_{ВП}$. Послідовність, отриману до початку k-ї ітерації алгоритму будемо позначати як σ_k. $\sigma_1 = \sigma_{ВП}$.

Означення 10. Межею оптимальності послідовності σ_k на ітерації k етапу оптимізації ПДС-алгоритму буде позиція η_k така, що роботи $([1],\ldots,[\eta_k]) \subset \sigma_k$ утворюють оптимальну підпослідовність.

Введемо правила призначення міток при виконані перенесень робіт в послідовності, що розглядається на k-й ітерації етапу оптимізації. В цій послідовності символом «*» позначаються роботи, які запізнювались в σ_k та в результаті виконання перенесень переміщуються на більш ранні позиції, використавши при цьому резерви часу передуючих робіт. Символом «**» помічаються роботи, до цього позначені «*» (вони запізнювались в послідовності $\sigma_{ВП}$), які в результаті перенесень перемістилися на пізніші позиції, але ці позиції не відповідають пріоритетам цих робіт. За роботами, що помічені «*» та «**» в послідовності σ_k можуть слідувати роботи із більшим пріоритетом, які запізнюються.

Таким чином структура σ_k наступна. У вихідній послідовності $\sigma_{ВП}$ для робіт, що запізнюються, зберігається порядок відповідно до (28). Пріоритетну впорядкованість порушують завдання із мітками, які в результаті перестановок та вставок використали резерви часу передуючих робіт та перемістились з позицій, що відповідають їхнім пріоритетам, на більш ранішні позиції, що дозволило покращити значення цільової функції.

Теорема 2. Нехай в послідовності σ_k [g] – робота, що запізнюється. Тоді зменшення значення цільової функції при переміщенні [g] на більш ранні позиції в результаті перенесень робіт можливі лише за виконання умов 1-4 теореми 1, або ж якщо

$$\forall i = \overline{1, g-1},\ S_{[i]} \le 0,\ \exists [m]^* \left([m]^{**}\right),$$

$$\frac{\omega_{[m]}}{p_{[m]}} \le \frac{\omega_{[g]}}{p_{[g]}},\ m < g.$$

Доведення. Випливає із структури σ_k та леми 8.

Доведено.

На кожній ітерації алгоритму виконуються наступні типи перенесень робіт. Виконуються перестановки (тільки якщо вони продуктивні):

- робіт, для яких резерв часу додатний;
- робіт, що використали резерви в результаті перестановок та вставок, за умови, що за ними слідують роботи, які запізнюються та мають не менший пріоритет.

Виконуються наступні типи вставок (якщо вони знижують запізнення відповідної роботи):

- вставки робіт за наявності резервів часу на інтервалі вставки;
- вставки після внесення резервів на відповідний інтервал за твердженням 4 теореми 1.

Примітка (до теореми 2). Теорема 2 не тільки постулює умови оптимальності послідоності σ_k, а й стверджує, що за неможливості виконання жодного типу перенесень із описаного набору отримана оптимальна послідовність. Таким чином за допомогою комбінації запроваджених перенесень завжди можна отримати оптимальну послідовність.

Можна виключати деякі операції перенесення, використовуючи так звані правила домінування. Правилом домінування називається твердження про те, що при виконанні певного співвідношення між параметрами задачі одна робота обов'язково передує іншій хоча б в одному оптимальному розкладі. Багато правил домінування були запропоновані в [8, 9, 10]. Використовуючи ці правила будується асиметричне транзитивне відношення передування, що відображає частковий порядок слідування робіт в деякому оптимальному розкладі.

Нехай таке відношення було побудоване на множині J, позначимо його M. Воно може задаватись, наприклад, за допомогою матричного представлення. Якщо $(j,k) \in M$, будемо це записувати як $j \to k$. Роботи $j,k \in J$ не обов'язково суміжні. Наступна теорема вказує стратегію використання цього відношення для скорочення кількості перенесень.

