

Г.Б. Мороз

МЕТАЕВРИСТИЧНІ МЕТОДИ ОПТИМІЗАЦІЇ ЯКОСТІ ОБСЛУГОВУВАННЯ КОМПОЗИТНИХ ВЕБСЕРВІСІВ

З появою сервіс-орієнтованих архітектур стало можливим реєструвати, викликати та об'єднувати вебсервіси за їхніми ідентичними атрибутами якості обслуговування для створення композитних вебсервісів з доданою вартістю, які відповідають потребам користувачів. Проте швидке впровадження нових вебсервісів у динамічне бізнес-середовище може негативно вплинути на їхню якість обслуговування. Тому питання про те, як залучити, агрегувати та використати інформацію про якість обслуговування окремих вебсервісів для отримання оптимальної наскрізної якості обслуговування композитного вебсервіса, є наразі одним із пріоритетних напрямків дослідження в програмній інженерії та сервіс-орієнтованих обчисленнях. У цій роботі представлені базова теоретична інформація, необхідна для розуміння важливості, багатогранності та складності проблеми композиції вебсервісів з урахуванням їхньої якості обслуговування, а також репрезентативний огляд використання метаевристичних методів глобальної оптимізації, які протягом останніх двох десятиліть є домінуючими методами вирішення даної проблеми. Мета роботи - привернути увагу студентів та наукової спільноти до актуальних проблем композиції вебсервісів, які виникають в Інтернеті речей, хмарних обчисленнях, соціальних мережах, технологіях мобільних комп'ютерів, смартфонів тощо, і залучити їх до активної участі в розв'язанні цих проблем.

Ключові слова: Сервісно-орієнтоване обчислення, якість обслуговування, композиція вебсервісів, композитний вебсервіс, метаевристика.

H.B. Moroz

METAHEURISTIC METHODS FOR OPTIMIZING THE QUALITY OF SERVICE OF COMPOSITE WEB SERVICES

With the advent of service-oriented architectures, it has become possible to register, invoke, and aggregate web services based on their identical quality of service attributes to create composite web services with added value that meet user needs. However, the rapid introduction of new web services into a dynamic business environment can negatively affect their quality of service. Therefore, the question of how to capture, aggregate, and use information about the quality of service of individual web services to obtain an optimal end-to-end quality of service of a composite web service is currently one of the priority research areas in software engineering and service-oriented computing. This paper presents the basic theoretical information necessary to understand the importance, multifacetedness, and complexity of the problem of web service composition taking into account their quality of service, as well as a representative overview of the use of methods global optimization metaheuristic, which have been the dominant methods for solving this problem over the past two decades. The purpose of the work is to draw the attention of students and the scientific community to the current problems of web service composition that arise in the Internet of Things, cloud computing, social networks, mobile computer and smartphone technologies, etc., and to involve them in active participation in solving these problems.

Key words: Service-oriented computing, quality of service, web service composition, composite web service, metaheuristics.

Вступ

Із кінця XX століття сервіс-орієнтоване обчислення (SOC) є важливою обчислювальною парадигмою, яка змінила спосіб розробки та використання програмних додатків [1], а також стала підґрунтям для еволюції компонентно-орієнтованої інженерії програмного забезпечення до більш прогресивної сервіс-орієнтованої програмної інженерії. У цій парадигмі *вебсервіси* (або

сервіси) розглядаються як фундаментальні будівельні блоки для підтримки швидкої, гнучкої та економічно ефективної розробки розподілених програм у гетерогенних середовищах [2]. Із появою сервіс-орієнтованих архітектур (SOA) стало можливо сервіси реєструвати, виявляти, викликати в розподілених середовищах, публікувати та вільно об'єднувати між собою в один потуж-

ніший, так званий, композитний вебсервіс (CWS, composite web service). Об'єднання (або композиція) сервісів є однією з ключових проблем для технології SOA та сервісів. Воно дозволяє організаціям створювати альянси, передавати функціональні можливості аутсорсингу та надавати своїм клієнтам комплексне обслуговування. З точки зору бізнесу композиція вебсервісів (WSC, *web services composition*) різко знижує вартість і ризики створення нових бізнес-додачків у тому сенсі, що існуючі бізнес-логіки представлені як сервіси та можуть бути повторно використані [3].

З роками популярність і використання сервісів експоненційно зростає. Швидко впровадження соціальних мереж, хмарних обчислень, мереж речей тощо тільки сприяє стрімкому збільшенню доступних в Інтернеті сервісів. Щоденно клієнти по всьому світу генерують мільярди запитів на різноманітні сервіси, а тисячі провайдерів (постачальників, розробників) пропонують нові або модифіковані існуючі сервіси. Серед величезної кількості наявних нині сервісів від різних провайдерів сотні, а то і тисячі можуть мати однакову або дуже схожу функціональність. За такого розвитку подій першорядного значення, як для провайдерів, так і для клієнтів, набуває якість обслуговування сервісів (QoS, *quality of service*) або, у більш вузькому сенсі, нефункціональні властивості сервісів. З одного боку, диференціація сервісів відносно QoS надає можливість провайдерам відрізняти свої сервіси від інших функціонально подібних сервісів, а, отже, QoS стає ключовим інструментом конкуренції провайдерів. З іншого боку, клієнти, оцінюючи QoS сервісів з однаковою функціональністю, отримують можливість визначити, який сервіс, а отже і провайдер, їх найбільше влаштовує. Проте в реальному житті це зробити незавжди просто, оскільки QoS, що надається клієнту, залежить від багатьох як внутрішніх, так і зовнішніх факторів. Зокрема, таких як обчислювальні ресурси провайдера, продуктивність самого сервісу, хостингової платформи, географічне положення користувача, швидкість підключення до Інтернету між користувачами та сервісами тощо [2]. На цьому фоні критичною для техноло-

гії SOA та сервісів стає *проблема композиції вебсервісів з урахуванням QoS (QWSC)*, суть якої зводиться до пошуку серед величезної кількості можливих CWS з однаковою функціональністю такого, наскрізна QoS якого буде оптимальною (з точки зору клієнта) [4]. На ранніх етапах розвитку SOC, коли кількість провайдерів і розміри репозиторію сервісів були незначні, проблему QWSC можна було вирішити за допомогою точних та евристичних методів [5,6]. Ефективність цих методів, які погано масштабувалися, почала швидко зменшуватись при стрімкому зростанні як кількості сервісів, так і їх користувачів. Наразі для проблеми QWSC характерні багатовимірність, нелінійність цільових функцій, конфліктуючі атрибути QoS, величезний пошуковий простір та значний об'єм вхідних даних тощо. Отже ця проблема є NP-складною і для свого вирішення потребує нових підходів та методів. Наразі серед найбільш успішних розглядаються метавристичні методи глобальної оптимізації. На відміну від точних методів оптимізації, метавристика не гарантує оптимальність отриманих рішень. У порівнянні з методами наближення вони не забезпечують доказову точність розв'язку та доказові межі часу виконання. Проте вони здатні забезпечити "практично прийнятні" рішення протягом задовільної тривалості часу.

У статті подано базові знання, необхідні для розуміння особливостей проблеми QWSC, а також представлені сучасні метавристичні методи її вирішення. Завершується робота висновком.

Передмова

В даному розділі надається коротка базова інформація, яка допоможе краще зрозуміти особливості та складність проблеми QWSC.

Сервіси та їхні властивості. Сервіси — це самоописні, самодостатні, слабо пов'язані, незалежні від платформи та багаторазово використовувані компоненти програмного забезпечення, призначені для підтримки взаємодії між машинами в мережі [3,4]. Вони описуються, публікуються, виявляються та викликаються в розподілених середовищах за допомогою набору стандар-

ртів на основі XML (розширювана мова розмітки), включаючи WSDL (мова опису сервісів), SOAP (простий протокол доступу до об'єктів) і UDDI (універсальне виявлення та інтеграція описів).

SAO, яка підприимує життєвий цикл сервісів, складається з трьох основних компонентів: провайдера (постачальника), реєстру сервісів та замовника (користувача, клієнта). На Рис.1 представлена абстрактна модель такої архітектури та взаємозв'язки між її компонентами. Провайдер розробляє сервіс, генерує його опис (WSDL) і публікує його в загальнодоступному реєстрі сервісів (UDDI), роблячи його доступним для виклику. Реєстр містить інформацію для ідентифікації сервісу, включаючи URL-адресу, яка вказує на розташування файлу WSDL, а також технічні відомості про сервіс. Клієнт отримує інформацію про сервіс з реєстру і, якщо вона його влаштовує, використовує відповідний файл WSDL для взаємодії з сервісом через повідомлення SOAP [4].



Рис.1. Модель SOA, що використовується сервісами (адаптовано з [2])

Сервіс вважається повністю представлений, якщо добре описані його як функціональні, так і нефункціональні властивості, виражені через певні атрибути QoS [3]. Функціональні властивості описують операційну поведінку сервісів, тобто, яка інформація потрібна для успішного виклику сервісу та яка інформація буде повернена після його виконання. З формальної точки зору функціональність сервісу як чорного ящика можна розглядати як певне відображення $\mathcal{I}$ набору вхідних даних I в набір вихідних даних O , ($\mathcal{I}:I \rightarrow O$). Оскільки користувача передусім цікавить, *що робить*

сервіс, то пари (I, O) здебільшого достатньо для розуміння цього. В більшості літературних джерел дотримуються саме такого погляду на функціональність сервісів.

