

DOI: <https://doi.org/10.35681/1560-9189.2026.28.2.363158>

УДК 004.89

Є. М. Федорченко, А. О. Олійник, Т. В. Федорончак,
О. О. Степаненко, М. О. Чорнобук, Т. А. Зайко, Ю. В. Федорченко
Національний університет «Запорізька політехніка»
вул. Жуковського, 64, 69063 Запоріжжя, Україна

Модифікована інтерактивна система генетичних алгоритмів для генерації музики

Описано систему генерації музики на базі інтерактивного генетичного алгоритму (IGA). Її головна особливість — вміння відстежувати, наскільки користувачеві цікаво працювати з програмою. Система створює монофонічні MIDI-треки, спираючись на відгуки. Щоб людина не втомлювалася від нескінченного оцінювання, модель сама змінює частоту мутацій залежно від рівня її залученості. Це дійсно працює: під час тестів п'ятеро добровольців отримували приємні для них мелодії у середньому всього за 4,6 ітерації. При цьому нове покоління варіантів генерується практично миттєво — менш ніж за 1 мілісекунду. Програма має лаконічний інтерфейс на .NET MAUI та прямий експорт у MIDI. Завдяки такій швидкості та гнучким налаштуванням, інструмент добре підходить для генерації музики в реальному часі без зайвого навантаження на користувача.

Ключові слова: MIDI, генетичний алгоритм, генерація музики, машинне навчання.

Вступ

Генерацію музики в реальному часі вже давно використовують в іграх та інтерактивних додатках. Замість статичних треків такі системи створюють аудіо, яке безпосередньо підлаштовується під дії гравця або зміни в самому середовищі. Це робить звукове оформлення унікальним для кожної сесії та допомагає глибше зануритися в процес [1].

Методи штучного інтелекту здатні генерувати складні вокальні та інструментальні композиції за звичайним текстовим описом або заданим стилем. Проте ця можливість має високу обчислювальну ціну. Сучасні генеративні моделі вимагають спеціалізованої серверної інфраструктури. Їх практично неможливо запустити на звичайному пристрої користувача для роботи в реальному часі. Щоб вирішити цю проблему, необхідні альтернативні алгоритми. Вони мають формувати музичний контент локально і оперативно, без критичного навантаження на систему [2].

© Є. М. Федорченко, А. О. Олійник, Т. В. Федорончак,
О. О. Степаненко, М. О. Чорнобук, Т. А. Зайко, Ю. В. Федорченко

Інтерактивні генетичні алгоритми (ГА) вирішують цю проблему, використовуючи оцінки користувача як функцію пристосованості. У класичних методах критерій оптимізації задається математично, тоді як в ГА генерація музичного матеріалу керується безпосередньо суб'єктивною реакцією слухача. Це дозволяє створювати композиції на основі індивідуальних смаків, які складно або неможливо описати традиційними метриками [3–5].

Еволюційні методи застосовуються для генерації музики, проте класичні алгоритми мають суттєві недоліки: вони сходяться повільно, а користувачі швидко втомлюються від постійного оцінювання результатів. Вирішити цю проблему дозволяють адаптивні інтерактивні алгоритми. Вони динамічно коригують параметри мутації та ін'єкції, безпосередньо реагуючи на поточні відгуки слухача. Це прискорює оптимізацію і допомагає зберігати інтерес людини до процесу генерації [5].

Аналіз літературних даних і постановка проблеми

У дослідженні [6] розглянуто GenJam — одну із перших систем, що застосовує інтерактивний генетичний алгоритм для створення джазових соло. Програма працює у трьох режимах. Під час навчання користувач оцінює випадково згенеровані композиції, але еволюційний процес при цьому не запускається. Демонстраційний режим відтворює найкращі зі збережених результатів. В еволюційному режимі система безпосередньо використовує генетичні оператори, щоб динамічно формувати нові популяції мелодій.

