

## ДОСЛІДЖЕННЯ МЕТОДІВ МІНІМІЗАЦІЇ СУМАРНОГО ЧАСУ ЗАПІЗНЮВАННЯ РОБІТ ІЗ ДИРЕКТИВНИМИ СТРОКАМИ З ВИКОРИСТАННЯМ ПРАВИЛ ДОМІНУВАННЯ НА ОДИНОЧНОМУ ОБЧИСЛЮВАЛЬНОМУ РЕСУРСІ

Розглянуто та проаналізовано алгоритми вирішення задачі мінімізації сумарного часу запізнювання робіт з директивними строками виконання та вагами на обчислювальному ресурсі. Досліджено евристичні алгоритми вирішення задачі на основі рангового підходу, а також їх модифікації з використанням правил домінування. Представлені результати обчислювальних експериментів щодо обґрунтування ефективності досліджуваних алгоритмів із правилами домінування на основі відносної похибки та часу виконання для різної кількості завдань вхідної черги. На основі проведених комп'ютерних експериментів для різних діапазонів параметрів запізнювання та директивних строків дані практичні рекомендації із застосування запропонованих алгоритмів.

**Ключові слова:** гамільтонів шлях, граф, оптимальний розклад, час запізнювання, директивний строк, вага, правило домінування, відносні похибки.

### Вступ

Сучасні інформаційно-комунікаційні системи визначаються наявністю багатої кількості різних типів ресурсів – інформаційних обчислювальних, тощо, їх розподіленістю, управління якими потребує підвищення ефективності та вдосконалення систем управління розподіленим обробленням, зокрема, на локальних ресурсах. Одним із завдань є розробка ефективних алгоритмів планування розкладу та оптимізації виконання робіт за обраними критеріями в системах реального часу.

Мета даної роботи – дослідження евристичних алгоритмів побудови оптимального розкладу виконання завдань з директивними строками на одиночному обчислювальному ресурсі за критерієм мінімізації сумарного часу запізнювання на основі рангового підходу [1, 2].

### Постановка задачі

У сучасних розподілених системах, що використовують у якості ресурсів кластери, процесорні елементи, робочі станції (наприклад, Грід-системах [1]), тощо, завдання, які поступили для вирішення на якийсь ресурс, мають різні вартості їх реалізації та директивні строки виконання. Часові відхилення від директивних строків приводять до подорожчання процесів проведення обчислень (штрафів) в обчислювальній системі та порушення умов виконання завдань.

Тому представляється актуальним розроблення ефективних алгоритмів, які повинні відповідати вимогам щодо оперативності (часової складності) їх реалізації та адекватності з точки зору мінімізації похибки відносно точного рішення.

Однією з таких задач є побудова оптимального розкладу виконання завдань на одиночному ресурсі (пристрої) обчислювальної системи з метою мінімізації сумарного часу запізнювання виконання усіх завдань, що

поступають безперервно, одночасно та згідно з обраною стратегією їх вирішення на ресурсі.

Постановка такої задачі для визначення оптимальної послідовності виконання завдань (розкладу) з мінімальним сумарним часом запізнювання формулюється наступним чином. Припустимо, що на одиночний ресурс надходить множина незалежних робіт  $J = \{j_1, j_2, \dots, j_n\}$ , кожна з яких безперервно виконуватиметься на ньому. При цьому відомі тривалість виконання кожного завдання  $l_j$  та директивний строк його виконання  $D_j$ . Необхідно визначити такий порядок (послідовність) виконання усіх завдань, які одночасно поступають на ресурс (пристрій), який мінімізує сумарний час запізнювання усіх завдань вхідної черги.

Формалізація наведеної постановки у вигляді математичної моделі виглядає так: необхідно побудувати розклад виконання завдань, який мінімізує функцію

$$f = \sum_{j=1}^n \max(0, C_j - D_j) \rightarrow \min, \quad (1)$$

де  $C_j$  – фактичний момент завершення завдання  $j$ ;  $D_j$  – директивний строк виконання завдання;  $n$  – кількість завдань вхідного потоку.