Теорема 3. Нехай з'ясовано, що $i \to j$. Тоді в ході виконання алгоритму на послідовності σ_k можна приймати наступні рішення, які виключають лише непродуктивні перенесення:

1) замість інтервалу вставки $B(p,g)$ роботи j = [g], яка запізнюється, береться інтервал $B(q,g)$, i = [q], $q > p$;

2) нехай j – чергова конкуруюча робота, що знаходиться на позиції g, $i \in B(g)$, причому $i \in T$, $i = [r]$, тоді оптимізація здійснюється на інтервалі $\overline{r+1, g-1}$;

3) нехай i – це k-та конкуруюча робота, що знаходиться на позиції g в послідовності σ_k, в послідовності σ_{k+1} вона займає позицію $q > p_{\min}(g)$, тоді робота j, що є наступною конкуруючою, в оптимальному розкладі не може зайняти позицію, меншу ніж q+1.

Доведення. Доведемо сформульовані правила по черзі.

Вставка роботи [g] на позицію $q > p_{\min}(g)$ є продуктивною відповідно до леми 5. Вставка на позицію $p < q$ порушить відношення передування, та буде в кінці кінців відмінена зворотною перестановкою або перестановками відповідно до теореми 2. Це доводить правило 1.

Правило 2 є коректним, оскільки резервів робіт, що знаходяться на інтервалі $\overline{1, r-1}$, виявилось недостатньо для зменшення запізнення роботи $i = [r]$ на попередніх ітераціях, і тому з цього інтервалу резерви не можуть бути внесені до власного інтервалу вставки роботи $j = [g]$. Залишився варіант, коли [r] – робота із міткою, але вона не може слідувати за j, отже її перестановка не виконується.

Правило 3 є очевидним для поточної ітерації. На всіх інших ітераціях воно справедливе щодо найбільшої позиції попередніх конкуруючих робіт.

Доведено.

Теорема 4. Якщо в σ_k конкуруюча робота i = [g] після виконання усіх можливих перенесень не перемістилася, оптимальне значення критерію відповідає позиції g, яку займає ця

робота, та з конкуруючих виключаються усі роботи, $[j]$, $j = \overline{g+1,r}$, якщо

$\forall j = \overline{g+1,r}\,[j] \in T, r \le n, [g] \to [j]$.

Доведення. Для доведення використовується твердження 3 теореми 3 та лема 5.

Доведено.

Елементарні операції алгоритму

Тут і далі поточною роботою будемо називати конкуруючу або породжену роботу, що запізнюється, та для якої виконуються перенесення, спрямовані на зменшення її запізнення.

Для полегшення опису та супроводження реалізацій досліджуваного ПДС-алгоритму розв'язання задачі, в ньому зручно виділити логічно окреслені елементарні операції, коректність роботи яких можна перевіряти окремо. Їх перелік наступний:

1) виконання усіх вільних перестановок на деякому інтервалі робіт на позиціях $\overline{p,g}$;

2) спроба зменшення цільової функції за рахунок використання заданою роботою $[g]$, яка запізнюється, резервів робіт на інтервалі $\overline{p,g}$, що не пов'язано із внесенням резервів часу в середину інтервалу вставки (виконання незалежної перестановки);

3) операція вставки поточної роботи, що запізнюється, із наступним упорядкуванням за незростанням пріоритетів усіх робіт на інтервалі $\overline{p,g}$;

4) виконання усіх можливих перестановок, які призначені для внесення додаткових резервів часу в середину інтервалу вставки за рахунок робіт, що мають ці резерви та передують фактичній позиції вставки поточної роботи, яка запізнюється (внесення додаткових резервів часу в інтервал $\overline{p^{\Phi},g}$, переглядаються роботи на інтервалі $\overline{1,p^{\Phi}-1}$);

5) оптимізація поточної роботи $[g]$, яка запізнюється (для неї знаходиться фактичний інтервал вставки, що позначається як $\overline{p^{\Phi},g}$, а також відбувається оптимізація за рахунок резервів робіт на позиціях до p^{Φ} – ця операція включає в себе попередні дві);

6) оптимізація інтервалу робіт на позиціях $\overline{1,g}$ за рахунок робіт із мітками (декомпозиційна складова алгоритму).

Окремими операціями можна вважати також перевірку умов оптимальності послідовностей робіт σ_{WSPT}, $\sigma_{ВП}$ (наслідок теореми 1), σ_k (наслідок теореми 1), але ці операції виконуються безпосередньо.

