

## KLASYFIKACJA ALGORYTMÓW WYBORU USŁUG W ARCHITEKTURZE SOA

Paweł KACZMAREK

Politechnika Gdańska, Wydział Elektroniki, Telekomunikacji i Informatyki  
tel: 58 347 24 89 fax: 58 348 61 25 e-mail: pkacz@eti.pg.gda.pl

**Streszczenie:** Wytwarzanie aplikacji w architekturze zorientowanej na usługi wymaga zastosowania odpowiednich algorytmów wyboru usług, które będą zapewniały wytworzenie aplikacji o wymaganych parametrach jakościowych. Biorąc pod uwagę, że problem wyboru usług ma wykładniczą złożoność obliczeniową, stosowane są zarówno algorytmy dokładne jak i przybliżone. W pracy przedstawiono porównanie i analizę istniejących algorytmów wyboru usług w kompozycji usług złożonych. Algorytmy porównano w zakresie złożoności obliczeniowej i dokładności uzyskiwanych wyników. Ponadto przeanalizowano algorytmy stosowane w technologiach Internetu przedmiotów (Internet of things) oraz algorytmy wykorzystujące metody sztucznej inteligencji. W niektórych przypadkach algorytmy zostały opracowane dla różnych poziomów złożoności modelu aplikacji, co również wpływa na ich charakterystyki.

**Słowa kluczowe:** architektura zorientowana na usługi, dobór usług, optymalizacja

### 1. WSTĘP

Wytwarzania aplikacji w Architekturze Zorientowanej na Usługi (Service Oriented Architecture - SOA) zakłada, że złożone aplikacje są wytwarzane przez integrację prostych usług dostarczanych przez niezależnych dostawców. Umożliwia to redukcję kosztów i czasu wytwarzania, jednak wymaga odpowiednich algorytmów, które będą zapewniały wybór modułów oraz wykonanie aplikacji o wymaganych parametrach jakościowych [1]. Parametry jakościowe aplikacji zależą zarówno od struktury aplikacji (pętle, warunki) jak i od parametrów modułów wybranych do jej konstrukcji. Wymagania mogą być określone zarówno w formie optymalizacji (np. jak najwyższa wydajność) jak też w formie ograniczeń, które mogą obejmować dostępne zasoby (np. koszt) lub wymagane progi jakościowe (np. bezpieczeństwo) [2].

W ogólnym przypadku problem optymalizacji ma wykładniczą złożoność obliczeniową. Dotychczas zaproponowano szereg algorytmów, których celem jest możliwie optymalny wybór usług przy zachowaniu wymaganych ograniczeń i akceptowalnej złożoności obliczeniowej [2] [3] [4] [5].

Biorąc pod uwagę, że większość opracowanych algorytmów daje przybliżone wyniki, w pracy porównano algorytmy w zakresie złożoności obliczeniowej oraz

dokładności uzyskiwanych wyników. Uwzględniono różne modele aplikacji złożonej przyjmowane przez algorytmy: model grafowy, gdzie algorytmy bazują na poszukiwaniu optymalnej ścieżki przy wielokryterialnych parametrach wyboru, i model kombinatoryczny, gdzie algorytmy bazują na problemie plecakowym w wersji wielokryterialnych ograniczeń i korzyści.

Ponadto przeanalizowano algorytmy stosowane w dziedzinie Internetu przedmiotów (Internet of things) jako nowym kierunku zastosowania SOA. Wśród analizowanych algorytmów uwzględniono również algorytmy bazujące na technikach sztucznej inteligencji. Algorytmy takie wykorzystują metody symulowanego wyżarzania, mrówkowe i inne dając wyniki przybliżone jednak zapewniając akceptowalny czas wykonywania obliczeń.

