

Bożena MAŁYSIAK-MROZEK, Alina MOMOT, Dariusz MROZEK  
Politechnika Śląska, Instytut Informatyki

Michał MOMOT  
Instytut Techniki i Aparatury Medycznej, ITAM, Zabrze

## **ARCHITEKTURA HIERARCHICZNEGO SYSTEMU WIELOAGENTOWEGO DLA PROCESU POSZUKIWANIA PODOBIENSTWA BIAŁEK**

**Streszczenie.** Poszukiwanie podobieństwa strukturalnego białek jest procesem niezwykle złożonym obliczeniowo i czasochłonnym, a jednocześnie istotnym z punktu widzenia zrozumienia działania i funkcji cząstek białkowych. Jednym ze sposobów przyspieszenia tego procesu jest jego zrównoleglenie przez rozproszenie obliczeń na wielu komputerach. W niniejszym artykule przedstawiono zaprojektowaną przez autorów architekturę hierarchicznego systemu wieloagentowego wykorzystywanego w poszukiwaniu podobieństwa białek. Zaprezentowano również różne scenariusze użycia i implementacji tej architektury w poszukiwaniu podobieństwa molekuł biologicznych. Dodatkowo przedstawiono wyniki eksperymentów numerycznych potwierdzających przydatność zaprojektowanej architektury w procesie eksploracji struktur białkowych.

**Słowa kluczowe:** bioinformatyka, dopasowanie, białka, struktura, podobieństwo

## **ARCHITECTURE OF THE HIERARCHICAL MULTI-AGENT FOR PROTEIN SIMILARITY SEARCHING**

**Summary.** Protein structure similarity searching is a very complex and time-consuming process. One of the possibilities how we can accelerate the process is parallelization by distributing the computational procedure on multiple computers. In the paper, we present the architecture of the hierarchical multi-agent system dedicated to protein structure similarity searching. We also demonstrate different scenarios how this architecture can be used and modified in similarity searching of biological molecules. Additionally, we show results of several numerical experiments confirming a suitability of the presented architecture in the exploration of protein structures.

**Keywords:** bioinformatics, alignment, protein structure, similarity searching

## 1. Wprowadzenie

Białka to podstawowe molekuly życia we wszystkich organizmach żywych [1]. Ponieważ są one substratami wszystkich reakcji biochemicznych, stanowią ważne cząstki z punktu widzenia procesów zachodzących w żywych komórkach. Funkcje białek w dużym stopniu zależą od konstrukcji tych cząstek, zbudowanych z setek aminokwasów, a co za tym idzie tysięcy atomów różnych pierwiastków chemicznych [2, 3]. Nie dziwi zatem fakt, że identyfikacja białek zwykle opiera się na analizie ich budowy.

### 1.1. Identyfikacja białek na podstawie ich konstrukcji

Identyfikacja białek i ich funkcji komórkowych może być prowadzona na różnych poziomach ich organizacji konstrukcyjnej – na poziomie sekwencji aminokwasów (struktury pierwszorzędowej) i struktury przestrzennej (włączając strukturę drugo-, trzecio- i czwartorzędową) [4, 5]. Proces identyfikacji zwykle przebiega przez porównanie zadanej cząstki białkowej z cząstkami znajdującymi się w dedykowanej bazie danych przy jednoczesnym poszukiwaniu podobieństw strukturalnych. Ponieważ w odróżnieniu od sekwencji aminokwasów struktury przestrzenne białek ujawniają więcej szczegółów dotyczących budowy białka, są one chętnie i szeroko analizowane podczas badania aktywności białek w reakcjach komórkowych, a także prowadzenia różnorodnych dociekań nad możliwością wejścia przez nie w interakcje z innymi molekułami (np. białko-lek).

### 1.2. Poszukiwanie podobieństwa strukturalnego białek

Poszukiwanie podobieństwa strukturalnego białek jest procesem eksploracji struktur białkowych (zwykle znajdujących się w bazie danych) w celu znalezienia cząstek mających identyczną lub zbliżoną budowę. W procesie tym poszukiwane są konkretne relacje pomiędzy poszczególnymi regionami porównywanych struktur białkowych, niezależnie od zmian konformacyjnych, które mogły się pojawić w wyniku zmian środowiskowych lub przyłączenia do badanej cząstki białkowej cząsteczki ligandu lub innej cząsteczki białkowej. Jednocześnie proces poszukiwania podobieństwa strukturalnego białek jest procesem złożonym obliczeniowo i czasochłonnym. Głównym tego powodem jest złożoność konstrukcyjna samych cząsteczek białkowych. Istniejące metody w tym zakresie, takie jak: VAST [6], DALI [7], CE [8], LOCK2 [9], PFSC [10], FATCAT [11], FAST [12] i inne, zazwyczaj redukują złożoną strukturę białkową i w procesie porównania reprezentują ją w znacznie prostszej formie. Następnie, po dokonaniu redukcji, metody te poszukują podobieństw tak reprezentowanych struktur przez porównanie parami zadanej struktury białkowej z poszczególnymi strukturami z bazy danych.

Kolejnym istotnym problemem, negatywnie wpływającym na efektywność procesu poszukiwania podobieństwa strukturalnego białek, jest rosnąca liczba struktur białkowych w światowych repozytoriach danych strukturalnych, takich jak Protein Data Bank [13] w Stanach Zjednoczonych. Baza ta obecnie (26 stycznia 2012 roku) przechowuje 78 867 struktur białkowych. Rosnąca liczba struktur przekłada się na spowolnienie procesu skanowania całych baz danych strukturalnych i w konsekwencji wydłuża czas identyfikacji funkcji białkowych na podstawie ich struktury, ponieważ za każdym razem należy porównać coraz więcej struktur.