GenJam застосовує дворівневу схему генетичного кодування композицій. Музичний матеріал ієрархічно поділяється на фрази та такти, які записуються у вигляді бінарних хромосом фіксованої довжини. Такий метод дозволяє безпосередньо поєднати ритм і висоту звуку. Наприклад, один такт розміру 4/4 кодується 32-бітною хромосомою. Вона містить вісім послідовних позицій для восьми нот, кожна з яких описується 4-бітною подією: паузою (Rest), утриманням (Hold) або новою нотою (New Note).

Попередня система має жорсткі обмеження: усі композиції генеруються з однаковим темпом (BPM) і базуються на заздалегідь визначених акордових послідовностях.

Значно більшу гнучкість демонструє архітектура GP-Music [8]. Її основа — деревоподібне подання музичних даних. Листя таких дерев містять базові елементи (окремі ноти, псевдоакорди або паузи), тоді як внутрішні вузли відповідають за музичні перетворення. У результаті кожен вузол формує локальну послідовність нот, а корінь дерева об'єднує їх у фінальний трек. Така ієрархія дозволяє напряму та без ускладнень застосовувати генетичні оператори мутації і кросоверу. Ще одна проблема інтерактивних систем — швидка втома людини. Для її вирішення автори GP-Music інтегрували просту неймережу. Вона навчається на початкових відгуках слухача, після чого починає самостійно оцінювати нові генерації. Це знімає рутинне навантаження з користувача. Цей метод, відомий як сурогатна фітнес-функція (Surrogate Fitness Function), потребує значних обчислювальних ресурсів для створення моделі, здатної достовірно імітувати оцінки користувача. Сама ж архітектура має суттєві обмеження, оскільки генерує виключно короткі монофонічні мелодії. Більше того, усі ноти в таких композиціях отримують однакову тривалість.

Через це алгоритм не здатен формувати складні ритмічні малюнки, притаманні повноцінній музиці.

Альтернативний підхід наведено в дослідженні [7], автори якого розробили систему DarwinTunes. Замість сурогатної моделі вони застосували масовий збір даних: алгоритм агрегував відгуки понад 6000 слухачів. Саме ці усереднені оцінки працювали як фітнес-функція для інтерактивного генетичного алгоритму. Еволюційний процес оперував популяцією деревоподібних цифрових геномів. Кожен такий геном діяв як локальна програма, що генерувала короткий восьмисекундний музичний цикл.

Для технічного представлення подібних композицій зазвичай використовують музичний цифровий інтерфейс (MIDI). Запроваджений у 1983 році, цей стандарт досі уніфікує зв'язок між електронним обладнанням від різних виробників. По суті, MIDI створює рівень абстракції: він не передає сам звук, а лише цифрові команди, відокремлюючи керування від безпосередніх аналогових чи цифрових процесів генерації аудіо.

Формат Standard MIDI File (SMF) забезпечує компактне збереження музичних даних. Замість запису безпосереднього аудіосигналу він фіксує послідовності дискретних подій, які програмні або апаратні інструменти потім перетворюють на звук. Структурно такий файл містить заголовок із глобальними налаштуваннями (зокрема темпом) і один або кілька треків. Події всередині треку впорядковані хронологічно. Кожна з них отримує часову мітку відносно попередньої, що дозволяє максимально точно позиціонувати ноти у часі.

Основою керування є події Note On та Note Off, які позначають початок і кінець звучання. Кожну ноту визначають чіткі числові характеристики: канал (0–15), висота (0–127) та інтенсивність (0–127). Цих параметрів цілком достатньо, щоб повністю описати ритмічну та звуковисотну структуру мелодії [9].

Ми обрали формат MIDI за основу нашої системи передусім через його простоту та універсальність. Оскільки розроблений алгоритм генерує базові монофонічні композиції і не потребує динамічної зміни гучності, формальний опис музичної послідовності s можна суттєво спростити та звести до такого вигляду:

$$s = (BPM, \{e_1, e_2, \dots, e_N\}), \quad (1)$$

$$e_i = (p_i, d_i) \text{ або } e_i = (\emptyset, d_i), \quad (2)$$

$$p_i \in P, d_i \in D, \quad (3)$$

де P — множина дозволених значень висоти ноти; D — множина дозволених значень довжини ноти.

