

## Intelligent information systems

УДК 004.02

doi: 10.20998/2522-9052.2021.1.14

А. В. Казмірчук, О. Г. Жданова, В. Д. Попенко, М. О. Сперкач

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

### РОЗВ'ЯЗАННЯ БАГАТОКРИТЕРІАЛЬНОЇ ЗАДАЧІ СКЛАДАННЯ РОЗКЛАДУ З ВИКОРИСТАННЯМ ГЕНЕТИЧНОГО АЛГОРИТМУ

**Анотація.** Робота присвячена багатокритеріальній задачі складання розкладу, в якій задана множина робіт повинна бути виконана декількома виконавцями різної продуктивності. За виконання роботи відповідному виконавцю нараховується певна кількість бонусів, яка залежить від часу виконання роботи. Критеріями оцінки розкладу є загальний час виконання усіх робіт та кількість витрачених бонусів. В ході проведених досліджень були проаналізовані основні підходи до розв'язання задач багатокритеріальної оптимізації, на основі яких обрано підхід Парето. В якості алгоритму було обрано генетичний алгоритм. **Метою даної роботи** є підвищення ефективності розв'язання задач багатокритеріальної оптимізації шляхом реалізації евристичного алгоритму та підвищення його швидкодії. **Завданнями роботи** є визначення переваг та недоліків підходів, що використовуються для розв'язання задач багатокритеріальної оптимізації, розробка генетичного алгоритму для розв'язання багатокритеріальної задачі складання розкладу та дослідження його ефективності. Розроблені оператори генетичного алгоритму, які враховують особливості досліджуваної задачі, та дозволяють отримувати в процесі роботи розв'язки Парето. Завдяки впровадженню паралельних обчислень в реалізації генетичного алгоритму вдалося підвищити його швидкодію в порівнянні із звичайною версією. Розроблений алгоритм може бути використано при вирішенні задачі оптимального розподілу ресурсів, що є частиною системи нарахування бонусів співробітникам.

**Ключові слова:** задача складання розкладу; багатокритеріальна оптимізація; евристика; генетичний алгоритм; фронт Парето.

#### Вступ

Сьогодні процеси оптимізації та автоматизації є невід'ємною складовою нашого життя. Кожного дня ми займаємося плануванням наших ресурсів таким чином, щоб за оптимальних витрат отримати бажані результати. Такими основними ресурсами, наприклад, виступають час та гроші. У реальному житті ми стикаємося із ситуаціями, коли необхідно оптимізувати декілька ресурсів одночасно, що вже є складнішою задачею. Так, наприклад, при плануванні робіт на підприємстві необхідно досягти максимальної ефективності роботи при мінімальних витратах ресурсів. Такі задачі відносяться до класу задач багатокритеріальної оптимізації.

Дана задача активно досліджується вченими всього світу. У роботі [1] наведено короткий огляд багатокритеріальної оптимізації з використанням метаевристики. Представлено опис всіх використаних алгоритмів. Не дивлячись на те, що основний наголос в даній роботі робиться на еволюційні алгоритми, такі метаевристики як оптимізація рою частинок, штучна імунна система та оптимізація мурашиних колон також досліджуються. У статті [2] аналізуються методи багатокритеріальної оптимізації, наводиться їх формулювання, переваги та недоліки. Ефективність обраних методів для цілей інженерного проектування визначається шляхом порівняння результатів, отриманих при рішенні тестових задач та реальної конструкторської інженерної задачі.

Робота [3] демонструє рішення задачі багатокритеріальної оптимізації за допомогою розробленої модифікації еволюційного алгоритму SPEA2.

Як видно з практики, евристичні методи широко застосовуються та мають переваги над точними. Одним із найпопулярніших представників евристичних методів є генетичний алгоритм.

**Мета та завдання роботи.** Метою даної роботи є підвищення ефективності розв'язання задач багатокритеріальної оптимізації шляхом реалізації евристичного алгоритму та підвищення його швидкодії. Завданнями роботи є визначення переваг та недоліків підходів, що використовуються для розв'язання задач багатокритеріальної оптимізації, розробка генетичного алгоритму для розв'язання багатокритеріальної задачі складання розкладу та дослідження його ефективності.

