

**ВИЗНАЧЕННЯ ОСОБЛИВОСТЕЙ ЗАСТОСУВАННЯ  
НАБЛИЖЕНИХ МЕТОДІВ ОПТИМІЗАЦІЇ ДЛЯ ВИРІШЕННЯ  
ПРИКЛАДНИХ ЗАДАЧ ІТ-ГАЛУЗІ**

Сібільов В.О.

e-mail: vladyslav.sibilov@nure.ua

Науковий керівник – канд. техн. наук, доц. Творошенко І.С.  
Харківський національний університет радіоелектроніки, каф. ІНФ  
м. Харків, Україна

This work is devoted to a review of solving applied optimization problems in the IT industry using various approximate methods, taking into account each one's implantation and usage features. Different groups of approximate methods were considered: from simple greedy algorithms to more complex evolutionary ones. After defining the operation principles of various approximate methods, their advantages and disadvantages were identified. Usage limitations for each of the considered methods and the possibility of combining them were also identified.

Невпинний розвиток інформаційних технологій супроводжується постійною потребою у вдосконаленні бізнес-процесів та оптимальному використанні обчислювальних ресурсів, що обумовлює важливість застосування різних методів оптимізації, орієнтованих на вирішення прикладних задач ІТ-галузі.

Побудова оптимального маршруту з урахуванням змін динамічного середовища, формування інвестиційного портфеля задля максимізації потенційного прибутку, формування плану виконання робіт з огляду мінімізації часових вікон та розподіл навантаження між серверами з метою оптимізації їх загальної продуктивності – все це лише мала частина задач прикладного характеру, вирішення яких лежить в основі роботи сучасних інформаційних систем різного призначення. Значна частина таких задач є дискретними за своєю природою, характеризуються великою розмірністю та величезною множиною можливих розв'язків, через що мають високу обчислювальну складність та належать до класу NP-складних, що унеможливорює їх ефективне розв'язання точними методами за прийнятний час [1]. У зв'язку з цим особливої актуальності набуває використання наближених методів оптимізації для вирішення такого класу задач.

Втім, хоч загальна мета наближених методів в аспекті розв'язання оптимізаційних задач є однаковою – знаходження розв'язку, близького до оптимального (за короткий час), кожен із них має власну стратегію її досягнення. Розглядаючи клас NP-складних оптимізаційних задач, загалом виділяють наступні групи наближених методів для їх вирішення:

- жадібні алгоритми;
- методи локального пошуку;

- стохастичні методи;
- еволюційні алгоритми.

Жадібні алгоритми є сімейством наближених методів, які на кожному кроці обирають локально оптимальне рішення, допускаючи, що послідовність таких кроків призведе до глобального оптимуму [2]. Такий підхід добре підходить для задач із чітко визначеним критерієм оптимальності на кожному кроці і дозволяє швидко знаходити рішення із мінімальними витратами ресурсів. Прикладами алгоритмів цієї групи є алгоритми Крускала та Прима для побудови мінімальних остовних дерев, алгоритм Дейкстри для пошуку найкоротшого шляху, алгоритм сортування вибором та багато інших алгоритмів власної реалізації, принцип роботи яких відповідає загальній філософії жадібності. Незважаючи на свою простоту, жадібні алгоритми є ефективним інструментом розв'язання багатьох складних оптимізаційних задач, адже дозволяють швидко знаходити доволі гарні рішення в умовах обмежень на обчислювальні ресурси.

Методи локального пошуку ґрунтуються на поступовому вдосконаленні поточного розв'язку шляхом дослідження його локального оточення. На кожному кроці вони досліджують сусідні варіанти, обираючи один з них як новий поточний розв'язок. Найвідомішими методами цієї групи є пошук зі спуском (Hill Climbing), імітація відпалу (Simulated Annealing) та табу-пошук (Tabu Search). Основними перевагами таких алгоритмів є швидке знаходження близьких до оптимальних рішень і можливість комбінувати різні стратегії локального поліпшення, а до недоліків відносять значну залежність від початкового розв'язку та ризик застрягання в локальних екстремумах.

Основна ідея стохастичних методів полягає у використанні випадкових процедур генерації та модифікації проміжних розв'язків, забезпечуючи більш ефективне дослідження великого простору можливих рішень та зменшуючи ризик зупинки в локальних екстремумах. Втім, випадковий характер пошуку нових рішень часто призводить до того, що різні запуски одного й того ж алгоритму можуть давати суттєву різницю в результатах, що робить методи цієї групи чутливими до умов використання та параметрів налаштування. Представниками цієї групи наближених методів є випадковий пошук (Random Search), випадковий перезапуск (Random Restart) та стохастичний підйом на пагорб (Stochastic Hill Climbing). Різні модифікації таких методів, як правило, входять до складу алгоритмів інших груп наближених методів.

Окремим класом наближених методів, заснованих на принципах природного відбору та біологічної еволюції, є еволюційні алгоритми. Серед найвідоміших алгоритмів цієї групи є генетичний алгоритм (Genetic Algorithm), алгоритм мурашиної колонії (Ant Colony Optimization) та алгоритм рою частинок (Particle Swarm Optimization).

На відміну від інших наближених методів, еволюційні алгоритми одночасно оперують цілою множиною потенційних рішень, а не зосереджуються на одному поточному розв'язку. Завдяки паралельному обробленню багатьох кандидатів та застосуванню механізмів відбору, схрещуванню та мутації ці алгоритми дозволяють ефективно досліджувати простір можливих рішень, підвищують здатність уникати локальних оптимумів та зменшують залежність від початкового розв'язку. Водночас, попри широкі можливості їх поєднання з іншими наближеними методами, доцільність використання еволюційних алгоритмів є не завжди виправданою через потреби налаштування великої кількості параметрів та високі обчислювальні витрати [3–5].

Таким чином, алгоритми кожної з розглянутих груп наближених методів мають свої особливості в аспекті вирішення складних оптимізаційних задач. Всі вони демонструють різну складність програмної реалізації, відрізняються за часом пошуку рішення, відносною точністю розв'язку та можливістю масштабування, через що стають дедалі доцільнішими спроби їх комбінації з метою компенсації недоліків та посилення сильних сторін кожного з них, що за умови поєднання з іншими технологіями та підходами, зокрема динамічним програмуванням та машинним навчанням, може сприяти отриманню більш ефективних методів вирішення різного роду оптимізаційних задач.

#### Список використаних джерел:

1. Складність алгоритму і її оцінювання під час розробки програмного забезпечення. Foxminded.ua. URL: <https://foxminded.ua/skladnist-alhorytmu/> (дата звернення 20.02.2026).
2. Чен С. Жадібний алгоритм із прикладом: що таке, метод і підхід. Guru99. URL: <https://www.guru99.com/uk/greedy-algorithm.html> (дата звернення 21.02.2026).
3. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2025) Image description compression in classification structural methods, *IEEE Access*, vol. 13, pp. 43631–43641.
4. Gorokhovatskyi V., Tvoroshenko I., Yakovleva O., and Hudáková M. (2026) Feature space quantization and application of metrics in structural methods of image recognition, *IEEE Access*, vol. 14, pp. 17812–17824.
5. Gorokhovatskyi V., Tvoroshenko I. (2024) An effective method for transforming an image description into a compact vector for classification. *Information Technology and Implementation (Satellite): Conference Proceedings, November 21, 2024, Kyiv, Ukraine / Ministry of Education and Science of Ukraine, Taras Shevchenko National University of Kyiv and [etc.]; Vitaliy Snytyuk (Editor)*. – Kyiv: Publishing House «Caravela», pp. 25–28.