Тоді формально задача розробки системи генерації музики зводиться до знаходження такої функції M :

$$S_{n+1} = M(S_n, Y_n), \quad (4)$$

$$Y_n = \{y_1, y_2, \dots, y_x\}, \quad (5)$$

$$S_n = \{s_1, s_2, \dots, s_x\}, \quad (6)$$

$$s \in T, \quad (7)$$

де y_i — оцінка користувача для i -ї композиції n -го покоління; $y_i \in [0,1]$; x — розмір покоління. S_N — n -не покоління; T — множина всіх можливих музичних композицій з урахуванням обмежень системи.

Найскладнішим викликом під час розробки системи є знаходження такої функції M , що здатна швидко генерувати композиції, які отримують високу оцінку від користувача.

Розробка модифікованого генетичного алгоритму

На базі модифікованого інтерактивного генетичного алгоритму розроблено систему, що генерує та відтворює короткі MIDI-композиції. Після того як алгоритм формує нове покоління S_n , користувач оцінює кожну з мелодій за 10-бальною шкалою. Отримані бали Y_n безпосередньо слугують фітнес-функцією для подальшої еволюції.

Для кодування композицій система використовує геном із трьох типів параметрів: якісних (гама, тоніка), кількісних (темп) і секвенційних (гармонічна й ритмічна послідовності, типи арпеджіо в тактах). На відміну від системи GenJam, такого набору даних достатньо для створення різножанрової музики. Архітектура також передбачає безпосереднє налаштування генерації: дослідник може вручну задавати межі для кількісних і можливі стани для дискретних змінних.

Система має наступний робочий цикл.

1. Згенерувати початкову популяцію S_1 випадкових музичних композицій (осіб). Значення x (розмір покоління) встановлено у 5 емпіричним шляхом
2. Продемонструвати користувачу поточну популяцію. Отримати значення оцінки y_i для кожної особи s_i від користувача.
3. Оцінити зацікавленість користувача δ
4. Згенерувати нову популяцію з x осіб. Кожна з осіб популяції S_N буде нащадком двох осіб з попередньої популяції S_{N-1} з вірогідністю $1 - \alpha$ або результатом ін'єкції з вірогідністю α
5. Провести мутації в новій популяції S_N . Вірогідність мутації кожного гену — β
6. Повернутися до кроку 2

Під час схрещування нащадок отримує унікальну комбінацію ознак. Формування кожного гена відбувається незалежно, а механізм успадкування спирається на тип параметра. Для кількісних характеристик алгоритм обчислює випадкове число в діапазоні між показниками батьків. Натомість дискретні параметри напряму переймають один із батьківських станів. Секвенційні ж гени утворюються шляхом випадкового злиття відповідних батьківських послідовностей. Самі особини для розмноження відбираються за таким принципом:

$$P(i) = \frac{y_i}{\sum_{j=0}^{n-1} y_j}, \quad (8)$$

де $P(i)$ — вірогідність вибору i -ї особи у якості батьківської, y_i — i -та оцінка, отримана від користувача, $y_i \in [0,1]$.

Модифікована система адаптує ймовірність ін'єкцій та мутацій відповідно до рівня зацікавленості користувача. Цей механізм необхідний для генерації музики,

оскільки в цій предметній області неможливо формально виміряти якість результату. На основі оцінок слухача алгоритм безпосередньо керує інтенсивністю генетичних змін: зі зростанням інтересу до процесу пропорційно підвищується і різноманітність створюваних композицій.

Параметр δ приймає значення від 0 до 1 та розраховується наступним чином:

$$\delta = \frac{\sigma}{0.5} * r_{\max}, \quad (9)$$

$$r_{\max} = \max(r_1, r_2, \dots, r_N), \quad (10)$$

де r_i — i -та оцінка, отримана від користувача, в діапазоні 0–1; σ — середнє квадратичне відхилення значень r .

