

DOI: <https://doi.org/10.36910/6775-2524-0560-2026-64-05>

УДК 658.51:004.94:519.8

Шкітов Андрій Анатолійович¹, аспірант

<https://orcid.org/0009-0005-4600-8467>

Грибнюк Олександр Миколайович², к.т.н, с.н.с.

<https://orcid.org/0009-0004-5207-4631>

¹ Відкритий міжнародний університет розвитку людини «Україна», м. Київ, Україна

² ННЦ Інститут аграрної економіки, м. Київ, Україна

ГІБРИДНИЙ ПЛАНУВАЛЬНИК ПРОМИСЛОВОГО ВИРОБНИЦТВА З ВИКОРИСТАННЯМ SIMPY ТА ГЕНЕТИЧНОГО АЛГОРИТМУ

Шкітов А. А., Грибнюк О. М. Гібридний планувальник промислового виробництва з використанням simpy та генетичного алгоритму. У статті розв'язується завдання планування промислового виробництва за умов змінної тривалості технологічних операцій, відмов обладнання, конфліктів ресурсів та обмеженої місткості виробничих буферів. Запропоновано гібридний планувальник, який поєднує генетичний алгоритм із дискретно-подієвим моделюванням у середовищі SimPy. Генетичний алгоритм здійснює глобальний пошук послідовності виконання операцій і варіантів призначення обладнання, тоді як симуляційна модель відтворює виконання кожного сформованого розкладу з урахуванням черг, простоїв, відмов, інтервалів відновлення та відхилень фактичної тривалості операцій від нормативних значень. Розроблено дворівневе представлення хромосоми, що кодує порядок виконання операцій і вибір альтернативних виробничих ресурсів. На відміну від традиційних схем симуляційної оптимізації, запропонована модель реалізує замкнений цикл зворотного зв'язку, у межах якого SimPy передає події показники до еволюційного модуля. Отримані показники використовуються не лише для визначення пристосованості хромосом, а й для адаптивної зміни ймовірності мутації та спрямованого перепризначення операцій на менш завантажене обладнання. Запропоновано робастну процедуру оцінювання розкладів за результатами повторних симуляційних запусків із використанням медіани та медіанного абсолютного відхилення. Експериментальне дослідження проведено для виробничої системи, що містить 30 замовлень, 120 операцій і п'ять груп ресурсів. Порівняння з методами FCFS, SPT та статичним генетичним алгоритмом показало, що у комбінованому сценарії запропонований планувальник скорочує тривалість виробничого циклу на 7,0 %, сумарне запізнення – на 23,7 %, простої обладнання – на 17,8 %, а нерівномірність завантаження – на 27,4 %. Отримані результати підтверджують, що використання подієвого симуляційного зворотного зв'язку підвищує здійсненість, стійкість та операційну ефективність виробничих розкладів.

Ключові слова: виробниче планування, генетичний алгоритм, дискретно-подієве моделювання, SimPy, оптимізація розкладів, промислове виробництво.

Shkitov A., Hrybnyiuk O. Гібридний планувальник промислового виробництва з використанням simpy та генетичного алгоритму. The study addresses industrial production scheduling under variable operation durations, equipment failures, resource conflicts and limited buffer capacity. A hybrid scheduler combining a genetic algorithm with discrete-event simulation in SimPy is proposed. The genetic algorithm performs a global search for operation sequences and equipment assignments, while the simulation model reproduces the execution of each candidate schedule, including queues, downtime, failures, recovery intervals and deviations from standard processing times. A two-level chromosome representation encodes both the order of operations and the selection of alternative production resources. Unlike conventional simulation-based optimization schemes, the proposed model implements a closed feedback loop in which SimPy returns event-based indicators to the evolutionary module. These indicators are used not only to calculate chromosome fitness but also to adapt the mutation probability and direct resource reassignment toward less loaded equipment. A robust evaluation procedure based on repeated simulation runs, median values and median absolute deviation is introduced to reduce the influence of individual random realizations. The experimental study was conducted for a system containing 30 orders, 120 operations and five resource groups under deterministic, stochastic and combined disturbance scenarios. The GA-SimPy scheduler was compared with FCFS, SPT and a static genetic algorithm. Under the combined scenario, it reduced the production makespan by 7.0%, total tardiness by 23.7%, equipment idle time by 17.8% and load imbalance by 27.4% compared with the static genetic algorithm. The results confirm that event-driven simulation feedback improves the feasibility, robustness and operational efficiency of schedules under changing production conditions.

Key words: production scheduling, genetic algorithm, discrete-event simulation, SimPy, schedule optimization, industrial production.

Постановка проблеми та її зв'язок із важливими науковими чи практичними завданнями. У Сучасні виробничі підприємства функціонують в умовах постійної зміни попиту, нерівномірного надходження замовлень, обмеженості виробничих ресурсів та високих вимог до своєчасності виконання технологічних операцій [1]. Навіть незначні відхилення у роботі окремих виробничих ділянок здатні спричинити накопичення черг, нерівномірне завантаження обладнання, збільшення часу простоїв і, як наслідок, зростання собівартості продукції. У таких умовах традиційні методи календарного планування дедалі частіше виявляються недостатньо ефективними, оскільки вони переважно базуються на статичних припущеннях і не враховують стохастичний характер виробничих процесів.

Особливої актуальності ця проблема набуває в умовах цифрової трансформації промисловості та концепції Industry 4.0, де виробничі системи генерують значні обсяги оперативної інформації про стан обладнання, виробничих ліній, транспортних операцій і технологічних ресурсів. Використання таких даних відкриває можливість побудови більш точних моделей планування, однак водночас значно ускладнює процес пошуку оптимального виробничого розкладу через збільшення розмірності задачі та кількості взаємозалежних обмежень.

У більшості сучасних досліджень задачі виробничого планування розв'язуються із застосуванням евристичних або метаевристичних алгоритмів, серед яких значного поширення набули генетичні алгоритми [2], [3]. Їх перевагою є здатність знаходити якісні рішення у задачах великої комбінаторної складності без повного перебору можливих варіантів. Проте ефективність таких алгоритмів істотно залежить від способу оцінювання кожної особи популяції. Найчастіше ця оцінка виконується за допомогою спрощених математичних моделей або статичних функцій пристосованості, які не відображають реальної поведінки виробничої системи, не враховують випадкові затримки, конфлікти за ресурси, утворення черг та інші події, що виникають під час виконання виробничого процесу.

Одним із перспективних напрямів подолання зазначених обмежень є інтеграція дискретно-подієвого моделювання з еволюційними методами оптимізації. Зокрема, бібліотека SimPy дозволяє моделювати функціонування виробничої системи як послідовність взаємопов'язаних подій із урахуванням часової динаміки, використання ресурсів і взаємодії між технологічними операціями. Поєднання можливостей дискретно-подієвого моделювання із генетичним алгоритмом створює передумови для більш достовірного оцінювання якості сформованих виробничих планів [4]. Водночас питання ефективної організації такої взаємодії та побудови функції пристосованості, що враховує результати імітаційного моделювання, залишаються недостатньо дослідженими. Крім того, потребують подальшого опрацювання підходи до мінімізації обчислювальних витрат під час інтеграції симуляційної та оптимізаційної складових.

Таким чином, актуальним науково-практичним завданням є розроблення гібридного планувальника виробничих процесів, який поєднує еволюційний пошук оптимального розкладу з дискретно-подієвим моделюванням виробничої системи для забезпечення більш реалістичного оцінювання виробничих сценаріїв і підвищення ефективності використання виробничих ресурсів.

