УДК 519.854.2

Ф. Г. ВАЩУК, д-р техн. наук, проф., ректор Закарпатського державного університету, м. Ужгород;
О. А. ПАВЛОВ, д-р техн. наук, проф., декан факультету інформатики та обчислювальної техніки НТУУ «КПІ», м. Київ;
О. Б .МІСЮРА, канд. техн. наук, ст. наук. співр. кафедри автоматизованих систем обробки інформації та управління НТУУ «КПІ», м. Київ;
О. О. МЕЛЬНИК, ст. викл. кафедри загальної інформатики та математичного моделювання Закарпатського державного університету, м. Ужгород

СКЛАДАННЯ РОЗКЛАДІВ ГРУП ДЛЯ ОДНОГО ПРИЛАДУ ІЗ НАЛАГОДЖЕННЯМИ ЗА КРИТЕРІЄМ МІНІМІЗАЦІЇ СУМАРНОГО ВИПЕРЕДЖЕННЯ І ЗАПІЗНЕННЯ

У даній статті розглядається задача складання розкладів із часами налагоджень сімейств (груп), де завдання в кожному сімействі виконуються разом. Налагодження, незалежне від послідовності, потрібне для виконання завдання з іншого сімейства. Ціль полягає в мінімізації сумарного випередження й запізнювання. Запропоновані евристичні алгоритми, дослідним шляхом вони оцінені в плані їх ефективності. Результати показують, що вони генерують розв'язки, достатньо близькі до оптимальних.

В данной статье рассматривается задача составления расписаний с временами наладок семейств (групп), где задания в каждом семействе выполняются вместе. Наладка, независимая от последовательности, требуется для выполнения задания из другого семейства. Цель состоит в минимизации суммарного опережения и запаздывания. Предложены эвристические алгоритмы, опытным путем они оценены в плане их эффективности. Результаты показывают, что они генерируют решения, достаточно близкие к оптимальным.

In this article the problem of scheduling with setup times of families (groups) is considered where the tasks in each family are performed together. The setup is sequence-independent and required for the task from another family. The goal is to minimize the total earliness and tardiness. Proposed heuristic algorithms are empirically evaluated in terms of their effectiveness. The results show that they generate solutions that are sufficiently close to optimal.

Вступ. Задачі складання розкладів груп, де всі роботи одного і того самого сімейства повинні бути призначені разом, привернули увагу багаточисельних дослідників через часте виникнення в реальному житті. Багато виготовлювачів реалізували концепцію групової технології (ГТ), щоб зменшити витрати на налагодження, час підготовки, витрати на складування незавершених робіт і вартість вантажно-розвантажувальних робіт. Групова технологія полягає в розділенні усієї множини робіт у декілька підмножин, названих сімействами, де сімейство – підмножина робіт, які мають схожі вимоги з погляду обладнання й налагоджень [1]. Так як різні сімейства робіт вимагають різного набору інструментів, налагодження часто необхідне, коли повинна бути виконана робота з іншого сімейства. Так як роботи призначаються на сімейства, засновані на вимогах до набору інструментів і налагодженню, звичайно є незначне або невелике налагодження при переході від однієї деталі до іншої в межах одного й того ж сімейства. Оскільки є велике переналагодження між сімействами, то існує перевага в обробці деталей, що належать тому самому сімейству як одна група. Це – ключова відмінність між задачею складання розкладів груп і традиційною задачею складання розкладів. Щоб розв'язати цю проблему більших великих налагоджень між сімействами, повинні бути знайдені оптимальна послідовність робіт у межах кожного сімейства, а також оптимальна послідовність сімейств, щоб оптимізувати заданий критерій ефективності.

У теперешній час значна увага в літературі приділяється складанню розкладів відносно директивних строків. Одна з причин цього явища – зростаючий тиск конкуренції на міжнародних ринках: фірми повинні запропонувати велику розмаїтість різних та індивідуальних виробів, у той час як клієнти очікують, що замовлені товари будуть поставлені вчасно. Принцип виробництва «точно в строк» установлює, що необхідна кількість товарів повинна бути вироблена або поставлена точно в заданий час. Виконання роботи з випередженням приводить до витрат на складування, у той час як запізнення робіт – до штрафів і, в остаточному підсумку, втраті доброзичливості клієнтів і репутації фірми.

