

Mateusz TWARDAWA^{1,2}, Piotr FORMANOWICZ¹

1 - Politechnika Poznańska

2 - Poznańskie Centrum Superkomputerowo-Sieciowe afiliowane przy Instytucie Chemii Bioorganicznej PAN

GRAFY PRZEPLYWU ZŁOŻONOŚCI - DEFINICJA, PRZYKŁADY I MOŻLIWE KIERUNKI ZASTOSOWAŃ*

Streszczenie. Analiza i monitorowanie złożoności układów dynamicznych są bardzo trudne do wykonania. Do dziś brak spójnej definicji złożoności systemów, która pozwalałaby na proste śledzenie zmian złożoności układu. Ta praca wprowadza szereg nowych konceptów (grafy przepływu złożoności, demoniczne krajobrazy, układy konstruujące kontekst), których celem jest wyprowadzenie relatywistycznej definicji złożoności algorytmicznej oraz względnej złożoności problemów. W celu zbadania zaproponowanych metod został opracowany i zaimplementowany model układu konstruującego kontekst oparty o automaty komórkowe, dla którego wyznaczono demoniczny krajobraz oraz graf przepływu złożoności. Analiza stworzonego grafu przepływu złożoności pozwoliła na zbadanie dynamiki zmian układu zależnych od złożoności algorytmicznej reguł i przełożenie jej na model ewolucji bakterii. Dodatkowo, opisano kilka potencjalnych aplikacji grafów przepływu złożoności.

COMPLEXITY FLOW GRAPHS - DEFINITION, EXAMPLES AND POSSIBLE APPLICATIONS

Summary. The analysis and monitoring of the complexity of dynamic systems are challenging tasks. To date, there is no cohesive definition of system complexity that allows for tracking changes in a system's complexity. This work introduces several new concepts, i.e., complexity flow graphs, demonic landscapes, and context-constructing systems, aimed at deriving a relativistic definition of algorithmic complexity and the relative complexity of problems. To investigate the proposed methods, a context-constructing system model based on cellular automata was developed and implemented. For this model, a demonic landscape and a complexity flow graph were determined. The analysis of the created complexity flow graph enabled the examination of the dynamics of system changes dependent on the algorithmic complexity of rules, translating them into a model of bacterial evolution. Additionally, several potential applications of complexity flow graphs are described.

* Badania częściowo sfinansowane ze środków pochodzących z programu "Młoda Kadra" przyznanego i realizowanego na Wydziale Informatyki i Telekomunikacji Politechniki Poznańskiej w latach 2022-2024, tytuł projektu: "Analiza złożoności algorytmicznej w modelu układu konstruującego kontekst opartego o automaty komórkowe za pomocą metod algorytmicznej dynamiki informacji".

1. Wprowadzenie

Przetwarzanie informacji jest kluczową zdolnością niemal wszystkich złożonych systemów zdolnych do adaptacji, jak np. komórki, mózgi lub społeczeństwa. Stąd wspólnym mianownikiem wszystkich systemów jest zdolność do przekształcania sygnałów otrzymanych na wejściu w odpowiedzi i reakcje układu, które mają charakter adaptacyjny względem środowiska. W tej pracy algorytmiczna złożoność systemu będzie względną miarą określającą zdolność systemu do poprawnego przekształcenia wejścia na poprawne wyjście (odpowiedź). Poprawność odpowiedzi układu jest ewaluowana przez środowisko, a dokładniej kontekst wynikający z problemu, jaki rozwiązuje i do którego stara się zaadaptować dany system. W tej pracy rozważane są jednocześnie dwa konteksty dla analizowanych układów, kontekst ewolucyjny, gdzie system jest rozumiany jako populacja, a problem jako nisza ewolucyjna oraz kontekst obliczeniowy gdzie system to przybliżony algorytm, starający się rozwiązać problem. Aby uściślić ewolucyjno-obliczeniową analogię, należy założyć, że algorytmy podlegają procesowi optymalizacji, podobnie jak ma to miejsce podczas ewolucji biologicznej dla populacji, przy czym sam schemat optymalizacji jest dowolny i nie musi odzwierciedlać ewolucji. Oznacza to też, że zawartość informacji i złożoność samego problemu stanowią górną granicę pojemności informacyjnej systemu, a system o maksymalnej poprawności ma najwyższą możliwą względną złożoność (zakładając, że jest tzw. "czarną skrzynką", bez wnikania w jego strukturę wewnętrzną). Innymi słowy definicja złożoności algorytmicznej systemu opiera się na zdolności układu do właściwego przetwarzania informacji, jednak wymaga ona dodatkowo kontekstu pozwalającego określić poprawność, na przykład w postaci definicji problemu dla algorytmu lub modelu niszy ewolucyjnej dla populacji organizmów.

Badanie złożoności algorytmicznej jest trudnym zadaniem, które było i ciągle jest podejmowane przez wielu badaczy. Zapoczątkowana przez Raya Solomonoffa, Andrieja Kołmogorowa oraz Gregorego Chaitina algorytmiczna teoria informacji dostarcza metody do badań złożoności problemów obliczeniowych i algorytmów, a nawet fizycznych i biologicznych układów [7]. Rozszerzeniem tej teorii jest algorytmiczna dynamika informacji, która stanowi zestaw metod do analizy złożoności algorytmicznej w układach zmieniających się w czasie, np. automatach komórkowych [23]. Wśród metod należących do algorytmicznej dynamiki informacji znajdują się techniki takie jak detekcja przyczynowości, metoda dekompozycji bloków, algorytmiczna analiza interwencyjna, czy rachunek informacji algorytmicznej [23] [22]. Metody te starają się oszacować obserwowany wzorzec pod kątem złożoności algorytmicznej, badając jego własności, np. kompresowalność w oparciu o pewne strategie, które są powiązane z koniecznością uwzględniania zmienności wzorca w czasie. Automaty komórkowe stanowią podstawowy przedmiot analiz metodami algorytmicznej dynamiki informacji z uwagi na to, że mogą symulować układy dynamiczne i są uniwersalnym modelem obliczeń [23] [8].