### 2. MODEL APLIKACJI W ARCHITEKTURZE ZORIENTOWANEJ NA USŁUGI

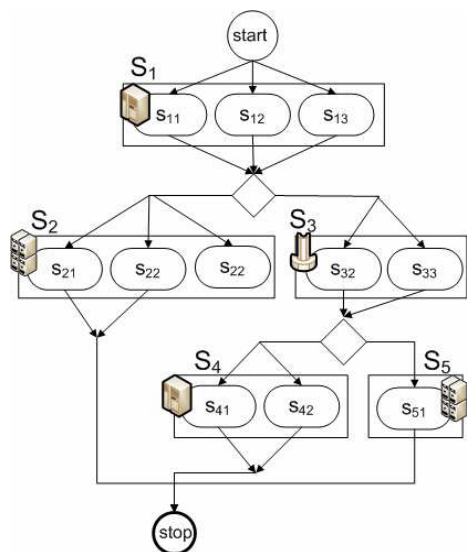
Istniejące algorytmy przyjmują następujące założenia dotyczące modelowania aplikacji w architekturze SOA [2] [3] [6] :

- aplikacja jest usługą złożoną (Complex Service - CS) składająca się z operacji prostych realizowanych przez usługi z klasy usług  $S_1, S_2, \dots, S_N$  jak pokazano na rysunku 1.
- dla każdej klasy usług istnieje jedna lub więcej alternatywnych usług prostych, różniących się parametrami jakościowymi QoS ( $q_{ij} = [q_{ij1}, \dots, q_{ijN}]$ )
- zostaje zdefiniowana funkcja celu ( $F$ ) jako wartość QoS końcowej aplikacji zależna od wartości QoS usług i struktury CS
- zostają zdefiniowane ograniczenia na wartości niektórych parametrów usługi złożonej  $Q_c = [Q_{c1}, \dots, Q_{cm}]$

Algorytm wyboru usług ma na celu optymalizację (osiągnięcie maksymalnej wartości) funkcji celu  $F$  oraz jednocześnie zachowanie ograniczeń dla złożonej usługi.

Wartość funkcji celu zależy od parametrów wybranych usług oraz ich kompozycji w aplikacji z wykorzystaniem konstrukcji programistycznych takich jak: warunki, pętle, wykonania równoległe [2]. Zależnie od typu atrybutu i struktury aplikacji końcowe parametry mogą być obliczane

jako suma (np. koszt), maksimum (np. czas wykonania dla wykonań równoległych) lub minimum (np. bezpieczeństwo).



Rys. 1. Przykładowa usługa złożona z zaznaczonymi klasami usług prostych  $S_i$  i usługami alternatywnymi  $s_{ij}$

Biorąc pod uwagę różnorodność algorytmów i podejść do wyboru usług, zidentyfikowano następujące atrybuty algorytmów:

- Złożoność obliczeniowa
  - i. wykładnicza dla dokładnych obliczeń
  - ii. wielomianowa dla obliczeń przybliżonych
  - iii. określona liczba kroków (np. dla sztucznej inteligencji)
- Model obliczeń
  - i. kombinatoryczny - wybór usług jest modelowany jako 0-1 wielokryterialny problem plecakowy (0-1 multidimensional multichoice knapsack problem - MMKP)
  - ii. grafowy - wybór jest modelowany jako wielokryterialny problem odnalezienia optymalnej ścieżki w grafie (multiconstraint optimal path selection - MCOP)
- Dokładność wyniku
  - i. przybliżone
  - ii. dokładne lokalnie (podaje optymalne rozwiązanie dla podproblemu)
  - iii. dokładne globalnie (o wykładniczej złożoności w ogólnym przypadku)
- Złożoność struktur aplikacji (warunki, pętle, zrównoleglenie)

### 3. ALGORYTMY OPTIMALIZACJI KOMBINATORYCZNEJ

Algorytmy optymalizacji kombinatorycznej stanowią dużą część algorytmów wyboru usług. Wybór jest modelowany jako problem plecakowy, gdzie usługi to elementy w plecaku zaś ograniczenia to dostępne zasoby. W ogólnym przypadku problem jest rozwiązywany przy zastosowaniu całkowitoliczbowego programowania liniowego (Integer Linear Programming - ILP).