Огляди недавніх досліджень у розв'язанні задач складання розкладів з налагодженнями сімейств Монма і Поттсом (1989), Поттсом і Ван Вассенговом (1992), Поттсом і Ковальовим (2000) і Вебстером і Бейкером (1995) показують, що більша частина досліджень в задачах складання розкладів груп присвячена сумарному (зваженому) часу проходження (знаходження в потоці) і максимальному запізнюванню. Наприклад, Чень, Гордон і Ковальов (1996) представили поліноміальний алгоритм для критерію мінімізації максимальної вартості розкладу за умови мінімуму зваженого сумарного часу проходження. Проте, наскільки нам відомо, жодна з опублікованих робіт не присвячена задачі складання розкладів груп за критерієм мінімізації сумарного випередження і запізнювання на одному приладі (МВЗ). Тому в даній статті розглядається складання розкладів груп з часами налагодження сімейств, не залежними від послідовності, та пропонуються евристичні алгоритми для знаходження розкладів з мінімальним сумарним випередженням і запізненням, в основу яких покладено новий підхід до розв'язання задачі МВЗ, викладений у [2, 3]. У даній статті ми розширюємо цей підхід для більш загальної та більш практичної задачі, що з'являється особливо у виробничих задачах складання розкладів – задачі із часами налагодження. В статті представлено алгоритми локального пошуку, що широко використовуються як практичний підхід для розв'язання задач комбінаторної оптимізації.

Постановка задачі. Задача складання розкладів груп з часами налагодження сімейств на одному приладі може бути сформульована таким чином:

задано число сімейств, позначене f, і кількість завдань в кожному сімействі, представлена числом n_i для сімейства $i = 1, \ldots, f$, які повинні бути виконані без переривання на одному приладі. Тривалість виконання і директивний строк j-го завдання з сімейства g_i визначені як p_{ij} і d_{ij} відповідно. Крім того, якщо завдання слідує за попереднім завданням з того ж сімейства, то між ними немає часу налагодження; інакше потрібен час налагодження сімейства S_{g_i} перед наступним процесом виконання. Звернемо увагу, що S_{g_i} не залежить від позиції, займаної сімейством. Тобто, час налагодження сімейства залежить тільки від наступного сімейства. До того ж передбачається, що є налагодження до виконання першої роботи в будь-якій послідовності. Всі завдання доступні в момент часу нуль, простої приладу допускаються, а переривання завдань заборонено. Прилад виконує не більш ніж одне завдання одночасно, і не може виконувати ніяке завдання, поки виконується наладка. Всі завдання в кожному сімействі мають бути призначені разом. Іншими словами, у будь-якій допустимої послідовності повинні бути тільки f наладок. Загальна кількість завдань $n = n_1 + n_2 + \ldots + n_f$. Для будь-якого заданого розкладу випередження та запізнення завдання *j* можуть бути визначені виразами (2) і (3). Ціль полягає в тому, щоб знайти розклад, який мінімізує сумарне випередження і запізнення всіх завдань:

$$\sum_{j=1}^{n} \left(E_j + T_j \right), \tag{1}$$

де випередження і запізнення визначаються як, відповідно,

$$E_j = \max\left(0, d_j - C_j\right) = \left(d_j - C_j\right)^+, \tag{2}$$

$$T_j = \max\left(0, C_j - d_j\right) = \left(C_j - d_j\right)^+. \tag{3}$$

Сформульована задача узагальнює задачу МВЗ і відноситься до класу *NP*-повних.

Пропонуються два алгоритму розв'язання задачі: для випадку, коли простої приладу допускаються (алгоритм А1) і для випадку, коли вони заборонені (алгоритм А2).

Умовні позначення:

$t_{g_i}^{нэ}$ – ефективний момент початку виконання сімейства g_i, що визначається алгоритмом розв'язання задачі МВЗ [2, 3];

C_{g_i} – момент закінчення виконання сімейства g_i;

T_{g_i} – сумарна тривалість виконання завдань сімейства g_i із налагодженням:

$$T_{g_i} = S_{g_i} + \sum_{j \in g_i} p_{ij};$$

U – множина всіх сімейств $\left\{ g_k \middle| k = 1, \ldots, f \right\}$;

V – множина всіх сімейств, призначених на виконання;

Алгоритми А1 і А2 засновано на використанні алгоритму розв'язання задачі МВЗ [2, 3]. У результаті в межах кожного сімейства g_i будується послідовність виконання завдань, оптимальна за вказаним функціоналом.