#### Постановка задачі

Нехай задано множину робіт  $J = \{1, 2, \dots, j, \dots, n\}$  і множину виконавців  $M = \{1, 2, \dots, i, \dots, m\}$ . Кожна робота може виконуватися будь-яким виконавцем. Передбачається, що виконавці мають різну швидкість виконання робіт  $v_i$ ,  $v_i > 0$ ,  $i = \overline{1, m}$ . Для  $j$ -ої роботи задано час виконання  $p_j$  та кількість бонусів  $b_j > 0$ ,  $j = \overline{1, n}$ . Якщо  $v_i > 0$  – швидкість  $i$ -го виконавця, то тривалість виконання  $j$ -ої роботи  $i$ -м виконавцем становить:

$$p_{ij} = p_j / v_i, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Кількість бонусів, нарахованих  $i$ -му виконавцю за  $j$ -ту роботу, залежить від швидкості роботи цього виконавця і обчислюється за формулою:

$$b_{ij} = b_j / v_i, \quad i = \overline{1, m}, \quad j = \overline{1, n}.$$

Загальну суму нарахованих бонусів за увесь обсяг робіт отримуємо з:

$$B = \sum_{i=1}^m \sum_{j=1}^n a_{ij} b_{ij} = \sum_{i=1}^m \frac{1}{v_i} \sum_{j=1}^n a_{ij} b_j,$$

де  $a_{ij} \in \{0, 1\}$ ,  $a_{ij} = 1$ , якщо  $j$ -та робота призначена  $i$ -му виконавцю.

Нехай маємо деякий розклад виконання робіт. Позначимо через  $C_{ij}$  – момент завершення  $j$ -ї роботи  $i$ -м виконавцем. Загальний час виконання робіт визначається таким чином:

$$C_{\max} = \max_{ij} \{C_{ij}\}.$$

Необхідно скласти розклад виконання робіт, при якому загальний час завершення робіт множини  $J$  буде мінімальним, і при цьому загальна кількість нарахованих бонусів також буде мінімальною:

$$C_{\max} \rightarrow \min, \quad (1)$$

$$B \rightarrow \min. \quad (2)$$

### Методи розв'язання задач багатокритеріальної оптимізації

Пошук одного чи більше оптимальних розв'язків називається багатокритеріальною оптимізацією, а задача називається задачею багатокритеріальної оптимізації [4-6]. Опис підходів до вирішення задач багатокритеріальної оптимізації наведено далі [7].

**Підхід з використанням згортки.** Цей підхід базується на перетворенні задачі багатокритеріальної оптимізації в задачу однокритеріальної оптимізації за допомогою скаляризації векторного критерію в один узагальнений, при цьому ваги кожного критерію визначаються користувачем. Це свого роду процес багатокритеріальної оптимізації, що базується на вподобаннях. Як тільки цільові функції (критерії) були визначені, цей метод призначає числове значення ваг кожній цільовій функції та потім комбінує значення зважених критеріїв в одну цільову функцію шляхом додавання або множення всіх зважених критеріїв, перетворюючи таким чином багатокритеріальну задачу оптимізації в однокритеріальну.

Переваги:

- простий для розуміння та використання.

Недоліки:

- вибір значень ваг є суб'єктивним та індивідуальним;

- втрачається можливість знайти інші корисні розв'язки, маючи кращий компроміс між критеріями;

- різні критерії з різними одиницями виміру вимагають відповідного процесу нормалізації, який часто є суб'єктивним;

- має проблему поєднання несумірних критеріїв;

- необхідність в декількох запусках алгоритму, заснованого на даному підході, для того, щоб отримати декілька компромісних розв'язків багатокритеріальної задачі.

**Лексикографічний підхід.** В його основу покладено розташування критеріїв в порядку пріоритетів. Це знову ж таки підхід, що базується на вподобаннях. Цей підхід працює наступним чином. Різні пріоритети призначаються різним критеріям, а потім критерії оптимізуються в порядку їх пріоритетів. Таким чином, коли два чи більше розв'язків порівнюються один з одним для того, щоб обрати найкращий, то їх показники ефективності порівнюються за критерієм, що має найвищий пріоритет. Якщо один розв'язок-кандидат краще іншого відповідно до цього критерію, то обирається саме він. Інакше, показники ефективності двох можливих розв'язків порівнюються відносно другого критерію. І знову ж, якщо розв'язок є кращим за другим критерієм, то він і обирається. Цей процес повторюється доти, поки не буде знайдено явного переможця або ж поки не будуть розглянуті всі критерії. В останньому випадку, якщо немає явного розв'язку-переможця, то можна обрати розв'язок, що оптимізує найбільш пріоритетний критерій.

Переваги:

- уникає проблеми поєднання несумірних критеріїв в одній формулі, розглядає кожен критерій окремо;

- володіє концептуальною простотою та зручністю використання.

Недоліки:

- визначення порогу допуску та відповідного ступеня важливості для кожного критерію суб'єктивне та довільне;

- послідовний за своєю природою.

**Підхід Парето.** Замість того, щоб перетворювати багатокритеріальну задачу оптимізації в однокритеріальну і потім вирішувати її за допомогою однокритеріального оптимізатора, використовується алгоритм багатокритеріальної оптимізації для вирішення оригінальної задачі. Замість одного оптимального розв'язку, як в задачах однокритеріальної оптимізації, даний підхід знаходить всі можливі оптимальні розв'язки і повертає набір недомінуючих оптимальних розв'язків, що називаються оптимальними розв'язками по Парето, або фронт Парето.