Нефункціональні властивості сервісів або атрибути QoS є обмеженнями, визначеними над їхньою функціональністю і використовуються для ранжування сервісів, які мають однакову функціональність. Існує безліч атрибутів QoS для оцінки сервісів з точки зору їхніх нефункціональних властивостей. Проте є декілька атрибутів QoS, які вважаються типовими та поширеними для переважної більшості сервісів від різних провайдерів [4,7]. Це, зокрема:

вартість, $C(S_i)$, що описує, скільки коштує виконання i -им сервісом завдання S_i ;

час відповіді, $T(S_i)$ - тобто час, протягом якого можна отримати відповідь від i -го сервісу після відправлення запиту на завдання S_i . Значення цієї метрики фактично залежать не тільки від QoS цільового сервісу, а й від якості проміжної мережі;

надійність, $R(S_i)$, що описує, наскільки ймовірно, що i -ий сервіс дасть очікувану відповідь на завдання S_i ;

доступність, $A(S_i)$, що описує, наскільки вірогідно отримати доступ до i -го сервіса при необхідності виконання завдання S_i ;

репутація, $Rep(S_i)$, що описує, наскільки завдання i -го сервісу S_i , користується перевагою або довірою серед користувачів.

Композиція сервісів з урахуванням QoS. Композиція сервісів — це керований QoS процес об'єднання (агрегації) кількох існуючих в мережі Інтернет сервісів в новий CWS з доданою вартістю, який має необхідні клієнту функціональність та наскрізну QoS. Наразі найбільш поширеним підходом до (напів)автоматичної WSC є підхід на основі робочого бізнес-процесу [4,5]. Підґрунтям цього підходу є подібність абстрактного CWS до абстрактного бізнес-процесу, який являє собою набір загальних сервісів (службових) завдань із визначеними залежностями між потоком управління та потоком даних.

На Рис. 2 показано два ключові етапи життєвого циклу WSC, а саме етапи **Виявлення** та **Вибору** сервісів. На етапі **Виявлення** для кожного i -го абстрактного сервіса з функціональністю O_i із множини всіх

доступних в Інтернет сервісів створюється клас конкретних сервісів-кандидатів, які також мають функціональністю O_i , але відрізнятися між собою за QoS. Вхідними даними етапу **Вибір** є набір класів сервісів-кандидатів, отриманих на попередньому етапі. Вибираючи з кожного класу по одному сервісу-кандидату, отримаємо множину всіх екземплярів CWS, серед яких необхідно знайти такий, що найкраще відповідає заявленим вимогам та обмеженням користувача до наскрізної QoS.

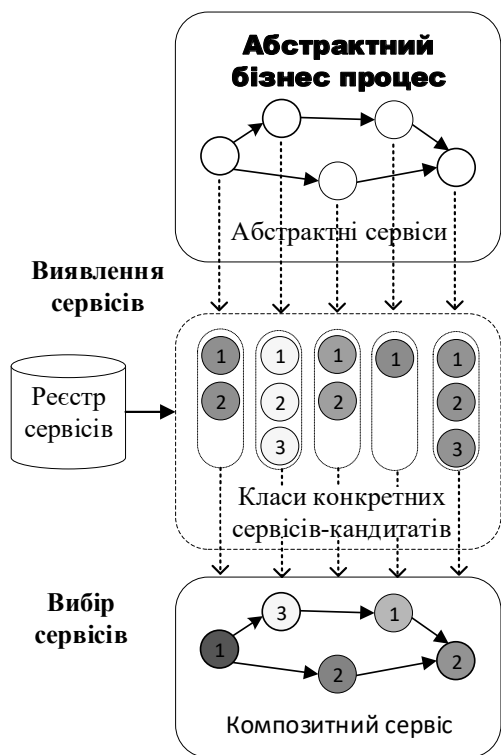


Рис.2. Ключові етапи життєвого циклу композиції сервісу (адаптовано з [4]).

Функції агрегації. В існуючих мовах моделювання бізнес-процесу, таких як BPEL-WS, OWL-S тощо виділяються чотири структури управління завданнями, а саме: послідовна, паралельна, розгалужена та циклічна [8]. Ці ж структури є базовими і для абстрактних CWS (Рис. 3). При *послідовній структурі* завдання S_i виконуються в послідовному порядку, $i=1, \dots, m$. В *паралельній структурі* всі паралельні завдання виконуються одночасно. Перехід до завдання за межами структури відбувається після завершення всіх паралельних завдань. В *розгалуженій структурі* з імовірністю p_i виконується завдання S_i і після його завершення відбувається перехід до завдання за межами структури, $\sum p_i = 1$. В *структурі циклу* сервіси, створені за її допомогою, виконуються неодноразово, доки не буде виконано певну умову, наприклад, кількість циклів l .

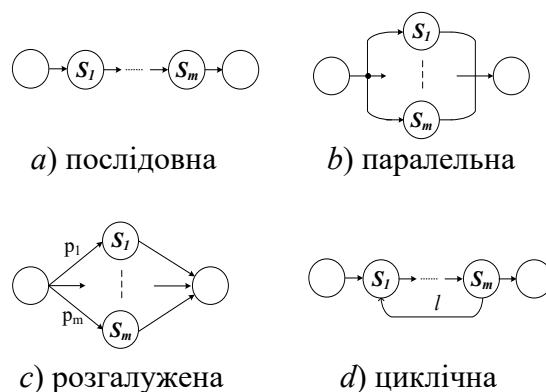


Рис. 3. Базові структури абстрактних CWS

В Табл. 1 наведені функції агрегації для базових структур абстрактних CWS і основних атрибутів QoS [4].

Таблиця 1.

Функції агрегації для QoS, де l — кількість циклів

Атрибут QoS	Структури абстрактних композитних сервісів			
	Послідовна	Паралельна	Розгалужена	Циклічна
Час відповіді	$\sum_{i=1}^m T(S_i)$	$\max_{i=1}^m T(S_i)$	$\sum_{i=1}^m T(S_i) \times p_i$	$l \times \sum_{i=1}^m T(S_i)$
Надійність	$\prod_{i=1}^m R(S_i)$	$\prod_{i=1}^m R(S_i)$	$\sum_{i=1}^m R(S_i) \times p_i$	$\prod_{i=1}^m R(S_i)$
Доступність	$\prod_{i=1}^m A(S_i)$	$\prod_{i=1}^m A(S_i)$	$\sum_{i=1}^m A(S_i) \times p_i$	$\prod_{i=1}^m A(S_i)$
Вартість	$\sum_{i=1}^m C(S_i)$	$\sum_{i=1}^m C(S_i)$	$\sum_{i=1}^m C(S_i) \times p_i$	$l \times \sum_{i=1}^m C(S_i)$
Репутація	$\frac{1}{m} \sum_{i=1}^m Rep(S_i)$	$\frac{1}{m} \sum_{i=1}^m Rep(S_i)$	$\sum_{i=1}^m P(S_i) \times p_i$	$l \times \sum_{i=1}^m P(S_i)$

Формальна постановка задачі QWSC.

Нехай

m – кількість абстрактних сервісів в абстрактному композитному сервісі;

k – кількість атрибутів QoS;

G_i – упорядкована множина конкретних сервісів-кандидатів, що мають функціональність i -го абстрактного сервіса, $1 \leq i \leq m$;

n_i – кількість сервісів в класі G_i ;

x_i – порядковий номер сервісу в класі G_i ,

$(1 \leq x_i \leq n_i)$;

$v_j(x_i)$ – значення j -го атрибута x_i -го конкретного сервісу із G_i , $1 \leq j \leq k$, $1 \leq x_i \leq n_i$;

$\mathcal{R}^m$ – множина всіх можливих CWS

$X = [x_1, x_2, \dots, x_m]$, $1 \leq x_i \leq n_i$, $i=1, \dots, m$;

$f_j(X) = f_j(v_j(x_1), \dots, v_j(x_m))$ – агрегатна функція j -го атрибута QoS CWS, $1 \leq j \leq k$;

Необхідно знайти такий вектор

$$X^* = [x_1^*, x_2^*, \dots, x_m^*], 1 \leq x_i \leq n_i,$$

який оптимізує вектор агрегатних функцій

$$F(X) = [f_1(X), f_2(X), \dots, f_k(X)]$$

У разі наявності обмежень

$$g_i(X) \leq 0, i = 1, \dots, r, \quad r \leq k;$$

$$h_j(X) \geq 0, j = 0, \dots, k-r,$$

де $X^* \in \mathcal{R}^m$ – (суб)оптимальний розв'язок (тобто прийнятний композитний сервіс).