### 1.3. Istniejące metody w zakresie zrównoleglenia poszukiwania podobieństwa strukturalnego białek

Z powodów wymienionych w poprzednim rozdziale społeczność naukowa, prowadząca badania w obszarze nauk o życiu (ang. *life sciences*), wciąż poszukuje odpowiednio wydajnych metod analizowania podobieństwa strukturalnego białek. Zrównoleglenie procesu poszukiwania podobieństwa białek jest jednym z podejść, które może przyspieszyć ten proces. Systemy, takie jak ProteinDBS [14], ProCKSI [15] oraz MAS-EAST [26], wykorzystują równoległość obliczeń w celu przyspieszenia procesu poszukiwania. ProteinDBS jest systemem porównania struktur białkowych w czasie rzeczywistym. W swoim działaniu używa on specjalnych rodzajów indeksów (*Entropy Balanced Statistical k-d tree index*), a także zbioru rozproszonych agentów, których zadaniem jest m.in.: zarządzanie indeksami, równoważenie obliczeń, efektywne pobieranie danych z bazy danych. ProCKSI jest system porównywania struktur białkowych, opartym na rozpraszaniu obliczeń w środowisku wieloprocessorowym. System może pracować w klasycznym środowisku równoległym, używając protokołu MPI (*Message Passing Interface*), lub w architekturze *grid*, korzystając z bibliotek MPICH-G2 [16]. MAS-EAST jest z kolei opartym na równolegle pracujących agentach programowych systemem poszukiwania podobieństwa energetycznego białek. Agenci obliczeniowi realizują w nim zadanie poszukiwania podobieństwa energetycznego białek zgodnie z założeniami algorytmu EAST [27].

## 2. Model teoretyczny

W niniejszym artykule zaprezentowano opracowaną i zaimplementowaną przez autorów architekturę rozproszonego, hierarchicznego systemu wieloagentowego, którego zadaniem jest równoległe, efektywne poszukiwanie molekuł białkowych o strukturze identycznej lub zbliżonej do zadanej przez użytkownika systemu. W prezentowanym systemie procedura obliczeniowa, a także zbiory danych, na których prowadzone są obliczenia, zostają rozproszone pomiędzy

wielu agentów obliczeniowych. Tego typu implementacja nie wymaga posiadania kosztownych rozwiązań sprzętowych poza zwykłymi komputerami klasy PC. W zamian powstaje system wielu komputerów, udostępniający kolektywnie zwiększoną moc obliczeniową. Moc obliczeniowa systemu może zostać dodatkowo powiększona w dowolnym momencie przez dołączenie innych komputerów lub całych farm stacji roboczych, na których pracują agenty obliczeniowe.

### 2.1. Założenia ogólne

W 1996 r. Franklin i Graesser zaproponowali definicję agenta programowego, w której podkreślali związek autonomicznego systemu agentowego ze środowiskiem, którego jest on częścią, na które oddziałuje i w którym działa zgodnie ze swoim własnym algorytmem [17]. Zdefiniowanie autonomicznego agenta wymaga zatem określenia środowiska jego działania, a także możliwości jego interakcji z tym środowiskiem. Ponadto należy zdefiniować właściwości agenta oraz opisać architekturę systemu wieloagentowego (ang. *multi-agent system*, *MAS*), który składa się z wielu agentów, współdziałających ze sobą przy realizacji określonego zadania. Agenci, będąc rozproszonymi w pewnym środowisku, działają na różnych komputerach.

Podstawowe własności systemów wieloagentowych obejmują [19]:

- każdy agent dysponuje niekompletną informacją i ma ograniczone możliwości,
- system kontrolujący jest rozproszony,
- dane są zdecentralizowane,
- obliczenia są asynchroniczne.

Systemy wieloagentowe, będąc systemami rozproszonymi, często powodują przyspieszenie realizacji danego zadania. Poprawiają również niezawodność systemu, gdyż awaria nawet pojedynczego agenta nie musi powodować zawieszenia realizacji procesu obliczeniowego. Systemy wieloagentowe można różnicować ze względu na sposób wykorzystania różnych rodzajów agentów lub ze względu na interakcje pomiędzy nimi, a także ze względu na środowisko, w którym działają. Jako przykłady zastosowania systemów wieloagentowych można wskazać: rynki elektroniczne, systemy transportowe, monitorowanie ruchu w czasie rzeczywistym i inne [18, 24]. Typowe własności agentów, metody ich działania oraz architektury definiujące systemy, w których funkcjonują, opisuje praca [21]. Można w niej znaleźć opis historii, a także bieżący stan wiedzy w zakresie różnych systemów wieloagentowych.

Interesujące wydają się przykłady zastosowań różnego typu hierarchicznych systemów wieloagentowych. Jednym z nich jest system opisany w pracy [22], w którym globalne decyzje podejmowane są na podstawie reguł decyzyjnych lokalnych ekspertów. Innym ciekawym przykładem jest system przedstawiony w pracy [23], związany z samoorganizującą się bazą

danych modyfikowaną w czasie rzeczywistym lub też hierarchiczny system wieloagentowy sterujący i monitorujący pracę złożonych układów, zaprezentowany w pracy [20].

## 2.2. Architektura hierarchicznego systemu wieloagentowego

W niniejszym rozdziale opisano architekturę dedykowanego hierarchicznego systemu wieloagentowego, którego zadaniem jest wyszukiwanie podobieństw struktur białkowych. W prezentowanym systemie procedura wyszukiwania oraz zbiory danych są rozproszone pomiędzy agentami różnych typów. Zaprojektowana architektura została zaimplementowana w języku Java przy użyciu środowiska wspomagającego budowę systemów wieloagentowych JADE. Wykonana aplikacja pozwoliła na zrealizowanie wielu eksperymentów numerycznych, których wyniki zostaną opisane w dalszej części artykułu.