Наведемо формальний опис виконання вільних перестановок на інтервалі $\overline{p,g}$. Ці перестановки завжди продуктивні за твердженням 2 теореми 1.

Елементарна операція 1 (виконання вільних перестановок).

1. Обчислити множину

$$Y = \left\{ (k,q): \begin{array}{l} k,q \in \overline{p,g},\, k < q, \\ C_{[q]} \le d_{[k]} \le C_{[q+1]}, \\ \exists i \in \overline{k+1,q} : S_{[i]} < 0 \end{array} \right\}.$$

2. Поки $Y \neq \varnothing$ виконувати: вилучити з Y пару (k',q'), де $[k']$ має найбільший директивний строк серед усіх робіт у Y, виконати перестановку $[k']$ на позицію q'.

Примітка (до елементарної операції 1). Якщо необхідно здійснювати лише вільні перестановки, що зменшують запізнення деякої заданої роботи $[g']$, роботи з Y, що стоять на позиціях $k > g' \vee q < g'$, не розглядаються.

Операція виконання незалежної перестановки базується на твердженні 3 теореми 1, Формальний опис її здійснення подано нижче.

Елементарна операція 2 (виконання незалежної перестановки).

1. Знайти роботу $[l], l \in \overline{p,g}$ (на її позицію в середині інтервалу не накладається жодних обмежень), для якої виконується

$d_{[l]} \ge C_{[g]} - p_{[g]}$,

$w_{[l]}\left(C_{[g]} - d_{[l]}\right) < w_{[g]} \min\left(p_{[l]}, C_{[g]} - d_{[g]}\right)$.

2. Якщо така робота знайдена, виконати перестановку роботи $[l]$ на позицію g, після чого помітити $[g]$ символом «*».

Після виконання вставки поточної роботи, що запізнюється, з позиції g на позицію p, потрібно усі роботи з інтервалу вставки впорядкувати у відповідності до їх пріоритетів, оскільки цьому інтервалі можуть знаходитись роботи, які були перенесені в результаті вільних перестановок та які порушують пріоритетну впорядкованість. Враховуючи таку необхідність, вставка роботи має виконуватись наступним чином.

Елементарна операція 3 (вставка поточної роботи).

1. Виконати вставку роботи $[g]$ на позицію p, помітити цю роботу символом «*».

2. Впорядкувати на інтервалі $\overline{p+1,g}$ роботи в порядку (28), знищуючи при цьому мітки «*» та «**» усіх робіт на цьому інтервалі.

3. Виконати на інтервалі $\overline{p+1,g}$ усі можливі вільні перестановки, кінець.

Внесення додаткових резервів часу включається в елементарну операцію оптимізації поточної роботи. Вона виконується після вставки поточної роботи і перед виконанням цієї операції поточна робота вже знаходиться на позиції p (тобто внесення резервів на інтервал супроводжується його розширенням).

Опишемо формально хід виконання цієї операції на інтервалі $\overline{1,p-1}$.

Елементарна операція 4 (внесення резервів часу на інтервал вставки поточної роботи):

1. Виконати вільні перестановки, що зменшують запізнення роботи $[p]$, після чого перевизначити p; якщо робота $[p]$ перестала запізнюватись, кінець; інакше перейти на крок 2.

2. Знайти роботу $[l]$, $l < p$, для якої виконується

$$S_{[l]} > 0,\ d_{[l]} > s_{[g]},\ d_{[l]} > C_{[p-1]};$$

якщо така робота була знайдена, перейти на крок 3; інакше кінець.

3. Виконати перестановку роботи $[l]$ на позицію p, покласти $p := p-1$; якщо $[l]$ перестала запізнюватись, кінець; інакше перейти на крок 1.

Примітка (до елементарної операції 4). Перегляд робіт-кандидатів відбувається починаючи з тих, що мають менший пріоритет.

Під час оптимізації робиться спроба усунути запізнення поточної роботи спочатку за рахунок резервів робіт на її фактичному інтервалі вставки, а якщо їх недостатньо, то за рахунок внесення додаткових резервів робіт, що передують цьому інтервалу.