Алгоритм розв'язання задачі МВЗ. Алгоритм базується на наступних твердженнях.

Нехай j і j_i позначають номер завдання відповідно до індексації, заданої функціоналом; $j_{[g]}$ – номер завдання, що стоїть в деякій розглянутій послідовності на позиції g. Позначимо $r_j = \max(0, d_j - C_j)$ – резерв завдання j. Очевидно, $r_j = E_j$.

Справедливі наступні леми [4]:

Лема 1. Послідовність, відсортована за неспаданням значень тривалості виконання, яка не має робіт, що випереджають, є оптимальною за критерієм МВЗ.

Лема 2. Послідовність, відсортована за незростанням значень тривалості виконання, яка не має робіт, що запізнюються, є оптимальною за критерієм МВЗ.

Алгоритм складається з трьох блоків. У *Блоці* 1 розв'язується задача мінімізації сумарного запізнення при виконанні незалежних завдань одним приладом (МСЗ) [3]. Алгоритм розв'язання задачі МСЗ побудований на перестановках і полягає в оптимальному використанні завданнями, що запізнюються, резервів завдань, що не запізнюються. Використовуються наступні типи перестановок:

1) Вільні перестановки.

Визначення 1. Вільною перестановкою завдання називається перестановка його на позицію з максимальним номером, на якій воно залишається не запізненим, при умові, що на інтервалі перестановки є запізнені завдання.

2) Перестановки (EFSR – extraction and forward-shifted reinsertion – извлечение и повторная вставка со сдвигом вперед). Перестановкою називається процедура переносу завдання $j_{[g]}$ на позицію $k \left(k > g\right)$ і, одночасно, завдань, що займають позиції $g+1, g+2, \ldots, k-1, k$, на позиції $g, g+1, \ldots, k-2, k-1$, відповідно.

3) Вбудовування (EBSR – extraction and backward-shifted reinsertion – извлечение и повторная вставка со сдвигом назад). Вбудовуванням називається процедура переносу завдання $j_{[g]}$ на позицію $p \left(g > p\right)$ і, одночасно, завдань $p, p+1, \ldots, g-2, g-1$ на позиції $p+1, p+2, \ldots, g-1, g$, відповідно.

У результаті розв'язання задачі МСЗ у Блоці 1 реалізується зменшення сумарного запізнення за рахунок зменшення сумарного випередження. Якщо в цій послідовності резерви відсутні, то ця послідовність оптимальна за критерієм МВЗ.

У *Блоці 2* здійснюється зменшення значення сумарного випередження і запізнення за допомогою послідовного збільшення моментів початку виконання завдань.

Позначимо $r_{\min} = \min\{r_j\}$; N_r – число завдань з резервами ($r_j > 0$); $N_з$ – число завдань, що запізнюються (в їх число включаються завдання з нульовим резервом). Встановлюємо $t_{g_i}^{нэ} = 0$.

Твердження 1 [5] (використання резервів завдань, що випереджають). Якщо в послідовності σ виконується $N_r > N_з$, то при збільшенні початку виконання завдань на величину, рівну $r_{\min}$, значення функціонала випередження/запізнення зменшується на величину $(N_r - N_з)r_{\min}$.

Розглядається послідовність, що отримана в результаті розв'язання задачі МСЗ [3]. Якщо в отриманій послідовності резерви відсутні, то ця послідовність оптимальна за критерієм МВЗ. На кожній ітерації за умови $N_r \geq N_з$ збільшуються моменти початку виконання завдань у поточній послідовності $t_{g_i}^{нэ}$ на $r_{\min}$. Такі процедури виконуються, поки у послідовності, отриманій після оптимізації за критерієм МСЗ, не виконається умова $N_r < N_з$. Отриману послідовність позначаємо σ^R і перевіряємо умови твердження 2.

Твердження 2 [5] (ознака оптимальності послідовності σ^R). Якщо в послідовності σ^R резерви відсутні, то вона оптимальна за критерієм МВЗ.

