

DOI: <https://doi.org/10.36910/6775-2524-0560-2026-62-30>

УДК 004.2

Пасічник Максим Юрійович, аспірант

<https://orcid.org/0009-0006-7885-3507>

Зайцев Володимир Григорович, д.т.н., професор

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна

ПРОГНОЗУВАННЯ НАВАНТАЖЕННЯ ДЛЯ ДИСПЕТЧЕРИЗАЦІЇ ЗАВДАНЬ У СИСТЕМАХ РЕАЛЬНОГО ЧАСУ

Пасічник М.Ю., Зайцев В.Г. Прогнозування навантаження для диспетчеризації завдань у системах реального часу. Системи реального часу поєднують періодичні та аперіодичні завдання, для яких критично важливим є своєчасне виконання. Періодичні завдання зазвичай мають наперед відомі параметри, тоді як аперіодичні запити виникають нерівномірно і можуть суттєво змінювати поточне навантаження на систему. Короточасні сплески аперіодичних запитів здатні спричинити зростання затримок і підвищувати ризик порушення дедлайнів. Водночас налаштування системи на найгірший випадок призводить до надмірної консервативності, знижує корисну пропускну здатність системи та погіршує якість обслуговування. У статті запропоновано підхід який використовує короткострокову оцінку навантаження для обережної адаптації параметрів обслуговування аперіодичних завдань. Зокрема, розглядається коригування бюджетів і періодів поповнення з урахуванням прогнозованого попиту. Щоб зберегти формальні гарантії своєчасності, оцінений попит доповнюється запасом на можливу похибку прогнозування, після чого виконується перевірка здійсненності розкладу для планування з фіксованими пріоритетами. Зміни параметрів допускаються лише за умови успішного проходження відповідних тестів, інакше система зберігає попередній, більш консервативний режим роботи. Розглянуто одноядерну систему з незалежними завданнями. Як показник оцінювання використано сумарний попит на процесорний час аперіодичних завдань у ковзному вікні $[t, t + \Delta t]$. Запропонований підхід дає змогу поєднати прогнозування навантаження з формальними методами аналізу розкладу, зменшити надмірну консервативність і підвищити ефективність обслуговування аперіодичних запитів без порушення вимог до періодичних завдань.

Ключові слова: системи реального часу, диспетчеризація завдань, прогнозування навантаження, оцінка навантаження, здійсненність розкладу, оптимізація ресурсів.

Pasichnyk M., Zaitsev V. Workload Prediction for Real-Time Task Scheduling. Real-time systems combine periodic and aperiodic tasks, for which timely execution is critical. Periodic tasks typically have predetermined parameters, whereas aperiodic requests occur irregularly and can significantly alter the current system workload. Short-term bursts of aperiodic requests can cause increased delays and heighten the risk of deadline misses. At the same time, configuring the system for the worst-case scenario leads to excessive conservatism, reduces effective system throughput, and degrades the quality of service. This paper proposes an approach that utilizes short-term workload estimation to carefully adapt the service parameters of aperiodic tasks. Specifically, it considers adjusting budgets and replenishment periods based on the forecasted demand. To maintain formal timeliness guarantees, the estimated demand is supplemented with a safety margin for potential forecasting errors, followed by a schedulability test for fixed-priority scheduling. Parameter changes are permitted only if the corresponding tests are successful; otherwise, the system retains its previous, more conservative operating mode. The study considers a single-core system with independent tasks. The total CPU time demand of aperiodic tasks within a sliding window $[t, t + \Delta t]$ is used as the evaluation metric. The proposed approach allows combining workload forecasting with formal schedulability analysis methods, reducing excessive conservatism, and improving the service efficiency of aperiodic requests without violating the requirements of periodic tasks.

Keywords: real-time systems, task scheduling, workload prediction, workload estimation, schedulability, resource optimization.