Переваги:

- ніколи не комбінує різні критерії в одній формулі, всі критерії розглядаються окремо;

- справляється із несумірними критеріями;

- результат у вигляді набору домінуючих розв'язків представлений у формі фронту Парето, що залишає можливість вибору користувачу підходящого оптимального розв'язку;

- уникає багаторазового запуску алгоритму.

Недоліки:

- набагато складніший, ніж інші два підходи.

Враховуючи переваги та недоліки всіх вищевказаних підходів, будемо використовувати підхід Парето.

### Опис генетичного алгоритму

В якості алгоритму для багатокритеріальної оптимізації було обрано генетичний алгоритм. Важливою перевагою генетичних алгоритмів є те, що вони

з самого початку працюють з великою популяцією кандидатів на розв'язки, і, відповідно, оптимальні по Парето розв'язки генеруються вже в першому поколінні. Таким чином, не зважаючи на складність поставленої задачі, можна отримати достатньо повну множину оптимальних по Парето розв'язків уже після першої ітерації процедури генетичного алгоритму при відносно невеликих затратах часу на обчислення [8].

Модифікація генетичного алгоритму для вирішення багатокритеріальної задачі оптимізації відрізняється від звичайної схеми оператором селекції, визначенням кращих особин популяції, обчисленням значення пристосованості особин та застосуванням підходу Парето. Отже, для кожної особини популяції (розкладу) обчислюються окремо значення кожного з двох критеріїв.

Після обчислення значень критеріїв застосовується підхід Парето. Для кожної особини популяції визначається ранг границі Парето. Особини (розклади), що знаходяться безпосередньо на границі мають ранг 1. Якщо ми видалимо ці особини із популяції та визначимо нову границю, то особини нової границі

матимуть ранг 2 і так далі. Отже, ранги визначаються таким чином: додаємо особину до границі, якщо вона не домінується іншими особинами, що вже перебувають на границі та видаляємо із границі всі особини, що домінуються новою особиною. Також для всіх особин кожного рангу обчислюється відстань до сусідніх особин (найближчих зліва та справа за значенням критерію (1)). Тоді для селекції особин поточної популяції та їх нащадків для створення нового покоління необхідно впорядкувати всіх особин по зростанню рангу і, після цього, по спаданню відстані до сусідніх особин. Таким чином до нового покоління потрапляють перші  $N$  особин. Відстань до сусідніх особин одного рангу необхідно обчислювати для того, щоб уникнути збіжності до локального мінімуму та зберегти різноманітність осіб популяції. Так, наприклад, позначені на рис. 1 особини мають в 2 рази більше шансів бути обраними в якості батьків для створення нащадків, що може призвести до хибних результатів.

Тому для наступного покоління обираються лише особини з великою відстанню до сусідніх особин.

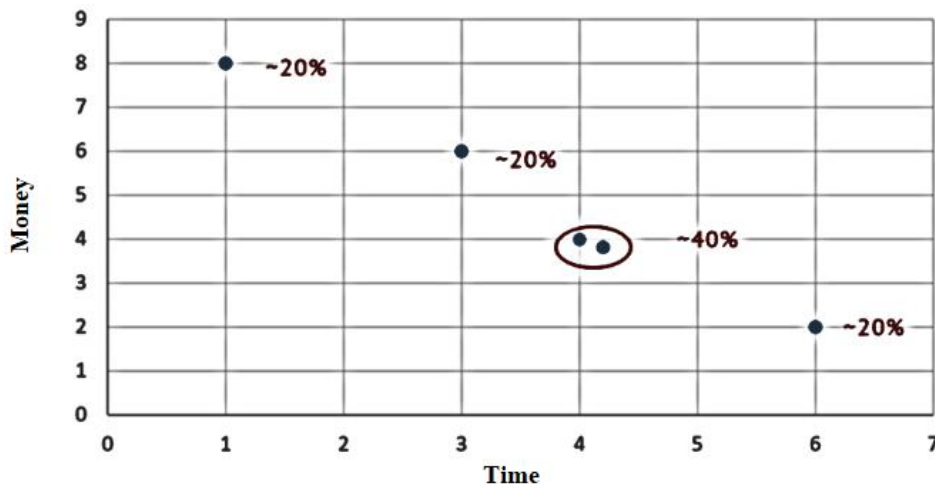


Рис. 1. Ілюстрація ймовірності селекції особин для створення нащадків  
(Fig. 1. Illustration of the probability of selection of individuals to create offspring)

Схема роботи генетичного алгоритму для розв'язання задачі багатокритеріальної оптимізації наведено нижче.

*Крок 1.* Згенерувати початкову популяцію.

*Крок 2.* Обробити популяцію.

*Крок 2.1.* Створити нащадків, використовуючи турнірний метод для вибору батьків.