Метаевристичні методи оптимізації QoS композитних сервісів

Одними з перших метаевристичних методів, використаних для вирішення проблеми QWSC, були метаевристики на основі траєкторії, такі як табу пошук [9] та моделювання відпалу [10]. Проте згодом вони поступились потужнішим метаевристичним методам на основі популяції, зокрема, алгоритмам ройового інтелекту, еволюційним алгоритмам тощо. Оскільки проблема QWSC має тенденцію ускладнюватися, то з часом виявилось, що застосування будь-якого одного

алгоритму оптимізації далеко не завжди приводить до успіху. Теоретичним підґрунтям для розуміння подібних емпіричних фактів є відома NFL-теорема Волперта - Макреді про відсутність безкоштовного обіду, яка стверджує, що не існує універсального алгоритму оптимізації, який може розв'язувати всі проблеми краще за інші алгоритми. Одним із очевидних шляхів подолання негативних наслідків NFL-теорема і підвищення ефективності вирішення завдань глобальної оптимізації є розробка гібридних алгоритмів, які об'єднують різні або однакові алгоритми, але з різними значеннями вільних параметрів. Мотивацією гібридизації є можливість подолання недоліків окремих алгоритмів, не втрачаючи їхніх переваг, що робить гібридні алгоритми ефективнішими за автономні.

Еволюційні алгоритми

Зазвичай під термінами «еволюційні алгоритми» або «еволюційне обчислення» розуміють велику групу алгоритмів до якої входять, зокрема, генетичні алгоритми (genetic algorithm, **GA**), еволюційні стратегії, еволюційне та генетичне програмування (genetic programming, **GP**), диференціальна еволюція тощо. Вони мають загальну концептуальну базу моделювання еволюції окремих структур і відрізняються за способом формулювання задачі, процесами відбору і використання операторів репродукції (відтворення).

Генетичні алгоритми. Canfora G. та ін. [11] першими застосували генетичний алгоритм для композиції сервісу з урахуванням QoS. Ними був описаний підхід для швидкої, грубозернистої WSC на основі GA, який виявився більш масштабованим, однак повільнішим за цілочисельне програмування. Мета підходу полягала в забезпеченні швидкого способу знаходження субоптимального композитного сервісу. В роботі [12] запропоновано GA, що характеризується схемою кодування хромосом спеціальною матрицею відношень і обробкою різноманітності популяції з моделюванням відпалу. Матриця відношень має здатність одночасно представляти перепланування композитного сервісу, циклічні шляхи та багато сценаріїв сервіса, таких як імовірні-

сний виклик, паралельний виклик, послідовна активація тощо. Zhang C., Ma Y. [13] для вибору сервісів із глобальними обмеженнями QoS запропонували швидкий конвергентний GA з розширеною політикою початкової популяції та еволюційною політикою на основі схеми кодування матриці відношень. А також динамічний GA, *представлений* політикою оцінки динамічної еволюції на основі схеми кодування генома матриці відношень, що забезпечує кращі плани CWS. Vanrompay Y. та ін. [14] для мобільних систем, що складаються з кількох вузлів, запропонували GA для ефективного пошуку майже оптимального щодо загальних атрибутів QoS складу композитного сервісу та його розгортання на наборі підключених вузлів таким чином, щоб розподіл відповідав заданим обмеженням якості обслуговування (QoS) та мінімізував вартість зв'язку між вузлами.

Gong X.R. та ін. в [15] представили GA композиції сервісу, який підтримує глобальне оптимальне та динамічне перепланування. Алгоритм використовує схему кодування матриці позицій, щоб одночасно виражати всі композитні шляхи та інформацію про перепланування QoS. Jiang Z.Y. та ін. [16] запропонували модель оптимізації запитів і відповідний GA на основі багатоатрибутного агрегування QoS різних сервісів для індивідуальності та ефективності в оцінках. Модель налаштовує QoS з глобальними обмеженнями та уподобаннями користувачів, динамічною схемою рейтингу та багаторівневою відповідністю. Авторами роботи [17] для вирішення проблеми QWSC по-перше, представлено GA відновлення з мінімальними конфліктами, в якому окремі особини можуть бути швидко виправлені за допомогою евристики впорядкування значень, коли вони порушують обмеження міжсервісних залежностей і конфліктів. По-друге, для вирішення проблеми розподілення CWS запропоновано GA на основі штрафів, оскільки розумне розподілення композитних сервісів між паралельними серверами може збільшити пропускну здатність. У GA вбудований швидкий локальний оптимізатор та ранговий відбір із елітарністю. Сао J. та ін. [18] запропонували удосконалений GA для оптимізації якості

моделі процесу обслуговування сервісів, яка спочатку перетворюється на дерево структури процесу для розрахунку якості, а потім GA використовується для уточнення процесу обслуговування. В роботі [19] представлено інтегровану структуру запитів до сервісів, яка включає в себе модель запитів до сервісів і двофазову стратегію оптимізації. Модель запиту визначає сервісні спільноти для організації великого та неоднорідного простору обслуговування. Двофазова стратегія оптимізації з жадібним алгоритмом і GA здатна «спільно розвивати» кілька можливих планів виконання одночасно. Klein A. та ін. [20] описали мережевий підхід до композиції сервісів у хмарі, що складається з мережевої моделі розрахунку QoS та алгоритму вибору, який ґрунтується на GA. Ukor R. та ін. в [21] представили задачу оптимізації вибору сервісу з поняттям маркерів запиту та трьома підходами (цілочисельне програмування, евристичний пошук по дереву та GA) для вирішення цієї задачі. Автори роботи [22], розглядаючи одночасно функціональні та нефункціональні вимоги, запропонували семантичний, заснований на GA, підхід до автоматичної композиції сервісів. У роботі [23] запропоновано коеволюційний GA для створення сервісу на основі QoS, який повністю враховує індивідуальні відносини між популяціями.

Багатоцільові GA. Claro D.B. та ін. [24] для вибору сервісів для оптимальної композиції реалізували багатоцільовий еволюційний підхід, заснований на NSGA-II (генетичний алгоритм непомітованого сортування), який ідентифікує набір оптимальних рішень за Парето без введення ранжирування серед різних параметрів QoS. В [25] також використано структуру NSGA-II для композиції сервісів з урахуванням QoS. Водночас для полегшення ухвалення рішення споживачам сервісів надавали кілька різних можливих розв'язків. Hashmi K. та ін. [26] на основі NSGA-II представили сервіс переговорів, який використовується як споживачем, так і сервісом провайдера для проведення переговорів щодо залежних параметрів QoS. Авторами [27] запропоновано алгоритм глобальної багатоцільової оптимізації WSSPEA2. Суть його полягає в

тому, що задача вибору вебсервісів на основі QoS трансформується в багатоцільову задачу оптимізації композиції сервісів з обмеженнями QoS. Силовий еволюційний алгоритм Парето (SPEA2) використовується для отримання набору оптимальних за Парето рішень за допомогою одночасної оптимізації ряду цільових функцій, і користувачі можуть вибрати одне з цих рішень за своїми перевагами. В [28] представлено алгоритм GODSS (глобальний оптимальний вибір динамічних сервісів) для вирішення динамічного вибору сервісів із глобальними оптимальними QoS у композиції сервісів. Суть алгоритму полягає в тому, що задача динамічного вибору сервісу з глобальними оптимальними QoS трансформується в багатоцільову оптимізацію композиції сервісів з обмеженнями QoS. Wang J. та ін. [29] запропонували багатоцільовий GA для оптимального вибору сервісів на основі QoS, враховуючи такі обмеження у процесі вибору сервісу, як структура управління в плані композиції, взаємозв'язок між конкретними сервісами та компроміс між кількома атрибутами QoS. В [30] запропоновано багатоцільовий GA, на ефективність якого не впливає еволюціонуючий набір рішень на кожній ітерації. Набір оптимальних рішень за Парето будується за правилом псевдобінарного дерева. Потім оптимальні за Парето рішення упорядковуються, і придатність кожного оптимального рішення за Парето визначається подібністю окремих рішень. Wagner F. та ін. [31] описали розширену операцію відновлення та надали багатоцільовий алгоритм оптимізації, який використовує базові знання для виявлення надійних оптимізованих за QoS виборів сервісів у відкритому сервісному середовищі. Алгоритм враховує вартість потенційних збоїв та ранжує рішення на основі переваг ризику особи, яка ухвалює рішення. Ramirez A. та ін. [32] досліджували придатність еволюційного алгоритму з багатьма цілями для вирішення проблеми зв'язування сервісів на основі реального тесту з дев'ятьма атрибутами QoS.

Гібридні GA. Ма Х. та ін. [33] вперше обговорили обмеження методу простих адитивних ваг через те, що ці ваги не мають фізичного значення і їх важко встановити.

