

Załącznik 2a: Autoreferat

1. Imię i Nazwisko

Szymon Wąsik

2. Posiadane Stopnie i Tytuły Naukowe

1. Stopień doktora nauk technicznych - Politechnika Poznańska, Wydział Informatyki; dyscyplina informatyka, 26 marca 2013 r., „Bioinformatyczne modele i algorytmy infekcji wirusowych”, promotor: prof. dr hab. inż. Jacek Błazewicz, promotor pomocniczy: dr inż. Piotr Łukasiak:
 - doktorat wyróżniony przez Radę Wydziału Informatyki,
 - osiągnięcia będące podstawą nadania stopnia doktora nagrodzone Nagrodą Ministerstwa Nauki i Szkolnictwa Wyższego za osiągnięcia naukowe.
2. Tytuł zawodowy magistra - Politechnika Poznańska, Wydział Informatyki; kierunek - informatyka; specjalizacja - inteligentne systemy wspomaganie decyzji, 28 sierpnia 2008 r., „Komputerowe modele infekcji wirusowych”, promotor: prof. dr hab. inż. Jacek Błazewicz:
 - medal „summa cum laude” (przyznano 5 medali na 4000 absolwentów).
3. Tytuł zawodowy inżyniera - Politechnika Poznańska, Wydział Informatyki i Zarządzania; kierunek - informatyka, 2006 r., „IntelliForest – siecio-centriczny system do monitorowania i ochrony lasów”, promotor: dr inż. Mikołaj Sobczak:
 - projekt realizowany w ramach pracy inżynierskiej jako 1 z 10 na świecie zakwalifikował się do prezentacji na finale konkursu CSIDC w Waszyngtonie (na ok. 300 projektów).

3. Dotychczasowe zatrudnienie w jednostkach naukowych

1. Instytut Informatyki, Wydział Informatyki, Politechnika Poznańska, adiunkt, 1 etat, 1.10.2013 – obecnie.
2. Instytut Chemii Bioorganicznej Polskiej Akademii Nauk, Poznań, adiunkt, ½ etatu, 1.08.2016 – obecnie.
3. Luksemburskie Centrum Medycyny Systemowej, Uniwersytet Luksemburski, staż naukowy, 1 etat, 1.06.2015 – 26.07.2015.
4. Instytut Informatyki, Wydział Informatyki, Politechnika Poznańska, asystent, 1 etat, 1.10.2008 – 30.09.2013.
5. Instytut Informatyki, Wydział Informatyki, Politechnika Poznańska, starszy referent techniczny, 1 etat, 1.02.2006 – 30.09.2008.
6. School of Computing Science, University of Nottingham, wykonawca w projekcie naukowym, ok. ½ etatu (rozliczane umowami zleceniami), 1.03.2008 – 30.06.2008.

4. Wskazanie osiągnięcia naukowego wynikającego z ustawy o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki

4.1. Tytuł osiągnięcia naukowego

Bioinformatyczne metody modelowania systemów dynamicznych w biologii

4.2. Lista prac wchodzących w skład osiągnięcia naukowego

[A1] S. Wasik, Modeling Biological Systems using Crowdsourcing, *Foundations of Computing and Decision Science*, vol. 43(3), 2018, doi:10.1515/fcds-2018-0012 [**Lista B MNiSW: 15 pkt., indeksowane w WoS bez IF**]

[A2] N. Szostak, S. Wasik, and J. Blazewicz, Hypercycle, *PLOS Computational Biology* 12(4): e1004853, 2016, doi:10.1371/journal.pcbi.1004853. [**IF 2016: 4.542**]

[A3] S. Wasik, F. Fratczak, J. Krzyskow, and J. Wulnikowski, Inferring mathematical equations using crowdsourcing, *PLOS ONE*, vol. 10(12), iss. e0145557, 2015, doi:10.1371/journal.pone.0145557. [**IF 2015: 3.057**]

[A4] S. Wasik, M. Antczak, J. Badura, A. Laskowski, T. Sternal, A Survey on Online Judges and Their Applications, *ACM Computing Surveys*, vol. 51(1), art. 3, 2018, doi:10.1145/3143560. [**IF 2017: 5.550**]

[A5] N. Szostak, J. Synak, M. Borowski, S. Wasik, and J. Blazewicz, Simulating the origins of life: The dual role of RNA replicases as an obstacle to evolution, *PLOS ONE*, 12(7): e0180827, 2017, doi:10.1371/journal.pone.0180827. [**IF 2017: 2.766**]

[A6] S. Wasik, P. Jackowiak, M. Figlerowicz, and J. Blazewicz, Multi-agent model of hepatitis C virus infection, *Artificial Intelligence in Medicine*, vol. 60(2), pp. 123-131, 2014, doi:10.1016/j.art-med.2013.11.001. [**IF 2014: 2.019**]

[A7] S. Wasik, T. Prejzendanc, and J. Blazewicz, ModelLang – a new approach for experts-friendly viral infections modeling, *Computational and mathematical methods in medicine*, vol. 2013, p. 8, 2013, doi:10.1155/2013/320715. [**IF 2013: 1.018**]

4.3. Omówienie celu naukowego ww. prac oraz osiągniętych wyników wraz z omówieniem ich ewentualnego wykorzystania

4.3.1. Cel badań

Głównym celem badań opisanych w pracach wchodzących w skład osiągnięcia naukowego było opracowanie nowych metod modelowania systemów dynamicznych w biologii. Modelowanie takich systemów wymaga często współpracy interdyscyplinarnej specjalistów z różnych dziedzin, takich jak matematyka, informatyka, chemia, fizyka, medycyna i oczywiście biologia. Dlatego głównym założeniem dla prowadzonych badań było, aby metody te były jak najbardziej zrozumiałe, nie tylko dla osób posiadających specjalistyczną wiedzę z dziedziny związanej z modelowanym systemem lub stosowaną metodyką modelowania, ale również dla ich współpracowników.

4.3.2. Kontekst badań

Modele, które były rozpatrywane w ramach prowadzonych badań pochodzą z dziedziny *biologii systemowej*, która jest stosunkowo nowym obszarem biologii. Został on zdefiniowany jako przeciwieństwo podejścia stosowanego przez wiele stuleci, zgodnie z którym rozwój biologii napędzany był podejściem redukcjonistycznym. Zgodnie z nim, aby zrozumieć działanie organizmu jako całości niezbędne było wyodrębnienie podstawowych procesów życiowych i zrozumienie każdego z nich osobno [1]. Naukowcy byli przekonani, że jeżeli uda im się szczegółowo zdefiniować każdy proces biologiczny, dostarczy to kompletnej wiedzy, z której będzie można bezpośrednio wyciągnąć wnioski jak działa cały organizm. To z kolei doprowadziło do ciągłego rozwoju metod obserwacyjnych, które pozwoliły na badanie organizmu z coraz większą liczbą szczegółów, od sposobu funkcjonowania całych organów, przez tkanki, komórki aż do poszczególnych procesów komórkowych, a dzięki metodom biochemicznym i biofizycznym nawet na poziomie poszczególnych cząsteczek i atomów. Powyższe podejście ostatecznie doprowadziło do precyzyjnego opisanie setek procesów zachodzących w organizmach żywych, ale nie udzieliło odpowiedzi na wiele kluczowych pytań. Dobrym przykładem są neurony w mózgu. Zasada transmisji impulsów elektrycznych jest bardzo dobrze rozumiana, jednak nie wyjaśnia ona jak powstają myśli lub jak przebiega proces myślowy [2].

Naukowcy zaczęli zauważać powyższy problem w drugiej połowie dwudziestego wieku, kiedy doszli do wniosku, że oprócz analizowania poszczególnych procesów potrzebne jest również spojrzenie bardziej globalne. Dało to początek nowej dziedzinie biologii nazwanej *biologią systemową* lub *biologią systemów*, która bada złożone interakcje występujące w systemach biologicznych. Jednym z głównych celów biologii systemowej jest analiza zachowań, które można zaobserwować jedynie z perspektywy globalnej (z angielskiego tak zwane *emergent properties*), czyli przeprowadzanie analiz, których brakowało w podejściu redukcjonistycznym. Dlatego też biologia systemowa może być traktowana jako najważniejsze z podejść holistycznych stosowanych w biologii [3].

Formalnie termin biologia systemowa został wprowadzony w 1966 roku przez Mihajlo Mesarovicę poprzez organizację międzynarodowego sympozjum *Systems Theory and Biology* [4]. Jednakże pierwsze badanie w tym obszarze były prowadzone dużo wcześniej, chociażby w celu opracowania modelu propagacji sygnału wzdłuż aksonu komórki nerwowej, który łączył opis zachowania cząsteczek sodu i potasu w celu zaobserwowania bardziej globalnego procesu transmisji impulsu elektrycznego [5]. Obecnie biologia systemowa to mocno interdyscyplinarna dyscyplina łącząca osiągnięcia nauk takich jak biologia, matematyka, informatyka, chemia i biochemia, fizyka i biofizyka, a nawet psychologia i socjologia. Badania przeprowadzane w ramach biologii systemowej są wykorzystywane w różnorodnych obszarach biologii, takich jak genomika, transkryptomika, proteomika, matabolomika i wielu innych.