У випадку, коли роботи характеризуються також ваговою характеристикою  $w_j$ , функція (1) має вигляд

$$f = \sum_{j=1}^n \max(0, C_j - D_j) w_j \rightarrow \min, \quad (2)$$

Поставлена задача є відомою задачею, основні результати та методи рішення цієї задачі приведені в роботах [3 – 8]. Згідно з ними, ця задача відноситься до NP-складних задач. Недоліком відомих методів є те, що вирішення цієї задачі в реальному часі ускладнене внаслідок експоненціального зростання кількості варіантів, що аналізуються, зі зростанням кількості робіт та, відповідно, й часової складності алгоритмів.

Відмітимо, що разом із часовою складністю алгоритмів у якості метрик ефективності використовуються такі характеристики, як кількість

робіт вхідного потоку, для якого за мінімальний час побудована оптимальна послідовність виконання робіт.

### Формалізація задачі мінімізації сумарного часу запізнювання завдань з директивними строками на обчислювальному ресурсі

Для формалізації моделі та методу вирішення приведеної задачі розглянемо граф  $G$ , в якому кожній вершині відповідає завдання, а ребро  $(i, j)$  характеризується ваговою характеристикою  $T_j = \max(0, C_j - D_j)$  (рис. 1).

Тоді задачу мінімізації функціонала (1) можна розглядати як задачу визначення такого обходу всіх вершин графа  $G$ , для якого довжина побудованого при цьому шляху була мінімальною.

Така постановка приводить до задачі визначення найкоротшого гамільтонового шляху в графі  $G$ .

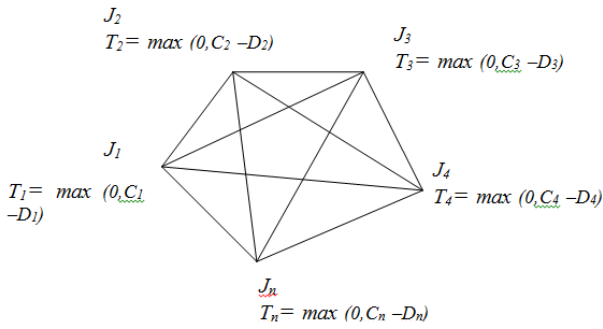


Рис. 1. Граф  $G$

Особливістю даної задачі є той факт, що ваги ребер графа, що витікають із довільної вершини  $J_i$ , не є постійними величинами, а залежать від того, за яким порядком пройдені ті вершини, що ведуть до вершини  $J_i$ , оскільки від цього залежить величина  $C_j$ , яка визначається сумою тривалостей робіт, які

попереджують виконання роботи  $J_i$ , тобто значенням [2]

$$C_j = \sum_{l_i \in \mu_{j_i}} l_i \quad (3)$$

В якості прикладу знаходження найкоротшого гамільтонового шляху в довільному графі розглянемо чотирихвершинний граф з такими характеристиками робіт  $J_j(C_j, D_j)$ :  $J_1(4,7)$ ,  $J_2(5,6)$ ,  $J_3(3,7)$ ,  $J_4(3,4)$ . Застосування алгоритму [2] дозволяє отримати такий гамільтонів шлях  $\{J_4, J_3, J_1, J_2\}$  (рис. 2).

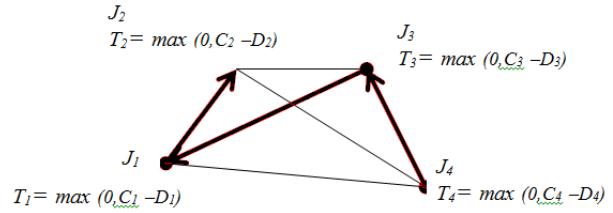


Рис. 2. Найкоротший гамільтонів шлях для довільного графу з чотирьох вершин

У випадку, коли мінімальна різниця  $C_i - D_j$  для будь-якого ярусу при визначенні гамільтонового шляху є однаковою для декількох шляхів-кандидатів, для вибору наступної вершини шляху з метою отримання локального оптимального рішення пропонується використовувати правила домінування, основний аналіз який наведений в роботах [6 – 8]. При цьому шлях обирається на основі вибору з множини шляхів-кандидатів такого, який задовольняє певному правилу домінування для послідовності вершин, які увійшли до гамільтонового шляху на певному ярусі. Таке включення дозволяє підвищити точність отриманого рішення.