Systemy konstruujące kontekst, czyli takie, w których posiadania oraz potencjalna złożoność mogą ulegać zmianom, nie funkcjonują w literaturze, a niniejsza praca po raz pierwszy je definiuje. Brak formalnej definicji nie oznacza jednak, że analizowane problemy i procesy nie były wcześniej podejmowane. Sama idea systemów konstruujących kontekst została zaczerpnięta z ekologii ewolucyjnej i jest generalizacją dla procesu konstrukcji niszy [16][13]. Sprzężenie między populacją a jej środowiskiem, które spr-

wia, że populacja sama pośrednio kształtuje swoje presje selekcyjne modyfikując własną niszę, zwróciło uwagę fizyków i innych badaczy zajmujących się problemem złożoności i teorią informacji [11][5][15]. Jedną z kluczowych konkluzji, jaka została zaproponowana mówi o tym, że populacje w trakcie ewolucji mogą (ale nie muszą) zmieniać swoją złożoność biologiczną (w sensie tożsamym ze złożonością algorytmiczną Kołmogorowa), pod warunkiem że ich złożoność jest nie większa niż złożoność presji selekcyjnych, jakie ją kształtują [11]. Stąd możliwość modyfikacji własnych presji selekcyjnych w ramach procesu konstrukcji niszy przekłada się na zmiany pojemności informacyjnej niszy [11]. Z kolei pojemność informacyjną niszy można rozumieć jako minimalną złożoność algorytmiczną populacji o maksymalnym względnym dostosowaniu do warunków i wymagań określonych przez niszę, analogicznie w przypadku problemu taką pojemność wyznaczałby najprostszy (najmniej złożony) algorytm, który w najlepszy możliwy do uzyskania sposób rozwiązuje problem.

W niniejszej pracy zakłada się, że mierzenie złożoności algorytmicznej systemu jest wykonalne jedynie względem problemu lub zbioru problemów, a systemy nie mogą być oszacowane i porównywane pod kątem złożoności bez odpowiednio zdefiniowanego kontekstu. Systemy rozumiane są tu jako przybliżone algorytmy, które mogą z różną dokładnością rozwiązywać wiele problemów lub jako populacje, które mogą radzić sobie lepiej lub gorzej w zależności od warunków środowiskowych określanych przez poszczególne nisze. Podobnie jest w przypadku złożoności problemu, którą można wyrazić poprzez wcześniej wspomnianą pojemność informacyjną. Określenie pojemności informacyjnej problemu może być wykonane jedynie w oparciu o referencyjny zbiór systemów (tj. algorytmów lub populacji). Celem tej pracy jest prezentacja i definicja dwóch nowych konceptów, tj. demonicznych krajobrazów, które pozwalają na określenie względnej złożoności systemów i problemów, a także powstałych na ich podstawie grafów przepływu złożoności. Grafy przepływu złożoności przedstawiają wzajemne relacje między złożonością algorytmiczną systemu a złożonością powiązanego problemu. W analizowanych układach złożoność nie jest stała. Potencjalna złożoność systemu jest określana przez problem, a system na drodze adaptacji dostosowuje swoją złożoność do aktualnego problemu (tak jakby problem stanowił naczynię, które napełnia złożoność algorytmiczna systemu). Z uwagi na to, że problem sam może ulegać modyfikacji, przyszłe stany systemu różnią się pod kątem zarówno obserwowanej jak i potencjalnej złożoności, a śledzenie tych zmian można określić mianem analizy przepływu złożoności. Dodatkowo w pracy przedstawiono przykład grafu przepływu złożoności, który został otrzymany dla symulacji modelu procesu konstrukcji niszy przez populację bakterii. Omówione zostały także wybrane własności oraz praktyczne zastosowania grafów przepływu złożoności.

2. Złożoność algorytmiczna i pojemność informacyjna w procesach kontekstualnie statycznych

2.1. Uproszczony model ewolucji

Zarówno podczas generowania algorytmów za pomocą metod optymalizacyjnych, jak i w trakcie ewolucji biologicznej częstym założeniem jest niezmiennosc kryteriów oceny rozwiązań, bądź odpowiednio krajobrazu adaptacyjnego. W takich przypadkach kontekst optymalizacji lub ewolucji nie podlega zmianom, stąd można powie-

dzieć, że cały proces jest kontekstualnie statyczny. Słowo “kontekst” jest tutaj celowo użyte, ponieważ termin ten może odnosić się do różnorodnych procesów. Jako jeden z nich można wyróżnić proces trenowania modeli uczenia maszynowego, np. głębokich sieci neuronowych. W trakcie procesu trenowania model sieci dostosowuje swoją złożoność algorytmiczną (w postaci architektury oraz rozkładu wag w sieci), do problemu (zdefiniowanego np. jako zbiór danych wejściowych z przypisanymi etykietami do klasyfikacji). *De facto*, ten proces można rozumieć jako transfer informacji od problemu zaszytego w zbiorze uczącym do modelu sieci neuronowej i zamodelować go zgodnie z modelem komunikacji Shannona-Weavera [3].

W ramach przykładu zostanie przedstawiony ogólny i bardzo uproszczony sposób, w jaki można ocenić złożoność algorytmiczną systemu w postaci populacji pałeczek *Escherichia coli*. Populacja bakterii może zostać potraktowana jako tzw. „czarna skrzynka”, której złożoność algorytmiczna może zostać oceniona przez dostosowanie do zbioru zdefiniowanych warunków środowiskowych, czyli zdolność do utrzymania się w nich i rozwoju. Traktując niszę jako fragment n -wymiarowej przestrzeni wszystkich warunków środowiskowych, można powiedzieć że złożoność algorytmiczna będzie po prostu szerokością niszy, którą definiuje się w ekologii jako rozpiętość obszaru n -wymiarowej przestrzeni warunków środowiskowych, w których populacja odznacza się nieujemnym współczynnikiem przyrostu populacji [4]. Podczas ewolucji biologicznej nisza będzie ulegać zmianom, które są konsekwencją zarówno zmian środowiska, jak i częstości poszczególnych genotypów w populacji oraz mutacji. Warto zwrócić uwagę na rolę częstości występowania poszczególnych warunków środowiskowych oraz interakcje z innymi populacjami (w tym konkurencję wewnątrz i międzygatunkową), co ma nieoobojętny wpływ na siłę oraz sposób działania doboru naturalnego na populację. Z przyjętej definicji złożoności algorytmicznej populacji wynika, że warunkiem do jej obliczenia jest poprawne zdefiniowanie wybranych referencyjnych warunków środowiska (niektóre z wymiarów określających warunki środowiskowe są nieskończone lub mogą obejmować niepraktyczne zakresy, natomiast sama populacja może znajdować się pod wpływem dryfu genetycznego, co w praktyce zmusza do tworzenia założeń, ograniczeń oraz doboru odpowiednich technik próbkowania i metod statystycznych). W celu wyprowadzenia podstawowej i prostej definicji złożoności algorytmicznej populacji, w tej pracy poszczególne zestawy warunków środowiskowych nie będą rozróżniane pod kątem trudności adaptacji do nich.