Вірогідність ін'єкції розраховується наступним чином:

$$\alpha = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) * (1 - \delta), \quad (11)$$

α_{\max} — максимальна та α_{\min} — мінімальна вірогідності ін'єкції, що задаються як параметри системи.

Вірогідність мутації розраховується подібним чином:

$$\beta = \beta_{\min} + (\beta_{\max} - \beta_{\min}) * (1 - \delta), \quad (12)$$

де β_{\max} — максимальна та β_{\min} — мінімальна вірогідності мутації, що задаються як параметри системи.

Кросинговер реалізовано наступним чином:

— для якісних генів:

$$g_k^{(child)} = \begin{cases} g_k^{(parent1)} \\ g_k^{(parent2)} \end{cases}; \quad (13)$$

— для кількісних генів:

$$g_k^{(child)} = g_k^{(parent1)} + (g_k^{(parent2)} - g_k^{(parent1)}) * U(0,1); \quad (14)$$

— для секвенційних генів:

$$L_{child} = U(\min(L_1, L_2), \max(L_1, L_2)), \quad (15)$$

$$g_{k,i}^{(child)} = \begin{cases} g_{k,i}^{(parent1)}, \text{ якщо } i \leq L_1 \text{ і } (U(0,1) < 0.5 \text{ або } i > L_2) \\ g_{k,i}^{(parent2)}, \text{ якщо } i \leq L_2 \text{ і } (U(0,1) < 0.5 \text{ або } i > L_1, \end{cases} \quad (16)$$

де g_k — значення гену k для згенерованої особи; $g_k^{(parent1)}$ та $g_k^{(parent2)}$ — значення гену k для першої і другої батьківської осіб відповідно; $U(x, y)$ — це випадкова величина з рівномірним розподілом на інтервалі $[x, y]$; L_{child} — довжина згенерованого секвенційного гену; L_1, L_2 — довжини цього гену для першої та другої батьківської особи відповідно; $g_{k,i}$ — i -й елемент k -го секвенційного гену.

Цей метод кросинговеру враховує специфіку кожного типу генів. Наприклад, темп (ВРМ) нащадка отримує значення в діапазоні між показниками батьків. Вод-

ночас гармонічна послідовність створюється шляхом безпосереднього злиття батьківських акордів.

Розроблене програмне забезпечення

Інтерфейс користувача реалізовано на базі кросплатформної бібліотеки .NET MAUI, що забезпечує роботу системи на операційних системах Windows та macOS [10].

Користувач може багаторазово та в довільному порядку прослуховувати й оцінювати композиції поточного покоління. Будь-який згенерований трек можна зберегти у форматі .mid. Розрахований рівень зацікавленості відображається на спеціальному індикаторі в нижній частині вікна. Крім того, для подальшого аналізу передбачено експорт загальної статистики, яка містить усі виставлені оцінки для кожного з поколінь.

Експерименти

Для тестування системи ми залучили п'ятьох добровольців. Їхнім завданням було взаємодіяти з алгоритмом, аж поки він не згенерує суб'єктивно приємну композицію.

На рис. 1 зображено приклад мелодії з першого покоління, яка отримала найнижчу оцінку. Вона звучить як хаотичний набір нот і повністю позбавлена музичної привабливості. Натомість рис. 2 демонструє результат уже п'ятої ітерації — цей трек користувач оцінив максимальним балом. Перша композиція містить неприродні переходи до шістнадцятих нот та невинуватих пауз. Їй бракує гармонійного розвитку, а в середині треку виникають монотонні повторення однієї ноти. Натомість друга мелодія демонструє чіткий гармонійний малюнок із чергуванням висотних підйомів та спадів. Алгоритм більше не генерує хаотичних коротких звуків чи механічного дублювання. На їхньому місці з'являються базові, але дієві прийоми створення музичної напруги та її розв'язання. Завдяки такій структурній цілісності цей варіант звучить значно природніше.



Рис. 1. Приклад суб'єктивно непривабливої композиції



Рис. 2. Приклад суб'єктивно більш привабливої композиції

На рис. 3 відображено динаміку максимальних оцінок у кожному поколінні, отриманих під час тестування за участю п'ятьох добровольців.