У випадку, коли поточна робота після оптимізації все одно запізнюється, зміни, що були здійснені, відміняються, і подальша оптимізація проводиться за рахунок робіт із мітками (реалізується декомпозиційна складова алгоритму).

Далі слідує формальний опис операції виконання оптимізації поточної роботи $j \in T$, $j = [g]$ (позначимо найпізніший момент початку виконання роботи j як $s_j = d_j - p_j$).

Елементарна операція 5 (оптимізація поточної роботи).

1. Знайти попередню позицію вставки p із співвідношення $C_{[p-1]} \le s_{[g]} < C_{[p]}$. Якщо $C_{[1]} > s_{[g]}$, $p := 1$.

2. Фактична позиція вставки p^{Φ} покладається рівною

$$p^{\Phi} := \max\{p, i+1, k+1\},$$

$$p < i+1 < g,\ p < k+1 < g,$$

$$\frac{w_{[i]^*}}{p_{[i]^*}}\left(\frac{w_{[i]^{**}}}{p_{[i]^{**}}}\right) < \frac{w_{[g]}}{p_{[g]}},\ [k] \to [g].$$

3. Запам'ятати поточну послідовність, позначивши її σ_{Φ}. Виконати вставку роботи j на позицію p^{Φ}. Якщо робота перестала запізнюватись, то кінець; інакше перейти на крок 4.

4. Здійснити спробу внести додаткові резерви часу в інтервал вставки роботи j; якщо в результаті цього робота j перестала запізнюватись, кінець; інакше повернутись до послідовності σ_{Φ}, перейти на крок 5.

5. Для кожної роботи $[i]$, $i = \overline{p^{\Phi}+1, g-1}$ в послідовності σ_{Φ} виконувати: якщо $S_{[i]} > 0$ та $d_{[i]} > s_{[g]}$, виконати вставку $j = [g]$ на позицію i, кінець (якщо на інтервалі будуть безрезультатно переглянуті усі роботи, поточна робота залишається на своїй позиції у розкладі).

Оптимізація заданого інтервалу робіт за рахунок вивільнення зайнятих ресурсів роботами із мітками – це елементарна операція, що реалізує декомпозиційну складову алгоритму. Ця елементарна операція включає рекурсивний виклик оптимізаційного етапу ПДС-алгоритму для підпослідовності робіт. Декілька таких підпослідовностей можуть бути проаналізовані паралельно, таким чином запроваджуючи у алгоритм масштабованість на паралельні обчислювальні системи.

Формальний опис проведення цієї операції подано нижче. Поточна робота призначена на позицію g.

Елементарна операція 6 (декомпозиційна складова алгоритму).

1. Знайти роботу $[m]^* \left([m]^{**}\right)$ на інтервалі $\overline{1,g-1}$, для якої

$w_{[m]}\left(C_{[k]}-d_{[m]}\right)<\sum_{i=m}^{g} w_{[i]}T_{[i]}$.

Якщо такі роботи не знайдено, кінець; інакше покласти

$$k:=\begin{cases} g, \text{ якщо } \dfrac{w_{[m]}}{p_{[m]}} \le \dfrac{w_{[g]}}{p_{[g]}}, \\ s, \text{ інакше,} \end{cases}$$

де $s+1$ – позиція першої роботи без мітки, пріоритет якої менше пріоритету роботи $[m]^*$ $\left([m]^{**}\right)$, $s>m$, перейти на крок 2.

2. Зберегти значення цільової функції та порядок робіт у поточній послідовності, позначивши її як σ_Φ.

3. Виконати перестановку роботи $[m]^*$ $\left([m]^{**}\right)$ на позицію k, що була визначена на кроці 1 та яка відповідає пріоритету цієї роботи.

4. Виконати етап оптимізації на інтервалі $\overline{1,g-1}$ (рекурсивний виклик), поклавши значення параметру $\eta:=m-1$, де η – межа оптимальності.

5. Обчислити значення цільової функції отриманої послідовності (позначивши її через σ_H) та порівняти його із значенням цільової функції для послідовності σ_Φ. Якщо значення цільової функції не зменшилось, повернутись до послідовності σ_Φ, та перейти на крок 1, виключаючи із розгляду роботу $[m]^*$ $\left([m]^{**}\right)$; інакше перейти на крок 6.