Stosowane są zarówno rozwiązania dokładne jak i przybliżone, które rozwiązują problem ILP podany wzorem:

$$\begin{aligned} & \text{Max} \sum_{i=1}^N \sum_{j \in S_i} F_{ij} x_{ij} \\ & \text{subject to} \sum_{i=1}^N \sum_{j \in S_i} q_{ij}^a * x_{ij} \leq Q_c^a \quad (a = 1, \dots, m) \quad (1) \\ & \sum_{j \in S_i} x_{ij} = 1, \quad x_{ij} \in \{0,1\}, \quad i = 1, \dots, N \end{aligned}$$

gdzie:  $N$  – liczba klas usług,  $x_{ij}$  – przyjmuje wartość 1 jeżeli usługa  $s_{ij}$  została wybrana, wartość 0 w przeciwnym przypadku,  $q_{ij}^a$  – parametry jakościowe usługi,  $Q_c$  – ograniczenia zasobów

Algorytmy dokładne obejmują między innymi: exhaustive search, branch-and-bound LP oraz algorytm WS\_IP [2]. Algorytmy dokładne mają wykładniczą złożoność obliczeniową.

Dużą część algorytmów przybliżonych ILP bazuje na algorytmie HEU [7] zaprojektowanym oryginalnie dla strumieni multimedialnych, ale stosowanym też w innych obszarach. Algorytm w pierwszym kroku odnajduje rozwiązanie spełniające ograniczenia na zasoby (ang. feasible solution) wybierając usługi o najmniejszym zużyciu ograniczonych zasobów. Następnie rozwiązanie jest poprawiane przez wymianę wybranych usług na usługi o wyższych parametrach QoS przy zachowaniu ograniczeń na zasoby. Przewidziany jest też krok pogarszania rozwiązania w celu uniknięcia lokalnego minimum. W tabeli 1. przedstawiono porównanie wybranych algorytmów.

Tabela 1. Algorytmy bazujące na programowaniu liniowym

Algorytm	Dokładność	Model aplikacji	Złożoność obliczeniowa
BBLP [2]	dokładny	sekwencyjny	wykładnicza
WS_IP [2]	dokładny	równoległy	wykładnicza
HEU [7]	94 %	SLA	wielomianowa
WS_HEU [2]	98,5 %	sekwencyjny	$O(N^2(l-1)2m)$
WFlow [2]	90 %	równoległy	$O(N^2(l-1)2m)$
RWSCS KP [6]	98 %	sekwencyjny	$O(N^2(l-1)2m)$
DIST HEU [3]	98 %	sekwencyjny	wielomianowa

Algorytm WS\_HEU [2] został opracowany do wyboru usług w kompozycji aplikacji sekwencyjnych. Algorytm bazuje na HEU wprowadzając dodatkowe usprawnienie w postaci kroku akceptującego tymczasowe rozwiązanie, które nie spełnia ograniczeń na zasoby (ang. infeasible solution). W ten sposób lepiej radzi sobie z problemem lokalnych optimum. Została również opracowana wersja algorytmu dla aplikacji rozproszonych WFlow. Algorytmy obliczają funkcję celu i warunki ograniczeń na zasoby zależnie od alternatywnych ścieżek wykonania zarówno dla aplikacji rozproszonych jak i sekwencyjnych.

Algorytm RWSCS KP [6] również bazuje na algorytmie HEU. Autorzy proponują bibliotekę reguł, które są wykorzystywane do wykluczenia usług, których parametry są zbyt niekorzystne, aby zostały włączone do końcowego rozwiązania.

Część algorytmów (np. DIST\_HEU [3]) bazuje swoje rozwiązanie na podziale usługi złożonej na mniejsze fragmenty i optymalizację tych fragmentów z zachowaniem odpowiednio podzielonych ograniczeń na zasoby. Algorytm DIST\_HEU

uwzględnia konstrukcje rozproszone i równoległe. Podobnie jak HEU, algorytm poszukuje inicjalnie rozwiązania spełniającego warunki ograniczeń na zasoby a następnie poszukuje usług, które mogą poprawić wartość funkcji celu.