Jednym z rodzajów modeli, które są najpowszechniej analizowane w ramach biologii systemowej są modele dynamiczne, czyli takie, które opisują zmianę pewnych wartości w czasie. Formalnie, modele takie mogą być zdefiniowane jako systemy, które dla każdego punktu czasowego $t \in T$ odwzorowują pewien punkt przestrzeni fazowej $m \in M$ w inny punkt $m' \in M$. Tak zdefiniowane systemy opisane są funkcją:

$$T \times M \rightarrow M,$$

i zazwyczaj definiowane są za pomocą równania różniczkowego lub układu równań różniczkowych, które modelują zmianę wartości punktów m w czasie:

$$\frac{dm}{dt} = f(m).$$

Opracowanie nowatorskich metod modelujących takie systemy było właśnie głównym celem przeprowadzonych badań.

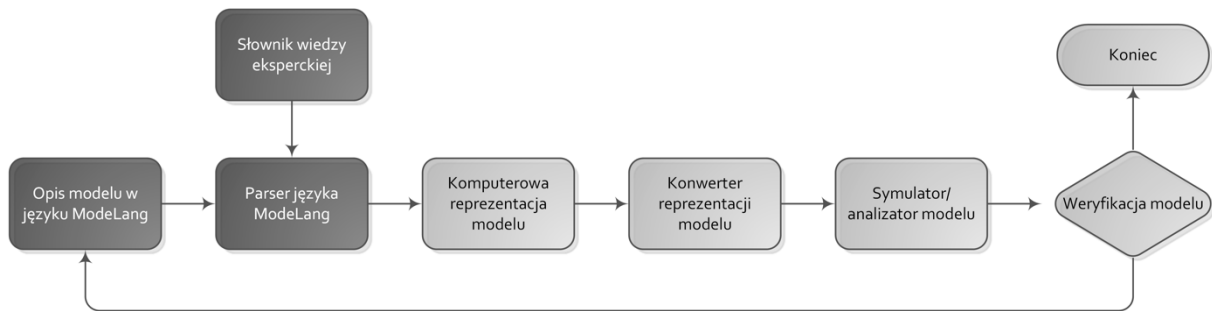
4.3.3. Osiągnięte wyniki

Zapis modeli biologicznych z wykorzystaniem kontrolowanych języków naturalnych

Do definiowania modeli systemów biologicznych opracowane zostało wiele formalizmów. Po pierwsze modele można zapisywać za pomocą wzorów matematycznych oraz ich komputerowej reprezentacji. Po drugie do opisywania modeli w biologii systemowej zaprojektowany został dedykowany format SBML [6], który zyskał duże uznanie w środowisku naukowym i wykorzystywany jest aktualnie przez kilkadziesiąt programów wspierających proces modelowania. Po trzecie zaprojektowane zostały dedykowane języki programowania wspierające modelowanie i symulowanie modeli, w tym modeli biologicznych, takie jak SimPy [7]. Wszystkie powyższe podejścia mają jednak podstawową wadę. Skierowane są one do konkretnej grupy naukowców, odpowiednio matematyków, biologów lub informatyków. Nie stanowi to zazwyczaj problemu dla bioinformatyków, którzy posiadając interdyscyplinarną wiedzę są w stanie wybrać najbardziej odpowiadającą im metodę modelowania z powyższej listy. Jednak, jak wynika z doświadczenia habilitanta, istotny problem pojawia się w przypadku współpracy interdyscyplinarnych zespołów, których członkowie posiadają doświadczenie w różnych obszarach nauki. Konkretnie, problem został zaobserwowany w trakcie współpracy, którą habilitant prowadził jako lider zespołu informatycznego opracowującego wieloagentowy symulator infekcji wirusowych [A6] we współpracy z biologami z Instytutu Chemii Bioorganicznej PAN w Poznaniu. W trakcie współpracy pojawił się znaczący problem komunikacyjny. Biolodzy nie byli w stanie zrozumieć formalizmów matematycznych używanych do modelowania, a informatycy nie byli w stanie uzyskać wystarczająco precyzyjnych informacji na temat procesów biologicznych zachodzących w zainfekowanym organizmie, które pozwoliłyby na ich odwzorowanie w implementowanym symulatorze.

Aby wspomóc komunikację pomiędzy naukowcami i rozwiązać powyższy problem opracowany został język ModeLang oparty na koncepcji kontrolowanych języków naturalnych (ang. Controlled Natural Language, CNL) [8]. Powyższa grupa języków definiowana jest jako podzbiór języków naturalnych, których gramatyka i słownictwo jest ograniczona w ten sposób, aby zmniejszyć lub wyeliminować niejednoznaczność w interpretacji języka. Dlatego też ModeLang jest to język, który korzysta z zalet języka naturalnego, dając biologom możliwość definiowania modelu korzystając ze zdań wprowadzanych w języku angielskim. Z drugiej strony informatycy mogą nadzorować proces modelowania, tak aby zapewnić jednoznaczność procesu parsowania opisu modelu, tak jak ma to miejsce w językach formalnych. Listing 1 przedstawia przykładowy model infekcji wirusem HCV zapisany w języku ModeLang, natomiast Rysunek 1 przedstawia szczegółowo algorytm modelowania z wykorzystaniem języka ModeLang. Opis modelu przygotowany przez eksperta jest przetwarzany przez parser języka, który wykorzystuje słownik wiedzy eksperckiej, aby być w stanie poprawnie zrozumieć wiedzę dziedzinową użytą w opisie modelu (np. słowa takie jak *infekuje*, *mnoży się*, *emituje*, *umiera*). Słownik taki musi być przygotowany dla każdej domeny, z której będą pochodzić modelowane systemy. Wynikiem działania parsera jest komputerowa reprezentacja modelu, która może być później przekształcona przez dedykowany konwerter do jednego ze standardowych typów danych używanych do analizowania modeli (np. SBML). Natomiast po konwersji model może być analizowany w dowolnym z programów komputerowych obsługujących docelowy format. Ostatecznie, następuje faza weryfikacji modelu i jeżeli okaże się, że należy

wprowadzić w nim pewne zmiany, cały proces powtarza się od początku. Szczegółowy opis działania języka ModelLang znajduje się w publikacji [A7].



RYSUNEK 1 ALGORYTM MODELOWANIA Z WYKORZYSTANIEM JĘZYKU MODELANG.

```

0 < s < 4
0.002 < rHU < 3.4
0 < rHI < 1.4e-6
1e-3 < dHU < 0.014
1e-3 < dHI < 0.5
4e6 < Umax < 1.3e7
1e-8 < beta < 1e-6
0.1 < pH < 44
0.8 < cv < 22
0 < cI < 1
Healthy hepatocytes are created by Healthy hepatocytes at speed rHU
Infected hepatocytes are created by Infected hepatocytes at speed rHI
The number of Healthy hepatocytes and Infected hepatocytes is less than
Umax
Blastic cells change into Healthy hepatocytes at speed s
Healthy hepatocytes die with mean time dHU
Infected hepatocytes die with mean time dHI
Healthy hepatocytes and Virions generate Infected hepatocytes with prob-
ability beta
Virions are emitted by Infected hepatocytes at rate pH
Virions die with probability cv
Infected hepatocytes change into Healthy hepatocytes with probability cI
  
```

LISTING 1 PRZYKŁADOWY ZAPIS MODELU INFЕКCJI HCV W JĘZYKU MODELANG.

Język ModelLang został przetestowany z grupą bioinformatyków, która korzystając z niego przygotowała sumarycznie ok. 300 testowych opisów modeli. Przeprowadzone testy wskazują, że język ModelLang może być skutecznie zastosowany do wsparcia współpracy interdyscyplinarnej z zakresu bioinformatyki. Jego koncepcja została opublikowana w czasopiśmie z listy JCR *Computational and Mathematical Methods in Medicine* [A7] oraz przedyskutowana na kilku konferencjach międzynarodowych. Koncepcja języka ModelLang została zdefiniowana przez habilitanta, natomiast dalsze prace nad językiem ModelLang prowadzone były we współpracy z doktorantem Tomaszem Prejzencancem, w którego przewodzie doktorskim habilitant pełni rolę promotora pomocniczego.

Wieloagentowy symulator modeli biologicznych

System wieloagentowy to system złożony z wielu wchodzących we wzajemne interakcje obiektów, tak zwanych „agentów”. Charakterystyczną cechą agentów jest to, że nie mają one nigdy pełnej wiedzy o całym systemie, a jedynie o najbliższych im sąsiadach. W ten sposób zachowanie systemu wieloagentowego jest zachowaniem wyłaniającym się z lokalnych interakcji, a nie analizy globalnego stanu środowiska [9].

Systemy wieloagentowe już od dłuższego czasu stosowane są do modelowania systemów biologicznych [10]–[13]. Z dużym sukcesem były przykładowo stosowane do modelowania procesu nowotworzenia [14]–[17]. Jednak w wielu obszarach biologii i medycyny ich popularność jest cały czas niewielka, a ich potencjał do analizowania skomplikowanych procesów zmiennych w czasie i przestrzeni niewykorzystany. Takimi przykładowymi obszarami, które zostały uwzględnione w badaniach prowadzonych przez habilitanta są modelowanie infekcji wirusowych oraz analiza początków życia na Ziemi. Zastosowanie modeli wieloagentowych do modelowania przebiegu infekcji wirusowych było dotychczas bardzo ograniczone [18]–[20] i obarczona pewnymi problemami metodycznymi. Natomiast symulacje w obszarze RNA World ograniczały się do uproszczonego względem metodologii wieloagentowej sposobu modelowania z wykorzystaniem automatów komórkowych [21], [22]. Dlatego też szczególnie tym dwóm obszarom poświęcono wiele uwagi.