W zaproponowanej przez autorów architekturze hierarchicznego systemu wieloagentowego, używanego w procesie poszukiwania podobieństwa strukturalnego białek, działają trzy następujące grupy agentów:

- Agent nadzorujący (*Supervisory Agent*), który jest odpowiedzialny za komunikację z użytkownikiem, przekazywanie danych do dalszego przetworzenia do agentów zarządzających (agenty niższego szczebla), scalanie rezultatów procesu poszukiwania podobieństwa białek otrzymanych od agentów zarządzających.
- Agenty zarządzające (*Control Agents*) – odpowiedzialne za komunikację i rozdział zadań pomiędzy agentów wyszukiwania oraz przesył danych do agenta nadzorującego.
- Agenty wyszukiwania (*Searching Agents*) – realizujące proces poszukiwania podobieństwa strukturalnego białek na fragmencie danych z określonej bazy danych przez porównanie struktury zadanej z określonymi strukturami z tej bazy.

Sposób działania systemu jest następujący. Przyjmijmy, że przez  $C$  będziemy oznaczać skończony zbiór agentów zarządzających  $C = \{C_i : i = 1, \dots, I\}$ . Każdy agent zarządzający  $C_i$  kontroluje skończony zbiór agentów wyszukiwania  $S_i = \{S_{ij} : j = 1, \dots, J\}$ . Każdy agent wyszukiwania  $S_{ij}$  ma dostęp do skończonego zbioru metod poszukiwania podobieństwa strukturalnego białek, np. FatCat [11], CE [8] lub innych. Używając wybranej metody, agent wyszukiwania prowadzi proces porównania i dopasowania  $V_{S_{ij}}(QP, DP)$  dla kwerendowego białka zadanego  $QP$  oraz białka z bazy danych  $DP$ , obliczając przy tym miary podobieństwa i inne, dodatkowe wartości (w zależności od algorytmu).

Każdy agent wyszukiwania  $S_{ij}$  oblicza podobieństwo (wartości miary podobieństwa) pomiędzy zadanym białkiem  $QP$  i grupą białek z bazy danych, która jest określona przez agenta zarządzającego  $C_i$  (*Control Agent*), zlecającego zadanie. Agent zarządzający  $C_i$  przekazuje do agenta wyszukiwania następujące informacje:

- strukturę kwerendową białka zadanego (jako plik w formacie PDB lub identyfikator PDB ID),
- metodę poszukiwania podobieństwa i wszystkie wymagane przez nią parametry,
- adresy baz danych (np. przez adres IP), z których może pobierać struktury potrzebne do prowadzenia procesu porównania,
- listę struktur białkowych z właściwej bazy danych, które powinny być porównane w pojedynczym procesie poszukiwania podobieństwa.

Opierając się na zdefiniowanych ustawieniach, każdy agent wyszukiwania  $S_{ij}$  przeprowadza procesy porównania i dopasowania, a następnie zwraca listę wyników poszukiwania podobieństwa  $L_{S_{ij}}(k)$  do agenta zarządzającego  $C_i$ , gdzie  $k$  jest numerem zwróconej listy. Gdy agent wyszukiwania zwróci pierwszą listę do agenta zarządzającego ( $L_{S_{ij}}(1)$ ), ten przydziela mu następną grupę struktur do porównania aż do momentu, kiedy zostanie przetworzony cały zbiór danych strukturalnych. Po otrzymaniu częściowej listy wyników od agenta wyszukiwania  $S_{ij}$  ( $j=1, \dots, J$ ) agent zarządzający scala otrzymaną listę wyników  $L_{S_{ij}}(k)$  z własną listą wyników  $SL_{C_i}$  (początkowo pustą) do nowej listy  $SL_{C_i}$ , która jest sortowana pod względem wartości miary podobieństwa od najbardziej do najmniej podobnej struktury w odniesieniu do struktury białka zadanego  $QP$ . Długość tej listy nie powinna przekroczyć wartości  $B$ , która jest określana przez użytkownika na początku procesu poszukiwania. Ze względu na fakt, iż niektóre struktury z bazy danych struktur białkowych mogą dawać takie same wartości miary podobieństwa, długość tej listy może być czasami większa. Jeśli dochodzi do takiej sytuacji i lista ta jest dłuższa, np. ma rozmiar  $B + G$ , musi być spełniony następujący warunek:

$$\forall_{g \in \{1, \dots, G\}} Sim(QP, SL_{C_i}(B)) = Sim(QP, SL_{C_i}(B + g)), \quad (1)$$

gdzie  $Sim(p_1, p_2)$  jest globalną wartością miary podobieństwa pomiędzy dwoma białkami  $p_1$  i  $p_2$ , natomiast  $SL_{C_i}(l)$  jest  $l$ -tą strukturą na posortowanej liście  $SL_{C_i}$ . Globalna wartość miary podobieństwa  $Sim(p_1, p_2)$  uwzględnia wyniki porównań i dopasowań białek parami  $V_{S_{ij}}(p_1, p_2)$ , wykonywanych przez agenty wyszukiwania  $S_{ij}$ , które obliczają miary podobieństwa struktur w zależności od przyjętej metody.