Постановка проблеми. Системи реального часу призначені для виконання дій у встановлені дедлайни, порушення яких розглядається як помилка. У типових застосуваннях співіснують періодичні завдання з відомими параметрами та аперіодичні завдання, інтенсивність яких змінюється у часі. Навіть за помірного середнього навантаження виникають короткі сплески надходжень: формується локальне накопичення завдань, зростають затримки, а в гіршому випадку можуть зриватися дедлайни періодичних завдань. Постійне налаштування системи на найгірший випадок зменшує корисну пропускну здатність і призводить до зайвих затримок у спокійних режимах. Проблема полягає в поєднанні двох вимог: використати інформацію про близьке майбутнє навантаження, аби тимчасово пом'якшувати обмеження тоді, коли це безпечно, і водночас не порушити гарантовану передбачуваність для періодичних завдань. Прогностична інформація не має замінювати перевірки придатності розкладання: зміни параметрів обслуговування допускаються лише після підтвердження передбачуваність для обраної політики диспетчеризації та з урахуванням додаткового запасу, що компенсує можливу похибку оцінювання. У межах цієї постановки виділяються такі складові:

1. Модель навантаження. Періодичні завдання дотримуються фіксованих правил виконання, а аперіодичні завдання отримують обслуговування в межах спеціально виділеного бюджету, щоб уникнути негативного впливу на дедлайн періодичних завдань. Характер навантаження може змінюватися: від стабільних ділянок до періодів із різкими сплесками інтенсивності.

2. Ризики без урахування майбутнього. За фіксованих параметрів системи реального часу існує компроміс: надто малий бюджет збільшує затримки аперіодичних завдань, надто великий — у фазах сплесків може відтіснити періодичні завдання та підвищувати ризик зриву дедлайнів.

3. Роль короткострокової оцінки. Необхідна оцінка найближчого навантаження в короткому часовому інтервалі, щоб вирішити, чи доцільно тимчасово змінити параметри обслуговування. Як оціночну величину доцільно використовувати або інтенсивність надходжень, або сумарний запит на процесорний час у цьому інтервалі.

4. Вимоги до безпеки рішень. Кожна оцінка супроводжується верхнім запасом, що компенсує похибку. Будь-які зміни застосовуються лише після успішної перевірки придатності для обраної політики; якщо перевірка не пройдена — зміни не вносяться. Частоту переналаштувань слід обмежити, щоб уникати коливань режиму.

5. Обмеження постановки. Розглядається одноядерна система з незалежними завданнями; за наявності блокувань їх слід або усунути, або врахувати консервативно як додаткові затримки. Накладні витрати на оцінювання та перевірки мають бути малими відносно інтервалів прийняття рішень; горизонт оцінювання — близький у часі.

6. Критерії доцільності. Підхід є доцільним, якщо вдається зменшити частку порушених дедлайнів у режимах зі сплесками, не погіршуючи дедлайни періодичних завдань у спокійних режимах, і знизити середню затримку завершення аперіодичних завдань без підвищення ризику зриву дедлайнів.

