

УДК 009.4

## МЕТОДИ ОПТИМІЗАЦІЇ РОЗМІЩЕННЯ ВЕКТОРНИХ ГРАФІЧНИХ ОБ'ЄКТІВ НА ПЛОЩИНІ

Д. Ю. Паламарчук, О. В. Тимченко, В. О. Демченко

*Українська академія друкарства,  
вул. Під Голоском, 19, Львів, 79020, Україна,*

*Проведено порівняльний аналіз трьох методів оптимізації розміщення векторних графічних об'єктів на площині: імітації відпалювання, імітаційного моделювання та генетичного алгоритму. Описано складнощі, які виникають під час вирішення цього завдання, а також особливості під час роботи з векторними зображеннями. Проведено аналіз останніх досліджень та публікацій, де розглянуто суть оптимальних рішень і методів для їх знаходження. Побудовано математичну модель задачі оптимального розміщення об'єктів на площині. Розглянуто структуру, особливості, алгоритми, переваги та недоліки методів імітації відпалювання, імітаційного моделювання та генетичного алгоритму. Наведено висновки з результатами досліджень цих методів як потенційні рішення для оптимізації розміщення векторних графічних об'єктів на площині.*

**Ключові слова:** *оптимізація, імітаційне моделювання, векторні графічні об'єкти, генетичний алгоритм, метод імітації відпалювання, порівняльний аналіз.*

**Постановка проблеми.** У сучасному світі існує багато видів виробництв, в яких використовуються нарізка або малювання векторних зображень на площині, зокрема за допомогою спеціальних пристроїв — плотерів. Цей вид виробництва може застосовуватися як в промислових масштабах поліграфії, так і для домашнього користування із застосуванням різних видів матеріалу: паперу, тканини, металу. Тож спектр застосування дуже великий і може вирішувати безліч бізнес-задач. При нарізці багатьох деталей постає проблема оптимального розміщення компонентів на полотні для економії ресурсів. Оптимізація полягає у розміщенні векторних зображень, які мають різні розміри і форми, з використанням найменшої необхідної площі на полотні. Вирішення цієї задачі допоможе бізнес-процесам економити ресурси, зменшити кількість відхідних матеріалів, збільшити кількість деталей, які можна виготовити. Оскільки для роботи плотерів використовуються векторні зображення, важливим є врахування їх особливостей. Зокрема, дані зображення можуть містити велику кількість контурів, що може збільшити час для пошуку оптимального розміщення деталі. Також важливим є їх розмір, позиція, обертання і допустима відстань між об'єктами. Оскільки існує певна невизначеність в отриманні результату, необхідно розглянути цю задачу як NP-складну і використати наближені методи пошуку оптимального рішення. Для вирішення задачі з розміщення векторних об'єктів на площині було розглянуто такі методи:

імітація відпалу, імітаційне моделювання і генетичний алгоритм. Проведено аналіз та розглянуто їх особливості, структуру, переваги та недоліки.

**Аналіз останніх досліджень та публікацій.** Оптимальне рішення в контексті оптимізації вказує на такий варіант або набір значень параметрів, які призводять до найкращого можливого результату в рамках певної задачі чи системи [1]. Це може бути максимізація прибутку, мінімізація витрат, оптимізація ефективності чи досягнення інших цілей. Метод оптимізації, спрямований на знаходження оптимального рішення, передбачає визначення цільової функції та обмежень, які визначають допустимий простір рішень. Задача оптимізації полягає в пошуку значень змінних, які максимізують чи мінімізують цільову функцію при врахуванні обмежень [2]. Оптимальне рішення може бути знайдене різними методами оптимізації, такими як градієнтні методи, методи наочного пошуку, еволюційні алгоритми тощо. У багатьох випадках використовують математичні методи для аналізу та знаходження оптимальних рішень. Важливо враховувати, що оптимальне рішення може залежати від умов, за яких проводиться оптимізація і враховується компроміс між різними цілями чи обмеженнями, які можуть бути в конкретній ситуації. Евристичні алгоритми — алгоритми, засновані на правдоподібних, але необґрунтованих математично припущеннях про властивості оптимального рішення задачі. Метаевристичні методи — це клас алгоритмів оптимізації, які призначені для розв'язання складних задач оптимізації, де традиційні методи можуть бути неефективними через велику кількість можливих рішень чи високу вимірність простору пошуку. Для порівняння було обрано три метаевристичні методи: метод імітації відпалювання, імітаційне моделювання та генетичний алгоритм.