6. Якщо робота $[m]^*$ ($[m]^{**}$) зайняла позицію, що відповідає її пріоритету; позначення «*» («**») для неї знімається; інакше вона безумовно позначається символом «**».

7. Покласти $g:=m$ (робота із міткою, що розглядалась, стає поточною), перейти на крок 1.

Примітка (до елементарної операції 6). Роботи із мітками розглядаються в порядку, зворотному їх маркуванню (спочатку розглядається робота, помічена останньою).

Формальний опис алгоритму

В попередньому розділі був даний формальний опис алгоритмів виконання усіх елементарних операцій ПДС-алгоритму. Тепер, використовуючи ці елементарні операції як операції більш високого рівня абстракції, наведемо формальний опис логіки ПДС-алгоритму в цілому, який складається із двох етапів: попереднього етапу та етапу оптимізації.

Попередній етап об'єднує наступні кроки:

1. Знайти початкове відношення M передування, використовуючи певний набір правил домінування.

2. Впорядкувати роботи в порядку (28), отримуючи послідовність σ_{WSPT}; перевірити умови оптимальності цієї послідовності (наслідок теореми 1), якщо вони справджуються, то кінець; інакше перейти до кроку 3.

3. Виконати усі можливі вільні перестановки в послідовності σ_{WSPT} отримавши таким чином послідовність $\sigma_{ВП}$; перевірити умови оптимальності цієї послідовності (наслідок 1 теореми 1), якщо вони справджуються, то кінець; інакше перейти до першого кроку етапу оптимізації.

Опишемо кроки етапу оптимізації. Нехай η – позиція передуючої роботи, що запізнювалась і яка розглядалась на попередній ітерації етапу оптимізації, вона буде межею оптимальності на поточній ітерації.

1. Знайти чергову роботу [g], яка запізнюється на інтервалі $\overline{\eta+1,n}$; якщо така робота відсутня, то кінець; інакше перейти на крок 2.

2. Якщо для [g] виконується

$\exists l<g: S_{[l]}>0, d_{[l]}>s_{[g]}$,

тобто вона є конкуруючою, виконати наступні кроки; інакше перейти на крок 3:

2.1) перевірити умови оптимальності послідовності σ_k, якщо вони справджуються, то кінець; інакше перейти на крок 2.2;

2.2) перевірити, чи можна виключити роботу [g] з множини конкуруючих робіт, для цього перевірити для неї умови теорем 3 та 4; якщо це так, покласти $\eta:=g$, доповнити відношення передування M, перейти на крок 1; інакше перейти на крок 2.3;

2.3) запам'ятати поточне значення цільової функції та послідовність σ_k, розпочинається нова ітерація алгоритму, перейти на крок 3.

3. Зробити спробу виконати незалежну перестановку; якщо перестановка продуктивна та робота [g] перестала запізнюватись, покласти $\eta:=g'=g-1$ (g' – нова позиція поточної роботи після незалежної перестановки), перейти на крок 1; інакше перейти на крок 4.

4. Провести оптимізацію роботи [g]. Якщо робота перестала запізнюватись, покласти $\eta := g'$, перейти на крок 1; інакше перейти на крок 5.

5. Оптимізувати поточну роботу за рахунок робіт із мітками на інтервалі $\overline{1, g'}$, перевизначити g', покласти $\eta := g'$ та перейти на крок 1.

Характеристика поліноміальної складової алгоритму

Якщо в процесі розв'язання довільної індивідуальної задачі виконуються строго визначені логіко-аналітичні умови, то ця індивідуальна задача розв'язується поліноміальним підалгоритмом (елементарні операції 1-5) точно із часовою складністю, що є поліномом від потужності розкладу.

В наступних твердженнях розглядаються окремі випадки, коли оптимальний розв'язок задачі досягається відпрацюванням поліноміальної складової алгоритму.

Лема 9. Якщо відношення передування на деякій ітерації утворює повний порядок, то оптимальну послідовність можна отримати, сортуючи роботи відповідно до цього порядку. Часова складність в цьому випадку – $O(n)$.

Лема 10. Нехай в послідовності σ_{WSPT} виконується:

$$\max_{i\in J}(S_i) \le p_k \ \forall k : C_k - p_k > 0,$$

тоді оптимальне значення критерію досягається за поліноміальний час із складністю $O(n^2)$.