Аналіз останніх досліджень і публікацій. Проблема поєднання прогнозування та диспетчеризації завдань у системах реального часу перебуває в центрі уваги сучасних досліджень. У низці робіт розглядаються різні аспекти використання прогностичної інформації для поліпшення ефективності планування. У статті [1] зроблено акцент на підвищенні стійкості алгоритмів планування тривалості виконання із прогнозами. Автори вказують, що навіть за наявності якісних прогнозів необхідні додаткові механізми для компенсації похибок, аби уникати зривів дедлайнів. Цей підхід важливий у контексті систем реального часу, де надійність є критичною. У дослідженні [2] розглядається можливість інтеграції прогнозів у політики диспетчеризації реального часу. Автори пропонують підходи, що дозволяють коригувати параметри планування на основі короткострокових оцінок навантаження, при цьому зберігаючи гарантію передбачуваності для періодичних завдань. Особливу увагу приділено питанням безпечності адаптації параметрів. У роботі [3] аналізується розклад на однорідних машинах із використанням прогнозів. Доведено, що навіть неточні прогнози можуть суттєво знизити середній час завершення завдань за умови правильного обліку похибки. Такий підхід є релевантним для сценаріїв, де аперіодичні завдання складають значну частку навантаження. У публікації [4] досліджуються алгоритми планування, де система не має повної інформації про майбутнє, але використовує наближені оцінки для прогнозування навантаження. Це дозволяє скорочувати затримки в середньому випадку, водночас зберігаючи прийнятну надійність у гірших сценаріях. Практичну складову вносить робота [5], у якій досліджено використання методів машинного навчання та реальних фабричних даних для прогнозування часу обробки. Хоча завдання відрізняється від класичних систем реального часу, ідея залучення машинного навчання для отримання більш точних оцінок може бути перспективною і в контексті диспетчеризації з гарантіями. Таким чином, сучасні дослідження сходяться в тому, що прогнози можуть істотно зменшити затримки і підвищити пропускну здатність, але лише за умови обережного застосування та наявності механізмів, що компенсують похибки. Найбільш релевантні для запропонованої постановки є роботи [2] і [4], де прогнозування розглядається як допоміжний інструмент у поєднанні з класичними гарантіями передбачуваності.

Постановка завдання. Ефективна диспетчеризація з урахуванням короткострокового навантаження має поєднувати обережне коригування параметрів обслуговування з незмінними гарантіями своєчасності. Мета — використовувати короткострокову оцінку навантаження як допоміжний сигнал для прийняття рішень і застосовувати зміни лише після підтвердженої придатності розкладання для обраної політики. Основні вимоги та виклики включають:

- Узгодженість із гарантіями. Будь-яке коригування параметрів можливе лише після успішної перевірки придатності. Якщо перевірка показує ризик порушення дедлайнів, зміни не вносяться.

- Метрика та часовий інтервал. Потрібно обрати один показник короткострокової оцінки: або інтенсивність надходжень, або сумарний попит на процесорний час у близькому інтервалі. Інтервал оновлення має бути помірним: надто короткий створює зайві накладні витрати і чутливість до шуму, надто довгий запізнює реакцію на зміни. Доцільно зафіксувати сталий крок оновлення й дотримуватися його.

- Запас на похибку. Для кожної оцінки встановлюють верхній запас, який покриває можливу похибку. Він має бути достатнім для безпеки, але не настільки великим, щоб знижувати пропускну здатність. На практиці зручно застосовувати або сталий консервативний запас, або такий, що визначається за нещодавніми відхиленнями; у будь-якому разі правила його вибору мають бути чіткими й передбачуваними.

- Правила адаптації. Потрібно заздалегідь окреслити, які саме параметри дозволено коригувати та в яких діапазонах. Зазвичай це бюджет системи реального часу для аперіодичних завдань і інтервал його поповнення, а також правила допуску нових аперіодичних завдань. Частоту переналаштувань слід лімітувати, щоб не спричинити коливання режиму.

- Режим при невизначеності. У разі невдачі перевірки, різкої зміни трафіку або перевищення допустимої похибки система негайно повертається до консервативних налаштувань. Це правило має спрацьовувати автоматично, без додаткових умов і винятків.

- Накладні витрати та пріоритетність. Обчислення оцінки й виконання перевірок повинні бути легковаговими порівняно з основними роботами. Пріоритет періодичних завдань не повинен погіршуватися через тимчасові послаблення для аперіодичних завдань; у разі конфлікту пріоритет надається дотриманню дедлайнів.

- Оцінка результатів і межі застосовності. Для підсумкової оцінки доцільно фіксувати частку порушених дедлайнів, середню затримку завершення аперіодичних завдань і агрегований показник похибки оцінювання.

Виклад основного матеріалу. Основна ідея запропонованого підходу полягає у введенні короткострокового прогнозу навантаження на проміжку $[t, t + \Delta t]$. При цьому базова політика диспетчеризації залишається незмінною і забезпечує фундаментальні гарантії своєчасності для періодичних завдань, а механізм прогнозування лише коригує доступний бюджет для аперіодичних завдань.