Метою дослідження є розроблення гібридного планувальника промислового виробництва на основі інтеграції дискретно-подієвого моделювання SimPy та генетичного алгоритму, який забезпечує пошук виробничих розкладів із мінімальним часом виконання замовлень, зменшеними простоями обладнання та збалансованим використанням виробничих ресурсів за рахунок оцінювання кожного кандидата безпосередньо у процесі імітаційного моделювання.

Аналіз останніх досліджень та публікацій. Проблематика планування промислового виробництва охоплює широкий клас задач, пов'язаних із визначенням послідовності виконання технологічних операцій, розподілом робіт між доступними ресурсами, дотриманням установлених строків і мінімізацією непродуктивних витрат. Особливе місце серед них посідають задачі типу Job Shop Scheduling Problem та Flexible Job Shop Scheduling Problem, які належать до класу комбінаторно складних задач і характеризуються стрімким зростанням простору можливих рішень зі збільшенням кількості замовлень, операцій і виробничого обладнання [5], [6]. У сучасних дослідженнях як основні критерії оптимізації переважно використовуються тривалість виробничого циклу, максимальне запізнення, сумарний час простою, завантаження обладнання, виробничі витрати та енергоспоживання.

Для розв'язання задач виробничого планування застосовуються методи математичного програмування, правила диспетчеризації, евристичні й метаевристичні алгоритми, імітаційне моделювання, машинне навчання та комбіновані підходи [7]. Методи змішаного цілочислового програмування забезпечують формалізоване подання технологічних обмежень і дають змогу знаходити оптимальні рішення для задач відносно невеликої розмірності. Зокрема, у дослідженнях із планування потокового виробництва запропоновано інтеграцію змішаного цілочислового програмування з дискретно-подієвим моделюванням і декомпозицією Бендерса [8]. Такий підхід дає змогу поєднати точність оптимізаційної моделі з реалістичним відтворенням виробничих процесів, однак зі збільшенням кількості операцій, альтернативних маршрутів і випадкових подій істотно зростає обчислювальна складність задачі.

Значне поширення у виробничому плануванні отримали генетичні алгоритми, перевагами

яких є можливість опрацювання великого простору пошуку, відсутність вимоги щодо диференційованості цільової функції та придатність до задач із дискретними змінними й нелінійними обмеженнями [9]. У межах таких підходів виробничий розклад кодується у вигляді хромосоми, після чого шляхом селекції, схрещування та мутації формується нова популяція потенційних рішень. Розвиток цього напрямку пов'язаний із модифікацією операторів еволюції, багатокритеріальним оцінюванням розкладів і гібридизацією генетичного пошуку з локальними евристичними. Наприклад, для багатокритеріального планування виробничих операцій запропоновано вдосконалений алгоритм NSGA-III, спрямований на одночасне покращення кількох показників функціонування виробничої системи [10]. Проте результативність таких алгоритмів значною мірою визначається способом формування функції пристосованості та відповідністю її компонентів реальним умовам виробництва.

У роботах, присвячених гнучкому виробництву, розглядається одночасний вибір послідовності операцій і призначення технологічного обладнання. Запропоновані моделі враховують альтернативні маршрути оброблення, різну продуктивність верстатів, обмеження доступності ресурсів і можливість паралельного виконання окремих операцій. У дослідженні гнучкого планування з подвійними ресурсними обмеженнями оптимізаційна модель охоплює не лише використання обладнання, а й доступність працівників, що підвищує відповідність отриманого розкладу реальним виробничим умовам [11]. Водночас у таких моделях тривалість операцій здебільшого задається детермінованими значеннями, а випадкові відмови, затримки, накопичення черг і зміни стану ресурсів або не враховуються, або подаються через спрощені коефіцієнти.

Перспективним напрямом розвитку методів виробничого планування є використання дискретно-подієвого моделювання [12]. На відміну від статичних оптимізаційних моделей, дискретно-подієва модель відтворює зміну стану виробничої системи в моменти виникнення визначених подій: надходження замовлення, початку або завершення операції, зайняття чи звільнення обладнання, утворення черги, відмови та відновлення ресурсу [13]. Завдяки цьому імітаційна модель дає змогу оцінити розклад з урахуванням динамічної взаємодії технологічних процесів і ресурсних конфліктів. Систематизація сучасних simulation-optimization підходів показує, що дискретно-подієве моделювання дедалі частіше застосовується як складова систем підтримки виробничих рішень, особливо за наявності збурень і невизначеності. Однак актуальними залишаються проблеми швидкодії, автоматизованого обміну даними між симулятором та оптимізатором, а також адаптації плану до зміни виробничого стану.

Для реалізації дискретно-подієвих моделей може бути використано середовище SimPy – процесно орієнтований фреймворк, у якому поведінка активних компонентів задається генераторними функціями, а модельний час змінюється шляхом переходу між подіями. SimPy підтримує подання спільних ресурсів, черг, процесів очікування, переривань і взаємодії між процесами, що дає змогу формалізувати роботу верстатів, транспортних засобів, накопичувачів, операторів та інших елементів виробничої системи [14]. Водночас SimPy є передусім інструментом моделювання, а не засобом автоматичного формування оптимального розкладу, тому його застосування у виробничому плануванні потребує інтеграції з окремим оптимізаційним модулем.

Поєднання дискретно-подієвого моделювання з метаввристичною оптимізацією сформувало окремий напрям simulation-based optimization. У таких системах оптимізаційний алгоритм формує набір можливих рішень, а симуляційна модель відтворює їх виконання та повертає кількісні показники для оцінювання. Наприклад, для стохастичної задачі гнучкого виробничого планування в напівпровідниковій промисловості запропоновано simulation-optimization framework, у якому враховано випадковий характер виробничих процесів і особливості функціонування вузьких місць. Результати такого дослідження підтверджують доцільність використання симуляції для оцінювання рішень у системах, де детерміноване подання тривалості операцій не забезпечує необхідної точності.

Інший simheuristic-підхід до стохастичного Job Shop Scheduling Problem поєднує пошук табу з моделюванням методом Монте-Карло [15]. У ньому враховуються випадкові інтервали між відмовами обладнання та тривалість його відновлення, а цільовим показником є очікуване максимальне запізнення. Проведення експериментів на 320 тестових прикладах засвідчило, що включення стохастичних факторів до процедури оцінювання дає змогу формувати стійкіші виробничі плани порівняно з оптимізацією лише для номінального сценарію [16]. Водночас

багаторазове моделювання кожного рішення збільшує тривалість обчислень і потребує спеціальних механізмів скорочення кількості симуляційних запусків.

У дослідженнях адаптивного планування для Industry 5.0 запропоновано гнучку взаємодію між виробничою симуляцією та оптимізаційним модулем. Така взаємодія передбачає використання результатів моделювання не лише для підсумкової перевірки сформованого плану, а й для коригування рішень на проміжних ітераціях. Подібний підхід підвищує адаптивність системи підтримки прийняття рішень, проте потребує узгодженого подання параметрів моделі, критеріїв оптимізації та умов повторного запуску процедури пошуку.

Окремі сучасні роботи безпосередньо поєднують генетичний алгоритм із виробничою симуляцією. Зокрема, для стохастичного планування складання електронних компонентів запропоновано *simulation-based scheduling method*, у якому генетичний алгоритм формує послідовність виробничих завдань, а симуляція оцінює тривалість виконання, сумарне запізнення та ризик порушення строків постачання. Такий підхід дає змогу врахувати відмінності між детермінованими та стохастичними виробничими стадіями, однак він орієнтований на конкретну технологічну структуру й не формує універсальної моделі взаємодії оптимізатора з дискретно-подієвим середовищем.