Потім пропонується новий ефективний обчислювальний підхід, лінійне фізичне програмування у співпраці з GA для вибору сервісу, керованого якістю. Liang W. та ін. [34] запропонували гібридний алгоритм для композиції сервісів, який включає GA та грубу теорію множин. Цей алгоритм може: (i) вирішувати проблеми, які можна розкласти на функціональні вимоги, і (ii) покращувати продуктивність GA, виявляти нездійснені рішення та винятки шляхом зменшення діапазону домену початкової популяції та обмеженого перетину, використовуючи грубу теорію множин. Liu Z. та ін. [35], базуючись на декомпозиції глобальних обмежень QoS, запропонували метод динамічної композиції, який складається з трьох етапів: (i) глобальні обмеження QoS розкладаються на локальні обмеження культурного GA; (ii) правила визначення QoS визначають значення QoS сервісів-кандидатів; (iii) найкращі сервіси, що задовольняють локальні обмеження, вибираються для кожного завдання протягом часу виконання. Que Y. та ін. [36] базуючись на ідеї декомпозиції QoS, реалізували гібридне рішення з інформаційною ентальпією, успадкованою від штучної імунної системи, для вирішення проблеми детермінованої ймовірності оператора GA, яка призводить до проблеми передчасної конвергенції. В [37] зроблено керований кластером GA шляхом впровадження k-середніх у процес генерації початкової популяції. Крім того, Jatoth C. та ін. [38] представили оптимальну композицію хмарного сервісу з використанням GA на основі адаптивної еволюції генотипу, що має справу з кількома атрибутами QoS і забезпечує рішення, які задовольняють баланс параметрів QoS і обмежень підключення композиції сервісу

Генетичне програмування. В роботі [39] запропоновано алгоритм GP композиції сервісів, який може об'єднувати сервіси, використовуючи різні керуючі структури, генерувати композиції відповідно до контекстно-вільної граматики та явно керувати оновленням атрибутів. А також мінімізує кількість сервісів і шукає композиції з мінімальним шляхом виконання. Авторами роботи [40] запропоновано алгоритм адаптивного GP, спрямова-

ний на створення бажаних виходів на основі доступних вхідних даних, а також на забезпечення того, щоб композитний сервіс мав оптимальне значення QoS. В роботі [41] представлено підхід на основі GP до композиції розподіленого сервісу з урахуванням QoS. Вплив каналів зв'язку на атрибути QoS вибраних сервісів розглядається явно. Щоб впоратися з розподіленим середовищем сервісів, була прийнята мережева система координат для оцінки часу, витраченого на зв'язок між сервісами, а також між сервісами та кінцевими користувачами.

Гібридне GP — це особливе застосування GA, яке широко використовується у повністю автоматизованій стратегії композиції з графовим поданням. У повністю автоматизованій композиції абстрактні робочі процеси створюють вихідно-вхідні зв'язки між сервісами, де оператор алгоритму використовується в еволюційному процесі для вибору сервісу композиції без обмежень. В [42] пропонується гібридний підхід до складання сервісів, який поєднує GP та випадковий жадібний пошук. Жадібний алгоритм використовується для генерації допустимих та локально оптимізованих особин для GP та для виконання операцій мутації під час генетичного програмування. Yu Y. та ін. [43] пропонують гібридний підхід, який об'єднує використання GP та табу-пошуку для композиції сервісу з інтенсивним використанням даних з урахуванням QoS. Ефективність запропонованого підходу оцінюється за допомогою загальнодоступних наборів даних. У статті [44] запропоновано комбінацію GP та випадкового жадібного пошуку для композиції сервісу. Жадібний алгоритм використовується для створення дійсних і локально оптимізованих особин для заповнення початкового покоління для GP та для виконання операцій мутації під час генетичного програмування.

Алгоритми роювого інтелекту

Це парасольковий термін для великої групи алгоритмів, до якої, зокрема, входять алгоритми *оптимізації рою частинок* (particle swarm optimisation, **PSO**), *оптимізації мурашиної колонії* (ant colony optimisation, **ACO**), оптимізації штучної

бджолоїної колонії, (artificial bee colony algorithm, **ABC**) тощо.

Оптимізація рою частинок. Wang W. та ін. [45] для вирішення проблеми QWSC запропонували вдосконалений алгоритм оптимізації рою частинок (iPSOA) із стратегіями нерівномірної мутації та адаптивного коригування ваги для покращення швидкості конвергенції на глобальному та локальному рівнях відповідно. А також представлено модель пулу сервісів, яка допомагає користувачам зберегти більше композитних планів за один процес вибору та ефективний алгоритм побудови пулу сервісів, заснований на вдосконаленій оптимізації дискретного рою частинок (IDPSO). Водночас стратегія нерівномірної мутації була введена в глобальну найкращу частинку в IDPSO для підвищення якості складних планів у пулі сервісів. Wang S. та ін. [46] запропонували швидкий PSO у середовищі хмарних обчислень для створення хмарних CWS. Спочатку оператор горизонту повинен видалити надлишкові кандидати CWS, а потім PSO вибирає CWS із решти кандидатів для створення композитних сервісів. В роботі [47] розроблено PSO для полегшення динамічного вибору сервісу, в якому запропоновано три типи операторів швидкості та одне рівняння еволюції положення. Критерій по-hope/re-hope гарантує різноманітність PSO та покращує можливість глобального пошуку. Wang L. та ін. [48] розробили автоматичний алгоритм композиції сервісів, заснований на глобальній оптимізації QoS та хаотичному PSO. Водночас модель вибору сервісів із глобальною оптимізацією QoS перетворюється на задачу багатоцільової оптимізації. В роботі [49], для реалізації ефективної інтеграції та уніфікованого управління ресурсами *логістичного центру* запропоновано метод вираження логістичних ресурсів та інкапсуляції сервісів, щоб знайти найкращий конкретний сервіс, прив'язаний до кожного відповідного абстрактного сервісу. Автори [50] запропонували групування PSO на основі переваг користувача у виборі сервісу. Алгоритм не тільки досягає балансу між швидкістю конвергенції та розмаїттям роїв, а й дозволяє максимально задовольнити користувачів за рахунок стратегій угруповання частинок, розвитку різних кластерів окремо,

своєчасного оновлення кожного кластера та використання нечітких обмежень для вираження уподобань користувача. В роботі [51] об'єднано PSO з алгоритмом присвоєння Мункреса для композиції сервісу, який відрізняється від споріднених робіт двома аспектами: (i) для вирішення проблеми оптимізації вводяться два метаевристичні методи, засновані на PSO та (ii) одночасно обробляються декілька запитів робочого процесу.

Багатоцільова PSO. Сао J. та ін. [52] запропонували багатоцільовий алгоритм вибору сервісів на основі PSO, який моделює проблему вибору сервісів як обмежену оптимізаційну задачу з кількома цілями. Сервіси в кожному наборі суб-сервісів спочатку сортуються відповідно до концепції домінування Парето. Потім створюється новий набір суб-сервісів, розмір якого набагато менший за початковий, і, нарешті, виводиться оптимальний набір за Парето. Автори роботи [53] перетворили задачу ранжирування top-k у задачу багатокритеріального програмування, коли переваги користувачів розглядаються разом із запитами користувачів. Потім пропонується вдосконалений дискретний PSO, щоб зведені рейтинги задовольняли як запити користувачів, так і їхні переваги. Guha T. та ін. [54] запропоновано та порівняно два алгоритми вибору сервісів в Grid: багатоцільовий алгоритм оптимізації рою частинок з кількома цілями з використанням методу відстаней скупченості (MOPSO-CD) до алгоритму встановлення відповідності на основі задоволення обмежень (CS-MM). Fan X. [55] побудував ефективний багатоцільовий алгоритм PSO з кількома обмеженнями за допомогою нішевої технології для вирішення проблеми вибору та композиції персоналізованого сервісу. Авторами [56], враховуючи кореляції між ресурсними сервісами, запропоновано багатоцільовий PSO для вирішення проблеми вибору та композиції ресурсів множинної мережі. Основні особливості: (i) він поєднує техніку недовіреного сортування для вибору найкращих глобальної та локальної позицій; (ii) формула оновлення частинок динамічно змінюється для досягнення компромісу між глобальним дослідженням та локальною експлуатацією; і (iii) для підтримки різноманітності популяції

застосовуються оператори обрізання популяції на основі перестановок і цілей.

Гібридна PSO — це одна із найвідоміших груп алгоритмів ройового інтелекту, яка у існуючій літературі згадується як друга за частотою гібридна метаевристика. Xu X. та ін. [57] для збільшення ефективності пошуку оптимального композитного сервіса запропонували гібридний алгоритм оптимізації хаосу (PS-CTPSO), який є поєднанням PSO та стратегії хижацького пошуку. Liu Y. та ін. [58] для вирішення проблеми композиції сервісів пропонують гібридний алгоритм оптимізації рою квантових частинок (HQPSO), який поєднує PSO з деякими принципами квантової механіки. Yin H. та ін. [59] для покращення різноманітності роїв впроваджують генетичні оператори в PSO. Крім того, Wang S. та ін. [60] запропонували покращити ефективність PSO за допомогою методів горизонту для видалення надмірностей у сховищі *сервісів*. У цьому ж контексті Hossain M. S. та ін. [61] для розв'язання проблеми отримання оптимальної композиції за мінімальний час із численних наборів динамічних сервісів, запропонували гібридний алгоритм, який поєднує PSO і кластеризацію k-середніх. Він працює паралельно за допомогою MapReduce на платформі Hadoop. Це важливо для обробки великих обсягів різноманітних даних і сервісів із різних джерел у мобільному середовищі. Chifu V. та ін. [62] застосували метод кластеризації даних для визначення щільної області у просторі пошуку для створення надійної можливості пошуку для знаходження глобальних оптимумів та уникнення локальних пасток. Gharbi M. та ін. [63] запропонували метод під назвою реляційний аналіз концепції (RCA) для зменшення простору пошуку. Евристичні методи, такі як алгоритм клонального відбору [64] і гармонійний пошук [65], об'єднані з оператором горизонту та стратегією хижацького пошуку [57], для підвищення можливостей пошуку традиційного PSO. В [66] реалізовано повністю автоматичну композицію сервісів за допомогою алгоритму планування. Гібридний PSO з метаевристиками, включаючи алгоритм штучної імунної системи [67] і гармонійний пошук [65], запропонований