Склад правил домінування, що використовуються для вирішення задачі мінімізації сумарного часу запізнювання для незваженого та зваженого випадків, наведений в табл. 1.

Таблиця 1

Склад правил домінування

Правило	Назва англійською	Назва українською	Формула визначення пріоритету для незваженого варіанту задачі	Формула визначення пріоритету для зваженого варіанту задачі	Тип правила
EDD	Earliest Due Date	Найбільш раннього директивного строку	$\min d_j$	$\max \left\{ \frac{w_j}{d_j} \right\}$	статичне
MDD	Modified Due Date	Модифікованого директивного строку	$\min \max \{ \rho_j, d_j - t \}$	$\min \left\{ \frac{\max \{ \rho_j, d_j - t \}}{w_j} \right\}$	динамічне
SPT	Shortest Processing Time	Найкоротшого процесорного часу	$\min p_j$	$\min \left\{ \frac{p_j}{w_j} \right\}$	статичне
PD	Processing Time / Due Date	Часу виконання / директивного строку	$\min \rho_j d_j$	$\max \left\{ \frac{w_j}{p_j d_j} \right\}$	статичне

COVE RT	Weighted Cost Over Time	Зваженої вартості понад час	–	$\max \left( \frac{w_j}{p_j} \max \left[ 0, 1 - \frac{\max \{d_j - t - p_j\}}{kp_j} \right] \right)$	динамічне
ATC	Apparent Tardiness Cost	Видимої вартості запізнювання	–	$\max \left( \frac{w_j}{p_j} \exp \left[ -\frac{\max \{d_j - t - p_j\}}{kp} \right] \right)$	динамічне

### Порядок проведення експерименту та аналіз результатів

Для експериментальної перевірки отриманого в роботі [2] аналітичного виразу для часової складності базового алгоритму розроблено програмний продукт, який реалізує алгоритм оптимізації за напрямком, алгоритм оптимізації за напрямком з використанням правил домінування (див. табл. 1), а також алгоритм повного перебору для оцінки відхилення наближених рішень від точного.

Для тестування розробленого програмного забезпечення з метою підвищення рівня адекватності отриманих результатів був використаний генератор стохастичних послідовностей для отримання пакетів завдань з різними характеристиками: параметр запізнювання (tardiness factor, TF) та діапазон директивних строків (range of due dates, RDD). Для детальнішого дослідження визначенні такі класифікаційні ознаки: для TF директивний строк слабкий при TF = 0,2,

середній – TF = 0,6, жорсткий – TF = 1,0; для RDD діапазон директивних строків вузький при RDD = 0,2, середній – RDD = 0,6, широкий – RDD = 1,0 (див. табл. 2). Для зваженого варіанта вирішення задачі використовується рівномірно розподілені ваги  $w_j$  робіт в діапазоні 1 – 10.

Для оцінки чутливості досліджуваних алгоритмів відносно використання правил домінування до параметрів вхідних робіт було проведено наступний експеримент. Були згенеровані пакети з 50 завдань (instances) (150 робіт у кожному) для комбінацій близькості директивних строків до процесорного часу – слабкі, середні, жорсткі; ширини їхнього діапазону – вузький, середній, широкий; діапазону процесорного часу (тривалість виконання) – 1 – 10, 1 – 100, 1 – 1000 відповідно. В табл. 2 наведено результати обчислень згенерованих вхідних пакетів методами оптимізації за напрямком та методом оптимізації з правилами домінування EDD та MDD для незваженого випадку визначення сумарного часу запізнювання.