У роботах, присвячених перевірці виробничих розкладів засобами дискретно-подієвого моделювання, наголошується, що математично допустимий розклад може виявитися практично нездійсненним після врахування конфліктів ресурсів, залежностей маршрутів і реальних правил функціонування виробничої лінії [17], [18]. Запропонована в таких дослідженнях схема передбачає формування розкладу генетичним алгоритмом із подальшою його перевіркою у симуляційному середовищі. Разом із тим використання симуляції лише як зовнішнього засобу валідації обмежує її вплив на еволюційний пошук: інформація про черги, простой та локальні перевантаження не завжди використовується для спрямованої зміни наступних поколінь розкладів.

Результати аналізу наукових публікацій дають підстави виокремити три основні підходи до розв'язання задач виробничого планування. Перший підхід ґрунтується на детермінованій оптимізації та забезпечує точне врахування формальних обмежень, але має обмежену масштабованість і недостатньо повно подає динаміку виробництва. Другий передбачає використання генетичних та інших метаевристичних алгоритмів, які придатні для задач великої розмірності, проте часто оцінюють рішення за спрощеними аналітичними функціями. Третій підхід поєднує оптимізацію з дискретно-подієвим моделюванням, завдяки чому підвищується реалістичність оцінювання розкладів, але зростають обчислювальні витрати та ускладнюється організація двобічного обміну між оптимізаційною і симуляційною складовими.

Отже, попри значну кількість досліджень у сфері генетичного планування та *simulation-based optimization*, недостатньо опрацьованими залишаються питання безпосереднього використання подієвих показників SimPy у процесі еволюційного пошуку. У більшості відомих рішень симулятор застосовується для підсумкового оцінювання або валідації розкладу, тоді як виявлені в процесі моделювання черги, простой, перевантаження ресурсів і порушення строків не впливають на структуру генетичних операторів та параметри наступної ітерації. Також потребує подальшого розвитку формування інтегрованої функції пристосованості, яка поєднує тривалість виробничого циклу, запізнення замовлень, простой та нерівномірність використання обладнання без заміни цих показників спрощеними статичними оцінками. Невирішеним залишається і завдання скорочення кількості повних симуляційних запусків, оскільки оцінювання кожної хромосоми у середовищі дискретно-подієвого моделювання може суттєво збільшувати загальну тривалість оптимізації.

Подальше дослідження доцільно спрямувати на розроблення гібридного планувальника, у якому генетичний алгоритм формує виробничі розклади, SimPy відтворює їх фактичне виконання з урахуванням ресурсних конфліктів і динамічних подій, а отримані симуляційні показники безпосередньо використовуються для обчислення пристосованості та адаптивної зміни параметрів еволюційного пошуку. На відміну від послідовної схеми «оптимізація – перевірка», така організація утворює замкнений цикл взаємодії між оптимізаційною і симуляційною складовими та створює передумови для формування практично здійснених виробничих розкладів.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження. Розглядається виробнича система, що складається з множини замовлень $J = \{J_1, J_2, \dots, J_n\}$, множини технологічних операцій $O = \{O_{ij}\}$ та множини виробничих ресурсів $M = \{M_1, M_2, \dots, M_m\}$. Кожне замовлення J_i містить упорядковану послідовність операцій $O_{i1}, O_{i2}, \dots, O_{ir_i}$, які повинні

виконуватися відповідно до визначеного технологічного маршруту. Для кожної операції задається множина допустимого обладнання $M_{ij} \subseteq M$, тривалість виконання p_{ijk} на ресурсі M_k , момент готовності замовлення a_i та граничний строк завершення d_i .

Основним результатом планування є розклад $S = \{(O_{ij}, M_k, s_{ij}, c_{ij})\}$, де s_{ij} – відповідно моменти початку та завершення операції O_{ij} , а M_k – призначений для її виконання виробничий ресурс.

Для кожної операції повинна виконуватися умова $c_{ij} = s_{ij} + p_{ijk}$.

Технологічна послідовність операцій одного замовлення визначається обмеженням $s_{i,j+1} \geq c_{ij}, j = 1, \dots, r_{i-1}$.

Одночасне виконання двох операцій на одному ресурсі не допускається. Для операцій O_{ij} та O_{uv} , призначених на ресурс M_k , повинна виконуватися одна з умов $c_{ij} \leq s_{uv}$ або $c_{uv} \leq s_{ij}$.

На відміну від класичної детермінованої постановки, тривалість виконання операції у запропонованій моделі розглядається як випадкова величина $\tilde{p}_{ijk} = p_{ijk}(1 + \varepsilon_{ijk})$, де ε_{ijk} характеризує відхилення фактичної тривалості операції від її нормативного значення. Крім того, враховуються випадкові відмови обладнання, тривалість відновлення, черги перед ресурсами та затримки передавання напівфабрикатів між виробничими дільницями.

Запропонований гібридний планувальник поєднує генетичний алгоритм, який виконує глобальний пошук послідовності операцій і варіантів призначення обладнання, з дискретно-подієвою моделлю SimPy, призначеною для відтворення фактичного виконання сформованого розкладу. Особливістю моделі є організація замкненого циклу взаємодії: результати симуляції використовуються не лише для обчислення функції пристосованості, а й для адаптивного коригування операторів наступної ітерації генетичного алгоритму.

Структурно гібридний планувальник містить п'ять функціональних компонентів: модуль підготовки виробничих даних, модуль генетичного пошуку, декодер хромосоми, дискретно-подієву модель SimPy та модуль аналізу симуляційного зворотного зв'язку.

Архітектуру взаємодії основних компонентів запропонованого планувальника доцільно подати на рис. 1.

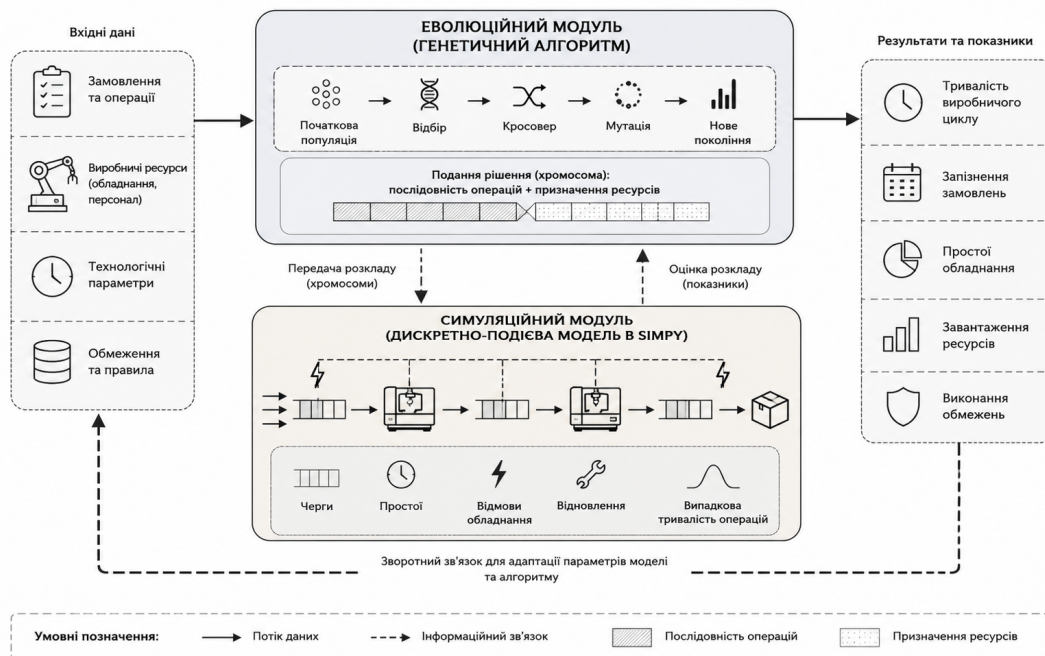


Рис. 1. Архітектура гібридного планувальника промислового виробництва

Вхідними даними є технологічні маршрути замовлень, множини допустимого обладнання, нормативна тривалість операцій, строки виконання, календар доступності ресурсів, статистичні параметри відмов і відновлення обладнання. На основі цих даних генерується початкова популяція хромосом.