У *Блоці 3* здійснюється оптимізація послідовності σ^R. Виконуються $n_i - 1$ ітерацій, на кожній ітерації визначається найбільш ефективна позиція вбудовування чергового завдання на більш ранню позицію, якщо це призводить до зменшення значення функціоналу. Нехай вже виконана $k-1$ ітерація. Для завдання $j_{[k]}$ перевіряємо кожну позицію $p = \overline{1, k-1}$ і шукаємо позицію p, на якій значення функціоналу зменшується на максимальну величину:

$$\sum_{i=p}^{k} \left| d_{j_{[i]}} - C_{j_{[i]}} \right| - \left| \left(d_{j_{[k]}} - C_{j_{[k]}} \right) - \sum_{i=p}^{k-1} l_{j_{[i]}} \right| - \sum_{i=p}^{k-1} \left| d_{j_{[i]}} - C_{j_{[i]}} + l_{j_{[k]}} \right| \to \max \qquad (4)$$

Для цього встановлюємо завдання $j_{[k]}$ на позицію p, виконуємо вільні перестановки (див. визначення 1), визначаємо зменшення функціоналу за формулою (4). Встановлюємо завдання $j_{[k]}$ на ту позицію p, на якій досягається максимальне зменшення значення функціоналу. $k = k+1$. Якщо $k > n$, кінець роботи алгоритму. Інакше переходимо до наступної ітерації.

Алгоритм А1. Алгоритм складається з трьох етапів.

Етап 1. Визначення найбільш ефективних моментів початку виконання сімейств $t_{g_i}^{нэ}, i = 1, \ldots, f$, при яких досягаються близькі до оптимальних значення функціоналу для кожного сімейства. На цьому етапі використовується алгоритм розв'язання задачі МВЗ [2, 3], наведений вище.

Етап 2. Побудова послідовності сімейств σ^0. Сімейства упорядковуються за незростанням значень їх пріоритетів і призначаються на виконання в цій послідовності таким чином, щоб початок виконання $t_{g_i}^{н}, i = 1, \ldots, f$, був як можливо ближче до моментів $t_{g_i}^{нэ}, i = 1, \ldots, f$, отриманих на попередньому етапі. Таке призначення дозволить отримати ефективні близькі до оптимального розв'язки за критерієм МВЗ.

Етап 3. Зменшення значення функціоналу за допомогою послідовного виконання попарних перестановок сусідніх сімейств.

Розглянемо виконання кожного етапу більш детально.

Eman 1. На цьому етапі для кожного сімейства g_i за допомогою реалізації алгоритму розв'язання задачі МВЗ із урахуванням тривалості налагодження будується послідовність виконання робіт в межах сімейства g_i, близька до оптимальної за значенням функціонала, що розглядається, і визначається найбільш ефективний момент початку виконання сімейства $t_{g_i}^{нэ}$.

Eman 2. Пропонований алгоритм включає f ітерацій по числу сімейств, які необхідно призначити на виконання. На кожній ітерації визначається інтервал виконання чергового сімейства з послідовності $\sigma^{уп}$.

Упорядковуємо сімейства за неспаданням значень їх тривалостей T_{g_i} (послідовність $\sigma^{уп}$). Побудова послідовності σ^0, що визначає черговість призначення сімейств на виконання, починаємо з сімейства, що має мінімальну тривалість. Визначаємо інтервал його вставки: $t_{g_1}^{н} = t_{g_1}^{нэ}; C_{g_1} = t_{g_1}^{н} + T_{g_1}$. Переходимо до наступного сімейства завдань послідовності $\sigma^{уп}$.

Нехай вже виконані k ітерацій, в результаті яких визначені позиції вставки в послідовність $g_1, g_2, \ldots, g_k$. Визначаємо інтервал вставки сімейства g_{k+1}: $t_{g_{k+1}}^{н} = t_{g_{k+1}}^{нэ}$; $C_{g_{k+1}} = t_{g_{k+1}}^{н} + T_{g_{k+1}}$. Якщо цей інтервал вільний, то сімейство g_{k+1} займе в послідовності σ^0 вказані позиції. Інакше шукаємо для вставки сімейства g_{k+1} новий інтервал, на якому $t_{g_{k+1}}^{н}$ максимально наближений до $t_{g_{k+1}}^{нэ}$.