Суть методу імітації відпалювання полягає в тому, що під час пошуку оптимального рішення допускаються як кращі, так і гірші переміщення (або зміни стану). Велика «температура» в початковій фазі пошуку дає змогу алгоритму «досліджувати» більшу частину простору пошуку, тобто робити значні зміни стану, зокрема і ті, які здаються «гіршими». З плином часу «температура» знижується, і алгоритм дедалі більше концентрується на «ускладненні» найкращих знайдених рішень, замість того, щоб «досліджувати» нові області [3].

Імітаційне моделювання — це процес побудови комп'ютерної моделі, яка відтворює динаміку реальної системи або процесу протягом часу [4]. Цей метод широко використовується для дослідження складних систем та сценаріїв, які можуть бути надто складними або небезпечними для дослідження в реальному світі.

Генетичний алгоритм є методом оптимізації, інспірованим процесом природного відбору, який містить механізми, такі як спадковість, мутація, селекція і кросовер (або рекомбінація). Основна ідея генетичного алгоритму полягає в тому, що велика популяція потенційних рішень для задачі оптимізації «еволюціонує» до більш «приспосованих» рішень через багато поколінь [5].

**Мета статті** — дослідження та аналіз ефективності методу імітації відпалювання, імітаційного моделювання та генетичного алгоритму в контексті оптимізації розміщення векторних об'єктів на площині.

**Виклад основного матеріалу дослідження.** Для побудови математичної моделі задачі оптимального розміщення об'єктів на площині використовуємо множину індексів  $I = \{1, \dots, n\}$ . Цільова функція — сумарна площа, використана для розміщення векторних об'єктів з урахуванням проміжків між ними, які вже непридатні для розміщення фігур. Параметрами задачі є елементи матриці  $C = (c_{ij}), i, j \in I$ .

Змінними задачі є елементи бінарної матриці розміщень об'єктів  $X = (x_{ij}), i, j \in I$ , які дорівнюють 1, якщо у побудованому розв'язку для задачі присутнє рішення  $(v_i, v_j)$ , 0 — інакше. Оптимальним розв'язком будемо називати найменш заповнену фігурами площу з урахуванням непридатних ділянок між фігурами.

$$E = \sum_{i \in I} \sum_{j \in I, j \neq i} c_{ij} x_{ij} \rightarrow \min$$

з обмеженнями

$$\sum_{j \in I, j \neq i} x_{ij} = 1, i \in I,$$

$$\sum_{i \in I, i \neq j} x_{ij} = 1, j \in I.$$

Алгоритми, спрямовані на вирішення проблеми визначення оптимального маршруту, можна розділити на дві категорії: точні та евристичні. Для малих завдань рекомендовано використовувати точні алгоритми, оскільки їх реалізація потребує великих обчислювальних потужностей, що не завжди можливо забезпечити для реальних задач. Зазвичай для останніх використовують евристичні алгоритми.

**Метод імітації відпалювання.** Метод імітації відпалювання (Simulated Annealing, SA) базується на аналогії з процесом відпалювання в металургії, де метал нагрівається до високих температур, а потім повільно охолоджується для досягнення стану мінімальної енергії [6].

В контексті оптимізації «енергія» системи може бути представлена як функція вартості (або цільова функція), яку потрібно мінімізувати.

Введемо такі позначення:

$S$  — множина станів системи;

$f(s)$  — функція зміни стану;

$s_i$  — стан системи на  $i$ -му кроці;

$s_k$  — новий стан (кандидат);

$t_{\min}, t_i, t_{\max}$  — мінімальна, поточна та вихідна температури, відповідно;

$T(t)$  — функція зміни температури;

$E(s)$  — цільова функція.