W związku z przedstawionym opisem powinny zostać spełnione następujące dwa warunki:

$$\forall_{i \in \{1, \dots, J\}} \forall_{l \in \{1, \dots, B+G_i-1\}} Sim(QP, SL_{C_i}(l)) \geq Sim(QP, SL_{C_i}(l+1)) \quad (2)$$

oraz

$$\forall_{i \in \{1, \dots, J\}} \forall_{l \in \{1, \dots, B+G_i\}} SL_{C_i}(l) \in \bigcup_{j \in \{1, \dots, J\}} \bigcup_{k \in \{1, \dots, K_j\}} \{L_{S_{ij}}(k)\}. \quad (3)$$

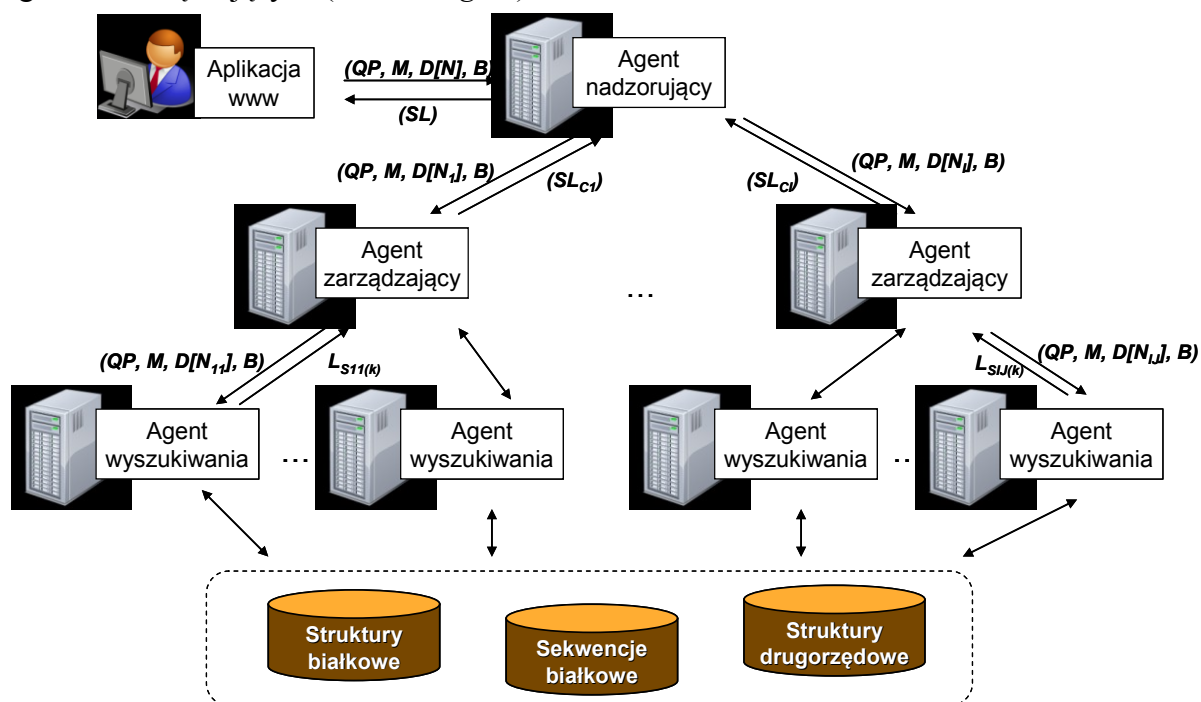
Kiedy wszystkie struktury z bazy danych przypisane do agenta zarządzającego  $C_i$  zostaną już przeanalizowane i porównane, posortowana lista wyników  $SL_{C_i}$  zostaje zwrócona do

agenta nadzorującego (*Supervisory Agent*), który tworzy ostateczną listę wynikową  $SL$ . Lista ta jest tworzona na podstawie list otrzymanych od wszystkich agentów zarządzających  $C_i$  ( $i = 1, \dots, I$ ), w podobny sposób jak to wykonywały agenty zarządzające. Ostateczna lista jest zwracana do użytkownika, który na tej podstawie dowiadyuje się, jakie struktury są w budowie przestrzennej najbardziej zbliżone do struktury przez niego zadanej. Przedstawiony przebieg procesu poszukiwania podobieństwa białek powinien spełniać następujący warunek:

$$\forall_{d \in D} \exists_{S_{ij}} V_{S_{ij}}(QP, d), \quad (4)$$

gdzie  $D$  jest zbiorem baz struktur białkowych, które podlegają porównaniu do zadanej struktury  $QP$ . Warunek ten oznacza, że każde białko ze zbioru baz danych zostanie porównane z białkiem zadanym przez agenta wyszukiwania  $S_{ij}$ .

Główna idea działania hierarchicznego systemu wieloagentowego dla poszukiwania podobieństwa strukturalnego białek została zaprezentowana na rys. 1. Użytkownik systemu zadaje białko kwerendowe  $QP$ , dla którego chce znaleźć białka o podobnej strukturze, a także wybiera metodę poszukiwania podobieństwa  $M$  w wybranym zbiorze baz danych  $D$  oraz określa długość listy  $B$ , czyli liczbę najbardziej podobnych struktur, które chciałby zobaczyć w wyniku przeprowadzenia procesu. Przez graficzny interfejs użytkownika (GUI) informacje te trafiają do agenta nadzorującego (*Supervisory Agent*). Agent nadzorujący dzieli zbiór danych ( $N$  struktur) i przypisuje rozłączne zakresy struktur ( $N_1, \dots, N_I$ ) do każdego z obecnych agentów zarządzających (*Control Agent*).