для гарантування тонкого балансу між дослідженням та експлуатацією. Naytamy S. та ін. [68] розробили двоступінчасту гібридну модель, в якій вихідні дані рекурентної нейронної мережі LSTM подавалися в PSO для перетворення даних QoS у послідовну інформацію. Крім того, Hosseinzadeh M. та ін. [69] розробили гібридний PSO зі штучною нейронною мережею для покращення параметрів QoS. Sun та ін. [70] запропонували підхід швидкого вибору сервісів, який повністю використовує PSO з адаптивними стрибками та нечітким логічним керуванням для адаптивного розкладання глобальних обмежень QoS на локальні з адаптивним рівнем якості. В [71] для проблеми QWSC запропоновано гібридний PSO з принципом клонального відбору та кількома корисними стратегіями. Удосконалена локальна стратегія «кращий перший» представлена для досягнення однакових впливів на локальну придатність компонентного завдання та функцію придатності композитного сервісу, коли замінюється інший сервіс-кандидат. У роботі [72] запропоновано новий алгоритм кооперативної еволюції на основі PSO та SA, який успадковує швидку конвергенцію PSO та глобальну конвергенцію SA, а також долає недоліки схильності до захоплення локального оптимуму PSO та повільної конвергенції SA. Багатоцільова PSO розроблена також для багатоцільової композиції сервісів із глобальною оптимізацією QoS. У роботі [73] представлена модель вибору сервісів на основі PSO для визначення оптимальної комбінації сервісів шляхом оцінки атрибутів QoS для динамічної програми електронного бізнесу. Модель реалізована багатоагентною системою, де агенти можуть представляти автономних запитувачів сервісів, постачальників сервісів та інші функції в програмах електронного бізнесу. Fethallah H. та ін. [74] для проблеми QWSC запропонували реактивне багатоагентне рішення на основі локальної версії PSO, у якій кожна група має повну сітчасту топологію, яка більш несприйнятлива до локальних оптимумів, ніж глобальна версія.

Оптимізація мурашиної колонії.

Zheng X. в [75] представив алгоритм вибору сервісу з урахуванням QoS, до якого вклю-

чена модель ступеня задоволеності користувача для грид-сервісів і розширений АСО з новим правилом мурашиного клонування. Авторами роботи [76] запропоновано модель мінімізації вартості композиції сервісів для додатків з інтенсивним об'ємом даних і розроблено відповідний алгоритм АСО для її реалізації. Сервіси навколишнього медіа в середовищі моніторингу розумного будинку повинні бути повсюдними, адаптивними та надійними щодо доступу та доставки. Hossain M. та ін. [77] запропоновано структуру вибору сервісів у “розумних” середовищах, яка використовує потенціал підходу до вибору сервісів на основі мурашок, враховуючи динамічність уподобань та задоволеності користувачів. А це є ключовим фактором для вибору медіа-сервісів, щоб знайти найкращий спосіб вибору відповідних медіа-сервісів. Це також дозволяє різним категоріям мешканців отримувати доступ до різноманітних медіа-сервісів таким чином, що їхній досвід оптимізується з огляду на навколишнє середовище. В роботі [78] запропоновано натхненний мурашками метод вибору семантичного сервісу, який використовує граф композиції та багатокритеріальну функцію для визначення оптимального рішення композиції відповідно до вподобань користувача щодо QoS та семантичної якості. Щоб підвищити ефективність і точність процесу виявлення сервісів, вони організували набір доступних сервісів зі схожою функціональністю в кластери сервісів. Wang R. та ін. [79] у поєднанні з механізмом позитивного зворотного зв'язку АСО спочатку проаналізували проблему вибору сервісів із базовим принципом АСО, а потім перетворили проблему вибору сервісів з урахуванням QoS у проблему найкоротшого шляху. Нарешті вони вказали кроки вирішення проблеми вибору сервісу на основі АСО та порівняли ефективність алгоритму за різними параметрами. В роботі [80] представлено жадібний алгоритм під назвою Greedy-WSC і алгоритм на основі оптимізації колонії мурашок під назвою АСО-WSC, які намагаються вибрати можливі комбінації хмар і використовувати мінімальну кількість хмар. Експериментальні результати показують, що запропонований метод оптимізації мурашиної

колонії може ефективно та результативно знаходити хмарні комбінації з мінімальною кількістю хмар.

Багатоцільова АСО. Fang Q. та ін. [81] застосували багатоцільовий алгоритм оптимізації мурашиної колонії для вирішення проблеми вибору динамічного сервісу. Модель вибору сервісу з глобальними обмеженнями QoS перетворюється на задачу багатоцільової оптимізації з обмеженнями користувача. Zhang W. та ін. [82] запропонували багатоцільове моделювання вибору оптимального шляху для композиції динамічних сервісів на основі QoS. Представлено стратегію декомпозиції композитного сервісу із загальною структурою потоку на паралельні шляхи виконання, яка є масштабованою для підтримки композиції дуже складних сервісів. Dahan F. та ін. в [83] розробили алгоритм оптимізації колонії літаючих мурашок під назвою FACO (Flying Ant Colony Optimization), який модифікує АСО для вирішення проблеми композиції сервісу з урахуванням QoS. Вони також представили алгоритм EFACO (Enhanced Flying Ant Colony Optimization), який зменшує складність обчислень FACO, запровадивши в ньому три вдосконалення 1) щоб уникнути проблеми з часом виконання, EFACO обмежує процес польоту лише тоді, коли якість рішення покращується; 2) щоб уникнути сканування всіх сусідніх вузлів застосували метод вибору сусідніх вузлів; 3) ввели третю модифікацію, яка перетворила алгоритм на мультиферомонний алгоритм.

Гібридна АСО. Yang Z. та ін. [84] розробили гібридний АСО, за допомогою якого використовувався генетичний алгоритм для вибору критичного параметра для АСО. Генетичний алгоритм відомий своїм гнучким, надійним механізмом пошуку, але інколи він страждає від відсутності надійних можливостей пошуку глобальних оптимумів. З іншого боку, АСО має репутацію глобального пошукового алгоритму, але страждає від повільної конвергенції, особливо великомасштабних проблем. Отже, Yang Y. та ін. [85] запропонували гібридизацію, де алгоритм мурашиної колонії служив зачатком генетичної операції. Liu Z.-Z. та ін. [86] інтегрували

систему мурахи в алгоритм культури. Погана стагнація була описана як підводне каміння АСО в існуючій літературі. Alayed H. та ін. [87] покращили алгоритм АСО, щоб збільшити різноманітність і уникнути стагнації, втіливши процес обміну. Гібридний метод композиції динамічних сервісів [88] представлено на основі методу оптимального шляху зваженого орієнтованого ациклічного графа, який поєднує АСО та GA для проблеми QWSC. Пропонується новий динамічний генетичний гібридний алгоритм мурашиної колонії [89] для визначення часу запуску генетичних алгоритмів і алгоритмів мурашиної колонії для найкращої стратегії оцінки злиття. Отже, здатність до оптимізації максимізується, а загальна швидкість конвергенції прискорюється. В роботі [90] для проблеми QWSC пропонується алгоритм оптимізації C-MMAS (culture max-min ant system), отриманий шляхом інтеграції системи Max-Min ant у структуру культурного алгоритму. Алгоритм складається з простору популяцій, простору переконань і протоколів зв'язку між ними. Популяційний простір містить ряд рішень. Простір переконань зберігає кращі рішення, отримані в еволюційному процесі популяційного простору; ці рішення розглядаються як досвід і знання та використовуються для скерування еволюції C-MMAS у популяційному просторі. Окрім того, розроблено комплексну модель (DGQoS) оцінки CWS, заснований на загальній QoS та доменній QoS, і новий алгоритм оптимізації C-MMAS для вирішення проблеми вибору сервісу на основі DGQoS.

В останні роки зі швидким розвитком мереж мобільного зв'язку продовжують з'являтися деякі нові сервіси, такі як хмарна віртуальна реальність, голографічний зв'язок тощо. Тому потрібні більш гнучкі та інтелектуальні алгоритми композиції сервісів. Виходячи з цього, в [91] запропоновано стратегію композиції сервісу в багатохмарному середовищі, а також алгоритм АСО на основі механізму мультиферомонів для оптимізації QoS. Щоб уникнути появи локальних оптимумів, ми додатково вводимо операцію мутації генетичного алгоритму.