Таблиця 2

Порівняння чутливості досліджуваних алгоритмів до параметрів вхідних робіт

Діапазон процесорного часу 1 – 10						
Директивні строкі, TF	Діапазон директивних строків, RDD	Середнє сумарне запізнювання за методом оптимізації, у.о.	Середнє сумарне запізнювання за методом оптимізації з EDD, у.о.	Покращення результату методу оптимізації з EDD відносно методу оптимізації, %	Середнє сумарне запізнювання за методом оптимізації з MDD, у.о.	Покращення результату методу оптимізації з MDD відносно методу оптимізації, %
слабкі	вузький	1941,56	558,46	71,2	541,00	72,1
	середній	3754,98	0	100,0	0	100,0
	широкий	6075,72	0	100,0	0	100,0
середні	вузький	16522,18	13590,90	17,7	13276,56	19,6
	середній	17604,58	9674,46	45,0	9463,08	46,2
	широкий	26502,92	14675,22	44,6	14356,12	45,8
жорсткі	вузький	45444,38	44857,82	1,3	44823,86	1,4
	середній	42844,9	41333,56	3,5	41334,44	3,5
	широкий	40377,64	37566,26	7,0	37641,76	6,8
Діапазон процесорного часу 1 – 100						
слабкі	вузький	18541,36	5236,22	71,8	5062,18	72,7
	середній	33763,68	0	100,0	0	100,0
	широкий	53761,08	0	100,0	0	100,0
середні	вузький	155526,7	128878,80	17,1	125494,00	19,3
	середній	169486,24	94403,34	44,3	92288,20	45,5
	широкий	236449,28	131938,50	44,2	131390,60	44,4

жорсткі	вузький	434547,4	432561,70	0,5	432202,50	0,5
	середній	401395,56	390861,70	2,6	389794,50	2,9
	широкий	389319,38	364650,60	6,3	363931,90	6,5

Діапазон процесорного часу 1 – 1000						
слабкі	вузький	181793,04	51703,6	71,6	49428,6	72,8
	середній	325464,38	0	100,0	0	100,0
	широкий	536319,7	0	100,0	0	100,0
середні	вузький	1527581,26	1290515,0	15,5	1251611,0	18,1
	середній	1721184,62	965727,7	43,9	938620,5	45,5
	широкий	2158866,06	1203855,0	44,2	1218920,0	43,5
жорсткі	вузький	4330889,12	4323351,0	0,2	4319343,0	0,3
	середній	4013879,44	3903451,0	2,8	3893413,0	3,0
	широкий	3890956,78	3647695,0	6,3	3652851,0	6,1

Як видно з табл. 2, евристичний алгоритм, який досліджується, дає кращий результат при досить великій різниці директивних строків та процесорного часу робіт та при широкому діапазоні директивних строків. Діапазон процесорного часу робіт на ефективність алгоритму та використання правил домінування суттєво не впливає.

Аналізуючи результати, наведені в табл. 2, можна зробити наступні висновки щодо практичного застосування базового та модифікованих алгоритмів:

для всіх діапазонів процесорного часу виконання завдань у разі слабого значення TF використання правил домінування дозволяє зменшити сумарний час запізнювання у найгіршому випадку на 71,2%;

для всіх діапазонів процесорного часу виконання завдань у разі середнього значення TF використання правил домінування дозволяє зменшити сумарний час запізнювання у найгіршому випадку на 15,5%;

для всіх діапазонів процесорного часу виконання завдань у разі жорсткого значення TF використання правил домінування дозволяє зменшити сумарний час запізнювання у найкращому випадку на 7,0%;

Отже, загальний висновок може бути сформульований таким чином: при жорстких строках параметру запізнювання та вузькому діапазоні директивних строків виконання завдань використання правил домінування майже не дає переваг у порівнянні з базовим алгоритмом; при слабких строках параметру запізнювання та широкому діапазоні директивних строків виконання завдань використання правил домінування дозволяє отримати результати з нульовим часом запізнювання, який взагалі не може бути покращений.

Для оцінки часової складності досліджуваного алгоритму було згенеровано пакети від 20 до 220 робіт з 75 завдань у кожному, оцінений час роботи алгоритму (рис. 3).

На рис. 4 наведено порівняння залежності часу сумарного запізнювання для алгоритму оптимізації за напрямком та його модифікацій з використанням правил домінування EDD та MDD для пакетів від 20 до 220 робіт з 75 завдань. Як бачимо, ефективність цих правил практично не відрізняється.

На рис. 5 наведено порівняння залежності часу зваженого сумарного запізнювання для алгоритму оптимізації за напрямком та його модифікацій з використанням правил домінування EDD, COVERT та ATC для пакетів від 20 до 150 робіт з 75 завдань. Як бачимо, ефективність правила ATC майже не відрізняється від ефективності правила EDD, у той час як правило COVERT дає результат, близький до алгоритму оптимізації без використання правил домінування.