W kontekście infekcji wirusowych habilitant zaprojektował i zaimplementował symulator wieloagentowy infekcji wirusowych, który następnie został przetestowany w oparciu o model infekcji HCV. Celem badań była weryfikacja modelu oraz opracowanie metody, która umożliwiłaby po pierwsze modelowanie rozkładu przestrzennego komórek, po drugie przypisanie komórkom specyficznych atrybutów o wartościach różnych dla każdej komórki. Taki model umożliwiłby przykładowo analizę jak proces mutacji wirusa wpływa na przebieg infekcji (taka zależność była analizowana w starszej publikacji habilitanta [05]). Zastosowany model infekcji został zaimplementowany na podstawie wyników prac zespołu profesora Perelzona [23] i został zdefiniowany następująco:

$$\begin{aligned}\dot{V} &= (1 - \epsilon)pI - cV, \\ \dot{I} &= r_I I \left(1 - \frac{U + I}{U_{max}}\right) + (1 - \eta)\beta VU - d_I I - qI, \\ \dot{U} &= s + r_U U \left(1 - \frac{U + I}{U_{max}}\right) - (1 - \eta)\beta VU - d_U U + qI.\end{aligned}$$

W powyższym modelu V oznacza liczbę wolnych wirionów, I liczbę zainfekowanych komórek wątroby (hepatocytów), a U liczbę niezainfekowanych hepatocytów. Parametr p opisuje jak szybko wiriony emitowane są przez zainfekowane komórki, a c jak szybko ulegają one rozpadowi. r_I oraz r_U to odpowiednio prędkość podziału zainfekowanych i niezainfekowanych hepatocytów, a d_I oraz d_U to prędkość ich umierania. β to prędkość infekowania zdrowych komórek, q to niewielka wartość oznaczająca proces usuwania wirusa z zainfekowanych komórek, a s niewielka wartość mogąca modelować powstawanie hepatocytów z komórek macierzystych. U_{max} oznacza maksymalną liczbę hepatocytów, które mogą występować w wątrobie (pojemność środowiska), natomiast ϵ oraz η modelują terapię (dla braku terapii oba parametry mają wartość 0).

Efektom przeprowadzonych testów modelu i symulatora było oprócz potwierdzenia poprawności działania symulatora znalezienie poważnego błędu w jednym z aktualnie istniejących modeli i zaproponowanie jego rozwiązania. Ponieważ w trakcie testów okazało się, że zadanie symulacji jest

bardzo złożone obliczeniowo opracowano i przebadano również wersję symulatora działającą równolegle. Dodatkowo opracowano metodę wyznaczania wartości parametrów występujących w tym modelu z wykorzystaniem algorytmu genetycznego, gdyż żadna z istniejących metod nie mogła być w tym celu zastosowana. Szczegółowe rezultaty badań zostały opublikowane w publikacji [A6]. Należy tutaj zauważyć, że mimo iż podstawy symulatora powstały jeszcze w ramach doktoratu habilitanta, to kluczowe jego funkcjonalności oraz zasadnicze analizy biologiczne zostały zaimplementowane i przeprowadzone już po uzyskaniu stopnia doktora w ramach grantu NCN, w którym habilitant pełnił rolę wykonawcy. Również po obronie doktoratu opracowana została publikacja [A6].

Symulator wieloagentowy wykorzystywany do modelowania infekcji wirusowych został zaprojektowany w sposób uniwersalny i elastyczny, umożliwiający jego łatwe dostosowanie do innych zastosowań. Dlatego też, w trakcie prac w tematyce początków życia na Ziemi, szybko pojawił się pomysł wykorzystania go do zamodelowania systemu replikaza-pasożyt, który hipotetycznie mógł występować na prebiotycznej Ziemi. System ten modeluje w jaki sposób replikujące cząsteczki RNA (replikaza) oraz cząsteczki, które w skutek mutacji utraciły zdolność replikacji (pasożyty) mogły ze sobą koegzystować. System ten może być zamodelowany z wykorzystaniem równań różniczkowych w następujący sposób:

$$\begin{aligned}\dot{R} &= -2k_R R^2 + [2(1 - k_R) + 3\kappa\omega + 2d]C_R \\ &\quad - k_P(1 - l)RP + [(1 - k_P) + \kappa\omega + d]C_{RP} - dR, \\ \dot{P} &= -k_P(1 - l)RP + [(1 - k_P) + 2\kappa\omega + d]C_{RP} - dP, \\ \dot{C}_{RR} &= k_R R^2 - [(1 - k_R) + \kappa\omega]C_{RR} - 2dC_{RR}, \\ \dot{C}_{RP} &= k_P(1 - l)RP - [(1 - k_P) + \kappa\omega]C_{RP} - 2dC_{RP},\end{aligned}$$

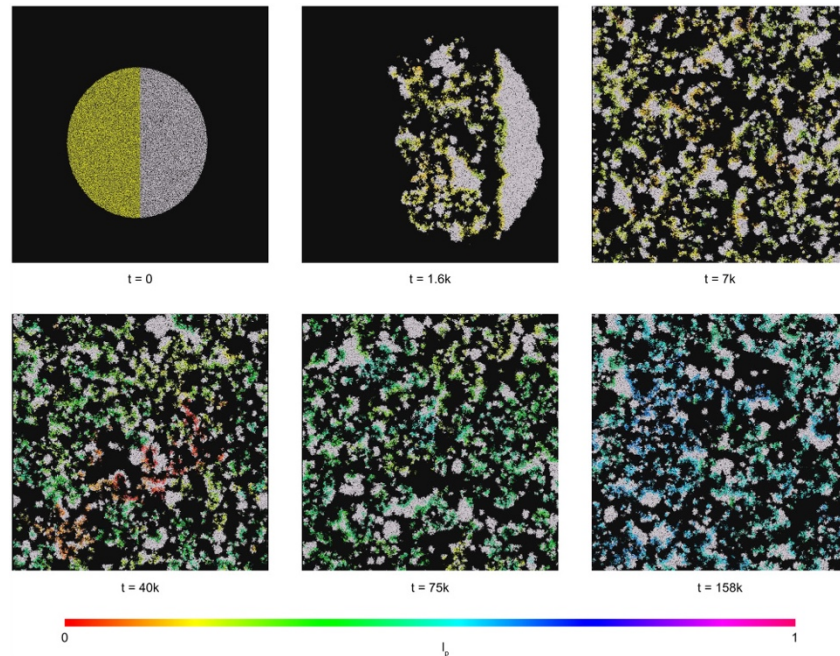
gdzie:

$$\omega = 1 - R - P - C_{RP} - C_{RR}.$$

W powyższym modelu R oznacza liczbę replikaz, P liczbę pasożytów, C_{RR} liczbę kompleksów replikaza-replikaza, czyli dwóch połączonych cząsteczek replikazy, z których jedna replikuje (kopiuje) drugą, a C_{RP} liczbę kompleksów replikaza-pasożyt, w których replikaza kopiuje bezużyteczną cząsteczkę pasożyta nie mając możliwości wykrycia typu kopiowanej sekwencji. Parametry k_r oraz k_p występujące w modelu oznaczają powinowactwo względem replikazy dla odpowiednio każdej innej replikazy oraz pasożytów. Regulują one czas życia kompleksu (im większe powinowactwo, tym mniejsze prawdopodobieństwo, że kompleks się rozpadnie). Dodatkowo każda replikaza i sekwencja ulega rozpadowi w tempie d . κ to stała replikacji opisująca jak szybko sekwencja jest kopiowana, a l to prawdopodobieństwo przebywania cząsteczki w stanie zwiniętym w którym jej skopiowanie jest utrudnione, za to jest to niezbędny stan, aby replikaza mogła kopiować inne sekwencje. W trakcie badań rozważano również wariant, w którym wartość parametru l była różna dla replikaz i pasożytów (odpowiednio l_R oraz l_P).

Celem prowadzonych badań była identyfikacja cech, które musiały charakteryzować prebiotyczny system replikaza-pasożyt, aby zagwarantować jego przeżywalność i dalszą ewolucję. Natomiast dzięki wykorzystaniu systemów wieloagentowych możliwe było bardziej realistyczne zamodelowanie przestrzeni względem wykorzystywanych wcześniej automatów komórkowych. Rysunek 2 przedstawia przykładowe wizualizacje symulowanego systemu wraz z postępowaniem symulacji. Analiza systemu pozwoliła potwierdzić wcześniejsze doniesienia, że stabilność systemu gwarantowana jest przez występowanie formacji wędrujących fal cząsteczek [21], jednak lepsze

odzworowanie przestrzeni pozwoliło doprecyzować, że ich struktura bardziej przypomina eksplozje życia występujące w różnych punktach środowiska. Dodatkowo udało się wykazać, że szybkie pojawienie się kwasu DNA na Ziemi mogło być jednym z krytycznych warunków niezbędnych do dalszego rozwoju życia. Być może pojawienie się tych cząstek było niezbędne dużo wcześniej niż dotychczas przypuszczano. Szczegóły implementacji symulatora oraz wyniki wszystkich przeprowadzonych analiz prezentowane są w publikacji [A5].



RYСУNEK 2 ZRZUTY EKRANA Z OKNA SYMULATORA PREZENTUJĄCE MODELOWANY SYSTEM W KOLEJNYCH KROKACH SYMULACJI. LICZBY POD OBRAZKAMI OZNACZAJĄ KROK SYMULACJI. KOLOR OZNACZA PRAWDOPODOBIEŃSTWO PRZEBYWANIA W STANIE ZWIŃĘTYM.