Інші метаевристики композиції сервісів

Jiang B. та ін. [92] для підвищення ефективності композиції сервісу запропонували підхід на основі вдосконаленого алгоритму феєрверків (fireworks algorithm). В [93] представлено розширений багатоцільовий алгоритм диференціальної еволюції для пошуку репрезентативного набору рішень із хорошою близькістю та дистрибутивністю. У [94] для вирішення проблеми композиції сервісу використовується варіант алгоритму пошуку гармонії (Harmony Search), який знаходить оптимальний композитний сервіс, що задовольняє локальні та глобальні обмеження користувача щодо атрибутів якості. Ghobaei-Arani M. та ін. [95] представили алгоритм *пошуку зозулі* (cuckoo search, CS), який здійснює композицію сервісу для покращення QoS у розподіленому хмарному середовищі. Dahan F. [96] запропонував покращений алгоритм оптимізації кита (IWOA), який містить три різні стратегії для підвищення продуктивності базового алгоритму оптимізації кита (Whale Optimization Algorithm, WOA) за рахунок збільшення швидкості збіжності, яка створює низьку точність рішення для проблеми композиції сервісів. У [97] для знаходження композиції хмарного сервісу з оптимальними значеннями QoS запропоновано гібридний алгоритм, який поєднує алгоритми стратегії орла (Eagle Strategy Algorithm) з WOA, що покращало швидкість збіжності та забезпечило належний баланс між дослідженням та експлуатацією. Dahan F. [98] пропонує алгоритм, заснований на поведінці мікрокажанів під час полювання на здобич. Запропонований алгоритм усуває деякі недоліки класичного алгоритму кажана (Bat Algorithm, BA) і визначає оптимальну комбінацію сервісів для задоволення потреб користувача. У роботі [99] вдосконалено алгоритм штучної бджолої колонії, щоб зробити його більш придатним для проблеми композиції сервісів. Запропоноване вдосконалення контролює стратегії експлуатації та дослідження таким чином, щоб заохочувати дослідження на ранніх стадіях, а експлуатацію на більш пізніх стадіях. Pop C. та ін. [100] представили

гібридний алгоритм, який поєднує навчання з підкріпленням із метаевристиками CS та табу пошук. У цьому алгоритмі навчання з підкріпленням відстежує заміну сервісу, тоді як табу пошук використовується для оптимізації процесу вибору з точки зору часу виконання та досліджуваного простору пошуку. Простір пошуку моделюється як структура розширеного графа планування, яка кодує всі можливі рішення композиції для заданого запиту користувача. Щоб встановити, чи є рішення оптимальним, атрибути QoS сервісів, а також семантична подібність між ними, розглядаються як критерії оцінки. Chifu V. та ін [101] запропонували алгоритм оптимізації спаровування медоносних бджіл у поєднанні з компонентами генетичного алгоритму, табу пошуку та навчання з підкріпленням. У роботі [102] запропоновано алгоритм пошуку вусиків жука (Beetle Antennae Searching Algorithm) інтегрувати в PSO для створення кращої початкової популяції та, як наслідок, досягнення швидшої конвергенції. Dahan F. та ін. [103] представили гібридний алгоритм між ABC та CS, щоб збільшити ефективність композиції сервісів. Алгоритм CS використовується для подолання повільної конвергенції ABC, дозволяючи покинутим бджолам покращити свій пошук і перекрити локальний оптимум. Ahanger T. та ін. [104] запропонували гібридний алгоритм між ABC і BA для досягнення кращого компромісу між локальним використанням і глобальним пошуком. Peng та ін. [105] запропонували багатокласстерну адаптивну оптимізацію мозкового штурму, об'єднану з подвійною опорною векторною машиною для зменшення простору пошуку.

Висновки

З появою і розвитком SAO еволюція складності проблеми композиції сервісів проходила в напрямку її зростання у міру збільшення кількості та асортименту сервісів, обчислювальних парадигм та можливостей мереж Інтернет. Водночас методи її розв'язання переходили від точних (без евристичних) підходів до майже оптимальних евристик і, нарешті, до метаевристичних глобальної оптимізації, які є домінуючими протя-

гом останніх двох десятиліть. Проведений в даній роботі репрезентативний огляд і аналіз метаевристичних методів дозволяє зробити ряд висновків, зокрема:

Фундаментальна мета дослідження композиції сервісу полягала в досягненні швидкої конвергенції та стабільних методів, які можуть знайти високоякісні рішення для користувачів, в умовах динамічно змінного середовища сервісів, незалежно від складності проблеми QWSC.

Метаевристики значною мірою виправдали покладені на них очікування. Проте поява останніми роками нових обчислювальних парадигм з обмеженими ресурсами, таких як інтернет речей, туманне та хмарне обчислення тощо суттєво ускладнили проблему QWSC. Емпіричні дані показують, що досягнення фундаментальної мети в рамках автономних метаевристик стає все складнішим. Швидка конвергенція, уникнення локального захоплення, робота з великим простором пошуку та досягнення високоякісного рішення зміщують дослідження композиції сервісів у бік розробки гібридних метаевристик, які можуть досягти тонкого балансу між дослідженням та експлуатацією. Гібридна метаевристика є багатообіцяючою спробою вийти за межі метаевристики через об'єднання декількох метаевристик таким чином, щоб подолати недоліки кожної з них, не втрачаючи водночас їхніх переваг. Поєднання машинного навчання з метаевристикою стало останньою тенденцією в цій галузі. Більше того, нещодавня гібридизація машинного навчання та генетичного програмування в рамках повністю автоматизованої композиції сигналізує про рух до парадигми гіперевристики.

References

1. Huhns M. N., Singh M. P., Service-oriented computing: Key concepts and principles, IEEE Internet Comput., vol. 9, no. 1, pp. 75–81, Jan. 2005.
2. Papazoglou M. P., Heuvel W.-J., Service oriented architectures: approaches, technologies and research issues, VLDB J. Vol.16 (3) (2007) 389–415.
3. O'Sullivan J., Edmond D., Hofstede A. T., What's in a service?, Distrib. Parallel Databases, 2002, 12, (2–3), pp. 117–133
4. Bouguettaya A., Sheng Q.Z., Daniel F. (Eds.), Web Services Foundations, Springer, 2013, P.739
5. Gabrel V., Manouvrier M., Murat C., Optimal and automatic transactional web service composition with dependency graph and 0-1 linear programming, in Proc. Int. Conf. Service-Oriented Comput. Berlin: Springer, Nov. 2014, pp. 108–122.
6. Li J., Zhao Y., Liu M., et al.: An adaptive heuristic approach for distributed QoS-based service composition, in Proc. IEEE Symp. Comput. Commun., Jun. 2010, pp. 687–694.
7. Dongre Y., Ingle R.: An investigation of QoS criteria for optimal services selection in composition, in Proc. 2nd Int. Conf. Innov. Mech. Ind. Appl. (ICIMIA), Mar. 2020, pp. 705–710.
8. Mili H., Tremblay G., et al.: Business Process Modeling Languages: Sorting Through the Alphabet Soup. ACM Computing Surveys, 11, 2010, P.57
9. Pop C., Vlad M., Chifu V., et al.: A tabu search optimization approach for semantic web service composition, in Proc. 10th Int. Symp. Parallel Distrib. Comput., Jul. 2011, pp. 274–277.
10. Liu Q., Zhang S.-L., Yang R., et al.: Web services composition with QoS bound based on simulated annealing algorithm, J. Southeast Univ., vol. 24, no. 3, 2008, pp. 308–311.
11. Canfora G., Penta M.D., Esposito R., et al.: A lightweight approach for QoS-aware service composition, Proc. 2nd Int. Conf. on Service Oriented Computing (ICSOC'04), NY, USA, 2004, pp. 36–47
12. Zhang C.W., Su S., Chen J.L.: DiGA: population diversity handling genetic algorithm for QoS-aware web services selection, Comput. Commun., 2007, 30, pp. 1082–1090
13. Zhang C.W., Ma Y.: Dynamic genetic algorithm for search in web service compositions based on global QoS evaluations. IEEE Int. Conf. on Scalable Computing and Communications, 2009, pp. 644–649
14. Vanrompay Y., Rigole P., et al.: Genetic algorithm-based optimization of service composition and deployment. 3rd Int. workshop on Services integration in pervasive environments, 2008, pp. 13–17
15. Gong X.R., Zhu Q.S., Wu C.L., et al.: Web services composition supporting global optimal and dynamic re-planning of QoS, Comput. Integr. Manuf. Syst., 2008, 14, (10), pp. 2068–2075
16. Jiang Z., Han J., Wang Z.: An optimization model for dynamic QoS-aware web services selection and composition, Chin. J. Comput., 2009, 32, (5), pp. 1014–1025
17. Ai L., Tang M., Fidge C.: Partitioning composite web services for decentralized execution using a genetic algorithm, Future Gener. Comput. Syst., 2011, 27, pp. 157–172
18. Cao J., Wang J., Zhao H., et al.: A service process optimization method based on model refinement, J. Supercomput., 2013, 63, pp. 72–88
19. Yu Q., Rege M., Bouguettaya A., et al.: A two-phase framework for quality-aware web service selection, Serv. Oriented Comput. Appl., 2010, 4, pp. 63–79
20. Klein A., Ishikawa F., Honiden S.: Towards network-aware service composition in the cloud. WWW 2012, Lyon, France, 2012, pp. 959–968