На рис. 6 наведено порівняння залежності часу сумарного запізнювання для алгоритму повного перебору та оптимізації за напрямком з використанням правил домінування EDD та ATC для пакетів від 3 до 11 робіт з 75 завдань. Така невелика кількість робіт зумовлена експоненційною часовою складністю алгоритму повного перебору [1].

На рис. 7 наведено залежність відносного покращення сумарного часу запізнювання з використанням правил домінування у порівнянні з базовим алгоритмом, яка свідчить про те, що зі збільшенням кількості робіт число локальних точних рішень збільшується.

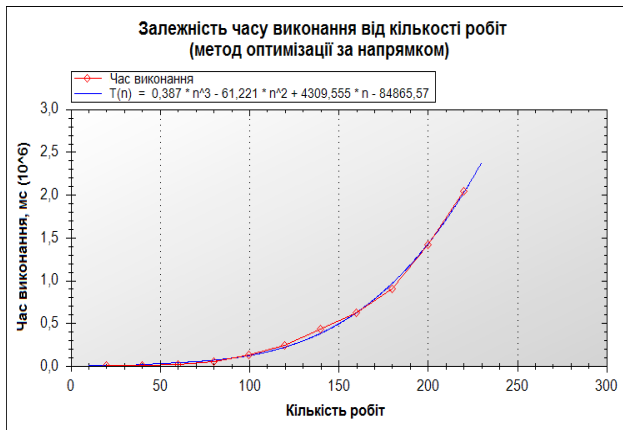


Рис. 3. Залежність часу виконання від кількості робіт для базового алгоритму

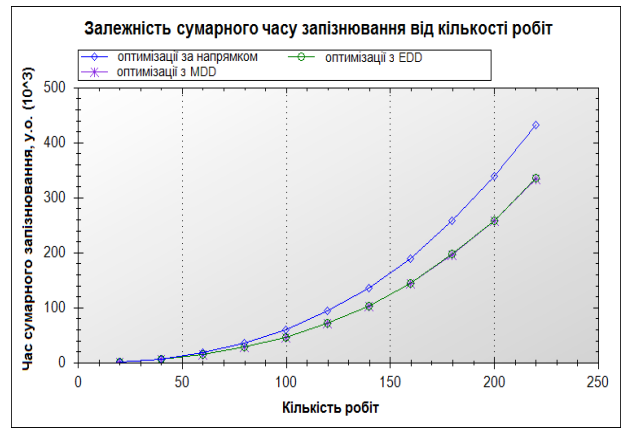


Рис. 4. Залежність сумарного часу запізнювання від кількості робіт для алгоритмів з правилами домінування

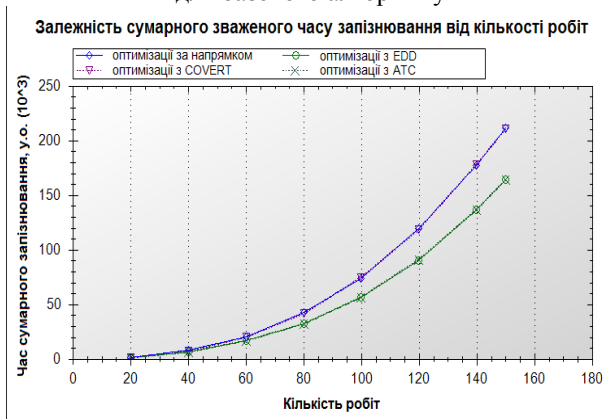


Рис. 5. Залежність сумарного зваженого часу запізнювання від кількості робіт для алгоритмів з правилами домінування

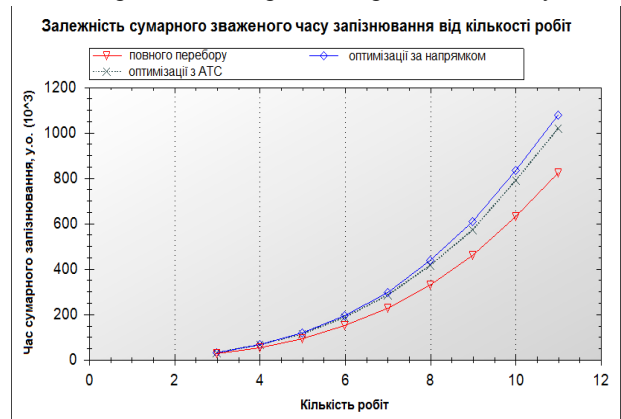


Рис. 6. Залежність сумарного зваженого часу запізнювання від кількості робіт для алгоритмів з правилами домінування та для алгоритму повного перебору