#### 4. ALGORYTMY OPTIMALIZACJI NA GRAFACH

W algorytmach optymalizacji na grafach kolejne usługi są przetwarzane zgodnie ze strukturą grafu wykonania aplikacji. Problem wyboru usług w notacji grafowej opisany jest w następujący sposób:

- każda alternatywna usługa prosta jest węzłem w grafie
- usługi proste są ze sobą połączone, jeżeli odpowiadające im usługi abstrakcyjne są połączone
- atrybuty QoS usług zostają przypisane do krawędzi i węzłów odpowiednio

Algorytm MCSP [2] jest jednym z algorytmów optymalizacji na grafach (tab. 2). Algorytm w pierwszym kroku dokonuje uporządkowania topologicznego wierzchołków zaś następnie przegląda je według uporządkowania topologicznego. W czasie przeglądania, algorytm utrzymuje listę podścieżek, które są kandydatami do optymalnej ścieżki. Jeżeli dodanie jakiegoś węzła może poprawić parametry QoS dotychczasowego rozwiązania, to jest dodawany do podścieżki. Algorytm MCSP jest dokładny, co powoduje, że zbiór ścieżek kandydatów może rosnąć wykładniczo podobnie jak złożoność czasowa.

Autorzy [2] zaproponowali modyfikację algorytmu (MCSP-K), w której ograniczyli długość listy ścieżek kandydatów do K. Powoduje to, że algorytm może zwracać wyniki nieoptymalne, ale czas przetwarzania zmniejsza się do wielomianowego. Algorytm MSCP-K osiąga skuteczność na poziomie 95% dla K równego 10 i 50 klas usług prostych. Ponadto zaprezentowano modyfikację algorytmu, w której uwzględniono konstrukcje równoległe fork i join (MCSP-K general). Zasady przetwarzania podczas łączenia ścieżek zakładają, że wszystkie parametry są agregowane gdyż wszystkie potencjalne ścieżki mogą zostać wywołane podczas przetwarzania.

Tablica 2. Algorytmy bazujące na modelu grafowym

Algorytm	Dokładność	Model aplikacji	Złożoność obliczeniowa
MCSP [2]	dokładny	sekwencyjny	wykładnicza
MCSP-K[2]	95 %	sekwencyjny	wielomianowa
MCSP-K general[2]	95 %	równoległy	wielomianowa

#### 5. ALGORYTMY SZTUCZNEJ INTELIGENCJI

Algorytmy sztucznej inteligencji umożliwiają rozwiązanie złożonego obliczeniowo problemu wyboru usług w określonych ramach czasowych, zazwyczaj przez zadanie wymaganej liczby kroków. W tabeli 3 przedstawiono porównanie wybranych algorytmów wyboru usług bazujących na sztucznej inteligencji.

[8] proponuje zastosowanie algorytmu genetycznego (GA) do wyboru usług wraz z odpowiednią reprezentacją koncepcji związanych z wyborem usług w nomenklaturze

GA. Chromosom odpowiada rozwiązaniu wyboru usług zaś populacja odpowiada alternatywnym rozwiązaniom. Zastosowano typowe kroki GA takie jak: ocena chromosomów na podstawie funkcji celu i tworzenie kolejnych generacji przez krzyżowanie.

Stosowane są również algorytmy mrówkowe, w [4] opisano algorytm ACO4WS, gdzie graf aplikacji jest wykorzystywany jako podstawa do odwiedzania kolejnych wierzchołków. Bazując na oznaczaniu krawędzi w stylu feromonów przeprowadzana jest aktualizacja ocen ścieżek, wietrzenie feromonów (ang. pheromone evaporation) oraz obliczanie ważonej funkcji celu. Metoda umożliwia uzyskanie rozwiązania zbliżonego do optymalnego w zadanej liczbie kroków.

[9] proponuje zastosowanie metod optymalizacji stada (Particle swarm optimization - PSO) do wyboru usług. Algorytm wybiera inicjalne rozwiązanie przypisując jednostkom kierunek i prędkość. W kolejnych krokach postępuje w pętli obliczając funkcję celu i wybierając lepszy kierunek. Algorytm uzyskuje różną dokładność od 30% do 80% zależnie od zadanej liczby kroków i liczby jednostek.