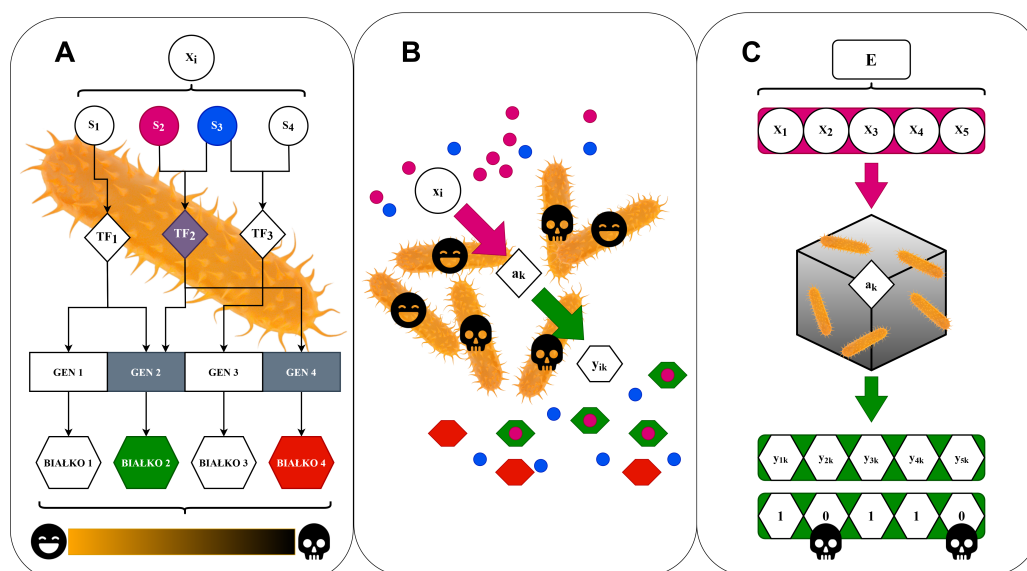
W uproszczonym przykładzie modelu ewolucji populacji bakterii środowisko może być rozumiane jako dyskretna przestrzeń wszystkich możliwych kombinacji substancji i ich stężeń na wejściu. Zakładając, że liczba stanów na wyjściu, tj. możliwych do wygenerowania przez bakterię odpowiedzi jest ograniczona do skończonego podzbioru i dyskretna, można wyznaczyć dostosowanie różnych populacji bakterii do zdefiniowanego i stałego środowiska. Określając zbiór wyników interakcji par wejście-wyjście, tj. warunków środowiska oraz powiązanego z nimi losu populacji bakterii, za pomocą wartości 0, gdy populacja ginie lub 1, gdy przeżywa, otrzymuje się prosty i binarny krajobraz dostosowania. Krajobraz ten ma postać dwuwymiarowej macierzy, w której wartości w jednym wymiarze odpowiadają dostosowaniu konkretnej populacji w różnych warunkach środowiska, natomiast w drugim wymiarze jest zawarta informacja o tym jak wszystkie możliwe populacje radzą sobie w specyficznych warunkach. W oparciu o tę macierz można śledzić ewolucję bakterii, zakładając że najbardziej złożona populacja to

taka, dla której suma wartości dostosowania we wszystkich warunkach środowiskowych jest równa ich liczbie (maksymalna szerokość niszy). Co do zasady ta macierz będzie też mogła nakładać ograniczenia na ewolucję, karząc niektóre ze stanów populacji, a także tworzyć lokalne optima, tak jak to jest w bardziej zaawansowanych modelach [9]. Ponadto, różnica między dostosowaniem dwóch różnych stanów populacji będzie wyrażać potencjalną siłę presji selekcyjnej, która wymusi zmianę stanu populacji, jeśli tylko pojawią się okoliczności pozwalające na zadziałanie doboru naturalnego.

2.2. Względna złożoność algorytmiczna

Zdefiniowany binarny krajobraz dostosowania pozwala określić względną algorytmiczną złożoność populacji bakterii jako szerokość niszy, czyli przeżywalność populacji pałeczek we wszystkich zdefiniowanych warunkach środowiska. Założenie o tym, że wszystkie warunki środowiska są jednakowo trudne oraz binarna ewaluacja dostosowania populacji, powoduje że analizie nie podlega stopień specjalizacji populacji. W rzeczywistości często rozróżnia się dwa ogólne rodzaje gatunków: generalistów, którzy radzą sobie wystarczająco dobrze, aby przeżyć w wielu różnych warunkach środowiska oraz specjalistów, którzy cechują się dużym sukcesem w wąskim zakresie wymagań siedliskowych, okupując to niską tolerancją na warunki środowiska z poza zakresu specjalizacji. Stąd dla tego przykładu maksymalna złożoność to taka, która zapewnia przeżycie bakterii w całym definiowanym zbiorze warunków środowiskowych. Złożoność ta nie może być większa, ponieważ posiadająca ją populacja ma wszystkie kluczowe informacje o warunkach środowiskowych potrzebne do przeżycia populacji. Oznacza to, że każda populacja, która będzie mapować w dokładnie taki sam sposób sygnały na wejściu w zbiory syntetyzowanych białek, nie będzie rozróżnialna pod kątem złożoności algorytmicznej. Dzięki temu unika się odchyień od wartości złożoności algorytmicznej układu, które mogą zostać wprowadzone przez redundantność komponentów systemu, a które nie wnoszą nic do generowanych wyników. Jednak z drugiej strony to podejście wymaga takiego zdefiniowania środowiska, aby uwzględniało w najlepszy sposób jego różnorodność i zmienność. Omawiany przykład zaprezentowano na rysunku 1.