У процесі побудови послідовності σ^0 можливі ситуації, коли сімейство g_{k+1} перетинається з іншим сімейством g_l за виконанням. Розглянемо такі випадки:

1) $t^н_{g_l} < t^н_{g_{k+1}} < C_{g_l}$. У цьому випадку сімейство g_{k+1} зсувається на більш пізні позиції: $t^н_{g_{k+1}} = C_{g_l}$; $C_{g_{k+1}} = t^н_{g_{k+1}} + T_{g_{k+1}}$. Якщо, в свою чергу, існує сімейство g_r, для якого $C_{g_{k+1}} > t^н_{g_r}$, сімейство g_r переміщується на більш ранні позиції після g_l, а сімейство g_{k+1} вставляється після нього: $t^н_{g_r} = C_{g_l}$; $t^н_{g_{k+1}} = C_{g_r} = t^н_{g_r} + T_{g_r}$; $C_{g_{k+1}} = t^н_{g_{k+1}} + T_{g_{k+1}}$.

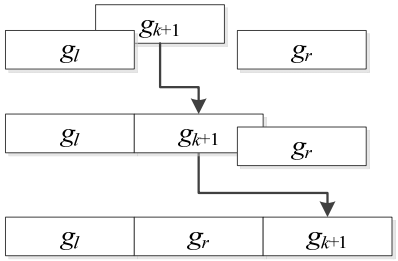

Рисунок 1 – Випадок перетину g_{k+1} з кінцем g_l

2) $t^н_{g_l} < C_{g_{k+1}} < C_{g_l}$. У цьому випадку сімейство g_{k+1} зсувається на більш ранні позиції: $C_{g_{k+1}} = t^н_{g_l}$; $t^н_{g_{k+1}} = C_{g_{k+1}} - T_{g_{k+1}}$. Якщо виявиться, що в результаті вставки сімейство g_{k+1} перетинається з сімейством g_r, і виконується $C_{g_r} > t^н_{g_{k+1}}$, то сімейство g_r переміщується на більш пізні позиції, а сімейство g_{k+1} вставляється перед ним: $C_{g_r} = t^н_{g_l}$; $t^н_{g_r} = C_{g_r} - T_{g_r}$; $C_{g_{k+1}} = t^н_{g_r}$; $t^н_{g_{k+1}} = C_{g_{k+1}} - T_{g_{k+1}}$.

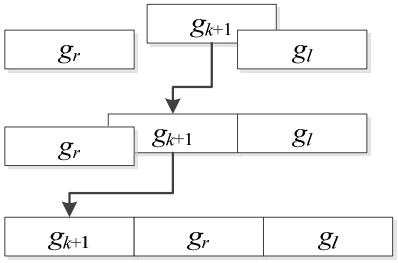

Рисунок 2 – Випадок перетину g_{k+1} з початком g_l

Така процедура виконується до тих пір, поки сімейство g_{k+1} не буде вставлено в послідовність σ^0. Якщо $t^н_{g_{k+1}} \neq t^{нэ}_{g_{k+1}}$, по алгоритму розв'язання задачі МВЗ перевпорядковуємо завдання в межах сімейства.

Переходимо до наступного сімейства завдань послідовності $\sigma^{уп}$. Якщо вставлені всі сімейства, послідовність σ^0 побудовано. Якщо $t^н_{g_i} = t^{нэ}_{g_i}$, $i = 1, \ldots, f$, то послідовність σ^0 оптимальна.

Eman 3. Полягає в послідовному виконанні попарних перестановок сусідніх сімейств. Перестановка виконується, якщо вона призводить до зменшення значення функціоналу.

Кінець алгоритму А1.

Алгоритм А2. Алгоритм А2 використовується у випадках, коли простої приладу заборонені.

Eman 1. Побудова послідовності σ^0. Побудова послідовності сімейств завдань здійснюється з визначення сімейства, яке буде займати останню позицію. З цією метою для кожного сімейства g_k, застосовуючи алгоритм розв'язання задачі МВЗ із урахуванням тривалості налагодження, визначаємо значення функціоналу задачі МВЗ для випадку, якщо це сімейство займе останню позицію в послідовності σ^0. Момент початку виконання сімейства визначається наступним чином:

$$t^н_{g_k} = \sum_{i=1}^{f} T_{g_i} - T_{g_k}.$$

Для призначення на останню позицію в σ^0 обираємо сімейство g_k з найменшим значенням функціоналу МВЗ з усіх $k = 1, \ldots, f$. Включаємо це сімейство в множину V призначених сімейств. Аналогічно визначаємо сімейство завдань g_l з множини $U \setminus V$, яке призначається на передостанню позицію в σ^0:

$$t^н_{g_l} = \sum_{i \in U} T_{g_i} - T_{g_k} - T_{g_l}.$$

У загальному випадку момент початку виконання чергового сімейства g_r визначається як:

$$t^н_{g_r} = \sum_{i \in U} T_{g_i} - \sum_{i \in V} T_{g_i} - T_{g_r}.$$

Така процедура виконується до тих пір, поки не будуть призначені всі сімейства і не буде побудована послідовність σ^0.