Як видно з результатів, головним недоліком алгоритму залишається певна непередбачуваність генерації. Попри це, модель здатна швидко сходиться до вдалого

рішення: найдовший пошук прийнятної композиції зайняв у користувачів 9 ітерацій, тоді як середній показник становить лише 4,6 ітерації.

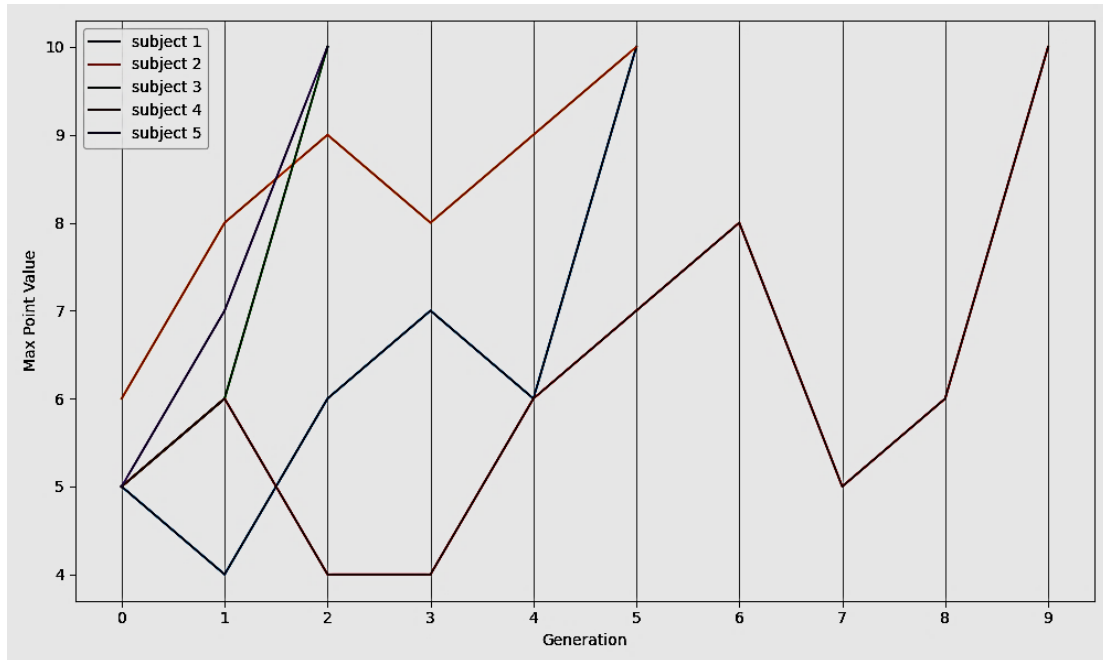


Рис. 3. Результати взаємодії 5 користувачів з системою

Тестування розробленої програмної системи проводилося на комп'ютері з операційною системою Windows 11, обладнаному центральним процесором Intel I7-12650H. У табл. 1 та на рис. 4 наведено результати тестування упродовж 10 поколінь. За результатами тестів встановлено, що середній час для генерації наступного покоління на даному комп'ютері складає менше 1 мілісекунди, що вказує на високу швидкість системи та невелику витрату ресурсів. Час, що витрачається на генерацію наступного покоління, є непомітним для реальних користувачів системи.

Таблиця 1. Результати тестування швидкості системи

Номер покоління	Витрачений час, мілісекунди
1	2,3868
2	0,3068
3	0,1194
4	0,119
5	0,0923
6	0,0747
7	0,0646
8	0,1081
9	0,0931
10	0,1471