Zastosowanie crowdsourcingu do modelowania systemów biologicznych

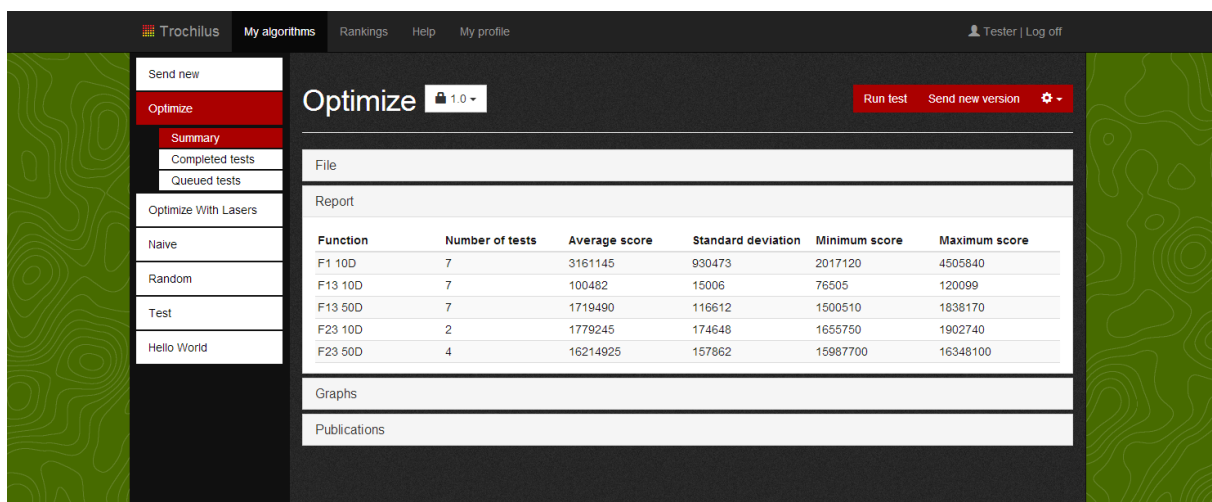
Crowdsourcing to technika zlecenia wykonania jakiegoś zadania dużej, często anonimowej grupie osób, które realizują je licząc na pewną nagrodę lub traktując to jako realizację swojej pasji. Dobrym przykładem jest największa na świecie encyklopedia – Wikipedia, która rozwijana jest wspólnym wysiłkiem milionów osób. Mimo że termin *crowdsourcing* został wprowadzony dopiero w roku 2006, jego zastosowanie jest dużo starsze i sięga początku XVIII wieku. Więcej informacji o crowdsourcingu oraz opis jego zastosowań do modelowania systemów biologicznych można znaleźć w samodzielnej publikacji habilitanta [A1]. Opisane w niej możliwości zastosowania crowdsourcing w biologii systemowej stanowiły ważny element badań habilitanta, dlatego osiągnięte wyniki są również krótko zaprezentowane poniżej.

W trakcie prac nad symulatorem wieloagentowym opisanym w poprzedniej sekcji zauważono, że jednym z większych problemów, które trzeba było rozwiązać, było wyznaczenie wartości parametrów rzeczywistych występujących w modelu. Istnieje co prawda wiele metod matematycznych, które są w stanie wykonać to zadanie dla modeli opisanych równaniami różniczkowymi (przykładowo [24]), ale nie można ich zastosować dla modeli wieloagentowych. Zachowanie tych modeli często opisane jest niestandardowymi, złożonymi i nieróżniczkowalnymi zależnościami, co zdecydowanie ogranicza liczbę możliwych do zaaplikowania metod. Dodatkowo ewaluacja zbioru wartości parametrów jest bardzo złożona obliczeniowo, gdyż wymaga przeprowadzenia symulacji wieloagentowej, która może trwać wiele minut, a nawet godzin. W końcu brakuje

ustandaryzowanego zestawu testów, które mogłyby wesprzeć opracowywanie efektywnych metod, które można by zastosować w przypadku modeli wieloagentowych.

Dobrym przykładem inicjatywy, która wspiera opracowywanie nowych metod optymalizacji ciągłej jest konkurs organizowany corocznie w czasie konferencji IEEE CEC [25], w czasie którego oceniane i porównywane są algorytmy rozwiązujące różne warianty problemu estymacji wartości parametrów rzeczywistych. Co prawda opracowywane tam metody nie mogą być wykorzystane wprost w przypadku systemów wieloagentowych, bo celem algorytmów jest optymalizacja stosunkowo prostych funkcji matematycznych, których czas ewaluacji jest mierzony w milisekundach, ale sama inicjatywa opracowania standardowych zestawów testów (benchmarków) oceniających algorytmy jest bardzo istotna z punktu widzenia naukowego.

Jedynym problemem koncepcyjnym konkursów organizowanych w ramach IEEE CEC jest brak możliwości łatwego porównania algorytmów po zakończeniu konkursu. Autorzy nowych lub ulepszonych metod, które powstaną po zakończeniu konkursu, nie mają łatwego dostępu do środowiska obliczeniowego, w którym mogliby porównać je z historycznymi metodami opracowanymi przez innych autorów. Aby rozwiązać ten problem, a jednocześnie umożliwić ewaluację metod przeznaczonych dla systemów wieloagentowych, przez zespół kierowany przez habilitanta zaproponowany został portal Trochilus. Jest to działająca w chmurze aplikacja Internetowa, w której zdefiniowano standardowe testy opracowane na podstawie problemów modelowania infekcji HCV oraz początków życia na Ziemi oraz umożliwiono naukowcom uruchamianie implementacji własnych metod w trybie on-line. Procedura wykorzystania platformy jest taka, że naukowiec po opracowaniu nowej metody zgłasza jej kod źródłowy przez dedykowaną stronę Internetową. Kod algorytmu jest następnie przez system kompilowany i automatycznie uruchamiany na niezmiennym zestawie testów. W ten sposób każdy twórca może na bieżąco porównać nowo opracowany algorytm z innymi dodanymi już do bazy systemu Trochilus. W ramach prac nad portalem opracowano również metodykę bezpiecznego uruchamiania kodu programów pochodzących z zewnętrznych źródeł w systemie Linux. Rysunek 3 przedstawia przykładowy zrzut ekranu z platformy Trochilus, a więcej informacji o jej działaniu można znaleźć w publikacji [A1].



The screenshot shows the Trochilus portal interface. The top navigation bar includes 'Trochilus', 'My algorithms', 'Rankings', 'Help', and 'My profile'. The user is logged in as 'Tester' and can 'Log off'. The main content area is titled 'Optimize' and features a 'Run test' button and a 'Send new version' button. A sidebar on the left contains navigation options: 'Send new', 'Optimize', 'Summary', 'Completed tests', 'Queued tests', 'Optimize With Lasers', 'Naive', 'Random', 'Test', and 'Hello World'. The main content area displays a 'Report' table with the following data:

| Function | Number of tests | Average score | Standard deviation | Minimum score | Maximum score |
|----------|-----------------|---------------|--------------------|---------------|---------------|
| F1 10D | 7 | 3161145 | 930473 | 2017120 | 4505840 |
| F13 10D | 7 | 100482 | 15006 | 76505 | 120099 |
| F13 50D | 7 | 1719490 | 116612 | 1500510 | 1838170 |
| F23 10D | 2 | 1779245 | 174648 | 1655750 | 1902740 |
| F23 50D | 4 | 16214925 | 157862 | 15987700 | 16348100 |

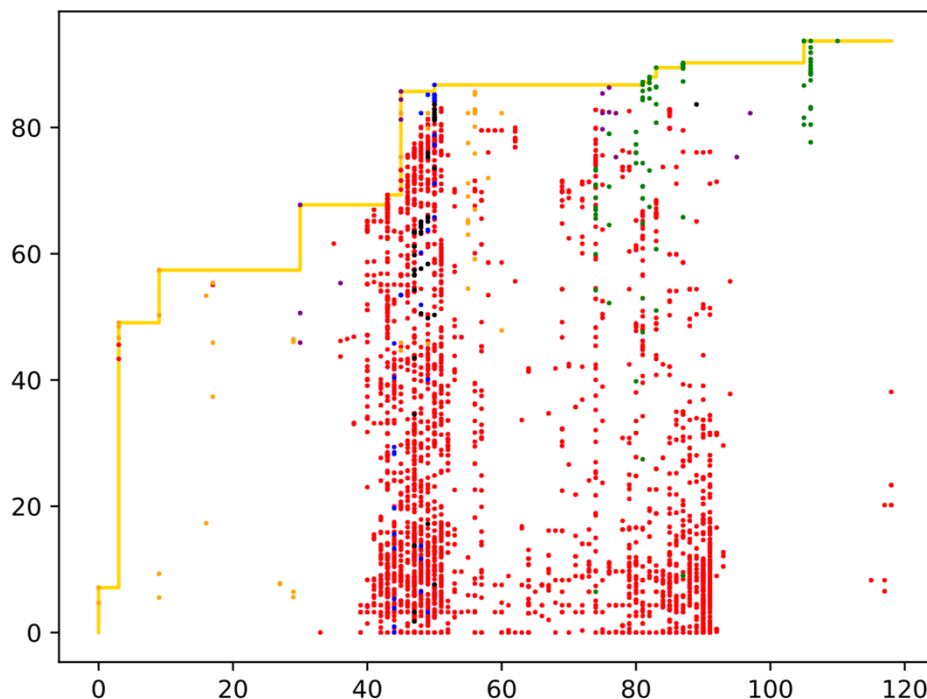
Below the table, there are sections for 'Graphs' and 'Publications'.

RYСУNEK 3 PRZYKŁADOWA STRONA PORTALU TROCHILUS PREZENTUJĄCA WYNIKI ALGORYTMU NA PIĘCIU RÓŻNYCH TESTACH.

