



DOI 10.28925/2663-4023.2025.27.718

УДК 004.8

Шоробура Владислав Вікторович

магістр спеціальності 121 «Інженерія програмного забезпечення»

Державний університет інформаційно-комунікаційних технологій, Київ, Україна

ORCID ID: 0009-0008-6018-8953

mag4049506@stud.duikt.edu.ua**Худік Богдан Олександрович**

доктор філософії, старший викладач кафедри інженерії програмного забезпечення

Державний університет інформаційно-комунікаційних технологій, Київ, Україна

ORCID ID: 0009-0005-1914-6418

b.khudik@duikt.edu.ua

РОЗРОБКА МЕТОДУ ОПТИМІЗАЦІЇ ПРОГРАМНОГО КОДУ НА ОСНОВІ ГЕНЕТИЧНОГО АЛГОРИТМУ

Анотація. Генетичні алгоритми дозволяють генерувати та комбінувати різні рішення, поступово покращуючи їх відповідно до певного критерію ефективності. У міжнародній практиці генетичні алгоритми використовуються для розв'язання різних задач оптимізації — зокрема, для розробки та удосконалення програмного забезпечення. Використання генетичних алгоритмів для оптимізації програмного коду є перспективним напрямком, що дозволяє автоматизувати пошук ефективних і продуктивних рішень у процесі розробки програмного забезпечення. Генетичні алгоритми сприяють підвищенню продуктивності коду, покращенню його структури та зменшенню споживання ресурсів, що є критично важливим у сучасному конкурентному світі. Незважаючи на те, що використання цього методу має певні обмеження, удосконалення алгоритмів та зростання обчислювальних можливостей роблять його дедалі доступнішим у реальних умовах розробки програмного забезпечення. Розробка нейронної мережі для автоматичної оптимізації коду, або ж створення програми, яка безпосередньо модифікує код, є надзвичайно ресурсоємною задачею. Навіть при використанні мережі потужних комп'ютерів такі підходи можуть призводити до сповільнення системи, періодичних зависань системи та необхідності постійних перезавантажень для стабілізації. Це не лише уповільнює процес розробки, а й робить його фінансово невигідним для багатьох компаній. Генетичні алгоритми надають можливість оптимізувати об'єм необхідних ресурсів, що є надзвичайно важливим фактором при розробці програмного забезпечення. В цілому, немає сумнівів в тому, що в майбутньому використання генетичних алгоритмів набуватиме все більш масового характеру. Спектр задач, які вирішуються з допомогою даних алгоритмів, а також якість та швидкість вирішення цих задач обґрунтовують довгострокову перспективність та популярність. З огляду на актуальність теми дослідження, стаття розглядає теоретичні основи генетичних алгоритмів та їх застосування для оптимізації коду, а також пропонує декілька шляхів розробки методу оптимізації програмного коду на основі генетичного алгоритму.

Ключові слова: програмний код; генетичний алгоритм; оптимізація програмного коду.

ВСТУП

Актуальність теми. Генетичні алгоритми активно застосовуються у задачах оптимізації завдяки своїй здатності імітувати природний процес еволюції, що дозволяє генерувати й комбінувати різні рішення, поступово вдосконалюючи їх на основі критерію придатності. В галузі оптимізації програмного коду це дає змогу створювати та обирати варіанти коду з найвищою продуктивністю. Проте, незважаючи на попередні дослідження, питання ефективного застосування ГА для автоматичної оптимізації коду



залишається відкритим. Існуючі підходи зазвичай сфокусовані на оптимізації вузьких аспектів (наприклад, продуктивності обраного фрагменту коду), але не охоплюють комплексну оптимізацію на рівні всієї системи, що є актуальним завданням для України з огляду на зростаючі вимоги до ефективного ПЗ у різних галузях.

Аналіз останніх досліджень і публікацій. У світовій практиці генетичні алгоритми використовуються для вирішення різних задач оптимізації, включаючи компіляцію та налаштування ПЗ. Зокрема, такі науковці, як Мітчелл та Голдберг, дослідили теоретичні основи ГА, тоді як інші, такі як Люкас і Тернер, розглядали їхнє застосування в конкретних галузях ПЗ. Проте більшість досліджень зосереджені на прикладних аспектах, а не на універсальних методах оптимізації коду. В цьому аспекті залишається нерозв'язаним завдання комплексної оптимізації, яка б враховувала як продуктивність, так і надійність коду.