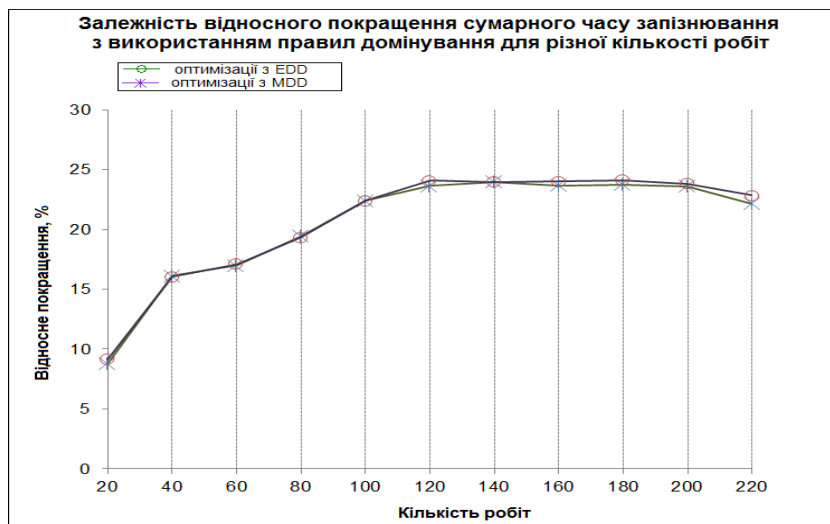


Рис. 7. Залежність відносного покращення сумарного часу запізнювання з використанням правил домінування від кількості робіт, %

Експеримент проводився на персональному комп'ютері з двох'ядерним процесором з тактовою частотою 3,2 ГГц та з оперативною пам'яттю 8 ГБ під управлінням ОС Windows 7. Для забезпечення точності вимірювання часу виконання обчислень експерименти проводилися на одному ядрі процесора.

## Висновки

1. Проведений аналіз існуючих методів вирішення задачі мінімізації сумарного часу запізнювання завдань на одиночному обчислювальному пристрої (ресурсі), який показав, що розроблення ефективних алгоритмів з малою

часовою складністю та точністю для використання в інформаційних системах реального часу є актуальним завданням.

2. Проведені дослідження евристичного алгоритму, запропонованого в роботі [2], на основі програмної реалізації алгоритму та тестових прикладів, які практично підтвердили теоретичну часову складність  $O(n^3)$ .

3. Для підвищення точності базового алгоритму запропоновано його модифікації на основі використання правил домінування, які використовуються для незваженого та незваженого варіантів задачі мінімізації сумарного запізнювання. При цьому порядок часової складності роботи алгоритмів не змінюється, що доводить, що вони є стійкими до комбінування з іншими алгоритмами, що підвищують точність отриманих результатів.

4. Отримана відносна похибка для евристичних алгоритмів свідчить про можливість їх використання при вирішенні задачі мінімізації сумарного часу запізнювання на одиночному обчислювальному пристрої (ресурсі).

5. Запропоновані модифікації базового евристичного алгоритму на основі правил домінування, використання яких дозволяє зменшити значення критерію (1), а часова складність алгоритмів при цьому не змінюється.

6. Розроблено рекомендації щодо використання запропонованих алгоритмів для різних діапазонів директивних строків та процесорного часу виконання вхідних робіт. Це дає змогу визначити сфери найбільш ефективного застосування запропонованих алгоритмів для різних діапазонів змін параметрів запізнювання та директивних строків виконання завдань на вузлах розподілених обчислювальних систем.

## Список літератури

1. Пономаренко В.С. *Методы и модели планирования ресурсов в Grid-системах: монография* / В.С. Пономаренко, С.В. Листровой, С.В. Минухин, С.В. Знахур. – Харьков: ИД «ИНЖЭК», 2008. – 408 с.