Доведення. Існуючі резерви кожного завдання в цьому випадку може використати лише одна конкуруюча робота. Складність визначається процедурою вибору найбільш ефективної позиції вставки кожної роботи, що запізнюється, і потребує $O(n^2)$ операцій.

Доведено.

Лема 11. Нехай в послідовності $\sigma_{ВП}$ [g] – найперша робота, яка запізнюється. Якщо на інтервалі $\overline{1, g-1}$ роботи зберігають порядок (28),

$$\max_{i\in J}(S_i) \le p_{[g]},$$

та

$$\forall s = \overline{g+1, n}\ S_{[s]} \le 0, \qquad p_s \ge p_g,\ d_s \ge d_g,$$

то задача розв'язується за поліноміальний час із складністю $O(n^2)$.

Доведення. При виконанні умов леми процедура вставки реалізується лише для єдиної конкуруючої роботи. Складність визначається операцією побудови послідовності $\sigma_{ВП}$, що складає $O(n^2)$.

Доведено.

Лема 12. Нехай під час виконання кожної ітерації оптимізації за вставки чергової роботи $[g] \in T$ виконується:

1) фактична позиція вставки p^{Φ} виявляється більшою за позицію усіх робіт із мітками;

2) на інтервалі $\overline{1, p^{\Phi}-1}$ відсутні роботи, для яких виконується твердження 1 теореми 1;

3) перестановки робіт із мітками з інтервалу $\overline{1, p^{\Phi}-1}$ на інтервал вставки чергової роботи, що запізнюється, є непродуктивними.

Тоді задача розв'язується за поліноміальний час із складністю $O(n^3)$.

Доведення. При виконанні цих умов для кожної поточної послідовності σ_k виконуються умови леми 11. Кількість ітерацій обмежена кількістю робіт в множині J, отже складність виконання алгоритму в цьому випадку не буде перевищувати $O(n^3)$.

Доведено.

Теорема 5. Поліноміальна складова алгоритму, заснована на лемах 10–12, має часову складність $O(n^3)$.

Доведення

Найскладніший випадок реалізації поліноміальної складової описаний в лемі 12, та складає $O(n^3)$. Таку саму складність має операція побудови матриці транзитивного відношення передування *M*.

Доведено.

Висновки

В цій статті на основі попередніх досліджень властивостей задачі мінімізації сумарного зваженого запізнення виконання робіт одним приладом та проведення додаткового теоретичного аналізу був сформульований ПДС-алгоритм її розв'язання, який може реалізовуватись у паралельному обчислювальному середовищі.

Подальші дослідження включатимуть формалізацію та обгрунтування додаткових умов реалізації поліноміальної складової алгоритму,

програмну реалізацію запропонованого алгоритму та проведення обчислювальних експериментів для порівняння його продуктивності із еталонним алгоритмом математичного програмування, в якому використовуються відомі результати [3, 4].

Перелік посилань

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. Mason S.J., Fowler J.W., Carlyle M. A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops // Journal of Scheduling. – 2002, Vol. 5 (3). – P.247-262.
3. Sen T., Sulek J.M. Dileepan P. Static Scheduling Research to Minimize Weighted and Unweighted Tardiness: A State-of-the-Art Survey // International Journal of Production Economics. – 2003, Vol. 83 – P.1-12.
4. Koulamas C. The Single-Machine Total Tardiness Scheduling Problem: Review and Extensions // European Journal of Operational Research. – 2010. – Vol. 202 (1). – P.1-7.
5. Lawler E.L. A "Pseudopolynomial" Algorithm for Sequencing Jobs to Minimize Total Tardiness // Annals of Discrete Mathematics. – 1977, Vol. 1. – P.331-342.
6. Згуровский М.З., Павлов А.А. Принятие решений в сетевых системах с ограниченными ресурсами: Монография.– К.: Наукова думка. – 2010. – 573 с.
7. 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.
8. 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.
9. Kanet, J.J. New precedence theorems for one-machine weighted tardiness // Mathematics of Operations Research. – 2007, Vol. 32. – P.579–588.
10. 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.