Мета і задачі дослідження. Мета дослідження полягає у розробці ефективного методу оптимізації програмного коду на основі генетичних алгоритмів, що дозволяє підвищити продуктивність програмного забезпечення та знизити споживання ресурсів, враховуючи сучасні вимоги до ефективності обчислень. Метою дослідження є аналіз генетичних алгоритмів для створення ефективного методу оптимізації програмного коду. Акцент уваги стоїть на аналізі наявних генетичних алгоритмів а також їх порівняння з іншими сучасними алгоритмами оптимізації. Для ефективного досягнення мети дослідження було поставлено наступні завдання:

1. дослідити основні принципи роботи генетичних алгоритмів та їхню придатність для вирішення задач оптимізації коду;
2. розробити методику застосування генетичних алгоритмів для автоматизованої оптимізації програмного коду.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Генетичні алгоритми (ГА) широко використовуються для вирішення задач оптимізації завдяки своїй здатності моделювати природний еволюційний процес. Це дозволяє створювати й комбінувати різноманітні рішення, поступово їх вдосконалюючи на основі заданого критерію придатності. У сфері оптимізації програмного коду такі алгоритми сприяють генерації та вибору найпродуктивніших варіантів. Проте, попри наявні дослідження, ефективне застосування ГА для автоматизованої оптимізації коду залишається відкритим питанням. Існуючі методи здебільшого орієнтовані на вузькі аспекти, як-от підвищення продуктивності окремих фрагментів коду, але не враховують комплексної оптимізації всієї системи. Це особливо актуально для України, де зростає попит на ефективне програмне забезпечення в різних секторах.

Оптимізація коду є ключовою задачею сучасного програмування, адже продуктивність програмного забезпечення прямо впливає на його швидкодію, ефективність використання ресурсів і масштабованість. Зі збільшенням обсягів даних, ускладненням програмних систем і необхідністю раціонального використання обчислювальних ресурсів значущість оптимізаційних методів лише зростає. Серед цих методів ГА займають важливе місце, оскільки дозволяють знаходити глобальні оптимальні рішення та працювати з великою кількістю варіантів, що відкриває нові можливості для вдосконалення коду без втрати його функціональності.

Генетичні алгоритми (ГА) є потужним методом оптимізації, що базується на принципах природного добору для пошуку оптимальних рішень. Ці алгоритми імітують

еволюційний процес, поступово вдосконалюючи покоління можливих розв'язків шляхом використання біологічно натхнених операцій, таких як селекція, схрещування та мутація. ГА довели свою ефективність у розв'язанні складних задач оптимізації, особливо тих, які передбачають широку площину можливих рішень, або в яких традиційні методи не дають задовільних результатів.

Основні принципи роботи генетичних алгоритмів наведені у табл. 1 [1].

Таблиця 1

Основні принципи роботи генетичних алгоритмів

№	Принципи	Пояснення
1	Імітація біологічної еволюції	ГА базується на концепції еволюції Чарльза Дарвіна, яка передбачає природний добір найбільш пристосованих особин для виживання. У ГА цей принцип застосовується для пошуку найкращих рішень шляхом «еволюції» набору можливих рішень або «популяції». Популяція складається з численних «особин», кожна з яких представляє можливий розв'язок задачі. Метою ГА є поступове вдосконалення популяції шляхом повторного застосування алгоритмічних кроків
2	Кодування розв'язків у вигляді генетичних представлень	У генетичних алгоритмах розв'язки задачі кодуються зазвичай у вигляді рядків з чисел, бітів або символів, які називають «хромосомами». Кожен елемент хромосоми, що відповідає певній характеристиці розв'язку, називається «геном». Вибір схеми кодування є ключовим для успішного функціонування ГА, оскільки вона визначає можливість і точність оцінки кожного рішення
3	Оцінка пристосованості рішень	У ГА кожен розв'язок оцінюється за допомогою спеціальної функції — «функції пристосованості», яка визначає, наскільки добре цей розв'язок відповідає критеріям задачі. Функція пристосованості відіграє вирішальну роль у відборі рішень для подальшої оптимізації. Ті рішення, які мають вищі значення функції пристосованості, отримують більшу ймовірність бути обраними для наступного покоління, що забезпечує поступове покращення популяції

Основні структурні елементи генетичного алгоритму показані на рис. 1.