21. Ukor R.: Service selection and horizontal multi-sourcing in process-oriented capability outsourcing, *J. Softw., Evol. Process*, 2012, 24, pp. 259–283
22. Fanjiang Y.-Y., Syu Y.: Semantic-based automatic service composition with functional and non-functional requirements in design time: a GA approach, *Inf. Softw. Technol.*, 2014, 56, (3), pp. 352–373
23. Li Y.Z., Hu J., et al.: Research on QoS service composition based on coevolutionary genetic algorithm, *Soft Comput.*, 2018, 22, pp. 7865–7874
24. Claro D.B., Albers P., Hao J.K.: Selecting web services for optimal composition. *Int'l Conf. Web Services (ICWS'05)*, Orlando, FL, USA, 2005
25. Yao Y.J., Chen H.P.: QoS-aware service composition using NSGA-II. *2nd Int. Conf. on Interaction Sciences*, Seoul, Korea, 2009, pp. 358–363
26. Hashmi K., Alhosban A., Najmi E., et al.: Automated web service quality component negotiation using NSGA-2. *ACS Int. Conf. on Computer Systems and Applications*, Ifrane, 2013, pp. 1–6
27. Li J.Z., Luo W.L., Zeng J.T., et al.: Application of SPEA2 algorithm in web services selection. *IEEE Youth Conf. on Information Computing and Telecommunications*, Beijing, 2010, pp. 387–390
28. Liu S., Liu Y.: A dynamic web services selection algorithm with QoS global optimal in web services composition, *J. Softw.*, 2007, 18, (3), pp. 646–656
29. Wang J.L. et al.: Optimal web service selection based on multi-objective GA. *Int. Symp. on Comp. Intellig. and Design*, 2008, pp. 553–556
30. Hu H., Dong W., et al.: Pareto optimality based genetic algorithm in web services composition, *J. Xi'an Jiao Tong Univ.*, 2009, 43, (12), pp. 50–54
31. Wagner F., Kloppe B., et al.: Towards robust service compositions in the context of functionally diverse services. *WWW*, 2012, pp. 969–978
32. Ramírez A., Parejo J., Romero J., et al.: Evolutionary composition of QoS-aware web services: A many-objective perspective, *Expert Syst. Appl.*, 2017, 72, pp. 357–370
33. Ma X.N., Dong B.T.: Linear physical programming based approach for web service selection. *2008 Int. Conf. on Information Management, Innovation Management and Industrial Engineering*, Xian, China, 2008, pp. 398–401
34. Liang W.Y., Huang C.C.: The generic genetic algorithm incorporates with rough set theory – An application of the web services composition, *Expert Syst. Appl.*, 2009, 36, pp. 5549–5556
35. Liu Z., Xue X., Shen J., et al.: Web service dynamic composition based on decomposition of global QoS constraints, *Int. J. Adv. Manuf. Technol.*, 2013, 69, pp. 2247–2260
36. Que Y., Zhong W., Chen H., et al.: Improved adaptive immune genetic algorithm for optimal QoS-aware service composition selection in cloud manufacturing, *Int. J. Adv. Manuf. Technol.*, vol. 96, nos. 9-12, pp. 4455-4465, Jun. 2018.
37. Sadeghiram S., Ma H., Chen G.: Cluster-guided genetic algorithm for distributed data-intensive web service composition, in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1-7.
38. Jatoth C., Gangadharan G.: Optimal fitness aware cloud service composition using an adaptive genotypes evolution based genetic algorithm, *Future Gener. Comput. Syst.*, vol. 94, pp. 185-198, 2019.
39. Rodriguez-Mier P., Mucientes M., Lama M., et al.: Composition of web services through genetic programming, *Evol. Intell.*, 2010, 3, pp. 171–186
40. Yu Y., Ma H., Zhang M.: An adaptive genetic programming approach to QoS-aware web services composition. *IEEE Congress on Evolutionary Computation*, Cancun, 2013, pp. 1740–1747
41. Yu Y., Ma H., Zhang M.: A Genetic Programming approach to distributed QoS-aware web service composition. *IEEE Congress on Evolutionary Computation 2014*: 1840-1846
42. Ma H., Wang A., Zhang M.: A hybrid approach using genetic programming and greedy search for QoS-aware web service composition, in *Transactions on Large-Scale Data and Knowledge-Centered Systems XVIII*. Springer, 2015, pp. 180-205.
43. Yu Y., Ma H., Zhang M.: A hybrid GP-tabu approach to QoS-aware data intensive web service composition, in *Proc. Asia-Pacific Conf. Simulated Evol. Learn.*: Springer, Dec. 2014, pp. 106-118.
44. Wang A., Ma H., Zhang M.: Genetic programming with greedy search for web service composition, in *Database and Expert Systems Applications*, vol. 8056, pp. 9-17.
45. Yang F.C., et al.: An improved particle swarm optimization algorithm for QoS-aware web service selection in service oriented communication, *Int. J. Comput. Intell. Syst.*, 2010, 4, (s), pp. 18–30
46. Wang S.G., Sun Q.B., Zou H., et al.: Particle swarm optimization with skyline operator for fast cloud-based web service composition, *Mobile Netw. Appl.*, 2013, 18, pp. 116–121
47. Fan X.Q., Jiang C.J., Fang X.W., et al.: Dynamic web service selection based on discrete PSO, *J. Comput. Res. Dev.*, 2010, 47, (1), pp. 147–156
48. Wang L., He Y.X.: Web service composition based on QoS with chaos particle swarm optimization. *6th Int. Conf. on Wireless Communications Networking and Mobile Computing*, 2010, pp. 1–4
49. Li W.F., Zhong Y., Wang X., et al.: Resource virtualization and service selection in cloud logistics, *J. Netw. Comput. Appl.*, 2013, 36, pp. 1696–1704
50. Li S.Z., Shen P., Yang S.X.: A grouping particle swarm optimization algorithm for web service selection based on user preference. *2011 IEEE Int. Conf. on Computer Science and Automation Engineering*, Shanghai, China, 2011, v3, pp. 427–431
51. Ludwig S.A.: Applying particle swarm optimization to quality-of-service driven web service composition. *IEEE 26th Int. Conf. on Advanced Information Networking and Applications (AINA)*, Fukuoka, Japan, 2012, pp. 613–620
52. Cao J.X., Sun X.S., Zheng X., et al.: Efficient multi-objective services selection algorithm based on particle swarm optimization. *IEEE Asia-Pacific Services Computing Conf.*, 2010, pp. 603–608
53. Yu W., Li S., Tang X., et al.: An efficient top-k ranking method for service selection based on ϵ -