Кожна хромосома передається до декодера, який перетворює її у попередній виробничий план. Декодований план не вважається остаточним результатом, оскільки він сформований без

повного врахування подієвої динаміки виробничої системи. Після декодування план надходить до моделі SimPy, де кожне замовлення подається як окремий процес, а верстати, оператори, транспортні засоби та накопичувальні буфери – як обмежені ресурси.

Під час симуляції реєструються такі події $E = \{e_{arr}, e_{req}, e_{start}, e_{finish}, e_{fail}, e_{repair}, e_{release}\}$, де e_{arr} означає надходження замовлення, e_{req} – запит ресурсу, e_{start} і e_{finish} – початок та завершення операції, e_{fail} – відмову обладнання, e_{repair} – завершення відновлення, а $e_{release}$ – звільнення ресурсу.

За результатами кожного симуляційного запуску формується вектор характеристик розкладу (1).

$$z(S) = [C_{max}, T_{\Sigma}, I_{\Sigma}, Q_{max}, D_U, N_{viol}] \quad (1)$$

де C_{max} – тривалість виконання всього виробничого плану; T_{Σ} – сумарне запізнення замовлень; I_{Σ} – сумарний час простою обладнання; Q_{max} – максимальна довжина черги; D_U – нерівномірність завантаження ресурсів; N_{viol} – кількість порушень виробничих обмежень.

Сумарне запізнення визначається як (2).

$$T_{\Sigma} = \sum_{i=1}^n \max(0, C_i - d_i) \quad (2)$$

де C_i – фактичний момент завершення замовлення J_i , отриманий у процесі симуляції.

Коефіцієнт завантаження ресурсу M_k обчислюється за виразом $U_k = \frac{B_k}{H}$, де B_k – фактичний час зайнятості ресурсу, H – горизонт моделювання. Нерівномірність використання обладнання визначається як середньоквадратичне відхилення (3).

$$D_U = \sqrt{\frac{1}{m} \sum_{k=1}^m (U_k - \bar{U})^2} \quad (3)$$

де $\bar{U}$ – середній рівень завантаження виробничих ресурсів.

Наукова новизна запропонованої архітектури полягає у використанні розширеного зворотного зв'язку від SimPy до генетичного алгоритму. Крім інтегрального значення пристосованості, оптимізаційний модуль отримує вектор локальних характеристик обладнання $b = \{b_1, b_2, \dots, b_m\}$, де $b_k = \alpha_q \frac{\bar{Q}_k}{Q_{max}} + \alpha_u U_k + \alpha_f \frac{F_k}{F_{max}}$. Тут $\bar{Q}_k$ – середня довжина черги перед ресурсом M_k , F_k – кількість відмов ресурсу протягом горизонту моделювання. Значення b_k використовується для визначення вузьких місць і спрямованої зміни призначень операцій у наступних поколіннях.

Для одночасного кодування послідовності операцій і вибору обладнання використано дворівневу структуру хромосоми $X = (X^{OS}, X^{MS})$, де X^{OS} – рівень послідовності операцій, а X^{MS} – рівень вибору виробничих ресурсів.

У частині X^{OS} кожний ген містить номер замовлення. Кількість повторень номера замовлення відповідає кількості його технологічних операцій. Наприклад, послідовність $X^{OS} = [1, 2, 1, 3, 2, 1, 3, 2]$ означає, що перша поява номера 1 відповідає першій операції замовлення J1, друга – другій операції цього замовлення, а третя – третій. Такий спосіб кодування автоматично підтримує технологічну послідовність операцій і скорочує кількість недопустимих хромосом.

У частині X^{MS} кожний ген містить номер обладнання, призначеного для відповідної операції $X^{MS} = [2, 1, 3, 2, 4, 3, 1, 2]$.

Приклад дворівневої хромосоми та її перетворення у виробничий розклад необхідно подати на рис. 2.

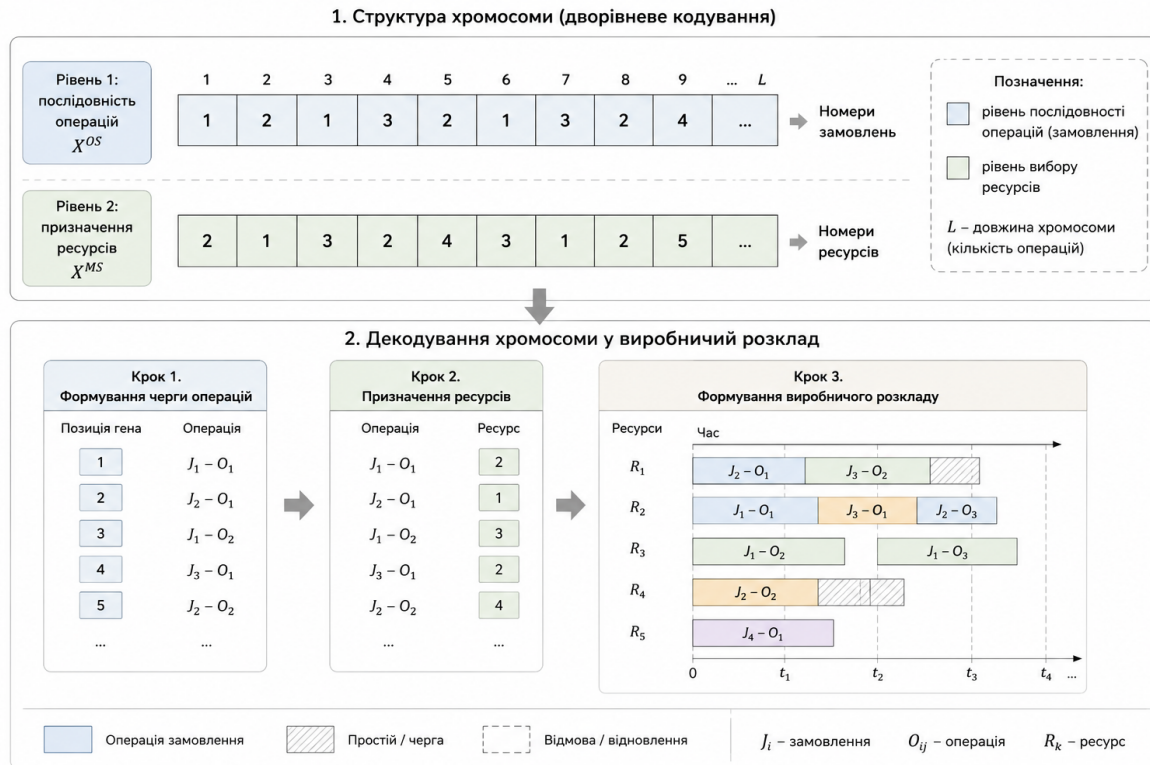


Рисунок 2 – Структура хромосоми та процедура декодування виробничого розкладу

Початкова популяція формується комбінованим способом. Частина хромосом генерується випадково, частина – на основі правила найкоротшої операції, а решта – за правилом мінімального прогнозованого завантаження обладнання. Такий підхід забезпечує одночасно різноманітність популяції та наявність у ній початкових рішень прийнятної якості.