*Крок 2.2.* Додати до поточного покоління всіх нащадків та нормалізувати значення фітнес-функцій – загальний час завершення робіт та загальна сума бонусів.

*Крок 2.3.* Визначити ранги особин відповідно до рівнів домінування.

*Крок 2.3.1.* Задати початкове значення рангу для всіх особин  $r_i = -1, i = \overline{1, N}$  та додати до множини перевірки на домінантність всіх особин популяції.

*Крок 2.3.2.* Встановити початкове значення рангу  $R = 1$ .

*Крок 2.3.3.* Для кожної особини множини перевірки на домінантність перевіряємо чи є вона недомінуючою, тобто кращою хоча б по одному критерію та не гіршою за іншими критеріями відносно інших особин із множини перевірки на домінантність.

*Крок 2.3.4.* Для всіх особин, що є недомінуючими, присвоїти  $r_i = R$  та видалити їх із множини для перевірки на домінантність.

*Крок 2.3.5.* Збільшити значення рангу  $R+ = 1$ .

*Крок 2.3.5.* Якщо множини перевірки на домінантність не порожня, то перейти на КРОК 2.3.3, інакше перейти на КРОК 2.4.

*Крок 2.4.* Для особин кожного рангу обчислити відстань до сусідніх особин.

*Крок 2.4.1.* Відстань обчислюється як сума відстаней до лівої та правої сусідньої особини. Для першої та останньої особини рангу відстань рівна нескінченності.

**Крок 2.5.** Впорядкувати всіх особин популяції спочатку по рангу, а потім по спаданню відстаней між особами в межах кожного рангу.

**Крок 2.6.** Створити нову популяцію, що складається з перших  $N$  особин впорядкованої множини.

**Крок 2.7.** Якщо не досягнуто максимальної кількості популяцій, то перейти на КРОК 2, інакше перейти на КРОК 3.

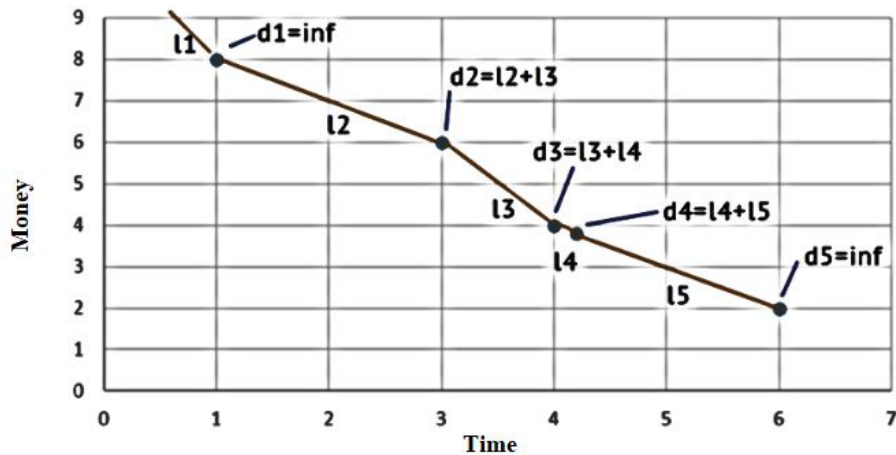
**Крок 3.** Визначити множину рішень як множину особин першого рангу популяції – Парето фронт.

На рис. 2 проілюстровано приклад визначення відстаней між особинами популяції одного рангу. Тут позначено:

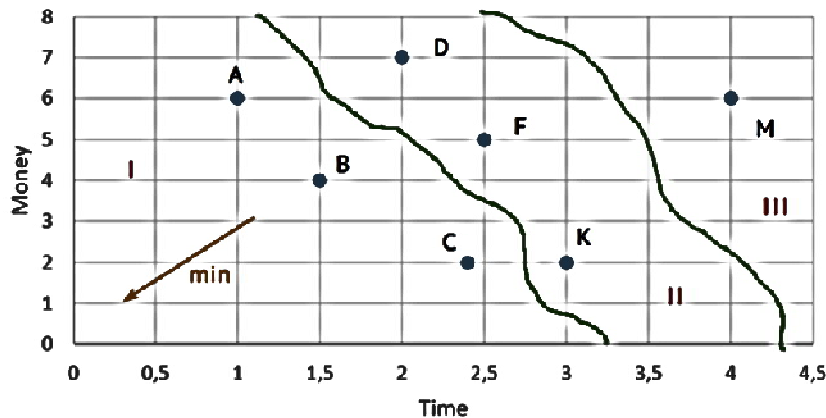
$l_1, \dots, l_5$  – відстані між двома особинами,

$d_1, \dots, d_5$  – сума відстаней від особини до сусідніх особин зліва та справа, при цьому для крайніх точок (особин) відстань до сусідніх особин визначається як  $\text{inf}$  (дуже велике число).