Architektura zastosowana w platformie Trochilus jest często nazywana Evaluation-as-a-Service (EaaS, [26], [27]) zgodnie z terminologią stosowaną dla usług działających w chmurze. W architekturze tej wszystkie dane oraz skrypty testowe przechowywane są w chmurze, a co

najważniejsze sama ewaluacja algorytmów w oparciu o te dane również odbywa się w chmurze, dzięki czemu algorytmy testowane są w dokładnie takich samych warunkach. Doświadczenia zebrane w trakcie implementacji architektury EaaS w portalu Trochilus nasunęły pomysł, że można by powyższą platformę rozszerzyć tak, aby umożliwiała ewaluację nie tylko algorytmów rozwiązujących jeden konkretny problem, ale dowolnych algorytmów optymalizacyjnych. Platforma taka byłaby unikalna w skali świata i byłaby niezwykle użytecznym narzędziem naukowym, gdyż problemy optymalizacyjne to bardzo popularne typy problemów kombinatorycznych występujące zarówno w biologii [28], [29], jak i w innych dziedzinach nauki [30]. W ten sposób zaproponowana została platforma Optil.io, która została zaimplementowana w ramach grantu finansowanego przez NCBiR w ramach program LIDER, którego habilitant był kierownikiem.


Architektura platformy Optil.io została zaprojektowana na podstawie doświadczeń zdobytych w trakcie implementacji portalu Trochilus i również wykorzystuje mechanizm EaaS. Jednak w odróżnieniu od portalu Trochilus platforma Optil.io umożliwia udostępnianie procedur ewaluacyjnych dla dowolnych problemów optymalizacyjnych. Co więcej, platforma ta umożliwia organizowanie konkursów, w których przez pewien zadany z góry czas uczestnicy mogą nasyłać algorytmy rozwiązujące pewien problem. Wszystkie zgłoszenia są na bieżąco oceniane, a uczestnikom prezentowany jest odświeżany na żywo ranking porównujący wszystkich uczestników. Wykres przedstawiający przebieg konkursu dla mającego wiele biologicznych zastosowań problemu komiwożera prezentuje Rysunek 4. Natomiast Rysunek 5 przedstawia zrzut ekranu platformy Optil.io prezentujący wyniki przykładowego konkursu programistycznego.




RYSUNEK 4 WYKRES OBRAZUJĄCY PRZEBIEG KONKURSU, W RAMACH KTÓREGO ROZWIĄZYWANY BYŁ WARIANT PROBLEMU KOMIWOJAZERA. PROBLEM TEN MA LICZNE ZASTOSOWANIA BIOLOGICZNE, NA PRZYKŁAD PRZY ROZWIĄZYWANIU PROBLEMU ASEMBLACJI SEKWENCJI DNA [31]. KAŻDA KROPKA OZNACZA POJEDYNCZE ZGŁOSZENIE. KOLORAMI INNYMI NIŻ CZERWONY OZNACZONO ZGŁOSZENIA PIĘCIU NAJLEPSZYCH UCZESTNIKÓW. NA OSI POZIOMEJ PRZEDSTAWIONY JEST CZAS TRWANIA KONKURSU (W DNIACH), A NA OSI PIONOWEJ ZAGREGOWANA WARTOŚĆ FUNKCJI CELU DLA WSZYSTKICH INSTANCJI TESTOWYCH. ŻÓŁTA LINIA OBRAZUJE NAJLEPSZE W DANYM MOMENCIE ROZWIĄZANIE.

Wyniki konkursów przeprowadzonych zarówno z wykorzystaniem platformy Optil.io, jak i Trochilus potwierdzają, że wykorzystanie crowdsourcingu do rozwiązywania problemów optymalizacyjnych, w tym tych związanych z modelowaniem systemów biologicznych, może przynieść bardzo dobre efekty. Po pierwsze możliwość konkurowania z innymi uczestnikami oraz obserwowania jak zmienia się ich ranking w czasie pozytywnie motywuje uczestników do opracowywania i wysyłania ulepszonych rozwiązań. Po drugie, agregując kilka rozwiązań, które osiągają najlepsze wyniki na różnych instancjach testowych, można uzyskać rozwiązanie lepsze niż bez wykorzystania crowdsourcingu.

SIGN IN SIGN UP



ABOUT ▾
CONTESTS ▾
PROBLEMS
STANDING



This is a problem with limited submission frequency. After each submission you will have to wait 10 minutes before you can submit it again.

OPTIMAL STEINER TREE LITE: PACE 2018 C

By Florian Sikora¹, Edouard Bonnet²

DESCRIPTION
RUNS
STANDING
SUBMIT
DISCUSS

TLE = Time Limit Exceeded, **WA** = Wrong Answer, **RTE** = Runtime Error, **MLE** = Memory Limit Exceeded, **OLE** = Output Limit Exceeded, **PLE** = Processes Limit Exceeded, [more help...](#)

First
1
Last

| # | USER | LANGUAGE | SCORE | TIME [S] | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|-----------------|---------------|-------|----------|--------|--------------|-----------|-----------|--------------|---------------|---------------|------|
| 1 | CIMAT_Team | C++ | 99.97 | 3,000.91 | 98.00 | 6,700,776.00 | 21,517.00 | 18,169.00 | 8,500,743.00 | 88,001,177.00 | 91,000,866.00 | 42,6 |
| 2 | Martin_J_Geiger | VB.NET | 99.89 | 2,662.96 | 98.00 | 6,700,776.00 | 21,565.00 | 18,264.00 | 8,500,742.00 | 88,001,175.00 | 91,000,866.00 | 42,6 |
| 3 | mucha | C++ | 99.76 | 2,996.23 | 99.00 | 6,700,778.00 | 21,589.00 | 18,252.00 | 8,500,742.00 | 88,001,175.00 | 91,000,866.00 | 42,7 |
| 4 | reko | CMake package | 99.73 | 2,118.79 | 100.00 | 6,700,776.00 | 21,517.00 | 18,133.00 | 8,500,739.00 | 88,001,175.00 | 91,000,866.00 | 42,5 |
| 5 | UWarsaw | C++ | 99.70 | 2,723.40 | 99.00 | 6,700,778.00 | 21,661.00 | 18,272.00 | 8,500,746.00 | 88,001,176.00 | 91,000,871.00 | 42,7 |
| 6 | Tarken | CMake package | 99.62 | 3,001.22 | 99.00 | 6,700,778.00 | 21,645.00 | 18,376.00 | 8,500,745.00 | 88,001,175.00 | 91,000,866.00 | 42,7 |
| 7 | swats | Jar file | 99.26 | 3,001.93 | 102.00 | 6,700,777.00 | 21,641.00 | 18,236.00 | 8,500,745.00 | 88,001,177.00 | 91,000,902.00 | 42,8 |
| 8 | Yold | Python 3 | 99.24 | 3,000.71 | 101.00 | 6,700,807.00 | 21,662.00 | 18,310.00 | 8,500,765.00 | 88,001,189.00 | 91,000,881.00 | 43,0 |
| 9 | sgls | C | 98.87 | 3,000.67 | 103.00 | 6,700,776.00 | 21,660.00 | 18,239.00 | 8,500,745.00 | 88,001,175.00 | 91,000,867.00 | 42,6 |

RYSUNEK 5 ZRZUT EKRANU PLATFORMY OPTIL.IO PREZENTUJĄCY WYNIKI PRZYKŁADOWEGO KONKURSU.

Dodatkowym ważnym wynikiem naukowym osiągniętym w czasie prac nad platformą Optil.io było formalne zdefiniowanie tak zwanej platformy *Online judge*, czyli platformy oceniającej w chmurze algorytmy zgodnie z architekturą EaaS i wykonanie przeglądu takich platform. Wyniki te zostały opublikowane we wiodącym czasopiśmie z obszaru informatyki *ACM Computing Surveys* wraz z opisem i przykładowym wykorzystaniem platformy Optil.io [A4].

Zupełnie odmienne zastosowanie crowdsourcingu zostało wykorzystane w trakcie implementacji gier Internetowych służących do modelowania systemów dynamicznych. Opracowane zostały dwie gry *Throw the Hamster* [A3] oraz *Race the Hamster* [A1], w których celem graczy było zaprojektowanie modelu infekcji HCV. Ponieważ opracowywanie modeli matematycznych jest zadaniem wymagającym specjalistycznej wiedzy z zakresu matematyki, zadanie to zostało przedstawione w dużo prostszej formie. Celem gracza było zaprojektowanie pojazdu (odpowiednio rakiety kosmicznej lub samochodu wyścigowego) poprzez skonstruowanie drzewa technologii, które mógł dodawać do swojego pojazdu. Natomiast wewnątrz w grze, drzewo to reprezentowało wzór matematyczny opisujący dany model. Po jego skonstruowaniu model był oceniany w oparciu o rzeczywiste dane kliniczne [32]. Dodatkowo wprowadzono możliwość przeglądania drzew opracowanych przez innych graczy i użycie ich do budowy własnych modeli co pozwoliło na wykorzystanie fenomenu wiedzy tłumu (ang. *wisdom of the crowd*), który umożliwia

wykorzystać zagregowaną wiedzę wielu indywidualnych osób w celu osiągnięcia jak najlepszego rozwiązania [33].