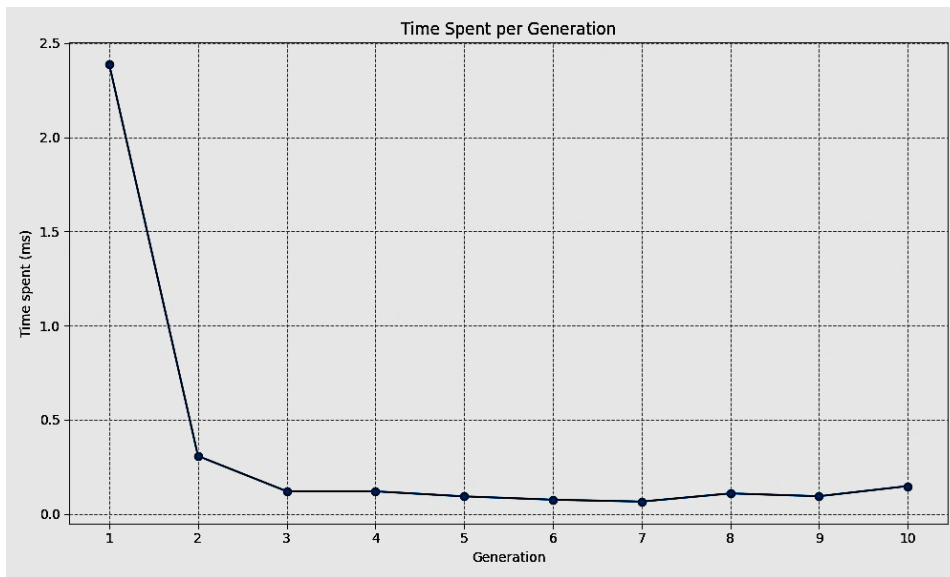


Рис. 4. Результати взаємодії 5 користувачів з системою

Обговорення результатів

У табл. 2 наведено порівняння розробленої системи з розглянутими вище аналогами.

Таблиця 2. Порівняння розробленої системи з аналогами

Система	Тип музики	Спосіб кодування композицій	Сурогатна фітнес-функція	Динамічне відстеження зацікавленості користувача	Лише монофонічна музика
Запропонована система	Довільні композиції	Складний геном, гени трьох типів	Ні	Так	Так
GenJam [6]	Джазові соло	Дворівнева ієрархія, обмежена кількість 4-бітних подій на такт	Ні	Ні	Так
GP-Music [8]	Компактні композиції, лише однакова довжина нот	Деревоподібна багаторівнева структура	Так	Ні	Так
DarwinTunes [7]	Компактні зациклені композиції	Деревоподібна багаторівнева структура	Ні	Ні	Так

Дана версія алгоритму створює лише монофонічні мелодії. Проте такого формату достатньо для фонового звуку в іграх, освітніх програмах чи арт-інсталяціях. Водночас система має технічні переваги, що вирізняють її з-поміж існуючих рішень.

Динамічне відстеження зацікавленості допомагає подолати втому користувача — критичну проблему інтерактивних генетичних алгоритмів. Алгоритм пос-

тійно коригує ймовірність мутацій та ін'єкцій, що скорочує кількість зайвих ітерацій. Це дає змогу отримати потрібний результат за меншу кількість кроків. Такий підхід є ефективною альтернативою сурогатним фітнес-функціям, оскільки він не потребує значних обчислювальних потужностей і легше підлаштовується під індивідуальні вподобання різних людей.

Обчислювальна ефективність моделі робить її практичною для застосування в режимі реального часу. Тести підтверджують, що нові покоління можуть бути створені менш ніж за одну мілісекунду, навіть на дешевому обладнанні. Таким чином, система може використовуватися з програмним забезпеченням, де продуктивність є критично важливою, без необхідності хмарної обробки або дорогого обладнання. Наприклад, у відеоіграх або інтерактивних виставах у реальному часі.

Користувач може самостійно задавати темп, гаму та гармонійні послідовності. Це дозволяє створювати музику в межах обраного жанру. На відміну від GenJam чи DarwinTunes, де музична форма має жорсткі рамки, запропонований підхід залишає вибір основних параметрів за людиною.

Хоча обрана структура геному обмежує вихідні дані монофонією, система працює швидко та адаптується до вподобань користувача для запобігання його втомі. Поєднання низької ресурсомісткості та інтерактивного контролю робить модель придатною для застосування в системах генерації музики в реальному часі.