Для відбору батьківських хромосом використовується турнірна селекція. Схрещування частини X^{OS} виконується методом precedence operation crossover, який зберігає коректну кількість операцій кожного замовлення. Для частини X^{MS} застосовується рівномірне схрещування, за якого призначення обладнання успадковується від одного з батьків відповідно до випадково сформованої двійкової маски.

Мутація послідовності операцій передбачає перестановку двох генів або переміщення одного гена в іншу позицію. Мутація рівня вибору ресурсів змінює призначення операції на альтернативний допустимий ресурс.

На відміну від традиційного генетичного алгоритму, ймовірність мутації не є сталою. Вона залежить від динаміки покращення популяції (4).

$$p_m(g) = \min(p_m^{max}, p_m^0 + \gamma \frac{g-g^*}{G_s}) \quad (4)$$

де p_m^0 – початкова ймовірність мутації; p_m^{max} – її граничне значення; g – номер поточного покоління; g^* – номер останнього покоління, в якому було покращено найкраще рішення; G_s – допустима кількість поколінь без покращення; γ – коефіцієнт адаптації.

Якщо значення функції пристосованості не покращується протягом кількох поколінь, інтенсивність мутації збільшується. Після виявлення кращого розкладу ймовірність мутації повертається до початкового значення. Це запобігає передчасній збіжності популяції.

Додатково запропоновано оператор керованої ресурсної мутації. Ймовірність вибору операції, призначеної на ресурс M_k , визначається пропорційно до рівня його перевантаження (5).

$$p_k^{mut} = \frac{b_k}{\sum_{h=1}^m b_h} \quad (5)$$

Після вибору перевантаженого ресурсу одна з призначених на нього операцій переноситься на альтернативне обладнання з найменшим прогнозованим навантаженням. Таким чином, результати подієвого моделювання безпосередньо впливають на структуру наступного покоління, а не лише на ранжування вже сформованих розкладів.

Для збереження найкращих рішень використовується елітарна стратегія. До наступного

покоління без змін переносяться хромосоми, які належать до множини недомінованих рішень за показниками C_{max} , T_{Σ} , D_U . Це дає змогу уникнути втрати розкладів, які мають різний компроміс між швидкістю виконання, своєчасністю замовлень і збалансованістю обладнання.

Метод оцінювання передбачає запуск окремої дискретно-подієвої симуляції для кожної хромосоми. Після декодування визначається пріоритет операцій і бажане призначення ресурсів, однак фактичні моменти їх початку та завершення встановлюються у середовищі SimPy з урахуванням поточного стану системи.

Кожне замовлення створюється як процес $P_i: O_{i1} \rightarrow O_{i2} \rightarrow \dots \rightarrow O_{ir_i}$. Перед виконанням операції процес формує запит до відповідного ресурсу. Якщо ресурс зайнятий, замовлення переходить у чергу. Після отримання ресурсу генерується затримка, що відповідає фактичному часу оброблення. У разі виникнення відмови операція переривається, ресурс переходить у стан відновлення, а невиконана частина операції продовжується після ремонту.

Послідовність оцінювання хромосоми в середовищі SimPy потрібно відобразити на рис. 3.

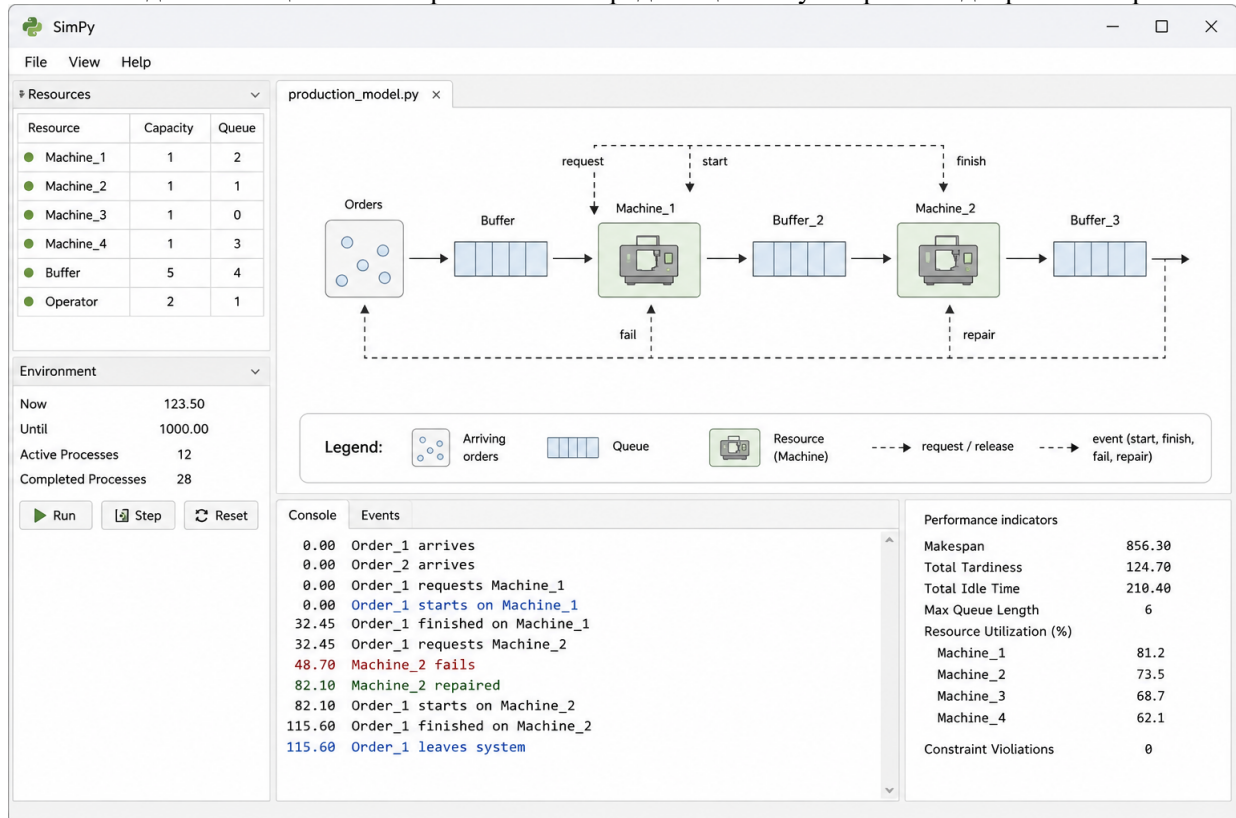


Рис. 3. Метод оцінювання виробничого розкладу на основі SimPy

Щоб зменшити вплив окремої випадкової реалізації, кожна перспективна хромосома оцінюється за кількома симуляційними прогонами. Для хромосоми X формується множина результатів (6).

$$Z(X) = \{z^{(1)}(X), z^{(2)}(X), \dots, z^{(R)}(X)\} \quad (6)$$

де R – кількість повторних запусків.

Для кожного критерію визначається робастна оцінка (7).

$$\hat{z}_l(X) = \text{median}\{z^{(1)}, \dots, z^{(R)}(X)\} + \eta \text{MAD}_l(X) \quad (7)$$

де MAD – медіанне абсолютне відхилення; η – параметр чутливості до нестабільності розкладу. Завдяки цьому перевага надається не лише швидким, а й стійким розкладам, результати виконання яких менше змінюються під впливом випадкових відмов і коливань тривалості операцій.