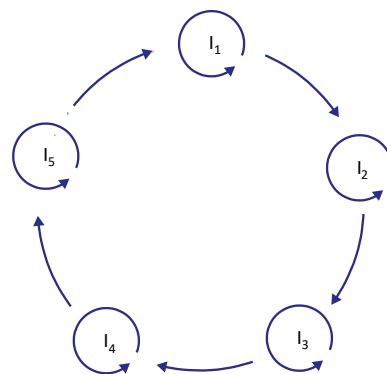
Opracowane zostały dwie wersje gry. Gra *Race the Hamster* usuwała kilka błędów w projekcie gry *Throw the Hamster*, które zostały zaobserwowane w czasie eksperymentów przeprowadzonych z jej udziałem. Były to przede wszystkim niska grywalność (gra była mało ciekawa dla graczy), brak podglądu funkcji odpowiadającej zbudowanemu drzewu oraz konieczność ręcznego ustawiania wartości parametrów. Eksperymenty przeprowadzone z wykorzystaniem obu gier udowodniły, że możliwe jest efektywne wykorzystanie crowdsourcingu w celu modelowania systemów biologicznych. Po pierwsze, udało się z wykorzystaniem kilkuset graczy opracować model lepszy niż model wygenerowany przez wiodący algorytm programowania genetycznego [34]. Po drugie, udało się wykorzystać wiedzę tłumu, gdyż najlepsze rozwiązanie powstało jako ciąg wielu ulepszeń pewnego rozwiązania zaproponowany przez wielu ludzi. Dla przykładu, najlepsza funkcja zaproponowana przez graczy w grze *Race the Hamster* to:

$$f(t) = 649.23 - 1.03 \cdot \log_{2.67} t - 0.01 \cdot t + t \cdot t \cdot (-0.01) \cdot 0.02 \cdot 0.01.$$

Wzór powyższej funkcji nie jest uproszczony, ponieważ została ona przedstawiona dokładnie w takiej formie, w jakiej została zdefiniowana przez gracza. Szczegóły dotyczące zastosowanej metodologii i wyników gry *Throw the Hamster* zostały opisane w publikacji [A3], a *Race the Hamster* w [A1]. W drugiej z tych publikacji znajduje się również spis dziesięciu zasad, które zostały zdefiniowane na podstawie analizy wyników obu gier, a które opisują w jaki sposób należy projektować gry wykorzystujące crowdsourcing w celu rozwiązania problemu naukowego (z angielskiego tak zwane *crowdsourced serious games* [35]).

Opracowanie nowej wersji teorii hipercykli

Teoria hipercykli opracowana została w latach 70-tych XX wieku przez Manfreda Eigena jako rozwiązanie problemu progu błędu (ang. *error threshold*) w pierwszych systemach replikujących cząstek RNA [36], [37]. Problem ten wynika z faktu, że cząsteczki RNA mutują zbyt szybko, aby być w stanie przechować większe ilości informacji bez udziału zaawansowanych mechanizmów korekty błędów. Szacuje się, że na prebiotycznej Ziemi (przed pojawieniem się życia) niemożliwe było powstanie sekwencji dłuższych niż 100 nukleotydów. W celu zwiększenia ilości możliwej do przechowania informacji zaproponowany został model hipercyklu, w którym informacja przechowywana jest w kilku, wzajemnie powiązanych sekwencjach (patrz Rysunek 6).



RYSUNEK 6 PRZYKŁADOWY MODEL HIPERCYKLU ZŁOŻONY Z PIĘCIU SEKWENCJI [BY NSZOSTAK - OWN WORK, CC BY-SA 4.0, [HTTPS://COMMONS.WIKIMEDIA.ORG/W/INDEX.PHP?CURID=41579114](https://commons.wikimedia.org/w/index.php?curid=41579114)].

Zaproponowany przez Eigena model Hipercyklu był przez wiele lat analizowany i rozwijany, przykładowo przez modelowanie dodatkowo etapu translacji lub błon ograniczających możliwości dyfuzji cząstek. Doprowadziło to do sytuacji, w której przez wielu różnych badaczy zaproponowane zostały różne, posiadające niewielkie różnice, definicje teorii hipercykli, w tym czasami warianty wzajemnie sprzeczne. Nawet definicja powyższej teorii w pracach Eigena była rozbita na cztery różne artykuły [36]–[39] stosujące różne oznaczenia. Zdecydowanie utrudniało to prowadzenie badań wykorzystujących teorię hipercykli. Dlatego też postanowiono wykorzystać kolekcję *Topic Pages* czasopisma *PLOS Computational Biology* w celu opracowania i opublikowania ujednoliconej wersji teorii hipercykli. Kolekcja *Topic Pages* to sekcja czasopisma poświęcona publikowaniu artykułów opisujących kluczowe z punktu widzenia biologii obliczeniowej zagadnienia. Publikowane w nich artykuły przechodzą bardzo rygorystyczny proces recenzji, a jej cechą charakterystyczną jest to, że artykuły publikowane są nie tylko w czasopiśmie, ale również jako strony Wikipedii. Dla autorów oznacza to dużo większy wysiłek związany z koniecznością zachowania podwójnych standardów – czasopisma i Wikipedii oraz konieczność opracowania artykułu w dwóch różnych formatach. Za to powyższy sposób publikacji zdecydowanie ułatwia dostęp do informacji wszystkim zainteresowanym osobom oraz podnosi jakość informacji przedstawianych w Wikipedii.

Opisane powyżej prace zaowocowały artykułem [A2], w którym zaproponowano ujednolicony model hipercyklu opisany poniższym równaniem różniczkowym:

$$\dot{x}_i = x_i \left(k_i + \sum_j k_{i,j} x_j - \frac{1}{x} \phi \right),$$

gdzie:

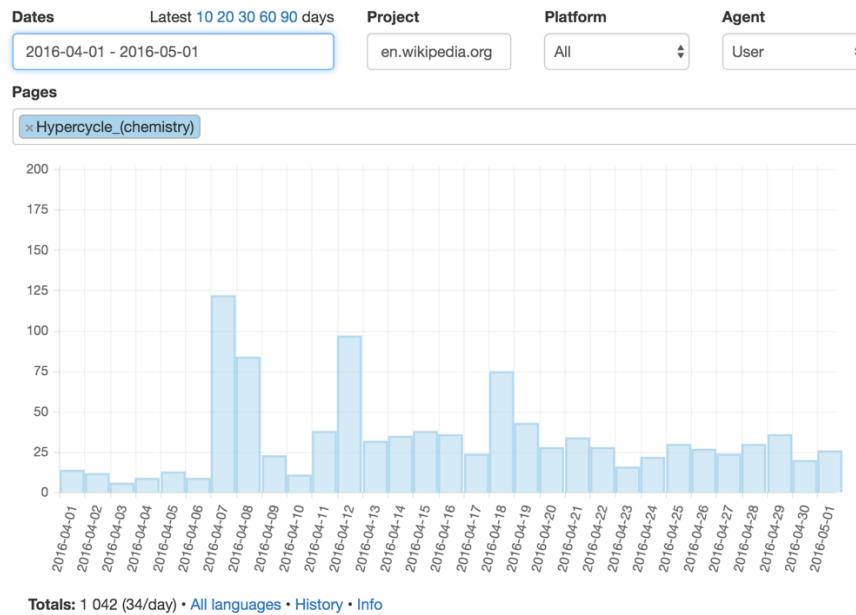
$$x = \sum_i x_i,$$

$$k_i = f_i - d_i.$$

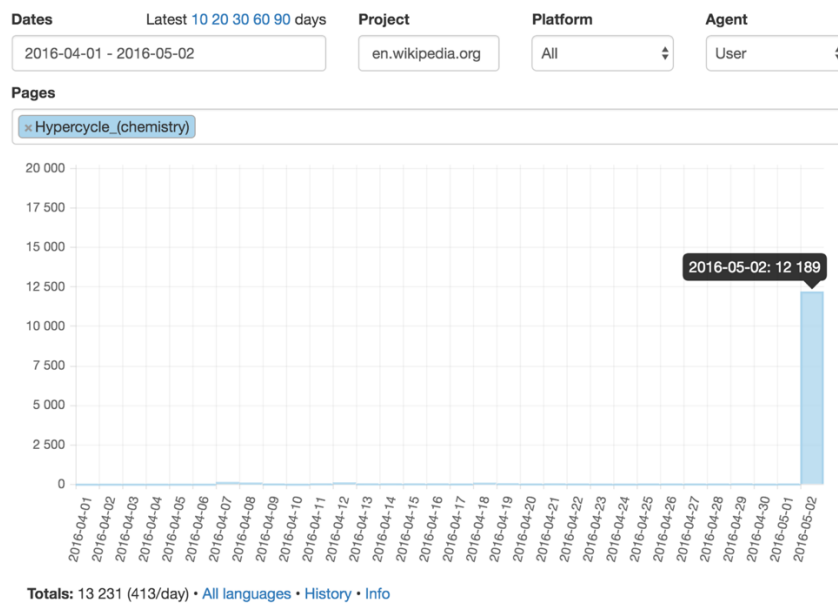
W powyższym modelu x_i oznacza koncentrację sekwencji i -tej, x oznacza koncentrację wszystkich sekwencji, k_i oznacza przyrost liczby sekwencji i -tej stanowiący różnicę pomiędzy liczbą sekwencji powstających w skutek samo-replikacji (f_i) oraz niszczonej w skutek degradacji (d_i). $k_{i,j}$ oznacza prędkość produkcji sekwencji i -tej, która jest katalizowana przez sekwencję j -tą. Natomiast ϕ oznacza upływ sekwencji z systemu, który gwarantuje zachowanie stałego stężenia sekwencji. Oprócz powyższego modelu w pracy [A2] opisane są jeszcze modele hipercyklu z translacją oraz kompartmentami.