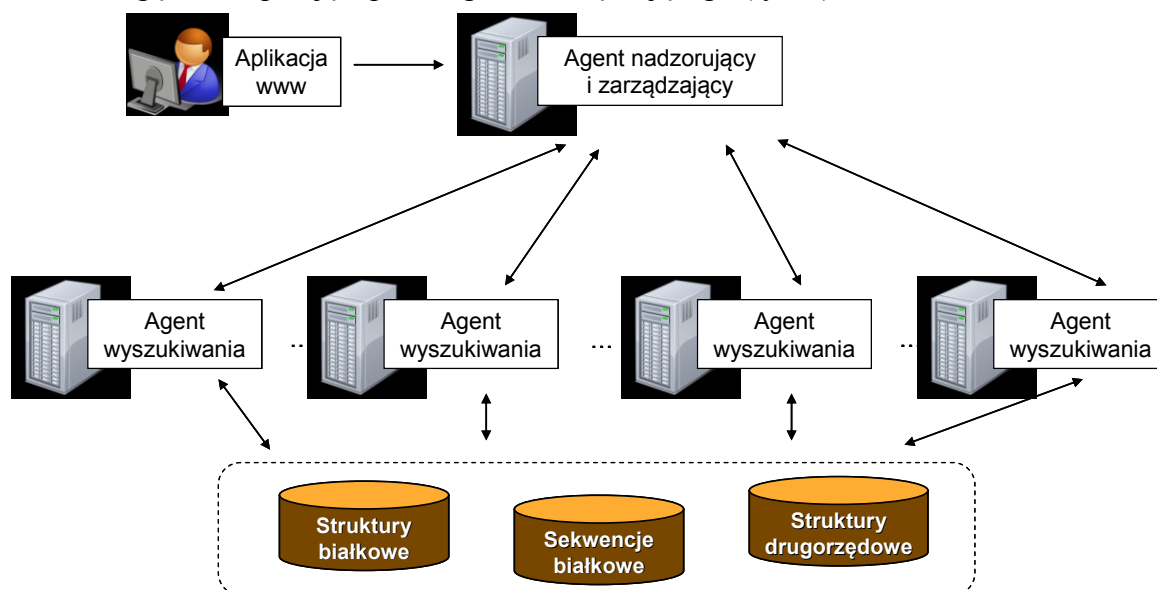
Rys. 1. Architektura hierarchicznego systemu wieloagentowego dla poszukiwania podobieństwa strukturalnego białek

Fig. 1. Architecture of the hierarchical multiagent system for protein structure similarity searching

Agenci zarządzający przekazują do wykonania zadanie przeprowadzenia procesu porównania struktury zadanej z mniejszą grupą struktur ( $N_{11}, \dots, N_{II}$ ) do agentów wyszukiwania (*Searching Agents*). Każdy z agentów wyszukiwania zwraca listę wyników procesu poszukiwania podobieństwa ( $L_{C_{11}}(k), \dots, L_{C_{II}}(k)$ ), które są scalane przez każdego z agentów zarządzających ( $SL_{C_1}, \dots, SL_{C_I}$ ) do nowej, posortowanej listy o długości  $B$ . W końcowej fazie procesu posortowane listy są zwracane do agenta nadzorującego, który dokonuje ostatecznego scalenia list do końcowej listy wynikowej  $SL$ . Lista ta jest zwracana do użytkownika.

### 2.3. Różne implementacje hierarchicznego systemu wieloagentowego

Architektura przedstawiona na rys. 1 może być różnie implementowana w zależności od potrzeb i zastosowania hierarchicznego systemu wieloagentowego w procesie poszukiwania podobieństwa molekuł biologicznych. W specjalnym przypadku, kiedy  $I = 1$ , co oznacza, że w systemie istnieje tylko jeden agent zarządzający, agent nadzorujący może być zbędny i jego zadania mogą zostać przejęte przez agenta zarządzającego (rys. 2).



Rys. 2. Zredukowana architektura hierarchicznego systemu wieloagentowego dla poszukiwania podobieństwa strukturalnego białek z pojedynczym agentem zarządzającym

Fig. 2. Reduced architecture of the hierarchical multiagent system for protein structure similarity searching with a single control agent

Architektura z pojedynczym agentem nadzorującym i wieloma agentami zarządzającymi jest przydatna w następujących przypadkach:

- różne grupy agentów wyszukiwania wykonują inne zadania, wykonują te same zadania przy użyciu różnych metod lub grupy agentów przetwarzają rozłączne zbiory danych;
- grupa agentów wyszukiwania dołącza się do trwających obliczeń i będzie kontrolowana przez osobnego agenta zarządzającego;



- obliczenia obejmują sporą grupę komputerów, a konsolidacja wyników i komunikacja pomiędzy agentami byłaby zbyt obciążająca dla pojedynczego agenta zarządzającego.

Architektura z pojedynczym agentem zarządzającym, odgrywającym również rolę agenta nadzorującego (rys. 2), jest odpowiednia w sytuacjach, gdy:

- liczba agentów wyszukiwania jest stosunkowo mała, a koordynacja ich działań i komunikacja nie stanowią problemu dla pojedynczego agenta;
- nie ma potrzeby odizolowywania grupy agentów ze względu na wykonywane zadania, używane metody lub przetwarzane dane.