Niech \mathbb{E} będzie nieskończonym zbiorem wszystkich par $e = (x, Y)$, w których skład wchodzi warunki środowiskowe x (instancje problemu) oraz powiązane z nimi zbiory akceptowalnych odpowiedzi populacji Y , przy czym warto zauważyć, że poprawna definicja zbioru Y może wymagać zakodowania kluczowych informacji o stanie populacji jako elementu warunków środowiskowych x . Niech E będzie skończonym podzbiorem \mathbb{E} , który obejmuje wybrane do analizy warunki środowiskowe z jakimi może mieć kontakt populacja bakterii. W analogiczny sposób można zdefiniować nieskończony zbiór wszystkich możliwych populacji \mathbb{A} oraz wybrany do analizy skończony podzbiór populacji A . Dla konkretnej populacji a_k ze zbioru A , można przeprowadzić ewaluację wszystkich warunków środowiskowych zdefiniowanych w zbiorze E . Instancję x_i można powiązać z odpowiedzią systemu $y_{ik} = a_k(x_i)$, tak aby wynik w postaci y_{ik} był równoważny z dostosowaniem a_k w warunkach x_i . Przez dostosowanie w opisanym przykładzie można rozumieć wartość współczynnika wzrostu populacji. Jeśli współczynnik ten wskazuje na nieujemny przyrost populacji, to wartość y_{ik} powinna być równa 1, a gdy współczynnik jest ujemny (zamieranie populacji), to populacja nie jest przystosowana do badanych warunków, a y_{ik} powinien wynosić 0. Analiza tego, jak populacja radzi sobie we wszystkich warunkach środowiskowych (przy uproszczeniu, że zachodzą z taką samą częstością) pozwala określić średnie dostosowanie populacji



Rys. 1. Uproszczony przykład określania złożoności algorytmicznej układu adaptującego się do środowiska. Obraz 1A: Każda z komórek w populacji bakterii a_k ma za zadanie odpowiedzieć na zestaw substancji (s_i), które definiują warunki środowiska x_i poprzez syntezę określonego zestawu białek. Jeśli odpowiedź jest poprawna to komórka przeżyje, a jeśli nie to umrze. Obraz 1B: Warunki środowiskowe x_i , kształtują los całej populacji poprzez uśmiercanie niedostosowanych komórek. Populację odtworzą tylko komórki dostosowane (zdolne do poprawnej odpowiedzi), dzięki czemu populacja przetrwa. Obraz 1C: Sprawdzenie, czy populacja przetrwa w całym zbiorze zdefiniowanych warunków środowiskowych pozwala ocenić jej złożoność algorytmiczną, zakładając że najbardziej złożona populacja to taka, która jest w stanie zaadaptować się do największej liczby warunków środowiska. Każda odpowiedź dla populacji a_k , tj. y_{ik} może przyjąć wartość 0 (populacja wymiera) lub 1 (populacja przeżywa).

do zbioru wybranych warunków środowiska E , co jest jednocześnie złożonością algorytmiczną populacji a_k względem E .

$$\bar{C}_k = \frac{1}{|E|} \sum_{x_i \in E} a_k(x_i) \quad (1)$$

W równaniu (1) średnia względna złożoność algorytmiczna populacji \bar{C}_k , gdzie k oznacza identyfikator populacji, to suma ewaluacji na wszystkich warunkach środowiskowych (instancjach) podzielona przez ich liczbę $|E|$. Przedstawione równanie, zawiera wiele uproszczeń, jak np. wynik ewaluacji algorytmu może mieć wartość 0 lub 1, a instancje nie są w żaden sposób ważone.

2.3. Względna trudność problemu

Podczas ewolucji, gdy zwiększa się średnie dostosowanie, zwiększa się też względna złożoność algorytmiczna populacji bakterii, która adaptuje się do środowiska i jego warunków. Można mówić tutaj o transferze informacji od środowiska do populacji bakterii, który przebiega zgodnie z zasadami doboru naturalnego [11]. Źródłem informacji są warunki środowiska w postaci niszy ewolucyjnej, która stanowi atraktor

ewolucyjny dla populacji, przez co presje selekcyjne wynikające z interakcji populacji ze środowiskiem, których względną złożoność można wyznaczyć, kształtują kolejne stany populacji.

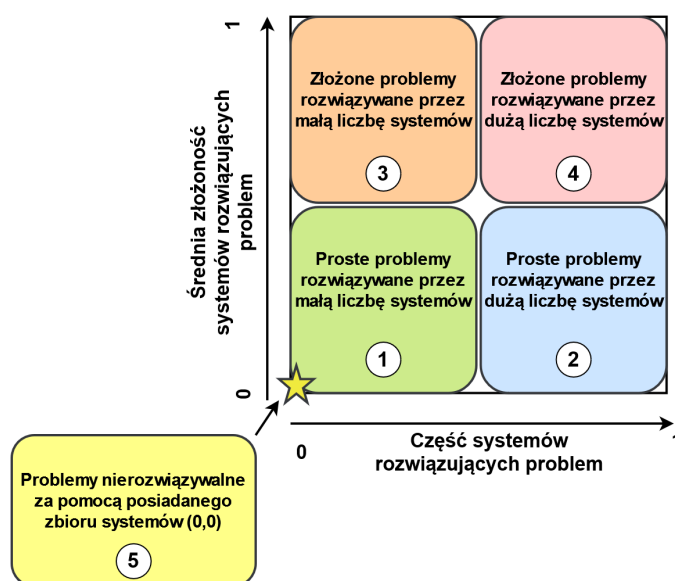
Trudność problemu ewolucyjnego (niszy) definiowanego przez warunki środowiska można określić jako średnią liczbę pokoleń potrzebą do przekształcenia losowo wybranej populacji a_k w populację, która posiada maksymalne względne średnie dostosowanie. Stąd trudność niszy ewolucyjnej i związanych z nią warunków środowiska można rozumieć jako długość średniej, bądź w zależności od potrzeb najkrótszej ustandaryzowanej sekwencji stanów populacji prowadzącej od losowo wybranego stanu do stanu o maksymalnym względnym średnim dostosowaniu. W literaturze można znaleźć wiele prób i propozycji metod służących do określania trudności problemów ewolucyjnych, dobrym przykładem mogą być: [1][12][14]. Warto w tym celu zauważyć, że nie każda populacja będzie miała możliwość go osiągnąć, a sekwencja stanów populacji musi uwzględnić zmiany puli genowej, wliczając w to zmiany częstości nowych i istniejących alleli, zależnych zarówno od doboru naturalnego, jak i od dryfu genetycznego. Taka definicja jest analogiczna do krajobrazów dostosowania, gdzie zwyczajowo przedstawia się względne dostosowanie do środowiska na płaszczyźnie określonej przez parametry warunków środowiskowych. Dyskretna wersja tego konceptu to właśnie graf, co też zostało przedstawione przez Sewall'a Wright'a, uważanego za kluczowego autora tego konceptu [20]. Opisana w tej pracy ewolucyjna trudność niszy ewolucyjnej lub problemu polega na przekształceniu początkowej niedostosowanej populacji w pełni dostosowaną. Wartość ta, policzona jest w oparciu o zbiór populacji A , który z założenia powinien stanowić odpowiednio dobraną i dużą próbę. Warto jednak zaznaczyć, że owa trudność warunków środowiska jest względna i istnieje tylko dla zdefiniowanych przestrzeni populacji. Powyższa relatywistyczna definicja wprost zakłada, że trudność problemu wymaga zdefiniowania przestrzeni systemów i jest własnością, która nie może być wyznaczona samodzielnie, gdyż nie można określić trudności niszy ewolucyjnej i powiązanych z nią konkretnych warunków środowiska bez odniesienia się do jakiejś referencyjnej populacji. W podobny sposób można przeanalizować trudność problemów, które akceptują rozwiązania przybliżone, w oparciu o referencyjny zbiór algorytmów, które je generują.