Po pozytywnym przejściu złożonego procesu recenzji artykułu uwzględniającego recenzję merytoryczną, ocenę jego wartości dla Wikipedii oraz weryfikację stylu artykułu na podstawie wytycznych Wikipedii, został on opublikowany, a jego sukces przerósł wszystkie oczekiwania. Po pierwsze, od momentu opublikowania artykułu oglądalność strony poświęconej hipercyklam wzrosła aż dwukrotnie (patrz Rysunek 7). Po drugie, artykuł został zgłoszony do publikacji na stronie głównej Wikipedii w sekcji *Did you know*, w której wyróżniające się artykuły mogą być promowane przez 12 godzin. Wymagało to przejścia kolejnych faz recenzji, tym razem dokonanych przez doświadczonych edytorów Wikipedii, jednak ostatecznie, gdy artykuł znalazł się na stronie głównej (Rysunek 9) liczba odwiedzin strony wzrosła z kilkudziesięciu do ponad 12 tysięcy (patrz Rysunek 8). W końcu statystyki wyświetleń i cytowań ze strony czasopisma *PLOS Computational*

Biology świadczą, że mimo że jest to stosunkowo nowy artykuł, jest on również bardzo przydatny dla środowisk naukowych.



RYSUNEK 7 LICZBA ODWIEDZIN ANGIELSKIEJ STRONY WIKIPEDII NA TEMAT HIPERCYKLI PRZED I PO OPUBLIKOWANIU ARTYKUŁU W *PLOS COMPUTATIONAL BIOLOGY* (7 KWIECZNIA 2016). MOŻNA ZAOBSERWOWAĆ, ŻE OPUBLIKOWANIE NOWEGO ARTYKUŁU PODNIOSŁO JEGO OGLĄDALNOŚĆ DWUKROTNIE.



RYSUNEK 8 LICZBA WYŚWIETLEŃ ARTYKUŁU O HIPERCYKLU W DNIU, GDY ZOSTAŁ ON OPUBLIKOWANY NA STRONIE GŁÓWNEJ WIKIPEDII (2 MAJA 2016). TEGO DNIA ARTYKUŁ ZOSTAŁ OBEJRZANY PONAD 12 TYSIĘCY RAZY.

The screenshot shows the Wikipedia main page. On the left is a sidebar with navigation links. The main content area features a 'Welcome to Wikipedia' message, a 'From today's featured article' section with a photo of a noisy miner bird, and a 'Did you know...' section. The first item in the 'Did you know...' section is highlighted with a red box: "... that the **hypercycle** model (*pictured*) could explain how life on Earth might have begun?". To the right of this text is a circular diagram labeled 'A hypercycle' showing a cycle of five nodes connected by arrows.

RYSUNEK 9 WIDOK GŁÓWNEJ STRONY WIKIPEDII Z ARTYKUŁEM O HIPERCYKLU OPUBLIKOWANYM W SEKCJI *DID YOU KNOW*.

5. Omówienie pozostałych osiągnięć naukowo-badawczych

5.1. Lista prac – pozostałe osiągnięcia

[01] D. Mietchen, S. Wodak, S. Wasik, N. Szostak, and C. Dessimoz, Submit a Topic Page to PLOS Computational Biology and Wikipedia, *PLOS Computational Biology* 14(5): e1006137, 2018, doi:10.1371/journal.pcbi.1006137. [IF 2017: 3.955]

[02] N. Szostak, S. Wasik, and J. Blazewicz, Understanding life: a bioinformatics perspective, *European Review*, vol. 25(2), pp. 231-245, 2017, doi:10.1017/S1062798716000570. [IF 2017: 0.434]

[03] T. Prejzendanc, S. Wasik, and J. Blazewicz, Computer representations of bioinformatics models, *Current Bioinformatics*, vol. 11, iss. 5, pp. 551-560, 2016, doi:10.2174/1574893610666150928193510. [IF 2016: 0.600]

[04] M. Popena, M. Szachniuk, M. Blazewicz, S. Wasik, E. K. Burke, J. Blazewicz, and R. W. Adamiak, RNA FRABASE 2.0: an advanced web-accessible database with the capacity to search the three-dimensional fragments within RNA structures., *BMC Bioinformatics*, vol. 11, p. 231, 2010, doi:10.1186/1471-2105-11-231. [IF 2010: 3.029]

[05] S. Wasik, P. Jackowiak, JB. Krawczyk, P. Kedziora, P. Formanowicz, M. Figlerowicz, and J. Blazewicz, Towards prediction of HCV therapy efficiency, *Computational and Mathematical Methods in Medicine*, vol. 11(2), pp. 185-199, 2010, doi:10.1080/17486700903170712. [IF 2010: 0.814]

- [06] S. Wasik, *Bioinformatyczne modele i algorytmy infekcji wirusowych (Bioinformatics models and algorithms of viral infections)*, Poznan, Poland: Nakom, 2012, vol. 10, ISBN: 978-83-89529-98-5.
- [07] S. Wasik, M. Antczak, J. Badura, and A. Laskowski, Evaluation as a Service architecture and crowdsourced problems solving implemented in Optil.io platform, in *Proceeding of Data Science Meets Optimization Workshop, Federated Artificial Intelligence Meeting*, Stockholm, 2018.
- [08] S. Wasik, M. Antczak, J. Badura, A. Laskowski, M. Olszowy, T. Sternal, and K. Wedrowicz, Optil.io Platform: Evaluation as a Service for Metaheuristics, in *Proceeding of the MIC and MAEB 2017 Conferences*, Barcelona, Spain, 2017, pp. 225-227.
- [09] S. Wasik, M. Antczak, J. Badura, A. Laskowski, and T. Sternal, Optil.io: Cloud Based Platform For Solving Optimization Problems Using Crowdsourcing Approach, in *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion*, San Francisco, CA, USA, 2016, pp. 433-436.
- [010] S. Wasik, I. Bladek, T. Chojnacki, K. Miazga, J. Szwachla, and J. Blazewicz, Platforma Trochilus: opracowywanie metod estymacji parametrów systemów biologicznych z wykorzystaniem crowdsourcingu, *Raport techniczny Instytutu Informatyki Politechniki Poznańskiej*, RB-5/16, 2016.
- [011] S. Wasik, F. Fraczak, J. Krzyskow, and J. Wulnikowski, Eksperymentalna weryfikacja metod modelowania systemów dynamicznych z wykorzystaniem crowdsourcingu, *Raport techniczny Instytutu Informatyki Politechniki Poznańskiej*, RB-7/16, 2016.
- [012] S. Wasik, Wstępna implementacja modelu systemu CRISPR-Cas, *Raport techniczny Instytutu Informatyki Politechniki Poznańskiej*, RB-7/15, 2015.
- [013] J. Synak, N. Szostak, S. Wasik, and J. Blazewicz, Application of chemical reaction simulation methods in order to verify RNA World Hypothesis, in *Machine Learning Reports 04/2015*, 2015, p. 6.
- [014] S. Wasik, Two crowdsourcing approaches for solving optimization problems, in *Machine Learning Reports 04/2015*, 2015, p. 16.
- [015] S. Wasik, P. Jackowiak, M. Figlerowicz, and J. Blazewicz, Modeling HCV infection using multi-agent simulation, in *Machine Learning Reports 01/2011*, 2011, pp. 37-41.
- [016] S. Wasik, P. Jackowiak, J. Krawczyk, P. Kedziora, P. Formanowicz, M. Figlerowicz, and J. Blazewicz, A certain model of HCV virus infection, *Institut fur Informatik, Technische Universitat Clausthal* 2009.

5.2. Opis pozostałych osiągnięć

Wiele z wylistowanych w poprzedniej sekcji prac związana jest z podstawową tematyką badawczą habilitanta prezentowaną w sekcji 4.3.3. Najwyżej punktowana z powyższych prac opublikowana w czasopiśmie *PLOS Computational Biology* opisuje doświadczenia z publikacją artykułu w sekcji *Topic Pages* czasopisma i analizuje zagadnienia jak możliwość ta wpływa na rozwój Otwartej Nauki [01]. Tematyka początków życia poruszana jest również w publikacji przeglądowej opublikowanej w *European Review*, która opisuje różne definicje życia i stara się oszacować prawdopodobieństwo powstania pierwszych systemów replikujących cząstek [02]. Natomiast podstawy matematyczne symulowania dyfuzji w takich systemach zostały poruszone w publikacji konferencyjnej [013]. Z kolei sama architektura systemów wieloagentowych mogących służyć do

przeprowadzania takich symulacji została opisana na podstawie modelu infekcji HCV w publikacji konferencyjnej [O15].

W trakcie opracowywania języka Modelang wykonano szczegółowy przegląd istniejących formatów umożliwiających zapisywanie modeli systemów biologicznych. Wyniki tego przeglądu zostały opublikowane przez *Current Bioinformatics* w publikacji [O3].

Bardzo wysoko cytowana publikacja opublikowana w *BMC Bioinformatics*, której habilitant jest współautorem powstała jeszcze w trakcie studiów doktoranckich i opisuje bazę danych fragmentów i struktur RNA nazwaną RNA Frabase [O4]. W ramach opisanych w niej prac habilitant był odpowiedzialny za opracowanie metod wizualizacji danych oraz optymalizację wydajności algorytmów wykorzystywanych przez moduł analityczny przeszukujący całą bazę danych.

Dwie publikacje opisują metodę analizy efektywności terapii infekcji HCV na podstawie różnorodności genetycznej populacji wirusa obecnej w zainfekowanym organizmie [O5, O16]. Metoda opisana w tych publikacjach stanowiła ważny element rozprawy doktorskiej. Natomiast wszystkie wyniki będące podstawą nadania stopnia doktora opisane zostały w monografii podsumowującej rozprawę doktorską wydanej przez wydawnictwo NAKOM [6]. Wyniki te były również podstawą do przyznania habilitantowi Nagrody Ministra Nauki i Szkolnictwa Wyższego.