### 3. Eksperymenty numeryczne

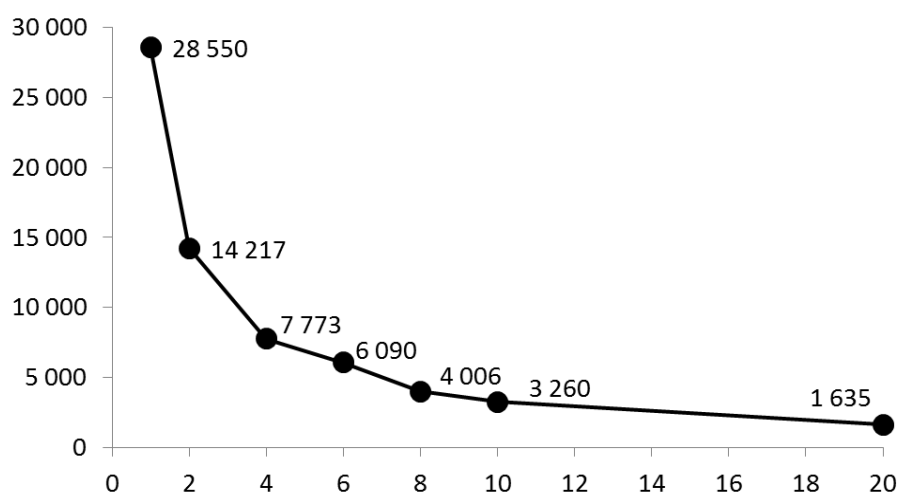
W celu oceny skuteczności zaproponowanej metody wyszukiwania podobieństw białek w hierarchicznym systemie wieloagentowym, opisanym w poprzedniej sekcji, przeprowadzono wiele eksperymentów numerycznych. Rozpraszenie niezbędnych obliczeń pomiędzy wieloma pojedynczymi agentami wyszukiwania (*Searching Agents*) miało na celu przyspieszenie procesu wyznaczania podobieństwa kwerendowego białka QP zadanego przez użytkownika do białek z wybranej przez niego bazy danych, a także innych, dodatkowych wartości w zależności od zadanego algorytmu.

Do implementacji metod poszukiwania podobieństwa strukturalnego białek wybrano język programowania Java, który wydaje się naturalnym rozwiązaniem ze względu na przyjęte założenia projektowe, takie jak to, że aplikacja umożliwiająca poszukiwania podobieństwa strukturalnego białek w środowisku wieloagentowym powinna być uruchamiana w przeglądarce internetowej oraz wymagane jest rozproszenie obliczeń na wielu komputerach. Do procesu równoleglenia obliczeń została użyta platforma JADE (Java Agent Development), która stanowi napisane w języku Java środowisko wspomagające budowę systemów wieloagentowych. Agenci platformy JADE mogą być uruchamiani na zdalnych komputerach, komunikują się pomiędzy sobą przy użyciu komunikatów, a ich działanie w środowisku rozproszonym można kontrolować, implementując klasy zachowań (ang. *Behaviour*) [25].

W przeprowadzonych eksperymentach użyto tylko jednego agenta zarządzającego (*Control Agent*), zatem agent ten przejął również zadania agenta nadzorującego (*Supervisory Agent*), który w takim przypadku nie był wykorzystywany. Procesy agentów wyszukiwania (*Searching Agents*) oraz agenta zarządzającego (*Control Agent*) były uruchamiane na różnych komputerach klasy PC, mających procesory i pamięć operacyjną o zbliżonych parametrach. Wszystkie komputery używane w eksperymentach pracowały pod kontrolą systemów operacyjnych Windows XP Professional lub Windows 7. Zgodnie z założeniami architektury agen-

ci wyszukiwania (*Searching Agents*) mają dostęp do skończonego zbioru metod poszukiwania podobieństwa strukturalnego białek; w tym przypadku zaimplementowano dwie, często stosowane metody: FatCat oraz CE.

W dalszej części zostaną omówione wyniki przeprowadzonych eksperymentów numerycznych, w których jako zadane białko użyto Muconate Lactonizing Enzyme z bazy danych Protein Data Bank (PDB), przy zadanej metodzie wyszukiwania CE. Białko to w bazie PDB jest identyfikowane przez 1MUC i składa się z dwóch aminokwasowych łańcuchów. Proces wyszukiwania podobieństwa obejmował podzbiór bazy PDB, zawierający 850 losowo wybranych struktur proteinowych, które zostały rozproszone na wielu komputerach. Liczba agentów wyszukiwania (*Searching Agents*) przyjmowała wartości z zakresu od 1 do 20, a mianowicie: 1, 2, 4, 6, 8, 10 oraz 20. Liczba białek przydzielanych każdemu z agentów w pojedynczym zadaniu była równa 5 ( $N_{11} = N_{12} = \dots = N_{1j}$ ). Rysunek 3 przedstawia wyniki przeprowadzonego eksperymentu w postaci wykresu obrazującego zależność czasu obliczeń od liczby agentów wyszukiwania.



Rys. 3. Zależność czasu obliczeń od liczby agentów wyszukiwania

Fig. 3. Time of computation as a function of Searching Agents number

Tabela 1

Przyspieszenie obliczeń dla zwiększającej się liczby agentów wyszukiwania

Liczba agentów wyszukiwania	Przyspieszenie
2	2,01
4	3,67
6	4,69
8	7,13
10	8,76
20	17,46

Dla lepszego zobrazowania skuteczności zaproponowanego rozwiązania w tabeli 1 przedstawiono uzyskane przyspieszenie obliczeń względem przypadku użycia wyłącznie jednego agenta wyszukiwania. Na szczególną uwagę zasługuje pierwszy wiersz tej tabeli, gdzie zmie-

rzony przyspieszenie było nieco ponaddwukrotne, co można tłumaczyć faktem, że zapewne drugi agent wyszukiwania był uruchomiony na komputerze o nieco wyższej wydajności. Dla liczby agentów powyżej dwóch wciąż można zaobserwować znaczne przyspieszenie względem przypadku użycia wyłącznie jednego agenta wyszukiwania, aczkolwiek przyspieszenie to nie jest liniową funkcją liczby agentów wyszukiwania, lecz przyjmuje wartości nieco mniejsze. Przyczyn tego zjawiska można dopatrywać się w potrzebie dodatkowej komunikacji między agentem zarządzającym (*Control Agent*) a agentami wyszukiwania, a także w zróżnicowanych możliwościach obliczeniowych komputerów, na których przeprowadzono eksperymenty numeryczne.