Структура методу імітації відпалювання:

1) ініціалізація: визначте початкове рішення  $s_1$  (може бути випадковим) і встановіть високу «температуру»  $t_1 = t_{\max}$ ;

2) генерація нового стану: на основі поточного стану системи генерується новий стан  $s_k = f(s_{k-1})$ . Це може бути зроблено за допомогою випадкового перебудування поточного стану;

3) оцінка стану: оцініть «енергію» (або вартість) нового стану за допомогою цільової функції  $\Delta E = E(s_k) - E(s_{k-1})$ ;

4) відхилення: порівняйте «енергії» старого і нового станів. Якщо новий стан має нижчу вартість  $\Delta E < 0$  (тобто він кращий), то він приймається як новий поточний стан  $s_{i+1} = s_k$ . Якщо новий стан має вищу вартість, він все ще може бути прийнятим із деякою ймовірністю, яка залежить від різниці вартостей та поточної

температури  $\exp\left(\frac{\Delta E}{t_i}\right) > M$ . Це дає змогу алгоритму «переборювати» локальні мінімуми;

5) зменшення температури: зменшіть «температуру» згідно з певним розкладом охолодження  $t_{i+1} = T(t_i)$  (наприклад, геометричним, лінійним або логарифмічним);

6) повторення: повторюйте кроки 2–5 до заданої умови зупинки, такої як досягнення певного числа ітерацій, досягнення певної «температури» або до того, як вартість не змінюється протягом певного числа ітерацій  $i = i + 1$  поки  $t_i > t_{\min}$ .

Після завершення процесу останній прийнятий стан розглядається як краще знайдене рішення.

Переваги методу імітації відпалювання:

- здатність знаходити глобальний оптимум: на відміну від багатьох інших методів оптимізації, метод імітації відпалювання може вийти з локальних мінімумів завдяки «гарячим» етапам, коли допускається тимчасове погіршення рішення;
- універсальність: метод імітації відпалювання може бути застосований до широкого спектра проблем оптимізації, включаючи такі, які мають нелінійні обмеження;
- можливість використання для різних типів даних: метод імітації відпалювання може працювати з дискретними, цілочисельними та неперервними змінними. Недоліки методу імітації відпалювання:
- параметричні налаштування: ефективність суттєво залежить від правильного вибору параметрів, таких як початкова температура, розклад охолодження, а також правил генерації нових рішень. Налаштування цих параметрів може бути непростим і потребує глибокого розуміння конкретного застосування;
- час виконання: метод може бути повільним, особливо для великих просторів пошуку, оскільки він потребує багато ітерацій для знаходження оптимального рішення;
- гарантія оптимальності: хоча метод добре справляється з виходом з локальних мінімумів, він не дає гарантій знаходження глобального мінімуму, особливо в задачах з великими просторами пошуку;
- використання в машинному навчанні: у контексті машинного навчання, де потрібно оптимізувати велику кількість параметрів, SA може бути непридатним або неефективним, порівняно з іншими методами оптимізації, такими як градієнтний спуск або методи, засновані на ньому.

**Імітаційне моделювання.** Імітаційне моделювання — це метод вивчення та аналізу систем, який полягає у створенні моделі системи та проведенні віртуальних експериментів на цій моделі. У цьому методі модель системи побудована таким чином, щоб вона відображала ключові аспекти реальної системи та її взаємодії [7].

Основна ідея імітаційного моделювання — це створення абстрактного представлення реальної системи, яке дає змогу вивчати її поведінку та взаємодії в умовах, які моделюють реальні обставини. Цей метод застосовується в різних галузях, таких як наука, інженерія, економіка, медицина та інші.

Імітаційне моделювання може включати в себе створення комп'ютерних програм, що відтворюють роботу системи та дають змогу проводити експерименти з різними сценаріями. Це дає змогу аналізувати вплив різних факторів, стратегій чи політик на функціонування системи та робити прогнози щодо її майбутнього розвитку.