Розглянемо ковзне часове вікно $[t, t + \Delta t]$, у межах якого аналізується навантаження від аперіодичних завдань. Позначимо через $A(t, \Delta t)$ множину аперіодичних завдань, що надійшли в цьому інтервалі. Для оцінки навантаження використовується сумарний попит на процесорний час:

$$C(t, \Delta t) = \sum_{i \in A(t, \Delta t)} C_i,$$

де C_i – оцінка часу виконання (мс) завдання i . Значення C_i може визначатися за останнім вимірним часом обслуговування або за ковзним середнім кількох попередніх виконань. Вибір конкретного способу залежить від характеру навантаження: для стабільних завдань достатньо простого усереднення, тоді як для високоваріативних потоків доцільно застосовувати методи, що швидко реагують на зміну статистики.

Щоб компенсувати невизначеність прогнозу, вводиться запас на похибку δ (мс), який враховує можливі відхилення між прогнозованим та фактичним навантаженням у вікні. З урахуванням цього запасу скоригована оцінка сумарного попиту набуває вигляду:

$$C^*(t, \Delta t) = C(t, \Delta t) + \delta.$$

Параметр δ може задаватися як константа, обрана на основі статистики найгірших спостережених сценаріїв, або адаптуватися динамічно (наприклад, пропорційно дисперсії похибок прогнозу). Збільшення δ підвищує надійність, але зменшує чутливість системи до потенційної економії ресурсів.

Базова політика планування визначає бюджет процесорного часу $V(t, \Delta t)$, доступний для обслуговування аперіодичних завдань у тому ж інтервалі $[t, t + \Delta t]$. Це може бути квота сервера аперіодичних завдань, фіксована частка гіперперіоду чи інший механізм обмеження. Для порівняння прогнозованого попиту з доступним бюджетом вводиться індекс навантаження:

$$\rho(t, \Delta t) = \frac{C^*(t, \Delta t)}{V(t, \Delta t)}$$

Індекс навантаження ρ інтерпретується як відношення очікуваного попиту до наявних ресурсів: $\rho < 1$ означає, що поточний бюджет є достатнім, тоді як $\rho > 1$ сигналізує про можливий дефіцит ресурсу й зростання затримок для аперіодичних завдань.

Безпосереднє коригування бюджету $V(t, \Delta t)$ при кожній зміні ρ може призвести до частого перемикання параметрів планувальника і погіршення передбачуваності системи. Для уникнення частих переключень використовується гістерезис із двома порогами τ_- та τ_+ ($\tau_- < \tau_+$), а також обмежується швидкість зміни бюджету. На основі цих порогів виділяються три режими роботи:

- якщо $\rho(t, \Delta t) < \tau_-$, система працює в базовому режимі, бюджет не змінюється.
- якщо $\tau_- \leq \rho(t, \Delta t) \leq \tau_+$, допускається тимчасове розширення бюджету для аперіодичних завдань за умови збереження здійсненності розкладу для періодичних завдань.
- якщо $\rho(t, \Delta t) > \tau_+$, застосовуються м'які обмеження: зменшення квот, відтермінування низькопріоритетних завдань або перенесення їх у фоновий потік.

Такий підхід дозволяє використовувати прогноз як механізм раннього попередження, не змінюючи фундаментальних гарантій системи реального часу. Важливо, що запропонований механізм не замінює класичних методів аналізу здійсненності для систем реального часу, а виступає надбудовою над обраною базовою політикою планування. Усі зміни бюджету для аперіодичних завдань виконуються лише в межах, сумісних із гарантованою здійсненністю розкладу для періодичних задач. Такий підхід дає змогу використати переваги прогнозування навантаження, зберігаючи формальні гарантії своєчасного виконання критичних завдань.