#### 4. Podsumowanie

Zrównoleglenie procesu poszukiwania podobieństwa strukturalnego białek w hierarchicznym systemie wieloagentowym prowadzi do skrócenia czasu tego procesu proporcjonalnie do skalowania systemu horyzontalnie, przez dodawanie większej liczby agentów wyszukiwania. Przeprowadzone eksperymenty dowodzą, że rozpraszanie procedury obliczeniowej na wielu komputerach prowadzi do znacznego przyspieszenia procesu poszukiwania. W odróżnieniu od konkurencyjnego systemu ProteinDBS i od tego, co twierdzą jego twórcy, przedstawiony w niniejszym artykule system nie należy do systemów zwracających wyniki w czasie rzeczywistym. Jednakże podczas przeprowadzonych eksperymentów udało się otrzymać lepszy współczynnik przyspieszenia niż w konkurencyjnym systemie ProCKSI. Zaobserwowano również, iż wraz ze wzrostem liczby agentów wyszukiwania przyspieszenie całego procesu poszukiwania rośnie, ale dynamika tego wzrostu nieznacznie spada. Nasze wnioski w tym zakresie są analogiczne do tych, które przedstawili twórcy systemu ProCKSI w artykule [16]. W hierarchicznym systemie wieloagentowym, przedstawionym w tym artykule, zaobserwowany spadek dynamiki jest jednak mniej znaczący i wynika prawdopodobnie z potrzeby komunikacji pomiędzy agentami w systemie.

Podczas tworzenia i testowania przedstawionego systemu użyto komputerów klasy PC o zbliżonych możliwościach obliczeniowych. Jednak autorzy założyli, iż w dowolnym momencie do obliczeń może zostać dołączony jakikolwiek inny komputer, który będzie pracował jako agent wyszukiwania. Do systemu mogą być dołączane nawet komputery ze znacznie gorszymi możliwościami obliczeniowymi. W takim, niezrównoważonym pod względem możliwości obliczeniowych, środowisku agent zarządzający dystrybuje pracę pomiędzy agentami wyszukiwania proporcjonalnie do ich możliwości. Zostało to osiągnięte przez podzielenie całego zbioru struktur białkowych, znajdujących się w bazie danych, na mniejsze pakiety (np. po pięć struktur). Agenci wyszukiwania rezydujący na komputerach o lepszych możliwościach zwykle zwracają

cają wyniki swoich obliczeń szybciej i otrzymują kolejny pakiet struktur do przetworzenia, podczas gdy agenci wyszukiwania, rezydujący na komputerach o słabszych możliwościach mogą, wciąż przetwarzać poprzednio otrzymany pakiet. W ten sposób agent zarządzający optymalizuje czas procesu poszukiwania podobieństwa, przypisując więcej pracy agentom na szybszych maszynach i mniej pracy tym agentom, którzy dłużej przetwarzają otrzymany pakiet.

Warto również zauważyć, że przez zastosowanie systemu wieloagentowego uzyskano nie tylko przyspieszenie procesu poszukiwania podobieństwa białek, ale również zwiększono niezawodność systemu wyszukiwania. Nawet odłączenie się agenta wyszukiwania od środowiska lub brak komunikacji przez dłuższy czas nie przerywa procesu poszukiwania, który w tym samym czasie jest prowadzony przez inne agenty. Przy ewentualnej awarii jednego z agentów brakujące pakiety struktur są przekazywane do ponownego obliczenia kolejnemu, aktywnemu agentowi na zakończenie procesu poszukiwania, kiedy prowadzona jest kontrola otrzymanych wyników zbiorczych. Ponadto przy większej liczbie agentów zarządzających dodatkowo wzrastają stabilność i niezawodność systemu.

Praca naukowa finansowana ze środków na naukę w latach 2008-2011 jako projekt badawczy N N516 265835: *Poszukiwanie podobieństwa strukturalnego białek w rozproszonym środowisku wieloagentowym.*

## BIBLIOGRAFIA

1. Lodish H. et al.: Molecular cell biology, 4th edition. W.H. Freeman and Company, New York 2001.
2. Allen J. P.: Biophysical chemistry. Wiley-Blackwell, Chichester 2008.
3. Branden C., Tooze J.: Introduction to protein structure. Garland, 1991.
4. Gu J., Bourne P. E.: Structural Bioinformatics (Methods of Biochemical Analysis), 2 edition. Wiley-Blackwell, 2009.
5. Gibas C., Jambeck P.: Developing bioinformatics computer skills, 1<sup>st</sup> edition. O'Reilly, Sebastopol 2001.
6. Gibrat J. F., Madej T., Bryant S. H.: Surprising similarities in structure comparison. Curr Opin Struct Biol, Vol. 6(3), 1996, s. 377÷385.
7. Holm L., Sander C.: Protein structure comparison by alignment of distance matrices. J Mol Biol., Vol. 233(1), 1993, s. 123÷138.
8. Shindyalov I. N., Bourne P. E.: Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Engineering, Vol. 11(9), 1998, s. 739÷747.