Для інтегрального ранжування хромосом використовується функція відхилення від адаптивної опорної точки (8).

$$D(X) = \max_l \left[\frac{\hat{z}_l(X) - z_l^*}{z_l^{\text{ref}} - z_l^* + \delta} \right] + \rho \sum_l \frac{\hat{z}_l(X) - z_l^*}{z_l^{\text{ref}} - z_l^* + \delta} \quad (8)$$

де z_l^* – найкраще значення критерію, досягнуте у поточній популяції; z_l^{ref} – опорне гірше значення; δ запобігає діленню на нуль; ρ – малий параметр регуляризації.

На відміну від звичайної зваженої суми, така функція не допускає компенсації критично поганого значення одного показника незначним покращенням іншого. Хромосома з надмірним запізненням або значним перевантаженням окремого ресурсу отримує нижчий ранг навіть за відносно малого загального часу виконання.

Функція пристосованості визначається як (9).

$$F(X) = \frac{1}{1 + D(X) + \lambda_v N_{viol}(X)} \quad (9)$$

Для скорочення обчислювальних витрат повне повторне моделювання застосовується не до всієї популяції. На першому етапі кожна хромосома проходить один симуляційний запуск. Додаткові $R - 1$ повторів виконуються лише для частки найкращих хромосом, що пройшли попередній відбір. Такий механізм знижує кількість симуляцій без втрати точності оцінювання перспективних рішень.

Експериментальну перевірку проведено шляхом моделювання виробничої ділянки з п'ятьма групами обладнання, на якій виконуються замовлення з альтернативними технологічними маршрутами. Програмну реалізацію гібридного планувальника виконано мовою Python. Дискретно-подієва модель побудована у середовищі SimPy, а реалізація генетичних операторів і керування популяцією виконана у вигляді окремого оптимізаційного модуля.

Обчислювальні експерименти проводилися на комп'ютері з багатоядерним процесором, 16 ГБ оперативної пам'яті та операційною системою Windows 11. Для забезпечення відтворюваності використовувалася фіксована множина початкових генераторів випадкових чисел. Кожний дослід повторювався 30 разів, після чого визначалися медіанні значення показників.

Параметри змодельованої виробничої системи наведено в табл. 1.

Таблиця 1 – Параметри експериментальної виробничої системи

Параметр	Значення
Кількість замовлень	30
Кількість виробничих ресурсів	5
Загальна кількість операцій	120
Кількість операцій в одному замовленні	3–5
Кількість альтернативних ресурсів для операції	1–3
Нормативна тривалість операції	8–45 хв
Відхилення фактичної тривалості	до $\pm 20\%$
Середній інтервал між відмовами обладнання	480 хв
Середній час відновлення	35 хв
Місткість проміжного буфера	8 одиниць
Розмір популяції	80 хромосом
Максимальна кількість поколінь	150
Початкова ймовірність мутації	0,08
Максимальна ймовірність мутації	0,30
Ймовірність схрещування	0,85
Кількість повторних симуляцій кращих хромосом	5
Кількість незалежних запусків	30

Для перевірки ефективності планувальника сформовано чотири експериментальні сценарії. У першому сценарії тривалість усіх операцій відповідала нормативним значенням, а відмови обладнання були відсутні. У другому сценарії фактичний час оброблення випадково змінювався в межах $\pm 20\%$. У третьому сценарії додатково моделювалися відмови та відновлення обладнання. Четвертий сценарій передбачав комбіноване навантаження: випадкову тривалість операцій, відмови ресурсів, обмежену місткість буферів і надходження термінових замовлень під час виконання плану.

Для порівняння використано такі базові методи: правило FCFS, за якого замовлення обробляються у порядку надходження; правило SPT, що надає пріоритет операції з найменшою тривалістю; статичний генетичний алгоритм, у якому функція пристосованості обчислюється без дискретно-подієвої симуляції; запропонований гібридний планувальник GA-SimPy із симуляційним зворотним зв'язком.

Узагальнені результати для комбінованого четвертого сценарію наведено в табл. 2.

Таблиця 2. Порівняння результатів планування виробництва

Метод	C_{max} , хв	T_{Σ} , хв	Простої обладнання, %	D_U	Середня довжина черги	Час планування, с
FCFS	1168	324	22,8	0,162	5,8	0,02
SPT	1096	281	19,4	0,147	4,9	0,03
Статичний генетичний алгоритм	1018	198	15,7	0,113	3,7	12,8
Запропонований GA-SimPy	947	151	12,9	0,082	2,6	37,4

Отримані результати показують, що використання лише диспетчерських правил не забезпечує збалансованого розподілу навантаження. Метод FCFS характеризувався найбільшим часом завершення виробничого плану та найбільшим сумарним запізненням. Правило SPT зменшило тривалість циклу на 6,2 % порівняно з FCFS, однак спричиняло відкладення довготривалих операцій і не враховувало можливі перевантаження окремих ресурсів.

Статичний генетичний алгоритм забезпечив кращі результати завдяки глобальному пошуку послідовності операцій і призначень обладнання. Проте розклад, сформований на основі нормативної тривалості операцій, втрачав частину своєї ефективності після моделювання відмов і черг. Це підтверджує, що аналітично прийнятний план не завжди залишається ефективним у динамічному виробничому середовищі.

Запропонований гібридний планувальник скоротив загальний час виконання виробничої програми на 71 хв, або 7,0 %, порівняно зі статичним генетичним алгоритмом. Сумарне запізнення замовлень зменшилося на 47 хв, що становить 23,7 %. Частка простоїв обладнання була знижена з 15,7 до 12,9 %, а нерівномірність завантаження ресурсів – на 27,4 %.

Покращення показників пояснюється тим, що оператор керованої мутації використовував інформацію про фактичні черги та перевантажені ресурси. Унаслідок цього операції частіше переносилися не на випадкове альтернативне обладнання, а на ресурси з нижчим подієвим навантаженням. Це дозволило зменшити середню довжину черги з 3,7 до 2,6 замовлення.

Динаміку збіжності статичного генетичного алгоритму та запропонованого планувальника, а також фрагмент отриманої діаграми Ганта представлено на рис. 4.

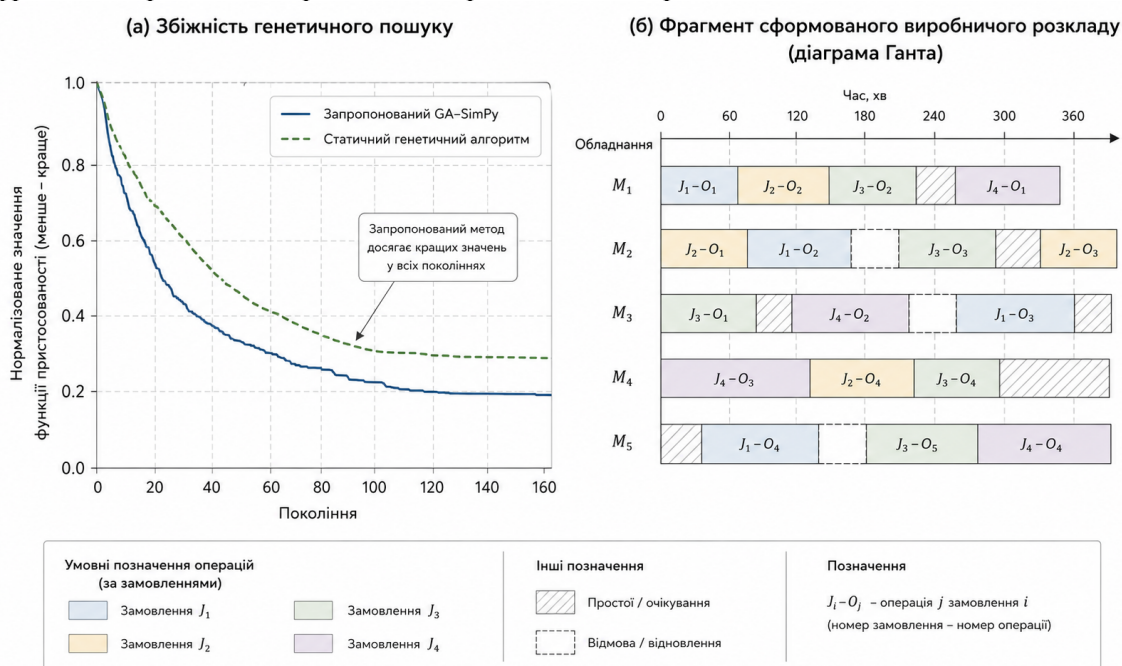


Рис.4. Збіжність генетичного пошуку та фрагмент сформованого виробничого розкладу