W praktyce można uprościć te miary i ograniczyć je do analizy wyników ewaluacji zbioru stanów populacji (algorytmów), co pozwoli także na ujęcie innych układów niż ewolucyjne. W przypadku zaproponowanego modelu opracowano w ramach tej pracy nową dedykowaną metodę estymacji polegającą na nanoszeniu problemu na wyznaczoną względną przestrzeń trudności problemu. Przestrzeń jest dwuwymiarowa, gdzie oś rzędnych to wartości od 0 do liczby zdefiniowanych algorytmów $|A|$, a oś odciętych dotyczy liczby wymaganych kroków potrzebnych do spełnienia warunków problemów (uważa się, że proste problemy wymagają mało kroków, a trudne dużo, przy czym nie da się ich określić, jeśli algorytm nie rozwiązuje problemu). Stąd w tej pracy przyjęto dwuwymiarowy system mierzenia trudności problemu:

$$D_i = \left(\frac{|A_i^*|}{|A|}, \frac{1}{|A_i^*|} \cdot \sum_{a_k \in A_i^*} \bar{C}_k \right) \quad (2)$$

W równaniu (2) zdefiniowano względną trudność problemu i (czyli D_i) jako parę dwóch wartości. Pierwszą z nich jest część poprawnie rozwiązujących problem

systemów wyrażona przez wynik dzielenia liczby elementów zbioru $|A_i^*|$ przez zbiór wszystkich zdefiniowanych systemów $|A|$ (zbiór A_i^* jest podzbiorem zbioru systemów A , który składa się wyłącznie z systemów rozwiązujących aktualny problem i w zadowalający sposób). Drugą z nich jest średnia złożoność algorytmiczna układów zdolnych do rozwiązania problemu \bar{C}_k . Dzięki podanej definicji złożoności problemu możliwe jest zdefiniowanie dwuwymiarowej przestrzeni (rys. 2), dla której można określić trudność problemu. Proste problemy będą mogły być rozwiązane przez układy o niskiej złożoności, a trudne przez niewielką liczbę złożonych systemów (rozwiązania nie muszą być dokładne). Dobrą praktyką przy wyliczaniu względnej trudności problemu jest obliczenie jej dla zbioru wielu systemów (np. populacji lub algorytmów).



Rys. 2. Kategorie trudności problemu zmapowane na dwuwymiarową przestrzeń opartą o średnią złożoność algorytmiczną oraz część poprawnie rozwiązujących problem systemów. Można wyróżnić pięć podstawowych kategorii trudności problemów względem posiadanego zbioru systemów (1: proste problemy, które można rozwiązać małą liczbą systemów, 2: proste problemy, które są rozwiązywane przez dużą część systemów, 3: złożone problemy, dla których istnieje niewielka liczba układów zdolnych do ich rozwiązania, 4: złożone problemy, które są rozwiązywane przez dużą część systemów, 5: Problemy nierozwiązywalne za pomocą posiadanego zbioru systemów).

3. Procesy kontekstualnie dynamiczne

3.1. Dynamika i konstrukcja kontekstu

W rzeczywistości warunki środowiska, które współtworzą kontekst dla populacji bakterii (czyli takie, w których zachodzi proces ewolucji biologicznej) nie są statyczne. Jest tak, ponieważ podzbiór dopuszczalnych (prawdopodobnych) warunków może się

zmieniać. Na przykład, populacja bakterii może zwiększać swój zasięg o wcześniej niedostępne warunki środowiskowe lub one same mogą ulec zmianie na skutek zewnętrznych procesów, co w konsekwencji prowadzi do zmian presji selekcyjnych. W takiej sytuacji układ będzie kontekstualnie dynamiczny, co oznacza, że problem w nim zawarty będzie mógł się zmieniać, co w konsekwencji będzie wymuszać też zmiany w systemie.

Dodatkowo, zmiany środowiskowe mogą być sprzężone ze zmianami w puli genowej populacji, co wprowadza szczególny rodzaj zależności nazywany konstrukcją niszy. Środowisko wpływa na pulę genową populacji poprzez siły doboru naturalnego, co sprawia, że jest ona bardziej dostosowana do niego. Jednak dodatkowo populacja może w trwały sposób modyfikować środowisko, odkształcając presje selekcyjne, które działają na nią samą, tworząc tym samym sprzężenie. Innymi słowy, populacja zmienia własny kontekst środowiskowy, który decyduje o dostosowaniu osobników w populacji. System z takim sprzężeniem nazwiemy układem konstruującym kontekst.