9. Shapiro J., Brutlag D.: FoldMiner and LOCK2: protein structure comparison and motif discovery on the web. *Nucleic Acids Res.*, Vol. 32, 2004, s. 536÷541.
10. Yang J.: Comprehensive description of protein structures using protein folding shape code. *Proteins*, Vol. 71(3), 2008, s. 1497÷1518.
11. Ye Y., Godzik A.: Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, Vol. 19(2), 2003, s. 246÷255.
12. Zhu J. H., Weng Z. P.: FAST: A novel protein structure algorithm. *Proteins*, Vol. 58, 2005, s. 618÷627.
13. Berman H. M., Westbrook J., Feng Z., Gilliland G., Bhat T. N., Weissig H. et al.: The Protein Data Bank. *Nucleic Acids Res.*, No. 28, 2000, s. 235÷242.
14. Shyu C.-R. et al.: ProteinDBS: a Real-Time Retrieval System for Protein Structure Comparison. *Nucleic Acids Research*, No. 32, 2004, s. 572÷575.
15. Barthel D. et al.: ProCKSI: a Decision Support System for Protein (Structure) Comparison, Knowledge, Similarity and Information. *BMC Bioinformatics*, Vol. 8(416), 2007, s. 1÷22.
16. Folino G. et al.: Towards a Distributed Framework for Protein (Structure) Comparison, Knowledge, Similarity and Information (ProCKSI). UK eScience All Hands Meeting (AHM 2008), Edinburgh, UK 2008, s. 1÷2.
17. Franklin S., Graesser A.: Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. *LNCS*, Vol. 1193, 1996, s. 21÷35.
18. Jain L. C., Chen Z., Ichalkaranje N. (eds.): *Intelligent agents and their applications*. Physica-Verlag, Heidelberg 2002.
19. Jennings N. R., Sycara K., Wooldridge M.: A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, Vol. 1, 1998, s. 7÷38.
20. Tan V. V., Yoo D. S., Shin J. C., Yi M. J.: A Multiagent System for Hierarchical Control and Monitoring. *Journal of Universal Computer Science*, Vol. 15, No. 13, 2009, s. 2485÷2505.
21. Tolk A., Uhrmacher A. M.: Agents: Agenthood, Agent Architectures, and Agent Taxonomies, [in:] Yilmaz L. et al. (eds.): *Agent-Directed Simulation and Systems Engineering*. Wiley-VCH, Weinheim 2009, s. 75÷109.
22. Wakulicz-Deja A., Przybyła-Kasperek M.: Hierarchical Multi-Agent System, [in:] Kłopotek M. A., et al. (eds.): *Recent Advances in Intelligent Information Systems*. EXIT, Warszawa 2009, s. 615÷628.
23. Choiński D., Nocoń W., Metzger M.: Multi-agent system for hierarchical control with self-organising database, [in:] Nguyen N. T., Grzech A., Howlett R. J., Jain L. C. (eds.): *KES-AMSTA 2007*. LNCS (LNAI), Vol. 4496 Springer, Heidelberg 2007, s. 655÷664.

24. Weiss G. (ed.): *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge 1999.
25. Bellifemine F. et al.: *JADE, A White Paper*. 2003, <http://jade.tilab.com/papers/2003/WhitePaperJADEEXP.pdf>.
26. Mrozek D., Małysiak B., Augustyn W.: *Agent-supported Protein Structure Similarity Searching*. *Lecture Notes in Artificial Intelligence*, Vol. 5044, Springer-Verlag GmbH, 2008, s. 49÷61.
27. Mrozek D., Małysiak B., Kozielski S.: *EAST: Energy Alignment Search Tool*. *Springer-Verlag GmbH, LNAI*, Vol. 4223, 2006, s. 696÷705.

Wpłynęło do Redakcji 26 stycznia 2012 r.

## **Abstract**

Proteins are molecules of life in all living organisms. Since they act as substrates in all biochemical reactions, they are very important molecules in organisms' cells. Functions of proteins strictly depend on the internal construction of proteins, built up with hundreds of amino acids and therefore, thousands of atoms. Therefore, on the basis of the information regarding protein internal construction, we are able to identify functions of newly discovered proteins. This is very important for the understanding of evolution of organisms, since we are able to observe how they differentiated during thousands of years.

Protein structure similarity searching is a process of exploring protein structures in order to find molecules having an identical or related construction. In the process, we want to find certain relationships between particular regions of protein structures, regardless of the conformational changes that could appear as a result of environmental factors or binding the molecule to a ligand or other molecules. However, protein structure similarity searching is very complex and time-consuming. The main reason of this is that the construction of proteins is complex on its own. Due to these problems, the scientific community still seeks appropriate and efficient methods for protein structure similarity searching. The parallelization of the similarity searching is one of the techniques, which can be used to accelerate the search process.

In the paper, we present the architecture of the hierarchical multi-agent system dedicated to protein structure similarity searching. In the presented system, we distribute the computational procedure and data sets that are used in the search process. Using multi-agent system,

we do not need any expensive hardware solutions. On the contrary, we can build a powerful framework ourselves on basis on standard PC computers. This gives us the possibility to raise the power of our framework at any moment by inviting other computers or farms of workstations to participate in the search process. In the paper, we also demonstrate different scenarios how this architecture can be used and modified in similarity searching of biological molecules. Additionally, we show results of several numerical experiments confirming a suitability of the presented architecture in the exploration of protein structures.

### **Adresy**

Bożena MAŁYSIAK-MROZEK: Politechnika Śląska, Instytut Informatyki,  
ul. Akademicka 16, 44-100 Gliwice, Polska, bozena.malysiak@polsl.pl.

Alina MOMOT: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,  
44-100 Gliwice, Polska, alina.momot@polsl.pl.

Dariusz MROZEK: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,  
44-100 Gliwice, Polska, dariusz.mrozek@polsl.pl.

Michał MOMOT: Instytut Techniki i Aparatury Medycznej, ITAM  
ul. Roosevelta 118, 41-800 Zabrze, Polska, michal.momot@itam.zabrze.pl.