Висновки

У межах дослідження проаналізовано наявні підходи та системи генерації музики за допомогою генетичних алгоритмів. Окрему увагу приділено рішенням на базі інтерактивних методів, а також використанню сурогатних фітнес-функцій для керування еволюційним процесом.

Було запропоновано систему, що генерує музику за допомогою модифікованого інтерактивного генетичного алгоритму, а також використовує динамічне відслідковування зацікавленості користувачів. Запропонована система також відрізняється від аналогів здатністю генерувати музику в різних стилях, а також більші тонкими налаштуваннями, що доступні користувачам.

Подальший розвиток системи передбачає впровадження сурогатної фітнес-функції за аналогією з підходом GP-Music [8].

1. Hutchings P.E., McCormack J. Adaptive music composition for games. *IEEE Transactions on Games*. 2020. Vol. 12, No. 3. P. 270–280. URL: <https://research.monash.edu/en/publications/adaptive-music-composition-for-games/> (Last accesses: 08.04.2026).
2. Biles J.A. GenJam: A genetic algorithm for generating jazz solos. *International Conference on Mathematics and Computing*. 1994. URL: https://www.researchgate.net/publication/2342018_GenJam_A_Genetic_Algorithm_for_Generating_Jazz_Solos (Last accesses: 06.04.2026).
3. Biles J.A. Interactive GenJam: Integrating real-time performance with a genetic algorithm. URL: https://www.academia.edu/52228995/Interactive_GenJam_Integrating_real_time_performance_with_a_genetic_algorithm (Last accesses: 08.04.2026).
4. Biles J.A. A tentative taxonomy for genetic algorithm music and art systems. URL: <https://genjam.org/wp-content/uploads/2019/07/bilesgecco2000.pdf> (Last accesses: 08.04.2026).
5. Evolution of music by public choice. *PLoS One*. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC3409751/> (Last accesses: 08.04.2026).

6. MacCallum R.M., Mauch M., Burt A., and Leroi A.M. Evolution of music by public choice. *Proc. Nat. Acad. Sci.* Jun. 2012. Vol. 109, No. 30. P. 12081–12086. doi:10.1073/pnas.1203182109
7. Johanson B., and Poli R. GP-Music: An interactive genetic programming system for music generation with automated fitness raters. In *Genetic Program. 1998: Proc. Third Annu. Conf. San Francisco: Morgan Kaufmann, 1998.* P. 181–186.
8. Midi. MIDI. Accessed: Aug. 12, 2025. Available: <https://midi.org/>
9. .NET Multi-platform App UI documentation — .NET MAUI. Microsoft Learn. Accessed: Aug. 12, 2025. Available: <https://learn.microsoft.com/en-us/dotnet/maui>
10. Afifie N.A et al. Evaluation Method of Mesh Protocol over ESP32 and ESP8266. *Baghdad Sci. J.* Dec. 2021. Vol. 18, No. 4(Suppl.). P. 1397. doi:10.21123/bsj.2021.18.4(suppl.).1397.
11. Fedorchenko I., Oliinyk A., Stepanenko A., Zaiko T., Shylo S., and Svyrydenko A. Development of the modified methods to train a neural network to solve the task on recognition of road users. *Eastern-Eur. J. Enterprise Technol.* Apr. 2019. Vol. 2, No. 9(98). P. 46–55. Accessed: Aug. 20, 2025. doi:10.15587/1729-4061.2019.164789.
12. I. Fedorchenko, M. F. Yusof, A. Oliinyk, M. Chornobuk, M. Khokhlov, and J. A. J. Alsayaydeh. Detection of fungal diseases of plants from leaf images based on neural network technologies. *Int. J. Elect. Comput. Eng. (IJECE).* Oct. 2024. Vol. 14, No. 5. P. 5866. doi:10.11591/ijece.v14i5.pp5866-5873.

Надійшла до редакції 12.04.2026

Прийнята до друку 19.05.2026

Опублікована 17.06.2026