Також імітаційне моделювання є корисним інструментом для дослідження складних систем, де аналітичні або експериментальні методи можуть бути обмежені. Цей підхід дає змогу вивчати систему в різних умовах та допомагає ухвалювати обґрунтовані рішення на основі отриманих результатів експериментів.

Для опису математичної моделі введемо такі параметри:

$\Delta t$  — час, який пройшов між подіями;

$\lambda$  — кількість подій, які вводяться в систему за одиницю часу;

$\mu$  — кількість подій, яку система може обробити за одиницю часу;

$t$  — час моделювання;

$Q$  — кількість подій в черзі.

Покроковий алгоритм:

1) ініціалізація: необхідно встановити початковий час  $t = 0$  і початковий стан системи  $Q = 0$ ;

2) генерація подій: згенерувати час приходу нової події  $t_{arrival}$  на основі розподілу ймовірностей, наприклад, експоненційний розподіл з параметром  $\lambda$ ;

3) перевірка подій: перевірити, чи прийшла нова подія  $t_{arrival} < t + \Delta t$ . Якщо так, додати подію до черги та оновити стан системи  $Q + 1$ ;

4) обробка подій: перевірити, чи система може обробити подію  $Q > 0$  та  $t_{service} < t + \Delta t$ , де  $t_{service}$  згенеровано на основі розподілу обслуговування  $\mu$ . Якщо так, то видалити подію з черги та оновити стан системи  $Q - 1$ .

5) оновлення часу: оновити час  $t = t + \Delta t$ ;

6) повторення: повторити кроки 2–5 до досягнення кінцевого часу моделювання або до задоволення іншої зупинки.

Переваги імітаційного моделювання:

- велика гнучкість: імітаційне моделювання може бути використано для моделювання складних систем, які можуть бути важкими для аналітичного опису;
- відтворення динаміки процесів в часі: імітаційне моделювання дає змогу відображати динаміку процесів в часі і вивчати зміни стану системи протягом часу;
- можливість проведення експериментів: ви можете проводити «експерименти» з моделлю, змінювати вхідні параметри і вивчати вплив цих змін на поведінку системи;
- безпека: моделювання дає змогу вивчати потенційно небезпечні або дорогі вартісні ситуації без реального ризику.

Недоліки імітаційного моделювання:

- складність: імітаційні моделі можуть бути досить складними, особливо для великих систем. Це може зробити процес моделювання важким та часомістким;
- труднощі з валідацією і верифікацією: перевірка того, що модель правильно відтворює реальну систему, може бути викликом, особливо для складних моделей;
- залежність від вхідних даних: якість результатів моделювання значною мірою залежить від якості вхідних даних. Якщо вхідні дані є неточними або неповними, результати моделювання також будуть неточними.

**Генетичний алгоритм.** Генетичний алгоритм (ГА) — це оптимізаційний метод, інспірований процесами еволюції в природі. Алгоритм оперує популяцією потенційних рішень для задачі оптимізації та використовує оператори відбору, схрещування і мутації, щоб створювати нові покоління рішень. Кожен індивід у популяції представляється генетичною інформацією, яка піддавалася еволюційним операціям для досягнення оптимального розв'язку. Генетичні алгоритми широко використовуються у різних галузях, зокрема інженерії, штучному інтелекті, економіці та ін., завдяки їхній здатності ефективно вирішувати задачі оптимізації в складних просторах рішень [8]. Можна виділити такі етапи генетичного алгоритму:

1) ініціалізація: створення початкової популяції індивідів

$$P_0 = \{X_1, X_2, \dots, X_N\};$$

2) оцінка пристосованості: визначення функції  $f(X)$  пристосованості для кожного індивіда

$$F(X) = f(X);$$

3) відбір: вибір індивідів для схрещування на основі їхньої пристосованості. Ймовірність вибору індивіда  $X_i$  і  $X_j$  пропорційна його пристосованості

$$P(X_i) = \frac{F(X_i)}{\sum_{j=1}^N F(X_j)};$$

4) схрещування: схрещення обраних індивідів за допомогою оператора кросовера. Новостворені індивіди позначаються ;

5) мутація: випадкове внесення змін в генетичну інформацію індивідів для введення випадковості

$$Y_k = Mutation(X_k);$$

6) оцінка пристосованості нових індивідів: повторення кроку 2 для новостворених індивідів

$$F(Y_k) = f(Y_k);$$

7) формування нової популяції: обрання індивідів для нової популяції на основі їхньої пристосованості;

8) повторення: повторення кроків 3–7 протягом кількох поколінь або до досягнення критерію зупинки.