Для демонстрації запропонованого підходу виконано дискретно-часову симуляцію з кроком 1 мс для одноядерної системи з фіксованими пріоритетами та сервером аперіодичних запитів. Симуляцію реалізовано авторським дискретно-часовим симулятором, реалізованим мовою C++ (стандарт ISO/IEC~14882:2024). Планування періодичних задач виконується за алгоритмом планування Rate-Monotonic з фіксованими пріоритетами. Аперіодичні запити обслуговуються простим періодичним сервером із бюджетом $V(t, \Delta t)$ та періодом поповнення P_s . Генерація аперіодичних надходжень здійснюється за моделлю ON/OFF-процесу з пуассонівськими надходженнями у кожній фазі: у фазі ON (тривалість 2000 мс) та у фазі OFF (тривалість 8000 мс). Оцінювання проводилося у вікнах тривалості $\Delta t = 50$ мс, у межах кожного вікна $[t, t + \Delta t]$ конфігурація сервера вважалася сталою. Період поповнення бюджету сервера дорівнює $P_s = 50$ мс і збігається з Δt .

Використано набір періодичних задач, наведений у табл. 1, сумарне завантаження періодичними задачами становить $U = 0.707$, тобто близько 70.7% процесорного часу. Набір періодичних задач є синтетичним і сконструйованим для моделювання помірно завантаженої одноядерної вбудованої системи. Сумарне завантаження U обрано так, щоб залишити 29.3% процесорного часу для обслуговування аперіодичних запитів, причому це значення близьке до межі здійсненності $U = n * (2^{\frac{1}{n}} - 1)$ [7] для $n = 12$ задач: $U \approx 0.714$. Аперіодичні запити генерувалися у двох сценаріях: спокійному та зі сплесками. Для кожного сценарію порівнювалися два режими сервера — сталий і адаптивний.

Таблиця 1. Набір періодичних задач

№	T , мс	C , мс	$U = C/T$
1	8	1	0.1250
2	10	1	0.1000
3	16	1	0.0625
4	20	1	0.0500
5	25	1	0.0400
6	40	3	0.0750

7	50	3	0.0600
8	60	3	0.0500
9	80	3	0.0375
10	100	4	0.0400
11	125	4	0.0320
12	200	7	0.0350
Разом			0.7070

У сталому режимі сервер працює з фіксованим бюджетом $B(t, \Delta t) = 8$ мс, причому в межах кожного вікна $[t, t + \Delta t]$ це значення вважається сталим. В адаптивному режимі бюджет $B(t, \Delta t)$ переглядається дискретно кожні P_s та обмежується інтервалом $[B_{min}, B_{max}] = [8, 15]$ мс. Поведінку механізму адаптації визначає набір параметрів. Для оцінювання попиту та похибки використано експоненційне ковзне середнє з коефіцієнтом згладжування $\alpha = 0.25$. Під час формування S^* застосовується коефіцієнт запасу $\delta = 2.0$ мс. Щоб зменшити коливання, введено пороги гістерезису $\tau_- = 0.6$ та $\tau_+ = 1.2$. Параметр $W_{cd} = 2$ задає мінімальну кількість вікон між послідовними змінами бюджету. Коефіцієнт згладжування визначено за стандартною формулою експоненційного ковзного середнього [6]:

$$\alpha = \frac{2}{N + 1}$$

де N – кількість вікон для згладжування. Для симуляції було обрано $N = 7$, що дає $\alpha = 0.25$.

Таблиця 2. Результати симуляції

Сценарій	Режим	N_{miss}	N_{total}	N_{done}	R , мс	$R_{0.95}$, мс
спокійний	сталий	0	673	673	22.120	57
спокійний	адаптивний	0	673	673	21.869	57
зі сплесками	сталий	0	1605	1605	976.788	2128
зі сплесками	адаптивний	0	1605	1605	281.649	630