На рис. 3 наведено приклад визначення рангів для двокритеріальної задачі мінімізації. Точки A, B, C із області I не домінуються точками із I та II областей, тобто вони є кращими хоча б по одному із критеріїв та не гіршими по інших критеріях. Аналогічно, точки D, F, K із області II не домінуються точкою M із області III. Таким чином, точки A, B, C належать до рангу 1 і складають множину оптимальних розв'язків по Парето. Точки D, F, K належать до рангу 2. Точка M належить до рангу 3.



**Рис. 2.** Відстані між особинами одного рангу  
(Fig. 2. Distances between individuals of the same rank)



**Рис. 3.** Ранги відповідно до рівнів домінантності для двокритеріальної задачі мінімізації  
(Fig. 3. Ranks according to the levels of dominance for the two-criteria minimization problem)

Для дослідження ефективності роботи алгоритму було розроблено його програмну реалізацію. Так, на рис. 4-6 відображено процес роботи генетичного алгоритму, зокрема множину особин-кандидатів на рішення задачі двокритеріальної оптимізації у 1-му, 100-му та 200-му поколіннях.

Як видно з вищенаведених рисунків, генетичний алгоритм з кожною ітерацією покращує множину кандидатів-розв'язків. Поступово розв'язки покращуються та групуються в лінію, що містить мно-

жину оптимальних по Парето розв'язків (фронт Парето).

Також розроблений алгоритм було оптимізовано шляхом застосування паралельних обчислень в програмній реалізації генетичного алгоритму. Оскільки обчислення для кожної особини популяції відстані до сусідніх особин в межах рангу є незалежними операціями, то було застосовано розпаралювання обчислень для даного кроку алгоритму.

Результати експериментів наведено на рис. 7.

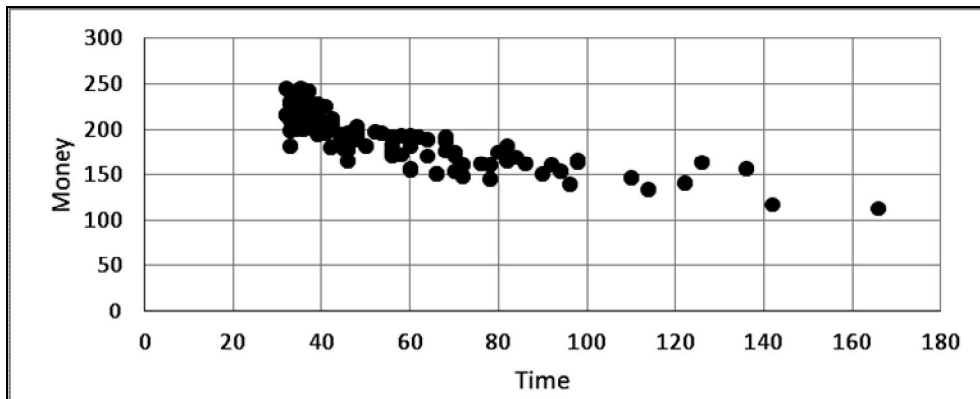


Рис. 4. Множина рішень-кандидатів в 1-му поколінні  
(Fig. 4. A quantity of candidate solutions in the 1st generation)

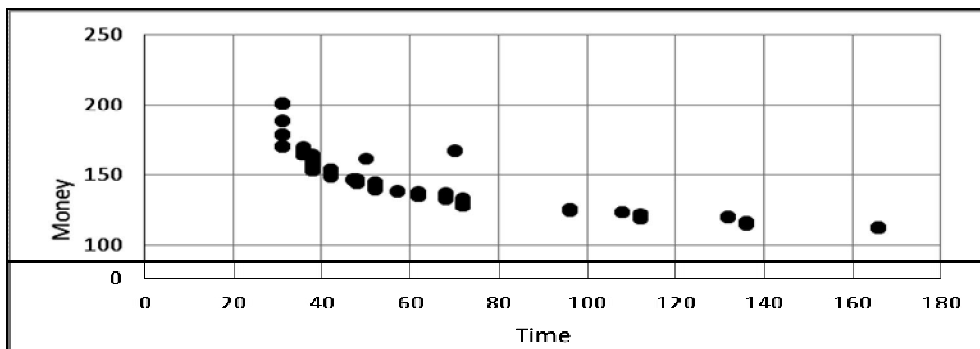


Рис. 5. Множина рішень-кандидатів в 100-му поколінні  
(Fig. 5. A quantity of candidate solutions in the 100 generation)