2. Минухин С.В. *Метод мінімізації часу виконання завдань з директивними строками на некластеризованому ресурсі обчислювальної системи* //

*Інформаційно-керуючі системи на залізничному транспорті*. – 2009. – №3. – С. 47 – 53.

3. Koulamas C. *The total tardiness problem: Review and extensions* // *Operations Research*. – 1994. – v.42. – P. 1025–1041.

4. Koulamas C. *The single-machine total tardiness scheduling problem: Review and extensions* // *European Journal of Operational Research*. – 2010. – v. 202. – No.1. – P. 1–7.

5. Sena Tapan. *Static scheduling research to minimize weighted and unweighted tardiness: A state-of-the-art survey* / Tapan Sena, Joanne M. Suleka, Parthasarati Dileepan // *Int. J. Production Economics*. – 2003. – v. 83. – No. 1. – P. 1–12.

6. Osman Ibrahim H. *Hybrid of the weighted minimum slack and shortest processing time dispatching rules for the total weighted tardiness single machine scheduling problem with availability constraints* / Ibrahim H. Osman, Hocine Belouadah, Krzysztof Fleszar, Maysaa Saffar // *Multidisciplinary International Conference on Scheduling: Theory and Applications. MISTA*. – 2009, Dublin, Ireland.

7. Baker K.R. *A dynamic priority rule for scheduling against due-dates.* / K.R. Baker, J.W.M. Bertrand // *Journal of Operations Management*. – 1982. – №3 (1). – P. 37-42

8. Akturk M. S. *A New Dominance Rule for the Total Weighted Tardiness Problem* / M. S. Akturk, M. B. Yildirim. // *Production Planning & Control*. – 1999. – №10 (2). – P. 138-149.

Надійшла до редколегії 5.03.2013

**Рецензент:** д-р техн. наук, проф. В.С. Харченко, зав. кафедри комп'ютерних систем і мереж Національного аерокосмічного університету ім. М.Є. Жуковського «ХАІ», Харків, Україна.

**ИССЛЕДОВАНИЕ МЕТОДОВ МИНИМИЗАЦИИ СУММАРНОГО ВРЕМЕНИ ЗАПАЗДЫВАНИЯ РАБОТ С ДИРЕКТИВНЫМИ СРОКАМИ С ИСПОЛЬЗОВАНИЕМ ПРАВИЛ ДОМИНИРОВАНИЯ НА ОДИНОЧНОМ ВЫЧИСЛИТЕЛЬНОМ РЕСУРСЕ**

С.В. Минухин, Д.С. Ленко

*Рассмотрены и проанализированы алгоритмы решения задачи минимизации суммарного времени запаздывания работ с директивными сроками выполнения и весами на одиночном вычислительном ресурсе. Исследованы эвристические алгоритмы решения задачи на основе рангового подхода, а также их модификации с использованием правил доминирования. Представлены результаты вычислительных экспериментов по обоснованию эффективности исследуемых алгоритмов с правилами доминирования на основе относительной погрешности и времени выполнения для разного количества заданий входной очереди. На основе проведенных компьютерных экспериментов для разных диапазонов параметров запаздывания и директивных сроков даны практические рекомендации по применению предложенных алгоритмов.*

**Ключевые слова:** гамильтонов путь, граф, оптимальное расписание, время запаздывания, директивный срок, вес, правило доминирования, относительная ошибка.

**INVESTIGATION OF METHODS FOR MINIMIZING TOTAL TARDINESS OF JOBS WITH DUE DATES BY USING THE DOMINANCE RULES ON A SINGLE COMPUTATIONAL RESOURCE**

S.V. Minukhin, D.S. Lienko

*The algorithms for solution of the task of minimizing of the total tardiness of jobs with execution due dates and weights on a single computational resource are considered and analyzed. The heuristic algorithms for solution of the problem based on the rank approach are investigated. The results of computational experiments to prove effectiveness of considered algorithms with dominance rules based on relative error and execution time for various numbers of input jobs are given. Based on conducted computational experiments for various tardiness factor and due date ranges practical recommendations for considered algorithms usage are given.*

**Keywords:** Hamiltonian path, graph, optimal schedule, tardiness time, due date, weight, dominance rule, relative error.