Eman 2. Послідовне виконання попарних перестановок сусідніх сімейств. Перестановки виконуються, якщо вони призводять до зменшення значення функціоналу. В алгоритмі А2 можливе використання також інших типів перестановок для отримання більш точного розв'язку, наприклад, NAPI, EBSR и EFSR [3].

Ілюстративний приклад. Алгоритм А1. В таблиці 1 наведено вхідні дані для 5 сімейств. В таблиці 2 – результати розрахунків, де $C^{э}_{g_i} = t^{нэ}_{g_i} + T_{g_i}$; σ_{g_i} і F_{g_i} – послідовність завдань та функціонал МВЗ у сімействі g_i після розв'язання задачі МВЗ, $\sigma^{ф}_{g_i}$ і $F^{ф}_{g_i}$ – після вбудовування сімейств на Етапі 2.

Послідовність сімейств $\sigma^{уп}$ = 1, 2, 5, 3, 4. σ^0 = 1, 2, 3, 5, 4. Значення функціоналу F = 585.

Таблиця 1 – Вхідні дані для алгоритму А1

j	1	2	3	4	5	6	7
Сімейство $g_1, S_{g_1} = 30$							
p_{1j}	10	10	10	15	20		
d_{1j}	75	85	95	65	50		
Сімейство $g_2, S_{g_2} = 40$							
p_{2j}	10	15	20	25	30		
d_{2j}	100	120	150	150	200		
Сімейство $g_3, S_{g_3} = 190$							
p_{3j}	25	30	35	40	40	45	50
d_{3j}	500	450	550	400	600	600	500
Сімейство $g_4, S_{g_4} = 150$							
p_{4j}	100	200	300	400	500		
d_{4j}	1300	1500	1800	2200	2700		
Сімейство $g_5, S_{g_5} = 10$							
p_{5j}	30	50	70	80	100		
d_{5j}	1030	750	820	900	1000		

За алгоритмом А2 отримана послідовність сімейств σ^0 = 1, 2, 3, 5, 4. Значення функціоналу F = 985. Функціонал збільшився за рахунок того, що в послідовності, отриманої за алгоритмом А1, мав місце простой обладнання перед виконанням сімейства 4.

Таблиця 2 – Результати розрахунків

i	1	2	3	4	5
$t^{нэ}_{g_i}$	0	55	205	1050	690
$C^{э}_{g_i}$	95	195	660	2700	1030
σ_{g_i}	5, 4, 1, 2, 3	1, 2, 3, 4, 5	4, 2, 1, 7, 3, 5, 6	1, 2, 3, 4, 5	2, 3, 4, 5, 1
F_{g_i}	0	35	200	0	0
$t^{н}_{g_i}$	0	95	235	1050	690
C_{g_i}	95	235	690	2700	1030
$\sigma^{ф}_{g_i}$	5, 4, 1, 2, 3	1, 2, 3, 4, 5	2, 1, 4, 3, 6, 5, 7	1, 2, 3, 4, 5	2, 3, 4, 5, 1
$F^{ф}_{g_i}$	0	205	380	0	0

Обчислювальні результати. Алгоритм був закодований мовою C# у середовищі розробки Visual Studio 2010 під бібліотеку Microsoft .NET 4.0. Випробування проводилися на персональному комп'ютері із процесором Pentium CORE 2 Duo 2.0 ГГц із оперативною пам'яттю 2 Гбайта під управлінням ОС Microsoft Windows Vista. Досліджувалися задачі розмірності до 500 завдань у сімействі.

Для визначення ефективності алгоритмів були проведені дослідження залежності часу розв'язання задачі від кількості сімейств та середньої кількості завдань в сімействі.

Схема генерації даних, запропонована Фішером [6], використовувалася для тестування алгоритму на різних типах прикладів, тип задачі визначається комбінацією фактора запізнення T і діапазону директивних строків R. Для кожної задачі спочатку генеруються тривалості виконання і часи налагоджень із рівномірного розподілу із заданими границями. Потім обчислюються директивні строки з розподілу, рівномірного на $\left[p^*(1-T-R/2), p^*(1-T+R/2)\right]$, де p^* – сума всіх тривалостей. Значення T і R вибираються з множин {0.2, 0.4, 0.6, 0.8} і {0.2, 0.4, 0.6, 0.8, 1.0}, відповідно, даючи по 20 задач кожного типу.