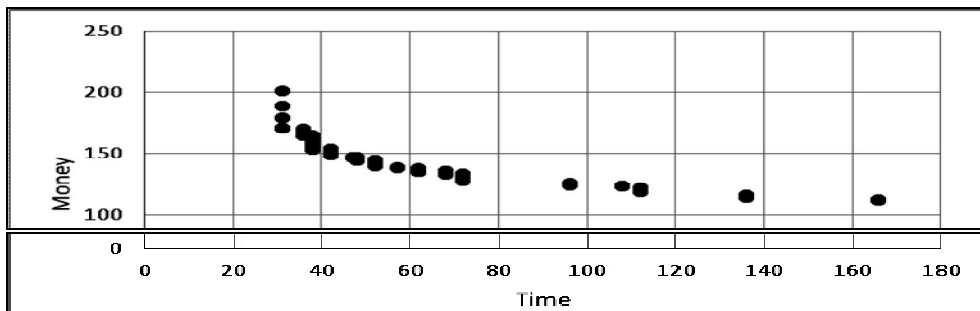


Рис. 6. Множина рішень-кандидатів в 200-му поколінні  
(Fig. 6. A quantity of candidate solutions in the 200 generation)

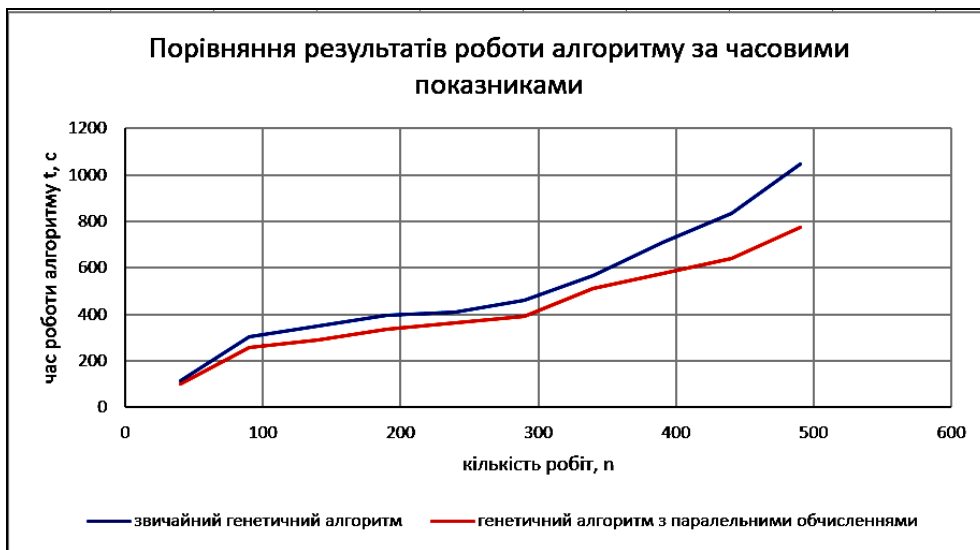


Рис. 7. Результати експериментальних досліджень  
(Fig. 7. The results of experimental studies)

Експерименти проводились для задач наступної розмірності: кількість виконавців  $m = 20$ , кількість робіт  $n$  змінювалась від 40 до 490 з кроком 40 робіт.

Як видно із результатів дослідження, застосування паралельних обчислень в генетичному алгоритмі дає вигоду у швидкодії майже на 16% в порівнянні із його звичайною версією.

### Висновки

В даній роботі було наведено огляд основних підходів для розв'язання задач багатокритеріальної оптимізації, зокрема, підхід з використанням згортки, лексикографічний підхід та підхід Парето, описані їх переваги та недоліки.

На основі проведеного аналізу було визначено, що підхід Парето має значні переваги над іншими підходами, тому для розробки алгоритму було використано саме його.

Також для розв'язання задачі багатокритеріальної оптимізації було обрано генетичний алгоритм, оскільки він з самого початку працює з великою популяцією кандидатів на розв'язки, і, відповідно, оп-

тимальні по Парето розв'язки генеруються вже в першому поколінні, що дозволяє при невеликих затратах часу на обчислення знайти множину оптимальних розв'язків.

Розроблено оператори генетичного алгоритму (селекція, визначення кращих особин популяції, обчислення значення пристосованості особин), які враховують особливості розв'язуваної багатокритеріальної задачі складання розкладу. Було проведено ряд досліджень ефективності роботи розробленого алгоритму. Завдяки впровадженню паралельних обчислень підвищено швидкість розробленого генетичного алгоритму приблизно на 16%.

Практична цінність отриманих результатів полягає в тому, що розроблений алгоритм може бути використано при вирішенні задачі оптимального розподілу ресурсів, що є частиною системи нарахування бонусів співробітникам.

Таким чином, вдасться отримати оптимальні розподіли завдань між працівниками, що дозволять при мінімальних витратах на бонусні виплати мінімізувати загальний час завершення виконання завдань.