Tablica 3. Algorytmy wyboru usług bazujące na metodach sztucznej inteligencji

Algorytm	Model obliczeń	Dokładność	Metoda AI
GA [8]	kombinatoryczny	przybliżony	genetyczny
ACO4WS [4]	grafowy	przybliżony	mrówkowy
PSO [9]	kombinatoryczny	przybliżony	optymalizacja stada

#### 6. ALGORYTMY STOSOWANE W INTERNECIE PRZEDMIOTÓW

Internet przedmiotów (Internet of things) jest nowym kierunkiem rozwoju, w którym urządzenia łączą się ze sobą i wymieniają dane przy użyciu istniejącej infrastruktury, zazwyczaj komunikacji bezprzewodowej. Dzięki takiemu rozwiązaniu możliwe jest tworzenie inteligentnych przestrzeni urządzeń wbudowanych takich jak czujniki, czytniki RFID, sprzęty codziennego użytku. Rozwój IoT wiąże się z koniecznością wyboru odpowiednich urządzeń i usług analogicznie do rozwiązań SOA. Urządzenia wbudowane posiadają często ograniczoną wielkość i możliwości zasilania, co wiąże się z koniecznością doboru odpowiednich usług pod kątem optymalizacji zużycia tych zasobów [10].

W pracy [5] przedstawiono algorytm wyboru usług optymalizacji na grafach. W rozwiązaniu zastosowano podejście jedno-źródło wiele-celów dla k najkrótszych ścieżek (ang. single-source many-target k shortest paths). Zaproponowany algorytm wyboru bazuje na algorytmie Dijkstry wyboru optymalnej ścieżki w grafie i przeszukiwaniu w szerz. W algorytmie autorzy zastosowali rozwiązanie MapReduce dla wielkich zbiorów danych i jego implementację Hadoop. Wykorzystanie zrównoleglenia obliczeń pozwala w znacznym stopniu usprawnić proces wyboru odpowiednich usług. Autorzy zaproponowali algorytmy BFSKNN oraz PruningBFSKNN jako rozszerzenie algorytmu Dijkstry, w którym więcej niż jeden węzeł jest rozwijany. Autorzy zwracają uwagę na trudności w szczegółowym określeniu wydajności systemu w środowisku rozproszonym.

Heurystyczne podejście do wyboru usług w IoT zostało zaprezentowane w [11]. Autorzy wykorzystali architekturę agentową do autonomicznego przetwarzania usługi złożonej, co obejmuje planowanie, kompozycję, wybór i szeregowanie

usług. Każdy z kroków wykonania jest realizowany przez dedykowane agenty, które zajmują się odpowiednim przetwarzaniem i wykonaniem usług prostych. Autorzy uwzględnili parametry QoS dla IoT: czas działania, niezawodność (ang. reliability) i dostępność (ang. availability).

Analogicznie jak w [2], pierwszym krokiem algorytmu poszukiwania optymalnego rozwiązania jest odnalezienie rozwiązania spełniającego ograniczenia na zasoby, zaś w kolejnych rozwiązaniu jest poprawiana. Rozwiązanie stosuje uczenie ze wzmocnieniem (ang. reinforcement learning). Fazy planowania wykonania i uczenia wykonywane są naprzemiennie w celu poprawy wyniku w dłuższym okresie.

W pracy [10] opisano kompleksowe rozwiązanie obejmujące poszukiwanie, odpytywanie, wybór i dostarczanie usług w IoT, co odróżnia rozwiązanie od wcześniej opisywanych. Autorzy zaproponowali cztery główne usprawnienia procesu kompozycji: ograniczenie narzutu na wykonanie usług, minimalizacja kosztów rejestracji i integracji, wsparcie dla wyszukiwania w czasie działania, wsparcie dla dostarczania usług na żądanie.

W ramach prac została zaimplementowana platforma integracji SOCRADES, która umożliwia wyszukiwanie, kompozycję i monitorowanie usług. Integracja usług odbywa się z wykorzystaniem standardów komunikacji SOA: Web services oraz RESTful services. Wykorzystany został również standard komunikacji Device Profiles for Web Services, który rozszerza typową funkcjonalność Web services.