Analiza różnych aspektów działania platformy Optil.io oraz prezentacja wniosków z różnych przeprowadzonych z wykorzystaniem tej platformy eksperymentów znajdują się w publikacjach [O7, O8, O9], natomiast podobne analizy dla platformy Trochilus przedstawiono w raporcie [O10]. Temat crowdsourcingu jest także poruszany w publikacji, w której opisano wyniki pierwszych analiz, które posłużyły do opracowania gry *Throw the Hamster* [O11]. Natomiast pierwsze propozycje architektury i projektu zarówno gry *Throw the Hamster*, jak i platformy Optil.io zostały przedstawione w publikacji [O14].

W końcu jeden z wymienionych raportów przedstawia pokrótce zastosowanie symulatora wieloagentowego do symulowania mechanizmu odpornościowego CRISPR-Cas [40] w zmieszanych populacjach różnych bakterii [O12]. Raport ten opisuje krótko implementację symulatora oraz otrzymane wyniki w trakcie prac prowadzonych w czasie wakacyjnego stażu na Uniwersytecie Luksemburskim.

Referencje

- [1] P. Nurse, „Reductionism: The ends of understanding”, *Nature*, t. 387, nr 6634, s. 657, 1997.
- [2] P. Fries, „A mechanism for cognitive dynamics: neuronal communication through neuronal coherence”, *Trends Cogn. Sci.*, t. 9, nr 10, s. 474–480, 2005.
- [3] H. Kitano, *Foundations of Systems Biology*. The MIT Press, 2001.
- [4] M. Mesarovic, *Systems Theory and Biology*. Springer Verlag, 1968.
- [5] A. L. Hodgkin i A. F. Huxley, „A quantitative description of membrane current and its application to conduction and excitation in nerve.”, *J Physiol*, t. 117, nr 4, s. 500–544, sie. 1952.
- [6] M. Hucka i in., „The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models.”, *Bioinformatics*, t. 19, nr 4, s. 524–531, mar. 2003.
- [7] K. Muller i T. Vignaux, „SimPy: Simulating Systems in Python”, *ONLampcom Python DevCenter*, luty 2003.

- [8] R. Schwitter, „Controlled natural languages for knowledge representation”, w *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, Beijing, China, 2010, s. 1113–1121.
- [9] M. Wooldridge i N. R. Jennings, „Intelligent agents: theory and practice”, *Knowl. Eng. Rev.*, t. 10, nr 2, s. 115–152, cze. 1995.
- [10] E. Merelli *i in.*, „Agents in bioinformatics, computational and systems biology”, *Brief. Bioinform.*, t. 8, nr 1, s. 45–59, sty. 2007.
- [11] D. Machado, R. S. Costa, M. Rocha, E. C. Ferreira, B. Tidor, i I. Rocha, „Modeling formalisms in systems biology”, *AMB Express*, t. 1, nr 1, s. 1–14, 2011.
- [12] J. O. Dada i P. Mendes, „Multi-scale modelling and simulation in systems biology”, *Integr. Biol. Quant. Biosci. Nano Macro*, t. 3, nr 2, s. 86–96, luty 2011.
- [13] M. T. Klann, A. Lapin, i M. Reuss, „Agent-based simulation of reactions in the crowded and structured intracellular environment: Influence of mobility and location of the reactants”, *BMC Syst. Biol.*, t. 5, s. 71, maj 2011.
- [14] L. Zhang, Z. Wang, J. A. Sagotsky, i T. S. Deisboeck, „Multiscale agent-based cancer modeling”, *J. Math. Biol.*, t. 58, nr 4–5, s. 545–559, kwi. 2009.
- [15] J. M. Osborne *i in.*, „A hybrid approach to multi-scale modelling of cancer”, *Philos. Trans. R. Soc. Lond. Math. Phys. Eng. Sci.*, t. 368, nr 1930, s. 5013–5028, lis. 2010.
- [16] Z. Wang, C. M. Birch, J. Sagotsky, i T. S. Deisboeck, „Cross-scale, cross-pathway evaluation using an agent-based non-small cell lung cancer model”, *Bioinforma. Oxf. Engl.*, t. 25, nr 18, s. 2389–2396, wrz. 2009.
- [17] P. Macklin, S. McDougall, A. R. A. Anderson, M. A. J. Chaplain, V. Cristini, i J. Lowengrub, „Multiscale modelling and nonlinear simulation of vascular tumour growth”, *J. Math. Biol.*, t. 58, nr 4–5, s. 765–798, kwi. 2009.
- [18] G. Zaiyi, H. H. Kwang, i T. J. Cing, „Sufficiency verification of HIV-1 pathogenesis based on multi-agent simulation”, w *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, Washington DC, USA, 2005, s. 305–312.
- [19] F. Mitha, T. Lucas, F. Feng, T. Kepler, i C. Chan, „The Multiscale Systems Immunology project: software for cell-based immunological simulation”, *Source Code Biol. Med.*, t. 3, nr 1, s. 6–6, 2008.
- [20] J. Itakura *i in.*, „Reproducibility and usability of chronic virus infection model using agent-based simulation; comparing with a mathematical model”, *Biosystems*, t. 99, nr 1, s. 70–78, 2010.
- [21] P. Hogeweg i N. Takeuchi, „Multilevel Selection in Models of Prebiotic Evolution: Compartments and Spatial Self-organization”, *Orig. Life Evol. Biosph.*, t. 33, nr 4/5, s. 375–403, 2003.
- [22] N. Takeuchi i P. Hogeweg, „Multilevel Selection in Models of Prebiotic Evolution II: A Direct Comparison of Compartmentalization and Spatial Self-Organization”, *PLoS Comput. Biol.*, t. 5, nr 10, s. e1000542, paź. 2009.
- [23] T. C. Reluga, H. Dahari, i A. S. Perelson, „Analysis of Hepatitis C Virus Infection Models with Hepatocyte Homeostasis”, *SIAM J. Appl. Math.*, t. 69, nr 4, s. 999–1023, 2009.
- [24] C. Zhan i L. F. Yeung, „Parameter estimation in systems biology models using spline approximation.”, *BMC Syst Biol*, t. 5, s. 14, 2011.

- [25] J. J. Liang, B.-Y. Qu, P. N. Suganthan, i A. G. Hernandez-Diaz, „Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session and Competition on Real-Parameter Optimization”, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China, 2013.
- [26] F. Hopfgartner *i in.*, „Report on the Evaluation-as-a-Service (EaaS) Expert Workshop”, *ACM SIGIR Forum*, t. 49, nr 1, s. 57–65, cze. 2015.
- [27] H. Müller *i in.*, „Report on the Cloud-Based Evaluation Approaches Workshop 2015”, *ACM SIGIR Forum*, t. 50, nr 1, s. 38–41, cze. 2016.
- [28] P. Festa, „On some optimization problems in molecular biology”, *Math. Biosci.*, t. 207, nr 2, s. 219–234, cze. 2007.
- [29] H. J. Greenberg, W. E. Hart, i G. Lancia, „Opportunities for Combinatorial Optimization in Computational Biology”, *Inf. J Comput.*, t. 16, nr 3, s. 211–231, cze. 2004.
- [30] H. Friedrich i J. Gump, „Simplified Modeling and Solving of Logistics Optimization Problems”, *Int. J. Transp.*, t. 2, nr 1, 2014.
- [31] P. Meksangsouy i N. Chaiyaratana, „DNA fragment assembly using an ant colony system algorithm”, w *The 2003 Congress on Evolutionary Computation, 2003. CEC '03.*, 2003, t. 3, s. 1756-1763 Vol.3.
- [32] H. Dahari, R. M. Ribeiro, i A. S. Perelson, „Triphasic decline of hepatitis C virus RNA during antiviral therapy”, *Hepatology*, t. 46, nr 1, s. 16–21, 2007.
- [33] F. Galton, „Vox Populi”, *Nature*, t. 75, nr 1949, s. 450–451, mar. 1907.
- [34] M. Schmidt i H. Lipson, „Distilling Free-Form Natural Laws from Experimental Data”, *Science*, t. 324, nr 5923, s. 81–85, kwi. 2009.
- [35] U. Tellioglu, G. G. Xie, J. P. Rohrer, i C. Prince, „Whale of a crowd: Quantifying the effectiveness of crowd-sourced serious games”, w *2014 Computer Games: AI Animation Mobile, Multimedia, Educational and Serious Games (CGAMES)*, 2014, s. 1–7.
- [36] M. Eigen i P. Schuster, „The Hypercycle”, *Naturwissenschaften*, t. 65, nr 1, s. 7–41, sty. 1978.
- [37] M. Eigen, „Selforganization of matter and the evolution of biological macromolecules”, *Naturwissenschaften*, t. 58, nr 10, s. 465–523, paź. 1971.
- [38] M. Eigen i P. Schuster, „A principle of natural self-organization”, *Naturwissenschaften*, t. 64, nr 11, s. 541–565, lis. 1977.
- [39] M. Eigen i P. Schuster, „Stages of emerging life —Five principles of early organization”, *J. Mol. Evol.*, t. 19, nr 1, s. 47–61, grudz. 1982.
- [40] J. Iranzo, A. E. Lobkovsky, Y. I. Wolf, i E. V. Koonin, „Evolutionary dynamics of archaeal and bacterial adaptive immunity systems, CRISPR-Cas, in an explicit ecological context”, *J. Bacteriol.*, s. JB.00412-13, cze. 2013.

Szymon Wąsik
(podpis)