У перших поколіннях обидва генетичні методи демонстрували близьку швидкість покращення рішень. Після приблизно 40–50 поколінь статичний алгоритм переходив до повільної

збіжності, оскільки повторно формував схожі призначення для перевантажених ресурсів. У запропонованому методі накопичення черг спричиняло збільшення ймовірності ресурсної мутації, унаслідок чого пошук продовжувався в напрямі менш завантажених конфігурацій.

Збільшення часу планування з 12,8 до 37,4 с є наслідком виконання дискретно-подієвих симуляцій. Однак отримана тривалість залишається прийнятною для оперативно-календарного планування, де план формується на період, значно більший за кілька десятків секунд. Крім того, дворівнева схема оцінювання, за якої повторні симуляції виконуються лише для перспективних хромосом, зменшила кількість повних запусків моделі приблизно на 58 % порівняно з оцінюванням кожної хромосоми за п'ятьма повтореннями.

Додатково встановлено, що перевага запропонованої моделі зростала зі збільшенням рівня невизначеності. У детермінованому сценарії різниця між статичним і гібридним генетичними алгоритмами за тривалістю виробничого циклу не перевищувала 2,5 %. За наявності випадкової тривалості операцій вона зросла до 4,8 %, а у комбінованому сценарії з відмовами, чергами та терміновими замовленнями – до 7,0 %. Це підтверджує доцільність використання SimPy саме для систем, у яких фактичний перебіг виробничого процесу істотно відрізняється від номінального плану.

Висновки та перспективи подальшого дослідження. У статті представлено гібридну модель планування промислового виробництва, що поєднує генетичний алгоритм із дискретно-подієвим моделюванням у середовищі SimPy. Запропонований підхід забезпечує формування виробничих розкладів з урахуванням не лише нормативної тривалості операцій, а й фактичних черг, простоїв, відмов обладнання, нерівномірності завантаження ресурсів і порушень установлених строків виконання замовлень.

Наукова новизна дослідження полягає у формуванні замкненого циклу взаємодії між оптимізаційним і симуляційним модулями, за якого результати дискретно-подієвого моделювання використовуються для обчислення пристосованості хромосом та адаптивного коригування генетичних операторів. Запропоновано дворівневе подання хромосоми, робастний метод оцінювання розкладів за повторними симуляційними прогонами та оператор керованої ресурсної мутації, що враховує локальне перевантаження обладнання.

Результати експериментального дослідження засвідчили, що гібридний планувальник скорочує тривалість виробничого циклу, сумарне запізнєння замовлень, простої обладнання та нерівномірність використання ресурсів порівняно з правилами FCFS, SPT і статичним генетичним алгоритмом. Найбільша перевага запропонованого підходу виявляється за наявності випадкових змін тривалості операцій, відмов обладнання та обмеженої місткості виробничих буферів.

Перспективи подальших досліджень пов'язані з реалізацією динамічного перепланування у режимі реального часу, паралельним виконанням симуляційних запусків, інтеграцією планувальника з цифровими двійниками виробничих систем та апробацією моделі на даних реального промислового підприємства.

Список бібліографічного опису

1. Yang, T., & Lai, S. (2024). Redefine manufacturing operations for modern production environments with the help of artificial intelligence enterprise information systems. *The International Journal of Advanced Manufacturing Technology*, 1-12. <https://doi.org/10.1007/s00170-024-14838-4>
2. Boj'ic, S., Maslaric, M., Mircetic, D., Nikolicic, S., & Todorovic, S. (2023). Simulation and Genetic Algorithm-based approach for multi-objective optimization of production planning: A case study in industry. *Advances in Production Engineering & Management*, 18(2), 250-262. <https://doi.org/10.14743/apem2023.2.471>
3. Xu, W., Sun, H. Y., Awaga, A. L., Yan, Y., & Cui, Y. J. (2022). Optimization approaches for solving production scheduling problem: A brief overview and a case study for hybrid flow shop using genetic algorithms. *Advances in Production Engineering & Management*, 17(1), 45-56. <https://doi.org/10.14743/apem2022.1.420>
4. De Felice, F., De Luca, C., Petrillo, A., Forcina, A., Ortiz Barrios, M. A., & Baffo, I. (2025). The role of digital transformation in manufacturing: Discrete event simulation to reshape industrial landscapes. *Applied Sciences*, 15(11), 6140. <https://doi.org/10.3390/app15116140>
5. Naderi, B., Ruiz, R., & Roshanaei, V. (2023). Mixed-integer programming vs. constraint programming for shop scheduling problems: new results and outlook. *INFORMS journal on computing*, 35(4), 817-843. <https://doi.org/10.1287/ijoc.2023.1287>
6. Schlenkrich, M., & Parragh, S. N. (2023). Solving large scale industrial production scheduling problems with complex constraints: an overview of the state-of-the-art. *Procedia Computer Science*, 217, 1028-1037. <https://doi.org/10.1016/j.procs.2022.12.301>
7. Kulkarni, A. J. (2025). Optimization Methods in Production Planning and Scheduling. In *Optimization Methods in Manufacturing Processes: A Machine-Generated Literature Overview* (pp. 195-279). Singapore: Springer Nature