### СПИСОК ЛІТЕРАТУРИ

1. Coello C.A.C. Multi-objective Optimization. *Handbook of Heuristics* / Martí, R., Panos, P., Resende, M. (eds). Chamm: Springer, 2018. DOI: [https://doi.org/10.1007/978-3-319-07153-4\\_17-1](https://doi.org/10.1007/978-3-319-07153-4_17-1).
2. Chianussi G., Codegone M., Ferreroc S., Varesio, F.E. Comparison of multi-objective optimization methodologies for engineering applications. *Computers & Mathematics with Applications*. 2012. Vol. 63, Is. 5. P. 912-942, DOI: <https://doi.org/10.1016/j.camwa.2011.11.057>.
3. Полковникова Н. А., Курейчик В. М. Многокритериальная оптимизация на основе эволюционных алгоритмов. *Известия ЮФУ. Технические науки*. 2015. № 2(163). С. 149-162.
4. Марченко И. В. Тема 8. Многокритериальная оптимизация. *Экономико-математические методы: конспект лекций*. Москва : МБИ, 2020. URL: [http://eos.ibi.spb.ru/umk/4\\_4/5/print/5\\_R1\\_T8.pdf](http://eos.ibi.spb.ru/umk/4_4/5/print/5_R1_T8.pdf).
5. Ashis K.M., Mohapatra Yo., Anil K.M. Multi-Objective Genetic Algorithm: A Comprehensive Survey. *International Journal of Emerging Technology and Advanced Engineering*. 2012. Vol. 3, Is. 2. pp. 81-90.
6. Chang K. Multiobjective Optimization and Advanced Topics. *e-Design: Computer-Aided Engineering Design*. Academic Press, 2015. P. 1105-1173. DOI: <https://doi.org/10.1016/B978-0-12-382038-9.00019-3>.
7. Tamil Selvi S., Baskar S., Rajasekar S. Application of Evolutionary Algorithm for Multiobjective Transformer Design Optimization. *Classical and Recent Aspects of Power System Optimization*. 2018. P. 463-504. DOI: <https://doi.org/10.1016/B978-0-12-812441-3.00017-3>.
8. Ouattara A., Pibouleau L., Azzaro-Pantel C., Domenech S. Eco-design and Life Cycle Assessment of a chemical process: the case study of HDA (Pareto Front). *Computer Aided Chemical Engineering*. 2013. Vol. 32. P. 727-732, DOI: <https://doi.org/10.1016/B978-0-444-63234-0.50122-6>.

### REFERENCES

1. Coello, C.A.C. (2018), "Multi-objective Optimization", Martí R., Panos P., Resende M. (eds), *Handbook of Heuristics*, Springer, Chamm, DOI: [https://doi.org/10.1007/978-3-319-07153-4\\_17-1](https://doi.org/10.1007/978-3-319-07153-4_17-1).
2. Chianussi, G., Codegone, M., Ferreroc, S. & Varesio, F.E. (2012), "Comparison of multi-objective optimization methodologies for engineering applications", *Computers & Mathematics with Applications*, Vol. 63, Is. 5, pp. 912-942, DOI: <https://doi.org/10.1016/j.camwa.2011.11.057>.
3. Polkovnykova, N.A., Kurejchyk, V.M. (2015), "Multiobjective optimization based on evolutionary algorithms", *Izvestiya SFedU. Engineering Sciences*, No. 2(163), pp. 149-162.
4. Marchenko, I.V. (2020), "Topic 8. Multiobjective optimization", *Economic and mathematical methods: lecture notes*, IBI, Moscow, available at: [http://eos.ibi.spb.ru/umk/4\\_4/5/print/5\\_R1\\_T8.pdf](http://eos.ibi.spb.ru/umk/4_4/5/print/5_R1_T8.pdf).
5. Ashis, K.M., Mohapatra, Yo., Anil, K.M. (2012), "Multi-Objective Genetic Algorithm: A Comprehensive Survey", *International Journal of Emerging Technology and Advanced Engineering*, Vol. 3, Is. 2, pp. 81-90.
6. Chang, K. (2015), "Multiobjective Optimization and Advanced Topics", *e-Design: Computer-Aided Engineering Design*, Academic Press, pp. 1105-1173, DOI: <https://doi.org/10.1016/B978-0-12-382038-9.00019-3>.
7. Tamil Selvi, S., Baskar S. & Rajasekar S. (2018), "Application of Evolutionary Algorithm for Multiobjective Transformer Design Optimization", *Classical and Recent Aspects of Power System Optimization*, pp. 463-504, DOI: <https://doi.org/10.1016/B978-0-12-812441-3.00017-3>.
8. Ouattara, A., Pibouleau, L., Azzaro-Pantel, C., & Domenech, S. (2013), "Eco-design and Life Cycle Assessment of a chemical process: the case study of HDA (Pareto Front)", *Computer Aided Chemical Engineering*, Vol. 32, pp. 727-732, DOI: <https://doi.org/10.1016/B978-0-444-63234-0.50122-6>.