Рис. 1. Основні структурні елементи генетичного алгоритму

Генетичні алгоритми — це універсальний інструмент, який ефективно вирішує складні задачі оптимізації, моделюючи процеси біологічної еволюції. Основи ГА охоплюють ключові концепції, такі як кодування рішень, оцінка їхньої пристосованості,

а також базові операції: відбір, схрещування та мутація. Завдяки цим механізмам алгоритми здатні адаптуватися та генерувати нові варіанти рішень, що забезпечує високу ефективність у випадках, коли традиційні методи виявляються малопродуктивними. Такий підхід дозволяє досягати кращої продуктивності та точності рішень, що робить ГА перспективним інструментом для вдосконалення програмного забезпечення та оптимізації коду.

Генетичні алгоритми (ГА) є потужним інструментом для вирішення задач оптимізації, і їхній потенціал активно використовується в різних галузях, зокрема для оптимізації програмного коду. Оптимізація коду має важливе значення в умовах сучасного програмного забезпечення, адже висока продуктивність і ефективність ресурсів є ключовими критеріями якості. Використання ГА дозволяє автоматизувати процес пошуку оптимальних рішень у складних багатоваріантних середовищах [5]. Цей підхід імітує еволюційні механізми природного відбору, сприяючи генерації та відбору найбільш ефективних варіантів коду.

Генетичні алгоритми (ГА) вже тривалий час привертають увагу дослідників як один із ефективних інструментів для розв'язання складних задач оптимізації. Особливо значущим є застосування генетичних алгоритмів у галузі оптимізації програмного коду, де вони дають змогу автоматизувати процес пошуку найбільш ефективних рішень. Досвід використання ГА в оптимізації програмного коду є різноманітним і включає численні підходи та методики, що були розроблені для вирішення різних аспектів проблеми.

Генетичний алгоритм є метаевристичним методом, що використовує механізми, подібні до біологічної еволюції, включаючи операції селекції, схрещування та мутації. У контексті оптимізації коду, він працює за алгоритмом, етапи якого представлені на рис. 2.

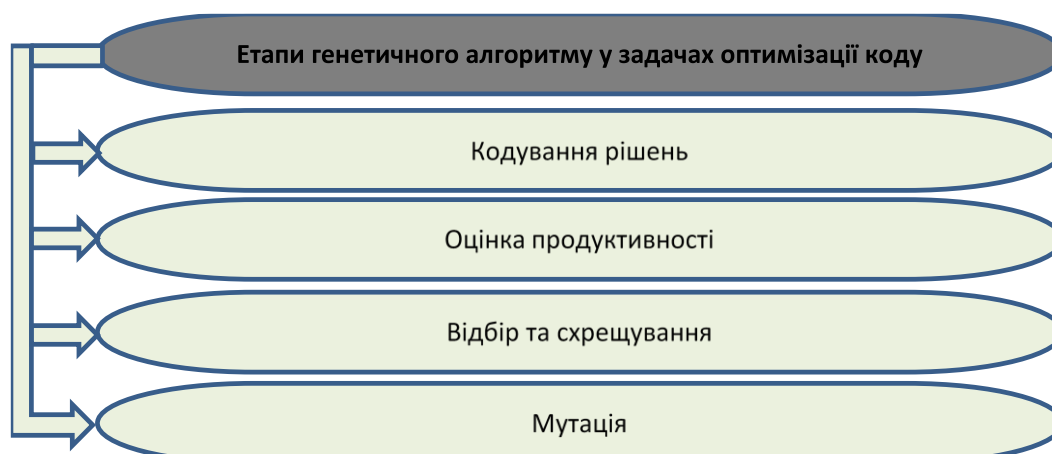


Рис. 2. Етапи генетичного алгоритму у задачах оптимізації коду

Оптимізація програмного коду за допомогою ГА включає ряд етапів.

На початковому етапі визначаються ключові метрики, які потрібно оптимізувати (наприклад, час виконання функції, обсяг пам'яті, використаний на обчислення).

На основі оригінального коду генерується початкова множина варіантів, кожен з яких представляє можливу модифікацію програми.