Przekładając te pojęcia na odpowiedniki w naukach obliczeniowych można operować na parze problem-algorytm. Problem, który jest odpowiednikiem warunków środowiska definiuje kryteria dla algorytmu. Algorytm natomiast wykonuje swoje zadania starając się najlepiej spełnić zadane kryteria, jednocześnie będąc przedmiotem optymalizacji (w takim samym rozumieniu, w jakim populacja bakterii jest przedmiotem ewolucji). Jednak kryteria i co za tym idzie definicja samego problemu, zmieniają się w czasie, a zmiany mogą być uzależnione od wersji aktualnego algorytmu (definicja problemu może zawierać informacje związane lub zależne od wersji programu), który jest dostosowywany dynamicznie zmieniającego się problemu. W ten sposób powstaje sprzężenie analogiczne do tego, które zostało opisane dla procesu konstrukcji niszy. Modyfikacja problemu może zmienić jego względną trudność. Populacje zdolne do trwałej modyfikacji warunków środowiskowych, mogą zostać za modelowane przez zmiany w definicji problemu spowodowane przez algorytm, a sposób wprowadzenia zmian będzie określać specjalna funkcja odkształcająca. Stąd, jedynie poprawnie zdefiniowana para problem-algorytm jest w stanie określić kontekst w danym momencie w czasie, a obserwując przebieg symulacji, można postarać się o analizę skali i tempa zmian problemu, algorytmu i kontekstu. Dodatkową konsekwencją jest też to, że dla tego samego problemu, docelowy kontekst może być odmienny, jeśli algorytmy na początku symulacji będą różne, co jest wynikiem (wrażliwości na warunki początkowe).

Procesy, w których zachodzi zjawisko konstrukcji kontekstu, mimo iż wydają się na pierwszy rzut oka dość abstrakcyjne, posiadają swoje odpowiedniki w procesach biologicznych, fizycznych oraz mogą być powiązane z problemami technicznymi i inżynierskimi. Ponadto można też je wyrazić przez metody znane z optymalizacji dynamicznej. Przykładami takich procesów mogą być:

1. ewolucja somatyczna nowotworu, gdzie populacja komórek nowotworowych adaptuje się do silnie zmienionego przez siebie mikrośrodowiska;
2. problemy szeregowania zadań w czasie rzeczywistym, w których kryteria uszeregowania zależą od aktualnego uszeregowania oraz samego algorytmu szeregującego;
3. dynamika gry rozgrywanej się i determinowanej przez dynamiczne środowisko, na które wpływają aktualnie stosowane strategie graczy.

3.2. Model ewolucji z konstrukcją niszy oparty o automaty komórkowe

Automaty komórkowe zostały pierwotnie opracowane przez Stanisława Ulama oraz Johna von Neumanna na pod koniec pierwszej połowy XX w. [17]. Ten dyskretny model obliczeń umożliwia zaawansowane symulacje wielu procesów naturalnych i przemysłowych. W latach 90-tych Matthew Cook wykazał, że automaty komórkowe są uniwersalne obliczeniowo (zupełne w sensie Turinga) [6].

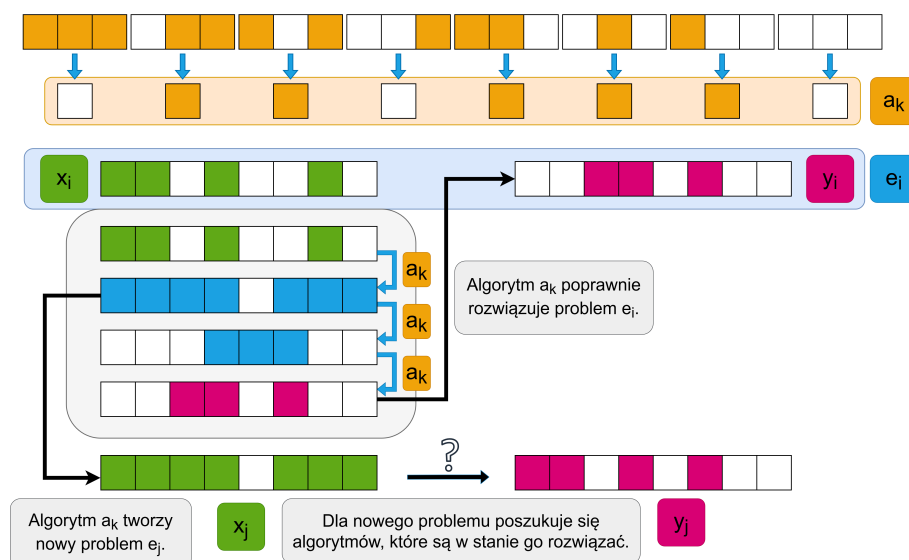
Jednowymiarowe automaty komórkowe stanowią dogodny sposób modelowania wybranych układów złożonych i zostały wykorzystane do zamodelowania procesu ewolucji bakterii *E. coli*, który został opisany w tej pracy. Niech 8-bitowa sekwencja x_i oznacza sygnały środowiskowe otrzymywane na wejściu przez populację bakterii a_k , a y_{ik} będzie oznaczać odpowiedź populacji na otrzymane wejście ($y_{ik} = a_k(x_i)$). Zakładając, że dla każdej 8-bitowej sekwencji x_i istnieje dokładnie jedna poprawna odpowiedź, tj. 8-bitowa sekwencja y_i (lista par sekwencji wyjściowych i poprawnych odpowiedzi (x_i, y_i) może być wygenerowaną wcześniej losowo bijekcją, która nie ulega zmianom podczas samej symulacji), to los populacji bakterii a_k (tj. reguł automatu kodowanych przez 8-bitową sekwencję) będzie uzależniony od tego, czy jest w stanie przekształcić sekwencję x_i w sekwencję y_i w skończonej liczbie kroków. Stąd, każda para wartości (x_i, y_i) stanowi problem e_i do rozwiązania ze zbioru wszystkich zdefiniowanych problemów E . Jeśli przekształcenie x_i na y_i nie jest osiągalne przy zdefiniowanej maksymalnej liczbie kroków to populacja ginie, co jest równoważne ze stwierdzeniem, że algorytm a_k nie rozwiązuje problemu e_i . Wszystkie reguły ze zbioru algorytmów A , zdolne do przeprowadzenia żądanej transformacji stanowią podzbiór A_i^* . Analogicznie można zdefiniować podzbiór A_i^- , w którym będą znajdować się algorytmy nie rozwiązujące problemu e_i . Warto zauważyć, że suma podzbiorów algorytmów rozwiązujących i nierozwiązujących problem e_i musi być taka sama jak zbiór algorytmów referencyjnych: $A_i = A_i^- \cup A_i^*$. Jest tak dzięki wprowadzeniu do definicji skończonej liczby kroków, która adresuje problem stopu. Opisany proces jest uproszczonym modelem ewolucji populacji bakterii (reguł automatu) do zdefiniowanych warunków środowiskowych, czyli jest to deterministyczny proces statyczny kontekstualnie dążący do stanu równowagi ewolucyjnej (określonych w zbiorze A_i^*).