Згідно з табл. 2, порушень дедлайнів періодичних задач не зафіксовано, оскільки $N_{miss} = 0$. Це узгоджується з тим, що сумарне завантаження періодичними задачами становить $U = 0.707$. Для аперіодичних запитів також не спостерігається втрат, оскільки всі завдання було виконано $N_{total} = N_{done}$. У спокійному сценарії оброблено 673 запитів і 1605 запитів у сценарії зі сплесками. Порівняння затримок аперіодичних запитів показує перевагу адаптивного режиму. У табл. 2 наведено середню затримку R та 95-й перцентиль $R_{0.95}$, який відображає затримку, яку не перевищують 95% запитів. У спокійному сценарії різниця невелика, що очікувано за рівномірного навантаження: R зменшується з 22.120 мс до 21.869 мс, а $R_{0.95}$ залишається на рівні 57 мс. У сценарії зі сплесками R зменшується з 976.788 мс до 281.649 мс, а $R_{0.95}$ зменшується з 2128 мс до 630 мс. Отже, адаптивний режим найефективніше зменшує затримки саме під час короткочасних перевантажень. Для перевірки стійкості результатів до вибору параметрів адаптації проведено аналіз чутливості на сценарії зі сплесками. У табл. 3 наведено показники для $N \in \{3, 5, 7, 9\}$ та $\delta \in \{1.0, 2.0, 3.0\}$ при незмінних інших параметрах сервера.

Таблиця 3. Чутливість до параметрів адаптації

№	N	α	δ , мс	N_{miss}	R , мс	$R_{0.95}$, мс	Покриття
1	3	0.500	1.0	0	285.769	647	0.890
2	3	0.500	2.0	0	271.191	642	0.918
3	3	0.500	3.0	0	273.464	633	0.935
4	5	0.333	1.0	0	283.280	644	0.888
5	5	0.333	2.0	0	281.781	634	0.919
6	5	0.333	3.0	0	273.331	630	0.937
7	7	0.250	1.0	0	293.017	640	0.890
8	7	0.250	2.0	0	281.649	630	0.919
9	7	0.250	3.0	0	277.441	627	0.937
10	9	0.200	1.0	0	301.730	636	0.890

11	9	0.200	2.0	0	279.152	626	0.917
12	9	0.200	3.0	0	280.983	630	0.937

Як видно з табл. 3, у всіх конфігураціях адаптивний режим забезпечує відсутність порушень дедлайнів і суттєве зменшення R порівняно зі сталим режимом. Збільшення δ підвищує покриття – частку вікон, де фактичний попит не перевищує $C^*(t, \Delta t)$, одночасно зменшуючи R . Збільшення N дещо знижує R та $R_{0.95}$, але ефект δ домінує. Обрання $N = 7$ та $\delta = 2.0$ мс є компромісом між швидкістю реакції та стабільністю оцінки.

Якість прогнозу оцінювалася за відхиленням оцінки $C(t, \Delta t)$ від фактичного попиту у кожному вікні спостереження в адаптивному режимі.

Таблиця 4. Точність прогнозування попиту

Сценарій	MAE, мс	$\max e $, мс	Покриття
спокійний	1.943	15.305	0.878
зі сплесками	2.388	34.619	0.919

Як видно з табл. 4, MAE коливається в межах 1.9 – 2.4 мс, що становить 4% – 5% тривалості вікна $\Delta t = 50$ мс, що свідчить про достатню точність оцінки для задач адаптації бюджету. Максимальна похибка $\max|e|$ досягає 15 мс у спокійному та 35 мс у сценарії зі сплесками. У вікнах із заниженим прогнозом порушень дедлайнів не зафіксовано, оскільки тест здійсненності діє як незалежний бар'єр безпеки, що блокує будь-які потенційно небезпечні зміни бюджету.

Висновок. У статті запропоновано підхід до обережної адаптації параметрів обслуговування аперіодичних завдань на основі короткострокової оцінки навантаження. Прогноз використовується лише як допоміжний сигнал: перед зміною параметрів обслуговування оцінку доповнюють запасом на похибку та виконують перевірку здійсненності розкладу для періодичних завдань. Це дозволяє зменшувати затримки аперіодичних запитів без порушення формальних гарантій своєчасності критичних задач.