Виконується тестування кожного варіанту коду, щоб визначити рівень відповідності встановленим критеріям оптимізації.



Процес відбору, схрещування та мутації повторюється доти, поки не буде досягнуто необхідного рівня оптимізації або не буде вичерпано ресурсів (часу або обчислювальної потужності).

Завершальним кроком є тестування та аналіз знайденого рішення на відповідність поставленим вимогам. Якщо рішення не відповідає очікуванням, можуть бути внесені коригування в налаштування алгоритму або цільову функцію.

Генетичні алгоритми мають низку переваг у задачах оптимізації програмного коду:

- гнучкість — ГА легко адаптуються до різних типів завдань і цільових функцій, що дозволяє налаштовувати їх для різних критеріїв оптимізації;
- пошук глобальних рішень — можливість уникати локальних мінімумів і досягати глобальних оптимумів особливо важлива для складних задач оптимізації;
- автоматизація процесу — використання ГА значно скорочує обсяг ручної праці та дозволяє досягти результатів навіть у випадках, коли точні аналітичні методи є недоступними.

Разом з тим, існують певні обмеження:

- високі вимоги до обчислювальних ресурсів — впровадження ГА в роботу може потребувати значної кількості часу та ресурсів, що є критичним чинником при обробці великих обсягів коду;
- чутливість до налаштувань — вибір параметрів, таких як розмір популяції, коефіцієнти мутації та селекції, суттєво впливає на якість і швидкість знаходження оптимального рішення.

В цілому, застосування генетичних алгоритмів для оптимізації програмного коду є перспективним напрямом, що дозволяє автоматизувати пошук ефективних і продуктивних рішень у процесі розробки програмного забезпечення. ГА сприяють підвищенню продуктивності коду, поліпшенню його структури та зниженню ресурсозатратності, що є критичним для сучасних додатків з високими вимогами до ефективності. Хоча використання цього методу має певні обмеження, вдосконалення алгоритмів та збільшення обчислювальних можливостей роблять його все більш доступним і застосовним у реальних умовах програмування.

Розробка нейромережі для автоматичної оптимізації коду чи створення програми, яка безпосередньо змінює код — це надзвичайно ресурсоємні завдання. Навіть на потужному комп'ютері такі підходи можуть призвести до зависання системи та необхідності постійного перезавантаження. Це не лише уповільнює процес розробки, але й робить його фрустраційним та непрактичним для більшості розробників.

Натомість, метод оптимізації параметрів пропонує збалансоване рішення, яке є ефективним, практичним та доступним для широкого кола розробників. Ось як він може допомогти оптимізувати код:

- налаштування критичних параметрів: багато алгоритмів та функцій мають параметри, які суттєво впливають на їх ефективність. ГА допомагає знайти оптимальні значення цих параметрів. Наприклад, стоїть задача розробки алгоритму сортування для великого масиву даних. Оптимальним варіантом буде використання гібридного алгоритму, який поєднує швидке сортування та сортування вставками. Ключовим параметром тут є поріг, при якому алгоритм «перемикається» з одного методу на інший. ГА може допомогти знайти оптимальне значення цього порогу, що призведе до найшвидшого виконання сортування;



- оптимізація ресурсоемних обчислень: для функцій, які виконують складні обчислення, навіть невеликі зміни в параметрах можуть призвести до значного покращення продуктивності. Наприклад, стоїть задача розробки програми для обробки зображень. Для цього використовується функція для застосування фільтру розмиття, яка має параметри розміру ядра та сили розмиття. ГА може допомогти знайти такі значення цих параметрів, які забезпечать найкращий баланс між якістю результату та швидкістю обробки;
- балансування між різними метриками: часто оптимізація одного аспекту може негативно вплинути на інший. ГА дозволяє знайти оптимальний баланс. Наприклад, стоїть задача розробити систему рекомендацій для онлайн-магазину. Використовується функція оцінки, яка має параметри, які впливають на точність рекомендацій та швидкість їх генерації. ГА може допомогти знайти такі значення параметрів, які забезпечать найкращий компроміс між точністю та швидкістю;
- адаптація до різних умов: оптимальні параметри можуть змінюватися залежно від вхідних даних чи умов виконання. ГА дозволяє досить легко перераховувати оптимальні параметри для різних сценаріїв. Наприклад, стоїть задача розробки алгоритму компресії даних. Оптимальні параметри компресії можуть відрізнятися для текстових файлів та зображень. Використовуючи ГА можна швидко знайти оптимальні параметри для кожного типу даних;
- покращення без переписування коду: ГА дозволяють суттєво покращити продуктивність без необхідності повного переписування алгоритму. Наприклад, для певних задач використовується складна функція для аналізу фінансових даних, яка працює повільно на великих обсягах даних. Замість того, щоб переписувати весь алгоритм, можна застосувати ГА для оптимізації ключових параметрів, таких як розмір вибірки чи поріг значущості, що може призвести до значного прискорення без змін у логіці роботи.