Received (Надійшла) 22.11.2020

Accepted for publication (Прийнята до друку) 13.01.2021

## ВІДОМОСТІ ПРО АВТОРІВ/ ABOUT THE AUTHORS

**Казмірчук Аліна Василівна** – магістрантка кафедри автоматизованих систем обробки інформації та управління, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна;  
**Alina Kazmirchuk** – master student of Computer-aided Management and Data Processing Systems Department, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine;  
e-mail: [girl.lina.k@gmail.com](mailto:girl.lina.k@gmail.com); ORCID ID: <https://orcid.org/0000-0003-3825-2441>.

**Жданова Олена Григорівна** – кандидат технічних наук, доцент кафедри автоматизованих систем обробки інформації та управління, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна;

**Olena Zhdanova** – PhD in Technical science, Associate Professor of Computer-aided Management and Data Processing Systems Department, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine;  
e-mail: [zhdanova/elena@hotmail.com](mailto:zhdanova/elena@hotmail.com); ORCID ID: <https://orcid.org/0000-0002-8787-846X>.

**Попенко Володимир Дмитрович** – кандидат технічних наук, доцент кафедри автоматизованих систем обробки інформації та управління, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна;

**Volodymyr Popenko** – PhD in Technical science, Associate Professor of Computer-aided Management and Data Processing Systems Department, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine;  
e-mail: [volodp@ukr.net](mailto:volodp@ukr.net); ORCID ID: <https://orcid.org/0000-0002-4500-2267>.

**Сперкач Майя Олегівна** – кандидат технічних наук, доцент кафедри автоматизованих систем обробки інформації та управління, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна;

**Maiia Sperkach** - PhD in Technical science, Associate Professor of Computer-aided Management and Data Processing Systems Department, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine;  
e-mail: [sperkachmaya@gmail.com](mailto:sperkachmaya@gmail.com); ORCID ID: <https://orcid.org/0000-0003-0251-2141>.

**Решение многокритериальной задачи составления расписания  
с использованием генетического алгоритма**

А. В. Казмірчук, Е. Г. Жданова, В. Д. Попенко, М. О. Сперкач

**Аннотация.** Работа посвящена многокритериальной задаче составления расписания, в которой определенное множество работ должно быть выполнено несколькими исполнителями разной продуктивности. За выполнение работы соответствующему исполнителю начисляется количество бонусов, которое зависит от времени выполнения работы. Критериями оценки расписания являются общее время выполнения всех работ и общее количество бонусов. В ходе проведенных исследований были проанализированы основные подходы к решению задач многокритериальной оптимизации, на основании которых был выбран подход Парето. В качестве алгоритма был выбран генетический алгоритм. **Целью данной работы** является повышение эффективности решения задач многокритериальной оптимизации с помощью реализации эвристического алгоритма и повышения его быстродействия. **Заданиями работы** являются определение преимуществ и недостатков подходов, которые используются для решения задач многокритериальной оптимизации, разработка генетического алгоритма для решения задачи многокритериального составления расписания и исследование его эффективности. Разработаны операторы генетического алгоритма, которые учитывают особенности исследуемой задачи и позволяют получать в процессе работы решения Парето. Благодаря внедрению параллельных вычислений в реализации генетического алгоритма удалось повысить его быстродействие в сравнении с обычной версией. Разработанный алгоритм может использоваться при решении задачи оптимального распределения ресурсов, что является частью системы начисления бонусов сотрудникам.

**Ключевые слова:** задача составления расписания; многокритериальная оптимизация; эвристика; генетический алгоритм; фронт Парето.

**Solving a multicriteria scheduling problem  
using a genetic algorithm**

Alina Kazmirchuk, Olena Zhdanova, Volodymyr Popenko, Maiia Sperkach

**Abstract.** The work is devoted to the multiobjective task of scheduling, in which a given set of works must be performed by several performers of different productivity. A certain number of bonuses is accrued for the work performed by the respective executor, which depends on the time of work performance. The criteria for evaluating the schedule are the total time of all jobs and the amount of bonuses spent. In the research the main approaches to solving multiobjective optimization problems were analyzed, based on which the Pareto approach was chosen. The genetic algorithm was chosen as the algorithm. **The purpose of this work** is to increase the efficiency of solving multicriteria optimization problems by implementing a heuristic algorithm and increase its speed. **The tasks of the work** are to determine the advantages and disadvantages of the approaches used to solve multicriteria optimization problems, to develop a genetic algorithm for solving the multicriteria scheduling problem and to study its efficiency. Operators of the genetic algorithm have been developed, which take into account the peculiarities of the researched problem and allow to obtain Pareto solutions in the process of work. Due to the introduction of parallel calculations in the implementation of the genetic algorithm, it was possible to increase its speed compared to the conventional version. The developed algorithm can be used in solving the problem of optimal allocation of resources, which is part of the system of accrual of bonuses to employees.

**Keywords:** schedule problem; multiobjective optimization; heuristics; genetic algorithm; Pareto front.