- ADMOPSO algorithm, *Neural Comput. Appl.*, 2018, 31, pp. 77–92
54. Guha T., Ludwig S.A.: Comparison of service selection algorithms for grid services: multiple objective PSO and constraint satisfaction based service selection. 20th IEEE Int. Conf. on Tools with Artificial Intelligence, 2008, pp. 172–179
 55. Fan X.Q. A decision-making method for personalized composite service, *Expert Syst. Appl.*, 2013, 40, pp. 5804–5810
 56. Tao F., Zhao D.M., Hu Y.F., et al.: Correlation-aware resource service composition and optimal-selection in manufacturing grid, *Eur. J. Oper. Res.*, 2010, v. 201, pp. 129–143
 57. Xu X., Rong H., et al.: Predatory search-based chaos turbo particle swarm optimisation (PS-CTPSO): A new PSO algorithm for web service combination problems, *Future Gener. Comput. Syst.*, v. 89, pp. 375–386, 2018.
 58. Liu Y., Miao H., Li Z., et al.: QoS-aware web services composition based on HQPSO algorithm, in *Proc. 1st ACIS/JNU Int. Conf. Comput., Netw., Syst. Ind. Eng.*, May 2011, pp. 400–405.
 59. Yin H., Zhang C., et al.: A hybrid multiobjective discrete PSO algorithm for a SLA-aware service composition problem, *Math. Problems Eng.*, vol. 4, 2014, pp.1-14
 60. Wang S., Sun Q., Zou H., et al.: Particle swarm optimization with skyline operator for fast cloud-based web service composition, *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 116-121, 2013.
 61. Hossain M. S., Moniruzzaman M., et al.: Big data-driven service composition using parallel clustered PSO in mobile environment, *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 806-817, Sep. 2016.
 62. Chifu V. R., Pop C. B., et al.: Web service composition technique based on a service graph and PSO, in *Proc. IEEE 6th Int. Conf. Intell. Comput. Commun. Process.*, 2010, pp. 265-272.
 63. Gharbi M., Mezni H.: Towards big services composition, *Int. J. Web Grid Services*, vol. 16, no. 4, pp. 393–421, 2020.
 64. Zhao X., Huang P., et al.: A hybrid clonal selection algorithm for quality of service-aware web service selection problem, *Int. J. Innov. Comput. Inf. Control*, vol. 8, no. 12, pp. 8527–8544, 2012.
 65. Fekih H., Mtibaa S., Bouamama S.: An efficient user-centric web service composition based on harmony particle swarm optimization, *Int. J. Web Services Res.*, vol. 16, no. 1, pp. 1–21, Jan. 2019.
 66. da Silva A. S., Mei Y., Ma H., et al.: Particle swarm optimization with sequence-like indirect representation for web service composition, in *Evolutionary Computation in Combinatorial Optimization*, vol. 9595, Chicano F., Hu B., García-Sánchez P., Eds. Cham, Switzerland: Springer, 2016, 8-14.
 67. Zhao X., Song B., et al.: An improved discrete immune optimization algorithm based on PSO for QoS-driven web service composition, *Appl. Soft Comput.*, vol. 12, no. 8, pp. 2208–2216, 2012.
 68. Haytamy S., Omara F.: A deep learning based framework for optimizing cloud consumer QoS-based service composition, *Computing*, vol. 102, no. 5, pp. 1117–1137, May 2020.
 69. Hosseinzadeh M., Tho Q., Ali S., et al.: A hybrid service selection and composition model for cloud-edge computing in the Internet of Things, *IEEE Access*, vol. 8, pp. 85939-85949, 2020.
 70. Sun Q., Wang S., Yang F.: Quick service selection approach based on particle swarm optimization. IEEE Fifth Int. Conf. on Bio-Inspired Computing: Theories and Applications, 2010, pp. 278–284
 71. Zhao X.C., Song B.Q., et al.: An improved discrete immune optimization algorithm based on PSO for QoS-driven web service composition, *Appl. Soft Comput.*, 2012, 12, (8), pp. 2208–2216
 72. Fan X.Q., Fang X.W., Jiang C.J.: Research on web service selection based on cooperative evolution, *Expert Syst. Appl.*, 2011, 38, pp. 9736–9743
 73. Yu C.X., Wang G.: A multi-agent based architecture for web service selection in E-business. Eighth IEEE Int. Conf. on e-Business Engineering, Beijing, China, 2011, pp. 245–250
 74. Fethallah H., Chikh M.A., Mohammed M., et al.: Qos-aware service selection based on swarm particle optimization. Int. Conf. on Information Technology and e-Services, Sousse, 2012, pp. 1–6
 75. Zheng X.: Ant colony intelligence based solution for grid services selection. 7th World Congress on Intelligent Control and Automation, 2008, pp. 2512–2517
 76. Wang L.J., Shen J., et al.: Towards minimizing cost for composite data-intensive services. Proc. of the 2013 IEEE 17th Int. Conf. on Computer Supported Cooperative Work in Design, 2013, pp. 293–298
 77. Hossain M.S., Hossain S.K.A., Alamri A., et al.: Ant-based services election framework for a smart home monitoring environment, *Multimed. Tools. Appl.*, 2013, 67, pp. 433–453
 78. Pop C.B., Chifu V.R., Salomie I., et al.: Ant-inspired technique for automatic web service composition and selection. 12th Int. Symp. on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania, 2010, pp. 449–455
 79. Wang R., Ma L., Chen Y.P.: The application of ant colony algorithm in web service selection. 2010 Int. Conf. on Computational Intelligence and Software Engineering (CiSE), Wuhan, China, 2010, pp. 1–4.
 80. Yu Q., Chen L., Li B. Ant colony optimization applied to web service compositions in cloud computing, *Computers & Electrical Engineering* Vol. 41, 2015, pp. 18–27
 81. Fang Q., Peng X., Liu Q., et al.: A global QoS optimizing web service selection algorithm based on MOACO for dynamic web service composition. Int. Forum on Information Technology and Application, Chengdu, China, 2009, v.1, pp. 37–42
 82. Zhang W., Chang C.K., Feng T.M., et al.: Qos-based dynamic web service composition with ant colony optimization. IEEE 34th Annual Computer Software and Applications Conf., Seoul, Korea, 2010, pp. 493–502
 83. Dahan F., Hindi K., Ghoneim A., et al.: An Enhanced Ant Colony Optimization Based Algorithm

- to Solve QoS-Aware Web Service Composition, *IEEE Access*, Vol.9, pp. 34098 – 4111, 2021
84. Yang Z., Shang C., Liu Q., et al.: A dynamic web services composition algorithm based on the combination of ant colony algorithm and genetic algorithm, *J. Comput. Inf. Syst.*, vol. 6, no. 8, pp. 2617–2622, 2010.
85. Yang Y., Yang B., Wang S., et al.: A dynamic ant-colony genetic algorithm for cloud service composition optimization, *Int. J. Adv. Manuf. Technol.*, vol. 102, (1-4), pp. 355–368, 2019.
86. Liu Z.-Z., Wang Z.-J., Zhou X.-F, et al.: A new algorithm for QoS-aware composite web services selection, in *Proc. 2nd Int. Workshop Intell. Syst. Appl.*, May 2010, pp. 1–4.
87. Alayed H., Dahan F., Alfakih T., et al.: Enhancement of ant colony optimization for QoS-aware web service selection, *IEEE Access*, vol. 7, pp. 97041–97051, 2019.
88. Yang Z.K., Shang C.W., Liu Q.T., et al.: A dynamic web services composition algorithm based on the combination of ant colony algorithm and genetic algorithm, *J. Comput. Inf. Syst.*, 2010, 6, (8), pp. 2617–2622
89. Yang Y., Yang B., Wang S., et al.: A dynamic ant-colony genetic algorithm for cloud service composition optimization, *Int. J. Adv. Manuf. Technol.*, 2019,102, (1–4), pp. 355–368
90. Liu Z., Wang Z., Zhou X., et al.: A new algorithm for QoS-aware composite web services selection. 2nd Int. Workshop on Intelligent Systems and Applications (ISA), Kyiv, 2010, pp. 1–4
91. Bei L., Wenlin L., Xin S., et al.: An improved ACO based service composition algorithm in multi-cloud networks. *Journal of Cloud Computing* (2024) Vol.13, №1, pp.1-12
92. Jiang B., Qin Y., Yang Y., et al.: Web Service Composition Optimization with the Improved Fireworks Algorithm. *Mobile Information Systems*. Volume 2022, pp.1-13
93. Peng S., Guo T.: Multi-Objective Service Composition Using Enhanced Multi-Objective Differential Evolution Algorithm. *Computational Intelligence and Neuroscience*. 2023(1), pp.1-10.
94. Jafarpour N., Khayyambashi M.: A new approach for QoS-aware web service composition based on harmony search algorithm. 11th IEEE Int. Symp. on Web Systems Evolution, 2009, pp. 75–78
95. Ghobaei-Arani M., Rahmanian A., et al.: CSA-WSC: cuckoo search algorithm for web service composition in cloud environments, *Soft Comput.*, 2018, 22, (24), pp. 8353–8378
96. Dahan F. An Improved Whale Optimization Algorithm for Web Service Composition: *Axioms* 2022, 11, 725. pp. 1-14
97. Gavvala S., Jatoth C., et al.: Qos-aware cloud service composition using eagle strategy, *Future Gener. Comput. Syst.*, 2019, 90, pp. 273–290
98. Dahan F. Neighborhood Search Based Improved Bat Algorithm for Web Service Composition. *Computer Systems Science & Engineering CSSE*, 2023, vol.45, no.2 pp.1343-1356
99. Dahan F., Hindi K.E.: Enhanced artificial bee colony algorithm for QoS-aware web service selection problem, *Computing*, 2017, 99, (5), pp. 507–517
100. Pop C. B., Chifu V. R., Salomie I., et al.: Cuckoo-inspired hybrid algorithm for selecting the optimal web service composition, in *Proc. IEEE 7th Int. Conf. Intell. Comput. Commun. Process.*, Aug. 2011, pp. 33-40.
101. Chifu V. R, Pop C. B., Salomie I., et al.: Hybrid honey bees mating optimization algorithm for identifying the near-optimal solution in web service composition, *Comput. Informat.*, vol. 36, no. 5, pp. 1143-1172, 2017.
102. Yang H., Xue F., Zhu H., et al.: Web service composition optimization based on adaptive mutant beetle swarm, *J. Phys., Conf. Ser.*, vol. 1651, no. 1, Nov. 2020, pp.347-501
103. Dahan F., Alwabel A.: Artificial Bee Colony with Cuckoo Search for Solving Service Composition. *Intelligent Automation & Soft Computing IASC*, 2023, vol.35, no.3 pp.3385-3402
104. Ahanger T. A., Dahan F.: Hybridizing Artificial Bee Colony with Bat Algorithm for Web Service Composition. *Computer Systems Science and Engineering CSSE*, 2023, vol.46, no.2, pp.2429-2445
105. Peng S., Wang H., Yu Q.: Multi-clusters adaptive brain storm optimization algorithm for QoS-aware service composition, *IEEE Access*, vol. 8, pp. 48822-48835, 2020.

Одержано: 14.07.2025

Внутрішня рецензія отримана: 07.08.2025

Зовнішня рецензія отримана: 15.08.2025

Про автора:

Мороз Григорій Борисович,

кандидат технічних наук,

с.н.с., провідний науковий співробітник.

<https://orcid.org/0000-0001-8666-9503>

Місце роботи автора:

Інститут програмних систем

НАН України