ВИСНОВКИ

Метод оптимізації параметрів — це практичний, ефективний та доступний спосіб покращення продуктивності коду. Він не вимагає надпотужних комп'ютерів, не призводить до зависань системи і, при цьому, може дати значні покращення у контексті швидкодії та ефективності програмного коду. Це робить даний метод ледь не ідеальним інструментом для розробників, які прагнуть оптимізувати свій код без надмірних витрат часу та ресурсів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bazhan, V. (2023). The Usage of Genetic Algorithms with Stochastic Processes for the Resolution of Optimization Problems. *Grail of Science*, (33), 253–261. <https://doi.org/10.36074/grail-of-science.10.11.2023.40>
2. Baranovskyi, V. (2015). The Usage of Evolutionary Algorithms for Complex Systems Optimization. *Cybernetics and System Analysis Journal*.



3. Horbenko, A. (2016). Comparison of Evolutionary Optimization Methods. *Compendium of Scientific Papers "Informatics and Computer Science"*.
4. Kryvulia, S. (2017). Genetic Algorithms in Context of Optimization Issues. *Ukrainian Scientific Journal "Computer Technologies and Systems"*.
5. Lysenko, P., & Ivanchenko, M. (2019). The Usage of Genetic Algorithms for Improving of the Software Systems. *Information Technologies in Education and Science*.
6. Shevchenko, V. (2018). The Usage of Genetic Algorithms for the Logistic Systems Optimization. *Science bulletin of KhNU, series: Informatics*.
7. Cheng, L., & Lee, H. (2010). Enhanced Genetic Algorithms for Code Optimization. *ACM Transactions on Software Engineering and Methodology*.
8. Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*.
9. Kannadasan, R., Manoj Kumar, K. N., & Sistla, K. (2016). Code Optimization using Genetic Algorithm. *SSRN Electronic Journal*, 4(11), 1–7.
10. Koza, J. R. (1992). Genetic Programming: On the Programming of Computers by Means of Natural Selection. *Stat Comput*, 4, 87–112. <https://doi.org/10.1007/BF00175355>
11. Nguyen, Q. (2017). Efficient Genetic Algorithm for Code Optimization. *Journal of Optimization Theory and Applications*.
12. Kramer, O. (2017). Genetic Algorithm Essentials. *Genetic Algorithm Essentials, Studies in Computational Intelligence*, 679, 11–19. https://doi.org/10.1007/978-3-319-52156-5_2
13. Pohlheim, H. (2005). *Genetic and Evolutionary Algorithms: Principles, Methods, and Applications*.
14. Simpson, A., & Priest, S. (1993). The application of genetic algorithms to optimisation problems in geotechnics. *Computers and Geotechnics*, 15(1), 1–19. [https://doi.org/10.1016/0266-352X\(93\)90014-X](https://doi.org/10.1016/0266-352X(93)90014-X)
15. Smithson, J., & Lee, H. (2001). *Parallel Genetic Algorithms for Code Optimization*. *IEEE Transactions on Parallel and Distributed Systems*.