Переваги генетичного алгоритму:

- глобальна оптимізація: алгоритм ефективно працює в просторах з великими розмірами та може знаходити глобальні оптимуми в проблемах з багатьма локальними оптимумами;

- паралелізм: алгоритм виконує пошук в багатьох точках простору рішень одночасно, що дає змогу ефективно використовувати паралельні обчислення;
- універсальність: алгоритм може застосовуватися до різних типів проблем, зокрема тих, які непридатні для інших методів оптимізації і не потребують спеціального знання про проблему;
- здатність до адаптації: алгоритм може адаптуватися до змін у проблемі в процесі виконання, що робить його корисними для динамічних проблем.  
Недоліки генетичного алгоритму:
- витрати часу та ресурсів: алгоритм може потребувати багато обчислень, особливо для великих просторів рішень або складних функцій придатності;
- передчасна конвергенція: алгоритм може зіткнутися з проблемою передчасної конвергенції, коли вся популяція занадто швидко стає однорідною і перестає генерувати нові, кращі рішення;
- налаштування параметрів: вибір відповідних значень для параметрів, таких як розмір популяції, швидкість мутації, може бути складним і може залежати від конкретної проблеми;
- відсутність гарантії знаходження оптимального рішення: хоча генетичні алгоритми здатні знаходити дуже хороші рішення, вони зазвичай не дають гарантії знаходження абсолютно оптимального рішення.

**Висновки.** Отже, розглянуто метод імітації відпалювання, імітаційне моделювання та генетичний алгоритм як потенційні рішення для оптимізації розміщення векторних графічних об'єктів на площині. Кожен з методів має свої переваги та недоліки. Вибір методу повинен здійснюватися на основі конкретних вимог до задачі, враховуючи час виконання, точність, здатність до глобального пошуку та простоту реалізації. Застосування методу імітації відпалювання показало його ефективність у задачах з великим простором пошуку, але потребує добре підбраного графіка охолодження для найкращих результатів. Імітаційне моделювання показало свою пристосованість до систем з великою кількістю параметрів та високим ступенем невизначеності, хоча воно може бути затратним за часом. Генетичний алгоритм виявився ефективним для пошуку оптимальних рішень у задачах, що дозволяють натуральне кодування рішень як хромосом, але вимагає правильного визначення генетичних операторів для кожної конкретної задачі. Подальше дослідження може передбачати більш детальне вивчення та експериментування з параметрами кожного методу, розробку гібридних методів, а також розробку більш ефективних методів для специфічних класів проблем розміщення векторних графічних об'єктів.

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. DeGroot Morris. Optimal Statistical Decisions. McGraw-Hill. New York, 1970.
2. Garey Michael R., Johnson David S. Computers and intractability. A Guide to the Theory of NP-Completeness. 1979.
3. Springer. Search Methodologies - Introductory Tutorials in Optimization and Decision Support Techniques. 2005.

4. Стеценко І. В., Бойко О. В. Система імітаційного моделювання засобами сіток Петрі. *Мат. машини і системи*. 2009. № 1. С. 117–124.
5. Hopcroft John E., Motwani Rajeev, Ullman Jeffrey D. *Introduction to Automata Theory, Languages and Computation*. 2001.
6. Гавриленко В. В., Івохіна К. Є., Рудоман Н. В. Про застосування методу імітації відпалу для розв'язання нечіткої задачі комівояжера. *Control, Navigation and Communication Systems*. 2022. № 3. С. 60–63.
7. Братушка С. М. Імітаційне моделювання як інструмент дослідження складних економічних систем. *Вісник Української академії банківської справи*. 2009. № 2 (27). С. 113–118.
8. Мартинова О. В., Степанова К. В. Генетичний алгоритм для розв'язання оптимізаційних задач. *Сучасні проблеми управління підприємствами: теорія та практика : матеріали міжнар. наук.-практ. конф. : тези допов. (18–19 бер. 2019 р.)*. 2019. С. 387–389.