Singapore. https://doi.org/10.1007/978-981-96-5257-0_4

8. Wallrath, R., Zondervan, E., & Franke, M. B. (2025). Integration of MILP and Discrete-Event Simulation for Flowshop Scheduling Using Benders Decomposition. *Systems and Control Transactions*, 936-941. <https://doi.org/10.69997/sct.180841>
9. Liu, Q., Wang, C., Li, X., & Gao, L. (2023). An improved genetic algorithm with modified critical path-based searching for integrated process planning and scheduling problem considering automated guided vehicle transportation task. *Journal of Manufacturing Systems*, 70, 127-136. [10.1016/j.jmsy.2023.07.004](https://doi.org/10.1016/j.jmsy.2023.07.004)
10. Torki, F. Z., Tigane, S., Kahloul, L., Hamani, N., Belaiche, L., & Benharzallah, S. (2026). Improving Reliability of Reconfigurable Manufacturing Systems using Genetic Algorithms: an NSGA-III-based approach for minimizing RMTs' failure in process planning. *International Journal of Computer Integrated Manufacturing*, 39(3), 357-382. <https://doi.org/10.1080/0951192X.2025.2550703>
11. Zonta, T., Da Costa, C. A., Zeiser, F. A., de Oliveira Ramos, G., Kunst, R., & da Rosa Righi, R. (2022). A predictive maintenance model for optimizing production schedule using deep neural networks. *Journal of Manufacturing Systems*, 62, 450-462. [10.1016/j.jmsy.2021.12.013](https://doi.org/10.1016/j.jmsy.2021.12.013)
12. Hubert, S., Meintschel, J., Bleidorn, D., Ortmanns, Y., & Wallrath, R. (2023). Production scheduling using deep reinforcement learning and discrete event simulation. *Chemie Ingenieur Technik*, 95(7), 1003-1011. <https://doi.org/10.1002/cite.202200242>
13. Zeng, Z., Wang, N., Xu, D., & Chen, R. (2025). Cascading failure modeling and resilience analysis of coupled centralized supply chain networks under hybrid loads. *Systems*, 13(9), 729. <https://doi.org/10.3390/systems13090729>
14. Pochelu, P. (2026). SerializableSimp: Parallel and Serializable Discrete-Event Simulation in Python. *Journal of Open Source Software*, 11(118), 8502. <https://doi.org/10.21105/joss.08502>
15. Rodríguez-Espinosa, C. A., González-Neira, E. M., & Zambrano-Rey, G. M. (2024). A simheuristic approach using the NSGA-II to solve a bi-objective stochastic flexible job shop problem. *Journal of Simulation*, 18(4), 646-670. <https://doi.org/10.1080/17477778.2023.2231877>
16. Coelho, D. G., Souza, M. J., & Cota, L. P. (2026). A simheuristic-based algorithm for the stochastic long-term maintenance scheduling problem. *International Transactions in Operational Research*, 33(1), 268-296. <https://doi.org/10.1111/itor.70021>
17. Álvarez, O. V., & López, P. P. R. (2023). Application of a discrete-event simulation model to improve the productivity of the production process in a manufacturing company. *Industrial Data*, 26(1), 303-332. <https://doi.org/10.15381/indata.v26i1.23717>
18. Elbasheer, M., Longo, F., Madden, M. G., Padovano, A., & Ullah, I. (2025). Towards a Digital Twin for Production Planning: Combining Discrete Event Simulation and ML for Flexible Job Shops. *Procedia Computer Science*, 253, 3078-3087. <https://doi.org/10.1016/j.procs.2025.02.032>

References

1. Yang, T., & Lai, S. (2024). Redefine manufacturing operations for modern production environments with the help of artificial intelligence enterprise information systems. *The International Journal of Advanced Manufacturing Technology*, 1-12. <https://doi.org/10.1007/s00170-024-14838-4>
2. Bojic, S., Maslaric, M., Mircetic, D., Nikolicic, S., & Todorovic, S. (2023). Simulation and Genetic Algorithm-based approach for multi-objective optimization of production planning: A case study in industry. *Advances in Production Engineering & Management*, 18(2), 250-262. <https://doi.org/10.14743/apem2023.2.471>
3. Xu, W., Sun, H. Y., Awaga, A. L., Yan, Y., & Cui, Y. J. (2022). Optimization approaches for solving production scheduling problem: A brief overview and a case study for hybrid flow shop using genetic algorithms. *Advances in Production Engineering & Management*, 17(1), 45-56. <https://doi.org/10.14743/apem2022.1.420>
4. De Felice, F., De Luca, C., Petrillo, A., Forcina, A., Ortiz Barrios, M. A., & Baffo, I. (2025). The role of digital transformation in manufacturing: Discrete event simulation to reshape industrial landscapes. *Applied Sciences*, 15(11), 6140. <https://doi.org/10.3390/app15116140>
5. Naderi, B., Ruiz, R., & Roshanaei, V. (2023). Mixed-integer programming vs. constraint programming for shop scheduling problems: new results and outlook. *INFORMS journal on computing*, 35(4), 817-843. <https://doi.org/10.1287/ijoc.2023.1287>
6. Schlenkrich, M., & Parragh, S. N. (2023). Solving large scale industrial production scheduling problems with complex constraints: an overview of the state-of-the-art. *Procedia Computer Science*, 217, 1028-1037. <https://doi.org/10.1016/j.procs.2022.12.301>
7. Kulkarni, A. J. (2025). Optimization Methods in Production Planning and Scheduling. In *Optimization Methods in Manufacturing Processes: A Machine-Generated Literature Overview* (pp. 195-279). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-96-5257-0_4
8. Wallrath, R., Zondervan, E., & Franke, M. B. (2025). Integration of MILP and Discrete-Event Simulation for Flowshop Scheduling Using Benders Decomposition. *Systems and Control Transactions*, 936-941. <https://doi.org/10.69997/sct.180841>
9. Liu, Q., Wang, C., Li, X., & Gao, L. (2023). An improved genetic algorithm with modified critical path-based searching for integrated process planning and scheduling problem considering automated guided vehicle transportation task. *Journal of Manufacturing Systems*, 70, 127-136. [10.1016/j.jmsy.2023.07.004](https://doi.org/10.1016/j.jmsy.2023.07.004)
10. Torki, F. Z., Tigane, S., Kahloul, L., Hamani, N., Belaiche, L., & Benharzallah, S. (2026). Improving Reliability of Reconfigurable Manufacturing Systems using Genetic Algorithms: an NSGA-III-based approach for minimizing RMTs' failure in process planning. *International Journal of Computer Integrated Manufacturing*, 39(3), 357-382. <https://doi.org/10.1080/0951192X.2025.2550703>
11. Zonta, T., Da Costa, C. A., Zeiser, F. A., de Oliveira Ramos, G., Kunst, R., & da Rosa Righi, R. (2022). A predictive maintenance model for optimizing production schedule using deep neural networks. *Journal of Manufacturing Systems*,

62, 450-462. [10.1016/j.jmsy.2021.12.013](https://doi.org/10.1016/j.jmsy.2021.12.013)

12. Hubert, S., Meintschel, J., Bleidorn, D., Ortmanns, Y., & Wallrath, R. (2023). Production scheduling using deep reinforcement learning and discrete event simulation. *Chemie Ingenieur Technik*, 95(7), 1003-1011. <https://doi.org/10.1002/cite.202200242>

13. Zeng, Z., Wang, N., Xu, D., & Chen, R. (2025). Cascading failure modeling and resilience analysis of coupled centralized supply chain networks under hybrid loads. *Systems*, 13(9), 729. <https://doi.org/10.3390/systems13090729>

14. Pochelu, P. (2026). SerializableSimpy: Parallel and Serializable Discrete-Event Simulation in Python. *Journal of Open Source Software*, 11(118), 8502. <https://doi.org/10.21105/joss.08502>

15. Rodríguez-Espinosa, C. A., González-Neira, E. M., & Zambrano-Rey, G. M. (2024). A simheuristic approach using the NSGA-II to solve a bi-objective stochastic flexible job shop problem. *Journal of Simulation*, 18(4), 646-670. <https://doi.org/10.1080/17477778.2023.2231877>

16. Coelho, D. G., Souza, M. J., & Cota, L. P. (2026). A simheuristic-based algorithm for the stochastic long-term maintenance scheduling problem. *International Transactions in Operational Research*, 33(1), 268-296. <https://doi.org/10.1111/itor.70021>

17. Álvarez, O. V., & López, P. P. R. (2023). Application of a discrete-event simulation model to improve the productivity of the production process in a manufacturing company. *Industrial Data*, 26(1), 303-332. <https://doi.org/10.15381/idata.v26i1.23717>

18. Elbasheer, M., Longo, F., Madden, M. G., Padovano, A., & Ullah, I. (2025). Towards a Digital Twin for Production Planning: Combining Discrete Event Simulation and ML for Flexible Job Shops. *Procedia Computer Science*, 253, 3078-3087. <https://doi.org/10.1016/j.procs.2025.02.032>

Історія статті:

Отримано: 29.05.2026 Доопрацьовано: 27.07.2026 Прийнято до друку: 23.09.2026 Опубліковано: 29.09.2026