Sprzężenie między populacją a środowiskiem wynikające z procesu konstrukcji niszy można wprowadzić do modelu poprzez modyfikacje warunków środowiska e_i za pomocą wybranej reguły ze zbioru A_i^* . Istnieje wiele sposobów aby to zrobić, w tej pracy zdecydowano się na pojedyncze wykonanie wybranej reguły na sekwencji x_i , tworząc w ten sposób nową parę (x_j, y_j) . Dzięki temu zabiegowi pojawią się nowe warunki środowiskowe, określające nowy problem e_j , a proces ewaluacji zbioru algorytmów (populacji bakterii), będzie trzeba wykonać na nowo. Jeśli zbiór A_i^* zawierał więcej niż jeden algorytm, to należy rozważyć osobno każdą z możliwości transformacji. Przykład układu opartego na automatach komórkowych zaprezentowano na rysunku 3.

3.3. Demoniczny krajobraz

W przypadku analizy układów ewolucyjnych, ewaluacja różnych populacji w przestrzeni warunków środowiskowych pozwala na zamodelowanie krajobrazu adaptacyjnego. W tym krajobrazie każda populacja będzie miała zdefiniowaną dla siebie wartość średniego dostosowania. Jeśli problem w postaci warunków środowiskowych się zmienia, wtedy krajobraz dostosowania również będzie dynamiczny.

W celu dokonania generalizacji różnych krajobrazów, ta praca wprowadza no-

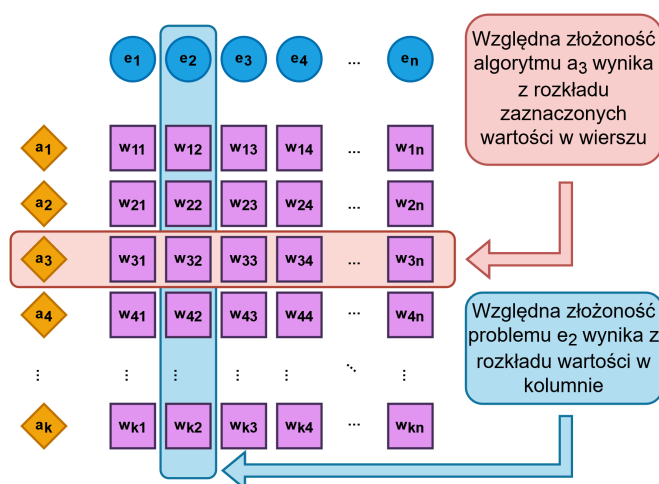


Rys. 3. Ilustracja działania układu konstruującego kontekst opartego na minimalistycznym modelu bazującym na jednowymiarowych automatach komórkowych. Algorytm a_k jest reprezentowany przez 8-bitową sekwencję, która jednoznacznie określa zasady transformacji sekwencji wejściowych na wyjściowe. Każda komórka automatu jest przekształcana w oparciu o nią samą oraz prawego i lewego sąsiada (lewym sąsiadem pierwszej komórki jest ostania, a prawym ostatniej pierwszej komórki). Problem e_i określa para sekwencji (x_i, y_i) , gdzie x_i to 8-bitowa sekwencja na wejściu, która reguła a_k ma za zadanie przekształcić w ściśle określoną 8-bitową sekwencję y_i wykonując się nie więcej niż 50 razy (sprawdzone są wszystkie reguły ze zbioru A , który liczy 256 elementów). Gdy tak się stanie nowy problem jest wyprowadzany na podstawie wyniku jednokrotnego wykonania a_k na x_i , tworząc nową parę (x_j, y_j) . W zaproponowanym modelu, funkcja mapująca $m(x) = y$ jest niezmienną, wcześniej losową wyznaczoną bijekcją.

wy rodzaj przestrzeni stanów tj. demoniczny krajobraz. Demoniczny krajobraz dotyczy wyników ewaluacji par system-problem, a jego nazwa wywodzi się bezpośrednio od znanego i współcześnie rozwiązanego paradoksu: demona Maxwella, tworząc analogię między demonem a algorytmem. Nawiązanie do demona Maxwella, zostało także użyte przez Davida C. Krakauera, który zaproponował tzw. zasadę demonicznego doboru. Reguła ta mówi, że złożoność populacji nie może przewyższać złożoności presji selekcyjnych, które na nie działają [11].

Demoniczny krajobraz pokazuje przestrzeń stanów dla zestawień system-problem, skupiając się na względnej kontekstualnie zawartości informacyjnej. Poprawnie dopasowana para system-problem charakteryzuje się wysokim stopień dopasowania systemu do problemu. Stąd, jeśli wykonamy ewaluacje zbioru systemów na zbiorze wielu problemów, otrzymując dla każdej pary wartość estymatora poprawności (dostosowanie), to wyznaczanie względnej złożoności problemu oraz względnej złożoności algorytmicznej stanie się możliwe. Jako estymator względnej złożoności problemu w tej pracy wykorzystano wczesniej zaproponowaną miarę względnej trudności problemu, jednak należy pamiętać, że trudność problemu nie jest tożsama ze złożonością. W przypad-

ku problemów ewolucyjnych, względna złożoność mogłaby być miarą charakteryzującą topologię krajobrazu dostosowania w oparciu o referencyjny zbiór stanów populacji. Wspomniane miary złożoności można wyznaczyć zgodnie ze wcześniej podanymi definicjami (równania 1 i 2). Warto zauważyć, że pełna informacja na temat opisywanych względnych złożoności problemu i systemu jest zawarta w rozkładach w krajobrazie demonicznym. Dla analizowanego w tej pracy modelu opartego na automatach komórkowych krajobraz ten otrzymamy dla wybranego problemu zestawionego ze wszystkimi algorytmami w postaci reguł automatu (względna złożoność problemu) oraz wybranego algorytmu przetestowanego dla wszystkich zdefiniowanych problemów (względna złożoność algorytmu), co pokazano na rysunku 4. Ponadto, w oparciu o te rozkłady na podstawie demonicznego krajobrazu można obliczyć, czy tranzycja z jednego problemu w inny (lub jednego algorytmu w inny) jest prawdopodobna (zakładając, że układ maksymalizuje względną poprawność tj. dopasowanie systemu do problemu).