Результати моделювання підтверджують, що за розглянутих умов запропонований механізм не призводить до порушення дедлайнів періодичних задач і водночас покращує показники затримки для аперіодичних запитів, особливо в режимах зі сплесками навантаження.

Подальші дослідження доцільно спрямувати на точніші методи прогнозування й оцінювання навантаження, аналіз накладних витрат механізму адаптації та визначення меж застосовності підходу для ширшого класу систем і політик диспетчеризації.

Список бібліографічного опису

1. Zhao T., Li C., Li W., Zomaya A. Y. Brief Announcement: Towards a More Robust Algorithm for Flow Time Scheduling with Predictions. Proceedings of the 34th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA). – July 2022. – DOI: 10.1145/3490148.3538557.
2. Zhao T., Li W., Zomaya A. Y. Real-Time Scheduling with Predictions. Proceedings of the 2022 IEEE Real-Time Systems Symposium (RTSS). – December 2022. – DOI: 10.1109/RTSS55097.2022.00036.
3. Zhao T., Li W., Zomaya A. Y. Uniform Machine Scheduling with Predictions. Proceedings of the Thirty-Second International Conference on Automated Planning and Scheduling (ICAPS). – July 2022.
4. Im S., Kumar R., Montazer Qaem M., Purohit M. Non-Clairvoyant Scheduling with Predictions. Proceedings of the 33rd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA). – July 2021. – DOI: 10.1145/3409964.3461790.
5. Yamashiro H., Nonaka H. Estimation of processing time using machine learning and real factory data for optimization of parallel machine scheduling problem. Operations Research Perspectives. – July 2021. – Vol. 8, Issue 4. – DOI: 10.1016/j.orp.2021.100196.
6. Roberts S.-W. Control Chart Tests Based on Geometric Moving Averages. Technometrics. – 1959. – Vol. 1, No. 3. – P. 239-250. – DOI: 10.1080/00401706.1959.10489860.
7. Liu C.-L., Layland J.-W. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. Journal of the ACM. – 1973. – Vol. 20, No. 1. – P. 46-61. – DOI: 10.1145/321738.321743.

References

1. Zhao T., Li C., Li W., Zomaya A. Y. Brief Announcement: Towards a More Robust Algorithm for Flow Time Scheduling with Predictions. Proceedings of the 34th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA). – July 2022. – DOI: 10.1145/3490148.3538557.
2. Zhao T., Li W., Zomaya A. Y. Real-Time Scheduling with Predictions. Proceedings of the 2022 IEEE Real-Time Systems Symposium (RTSS). – December 2022. – DOI: 10.1109/RTSS55097.2022.00036.

3. Zhao T., Li W., Zomaya A. Y. Uniform Machine Scheduling with Predictions. Proceedings of the Thirty-Second International Conference on Automated Planning and Scheduling (ICAPS). – July 2022.
4. Im S., Kumar R., Montazer Qaem M., Purohit M. Non-Clairvoyant Scheduling with Predictions. Proceedings of the 33rd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA). – July 2021. – DOI: 10.1145/3409964.3461790.
5. Yamashiro H., Nonaka H. Estimation of processing time using machine learning and real factory data for optimization of parallel machine scheduling problem. Operations Research Perspectives. – July 2021. – Vol. 8, Issue 4. – DOI: 10.1016/j.orp.2021.100196.
6. Roberts S.-W. Control Chart Tests Based on Geometric Moving Averages. Technometrics. – 1959. – Vol. 1, No. 3. – P. 239-250. – DOI: 10.1080/00401706.1959.10489860.
7. Liu C.-L., Layland J.-W. Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment. Journal of the ACM. – 1973. – Vol. 20, No. 1. – P. 46-61. – DOI: 10.1145/321738.321743.

Історія статті:

Отримано: 16.01.2026 Доопрацьовано: 22.03.2026 Прийнято до друку: 23.03.2026 Опубліковано: 29.03.2026