**Vladyslav Shorobura**

Master, Software Engineering

State University of Information and Communication Technologies, Kyiv, Ukraine

ORCID ID: 0009-0008-6018-8953

maq4049506@stud.duikt.edu.ua**Bohdan Khudik**

PhD, Senior Lecturer at Department of Software Engineering

State University of Information and Communication Technologies, Kyiv, Ukraine

ORCID ID: 0009-0005-1914-6418

b.khudik@duikt.edu.ua

DEVELOPMENT OF A METHOD FOR OPTIMIZING SOFTWARE CODE BASED ON A GENETIC ALGORITHM

Abstract. Genetic algorithms imply generating & combining of different solutions with the subsequent gradual improvement in accordance to a certain efficiency criteria. In international practice, genetic algorithms are used to solve various optimization issues — in particular, those which concern the software development. The utilization of the genetic algorithms for optimization of the program code is a very promising direction that offers extra automation in terms of search for effective and productive solutions in the software development process. Genetic algorithms contribute to increasing of the code performance, improving its structure and reducing resource consumption, which is crucial in today's competitive world. Despite the fact that this method has certain limitations and flaws, the constant improving of algorithms & increasing of computing capabilities step by step make it widely accessible in the software development circumstances of the real world. Creation of the neural network for automatic code optimization (as well as the development of a program that directly modifies the code) is an extremely resource-intensive task. Even provided that the network of powerful computers is involved into processing, such heavy tasks can lead to system slowdowns, periodic system freezes and the need for constant reboots for stabilization purposes. This not only slows down the development process, but also makes it financially unprofitable for many companies. Genetic algorithms provide an opportunity to optimize the amount of necessary resources, which is an extremely important factor in software development. In general, there is no doubt that in the future the utilization of the genetic algorithms will become increasingly widespread. The range of tasks that are solved using these algorithms (as well as the quality and pace of solving of these tasks) justify their [genetic algorithms'] long-term prospects and popularity. Given the relevance of the research topic, the hereby article considers the theoretical foundations of genetic algorithms and their utilization in regards of the code optimization, and also suggests several ways of development of a method for the optimization of the program code based on the genetic algorithms.

Keywords: software code; genetic algorithm; software code.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Bazhan, V. (2023). The Usage of Genetic Algorithms with Stochastic Processes for the Resolution of Optimization Problems. *Grail of Science*, (33), 253–261. <https://doi.org/10.36074/grail-of-science.10.11.2023.40>
2. Baranovskyi, V. (2015). The Usage of Evolutionary Algorithms for Complex Systems Optimization. *Cybernetics and System Analysis Journal*.
3. Horbenko, A. (2016). Comparison of Evolutionary Optimization Methods. *Compendium of Scientific Papers "Informatics and Computer Science"*.
4. Kryvulia, S. (2017). Genetic Algorithms in Context of Optimization Issues. *Ukrainian Scientific Journal "Computer Technologies and Systems"*.
5. Lysenko, P., & Ivanchenko, M. (2019). The Usage of Genetic Algorithms for Improving of the Software Systems. *Information Technologies in Education and Science*.



6. Shevchenko, V. (2018). The Usage of Genetic Algorithms for the Logistic Systems Optimization. *Science bulletin of KhNU, series: Informatics*.
7. Cheng, L., & Lee, H. (2010). Enhanced Genetic Algorithms for Code Optimization. *ACM Transactions on Software Engineering and Methodology*.
8. Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*.
9. Kannadasan, R., Manoj Kumar, K. N., & Sistla, K. (2016). Code Optimization using Genetic Algorithm. *SSRN Electronic Journal*, 4(11), 1–7.
10. Koza, J. R. (1992). Genetic Programming: On the Programming of Computers by Means of Natural Selection. *Stat Comput*, 4, 87–112. <https://doi.org/10.1007/BF00175355>
11. Nguyen, Q. (2017). Efficient Genetic Algorithm for Code Optimization. *Journal of Optimization Theory and Applications*.
12. Kramer, O. (2017). Genetic Algorithm Essentials. *Genetic Algorithm Essentials, Studies in Computational Intelligence*, 679, 11–19. https://doi.org/10.1007/978-3-319-52156-5_2
13. Pohlheim, H. (2005). *Genetic and Evolutionary Algorithms: Principles, Methods, and Applications*.
14. Simpson, A., & Priest, S. (1993). The application of genetic algorithms to optimisation problems in geotechnics. *Computers and Geotechnics*, 15(1), 1–19. [https://doi.org/10.1016/0266-352X\(93\)90014-X](https://doi.org/10.1016/0266-352X(93)90014-X)
15. Smithson, J., & Lee, H. (2001). *Parallel Genetic Algorithms for Code Optimization*. *IEEE Transactions on Parallel and Distributed Systems*.