#### REFERENCES

1. DeGroot, Morris. (1970). *Optimal Statistical Decisions*. McGraw-Hill. New York (in English).
2. Garey, Michael R., & Johnson, David S. (1979). *Computers and intractability. A Guide to the Theory of NP-Completeness* (in English).
3. Springer. *Search Methodologies - Introductory Tutorials in Optimization and Decision Support Techniques*. 2005 (in English).
4. Stetsenko, I. V., & Boiko, O. V. (2009). *Systema imitatsiinoho modeliuвання zasobamy sitok Petri: Mat. mashyny i systemy*, 1, 117–124 (in Ukrainian).
5. Hopcroft, John E., Motwani, Rajeev, & Ullman, Jeffrey D. (2001). *Introduction to Automata Theory, Languages and Computation* (in English).
6. Havrylenko, V. V., Ivokhina, K. Ye., & Rudoman, N. V. (2022). *Pro zastosuvannya metodu imitatsii vidpalu dlia rozv'iazannya nechitkoi zadachi komivoiazhera: Control, Navigation and Communication Systems*, 3, 60–63 (in Ukrainian).
7. Bratushka, S. M. (2009). *Imitatsiine modeliuвання yak instrument doslidzhennia skladnykh ekonomichnykh system: Visnyk Ukrainskoi akademii bankivskoi spravy*, 2 (27), 113–118 (in Ukrainian).
8. Martynova, O. V., & Stiepanova, K. V. (2019). *Henetychnyi alhorytm dlia rozv'iazannya optymizatsiinykh zadach. Suchasni problemy upravlinnia pidpriemstvamy: teoriia ta praktyka : materialy mizhnar. nauk.-prakt. konf. : tezy dopov. (18–19 ber. 2019 r.)*, 387–389 (in Ukrainian).

doi: 10.32403/0554-4866-2023-2-86-40-48

#### METHODS OF OPTIMIZING THE ARRANGEMENT OF VECTOR GRAPHIC OBJECTS ON THE PLANE

D. Yu. Palamarchuk, O. V. Tymchenko, V. O. Demchenko



*Ukrainian Academy of Printing,  
19, Pid Holoskom St., Lviv, 79020, Ukraine  
o\_tymch@ukr.net*

*The article is dedicated to the study and comparative analysis of three optimization methods: simulated annealing, simulation modelling, and genetic algorithm, in the context of the problem of optimal placement of vector graphic objects on a plane. The complexities encountered when solving this problem and the specifics of working with vector images are described. The analysis of the latest research and publications is conducted, where the essence of optimal solutions and methods for their finding are considered. A mathematical model of the problem of optimal placement of objects on a plane is presented. The structure, features, algorithms, advantages and disadvantages of the methods of simulated annealing, simulation modelling, and genetic algorithm are considered. The choice of method should be based on the specific requirements of the problem, including execution time, accuracy, global search capability, and ease of implementation. The application of the simulated annealing method shows its effectiveness in problems with a large search space, but requires a well-chosen cooling schedule for best results. Simulation modelling shows its suitability for systems with a large number of parameters and a high degree of uncertainty, although it can be time-consuming. The genetic algorithm proves to be effective for finding optimal solutions in problems that allow natural encoding of solutions as chromosomes, but requires the correct definition of genetic operators for each specific problem. Conclusions are provided based on the results of research of these methods as potential solutions for optimizing the placement of vector graphic objects on a plane. This research aims to assist scientists and engineers in selecting the most effective method for solving a specific problem, as well as to stimulate further research in the field of optimization of placement of vector graphic objects.*

**Keywords:** *optimization, simulation modelling, vector graphic objects, genetic algorithm, annealing simulation method, comparative analysis.*

*Стаття надійшла до редакції 24.05.2023.*

*Received 24.05.2023.*