## 7. PODSUMOWANIE I WNIOSKI KOŃCOWE

Wybór usług optymalizujący parametry aplikacji i zachowujący ograniczenia pozostaje trudnym zadaniem biorąc pod uwagę jego wykładniczą złożoność obliczeniową. Z przedstawionych danych wynika, że algorytmy przybliżone pozwalają uzyskać dokładność powyżej 90%. Niestety wiele algorytmów pozwala na analizę jedynie niektórych konstrukcji aplikacji pomijając często konstrukcje równoległe i rozproszone, co stanowi ciekawy temat dalszych prac.

Rozwiązania Internetu przedmiotów mogą stać się wiodącym kierunkiem rozwoju SOA. Specyfika takich rozwiązań wiąże się z ograniczonymi zasobami i koniecznością zmian usług w czasie działania systemów. Zmiana technologii wiąże się z koniecznością dostosowania istniejących algorytmów i rozwiązań. W

rozproszonych systemach Internetu przedmiotów dokładne obliczenie wydajności stanowi poważny problem ze względu na wiele czynników wpływających na działanie systemu [5]. Mierzenie i obliczanie wydajności w tego typu systemach jest ciekawym tematem dalszych prac badawczych.

## 8. BIBLIOGRAFIA

1. Singh M. P., Huhns M. N.: *Service-Oriented Computing*, John Wiley & Sons, 2005, ISBN 978-0-470-09148-7
2. Yu T., Zhang Y., Lin K.-J.: *Efficient Algorithms for Web Services Selection with End-to-End QoS Constraints*, ACM Transactions on the Web, 2007, ISSN 1559-1131
3. Alrifai M., Risse T., Dolog P., Nejdl W.: *A Scalable Approach for QoS-Based Web Service Selection*, Service-Oriented Computing, ICSOC 2008 Workshops, ISBN 978-3-642-01246-4
4. Wang X. L., Jing Z., Yang H. Z.: *Service Selection Constraint Model and Optimization Algorithm for Web Service Composition*, Information Technology Journal, 2001
5. Yin X., Yang J.: *Shortest Paths Based Web Service Selection in Internet of Things*, Journal of Sensors, Hindawi Publishing Corporation, 2014
6. Cao H., Feng X., Sun Y., Zhang Z., Wu Q.: *A Service Selection Model with Multiple QoS Constraints on the MMKP*, IFIP International Conference on Network and Parallel Computing, 2007, DOI 10.1109/NPC.2007.35
7. Khan S., Li K., F., Manning E. G., Akbar M.: *Solving the Knapsack Problem for Adaptive Multimedia Systems*, Studia Informatica Universalis, 2001
8. Cao L., Li M., Cao J.: *Using genetic algorithm to implement cost-driven web service selection*, Multiagent and Grid Systems - An International Journal, 2007, ISSN 1574-1702
9. Xia H., Chen Y., Li Z., Gao H., Chen Y.: *Web Service Selection Algorithm Based on Particle Swarm Optimization*, Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009, DOI 10.1109/DASC.2009.99
10. Guinard D., Trifa V., Karnouskos S., Spiess P., Savio D.: *Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services*, IEEE Transactions on Services Computing, 2010
11. Kouicem A., Chibani A., Tari A., Amirat Y., Tari Z.: *Dynamic services selection approach for the composition of complex services in the web of objects*, IEEE World Forum on Internet of Things (WF-IoT), 2014

## CLASSIFICATION OF SERVICE SELECTION ALGORITHMS IN SOA ARCHITECTURE

**Key-words:** Service Oriented Architecture, service selection, optimization

Application development in Service Oriented Architecture requires application of appropriate service selection algorithms that support achievement of required application quality. Both accurate and approximate algorithms are applied considering that the service selection problem has non-polynomial computational complexity. In this paper, existing service selection algorithms were analyzed and compared considering computational complexity and result accuracy. Additionally, algorithms that use artificial intelligence methods were described. Solutions dedicated to Internet of things were analyzed with respect to specific requirements and working conditions in this application area. In some cases, algorithms were designed for different complexity of the application model, which also influences their characteristics.