Результати досліджень наведені нижче в табл. 3 і 4.

Висновки. Ми представили два алгоритма розв'язання задачі МВЗГ, засновані на методі розв'язання задачі МВЗ, викладеному в [2, 3]. Алгоритм А1 призначений для розв'язання задачі у випадку, коли простої приладу дозволяються; алгоритм А2 – при їх забороні. Алгоритм А1 дає розв'язки, близькі до оптимальних. Дослідження показали, що алгоритми А1 і А2 дозволяють за прийнятний час ефективно розв'язувати задачі великої розмірності. Результати, що наведені в табл. 3 і 4, дані для випадку T = 0,4; R = 0,8. Майбутні

дослідження будуть спрямовані на вивчення поводження алгоритму при різних значеннях R і T.

Таблиця 3 – Середній час розв'язання задач (сек.) для алгоритму A1

Завдань в сімействі	Кількість сімейств					
	2	4	8	10	20	30
50	0.08	0.16	0.33	0.42	0.86	1.30
100	0.21	0.42	0.86	1.08	2.20	3.35
150	0.36	0.73	1.49	1.87	3.83	5.81
200	0.53	1.08	2.20	2.77	5.66	8.59
250	0.72	1.46	2.98	3.75	7.67	11.6
300	0.92	1.87	3.82	4.81	9.82	14.9
350	1.13	2.31	4.71	5.93	12.1	18.4
400	1.36	2.77	5.65	7.11	14.5	22.1
450	1.59	3.25	6.64	8.35	17.1	25.9
500	1.84	3.75	7.66	9.64	19.7	29.9

Таблиця 4 – Середній час розв'язання задач (сек.) для алгоритму A2

Завдань в сімействі	Кількість сімейств					
	2	4	8	10	20	30
50	0.16	0.65	2.62	4.11	16.7	37.8
100	0.41	1.66	6.72	10.6	42.8	97.1
150	0.71	2.88	11.7	18.3	74.3	168.5
200	1.05	4.25	17.3	27.1	109.8	249.2
250	1.42	5.76	23.4	36.7	148.8	337.5
300	1.82	7.38	30.0	47.0	190.7	432.5
350	2.25	9.11	36.9	58.0	235.1	533.4
400	2.69	10.9	44.3	69.5	282.0	639.6
450	3.16	12.8	52.0	81.6	330.9	750.7
500	3.65	14.8	60.0	94.2	381.9	866.3

Список літератури: **1.** *Gupta J. N. D.* Single Machine Group Scheduling with Setups to Minimize Total Tardiness / *J. N. D. Gupta, S. Chantaravarapan* // International Journal of Production Research, Volume 46 (6). – 2008. – P. 1707–1722. **2.** *Павлов О. А.* Дослідження властивостей та розв'язання задачі «Мінімізація сумарного штрафу як за випередження, так і за запізнення відносно директивних строків при виконанні незалежних завдань одним приладом» / *О. А. Павлов, О. Б. Місюра, О. В. Мельников* // Вісник НТУУ "КПІ". Інформатика, управління та обчислювальна техніка. – К. : "ВЕК+", 2008. – № 48. – С. 3–6. **3.** *Згуровский М. З.* Принятие решений в сетевых системах с ограниченными ресурсами / *М. З. Згуровский, А. А. Павлов* // Монография. – К. : Наукова думка. – 2010. – 573 с. **4.** *Ow P. S.* The Single Machine Early/Tardy Problem / *Ow P. S., Morton T. E.* / Management Science. – Vol. 35. – № 2. – 1989. – P. 177–191. **5.** *Згуровский М. З.* ПДС-алгоритмы и труднорешаемые задачи комбинаторной оптимизации / *М. З. Згуровский, А. А. Павлов, Е. Б. Мисюра* // Системні дослідження та інформаційні технології. – 2009. – №.4. – С. 14–31. **6.** *Fisher M. L.* A dual algorithm for the one-machine scheduling problem / Math. Programming, – №11. – 1976. – P. 229–251.

Надійшла до редколегії 19.08.2011