Rys. 4. Uproszczony schemat demonicznego krajobrazu w postaci dwuwymiarowej macierzy zawierającej wyniki ewaluacji wszystkich zdefiniowanych algorytmów na wszystkich zdefiniowanych problemach (sposób ewaluacji może być specyficzny dla wybranej domeny badań). Na podstawie demonicznego krajobrazu można określić złożoność każdego algorytmu, którą jest rozkład wyników ewaluacji wybranego algorytmu we wszystkich problemach. Analogicznie złożoność problemu jest określona tutaj na podstawie rozkładu wyników ewaluacji wszystkich algorytmów. W przypadku analizy rzeczywistych układów, demoniczny krajobraz może zostać oszacowany na podstawie odpowiednio dobranej próby.

4. Grafy przepływu złożoności

Grafy przepływu złożoności są strukturą, której celem jest przedstawienie zmian złożoności algorytmicznej systemu względem zmian złożoności kryteriów problemu, do których może dochodzić w trakcie procesów kontekstualnie dynamicznych. W szczególności dotyczy to procesów, w których zmiany wywoływane przez system oraz problem zachodzą w różnych skalach czasowych (np. adaptacja populacji bakterii do nowego środowiska zachodzi w czasie ewolucyjnym, a transformacja środowiska przez populację jest zjawiskiem zachodzącym w skali ekologicznej). Poniżej opisany graf dotyczy

procesu konstruującego kontekst, czyli takiego, w którym system dostosowywany jest do tego aby rozwiązywać problem, a problem może być modyfikowany przez system, który stara się zaspokoić określone przez niego kryteria, co opisano w sekcji 3.2.

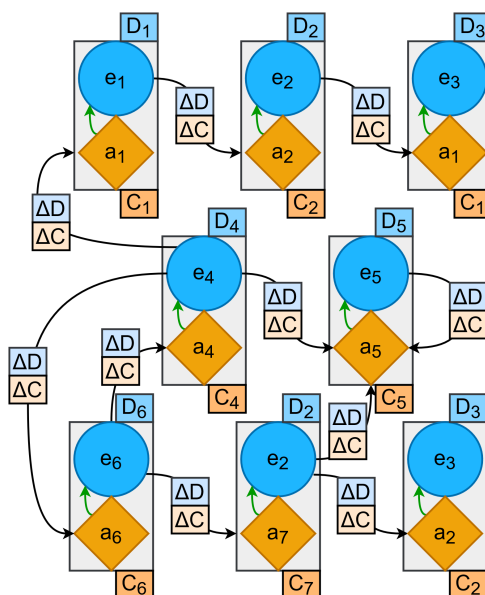
Wierzchołek v_i w grafie przepływu złożoności (ΔG) reprezentuje parę złożoną z: systemu w stanie początkowym oraz problemu do rozwiązania (a_i, e_i) . System a_i rozwiązuje problem określony w wierzchołkach dla których analizowany wierzchołek v_i jest incydentny i przyczynił się do sformułowania bieżącego problemu (np. modyfikując go), lecz sam go nie rozwiązuje. Wierzchołek v_i posiada łuk do innego wierzchołka v_j , wtedy i tylko wtedy, gdy problem związany z wierzchołkiem v_i tj. e_i jest rozwiązywany przez system zdefiniowany w wierzchołku v_j tj. a_j . Wierzchołki w grafie przepływu złożoności są ważone przez względną złożoność algorytmiczną systemu doprowadzającego C_i oraz względną złożoność problemów D_i . Obie wagi określające złożoność są wyznaczane w oparciu o wcześniej zdefiniowany demoniczny krajobraz. Dodatkowo przypisane wierzchołkom wagi pozwalają na wyliczenie różnic między wagami w wierzchołkach i naniesienie ich jako wagi na łukach grafu (stan obecny minus stan przeszły: ΔC_{ij} oraz ΔD_{ij}). Oznacza to, że analiza gradientów oraz zmian złożoności jest możliwa do przeprowadzania w oparciu o informacje zawarte w grafie przepływu złożoności. Schemat grafu przepływu złożoności opracowanego dla analizowanego modelu opartego na automatach komórkowych zaprezentowano na rysunku 5.

Dodatkowo, do stworzonego w ramach tej pracy grafu przepływu złożoności wprowadzono dwa rodzaje wag dla łuków, które są specyficzne wyłącznie dla analizowanego modelu. Pierwszą z tych wag stanowi liczba kroków potrzebnych do przekształcenia sekwencji na wejściu w oczekiwaną sekwencję na wyjściu. Waga ta w naturalny sposób łączy ze sobą problem zdefiniowany w wierzchołku, z którego wychodzi łuk do wierzchołka zawierającego poprawnie rozwiązujący go algorytm. Drugą wagą jest zmiana średnich długości liczby kroków wymaganych przy wspomnianej transformacji. Tą wagę można otrzymać wyliczając dla każdego wierzchołka średnią długość liczby kroków, następnie wyliczając ją jako różnicę średniej liczby kroków między wierzchołkiem do którego prowadzi łuk i wierzchołkiem, z którego on wychodzi.

5. Symulacje modelu konstruującego kontekst oraz wyniki

5.1. Implementacja modelu

W celu wygenerowania grafu przepływu złożoności zaimplementowano prosty model procesu konstrukcji kontekstu oparciu o automaty komórkowe. Zasady działania modelu zostały opisane wcześniej w tej pracy (podrozdział 3.2). Model zaimplementowano w języku Python 3.10 na 8 bitowych sekwencjach. Każdy problem definiowany jest przez parę dwóch sekwencji początkowej i oczekiwanej, przyjmujące rolę systemów algorytmu to także 8-bitowe sekwencje, które reprezentują reguły jednowymiarowego automatu komórkowego (zgodnie modelem przedstawionym na rysunku 3). Mapowanie 256 sekwencji wejściowych problemu na 256 oczekiwanych zostało wykonane losowo, jako bijekcja. Zadaniem algorytmu jest bezbłędnie odtworzenie oczekiwanej sekwencji na podstawie początkowej sekwencji w maksymalnie 50 krokach. Na potrzeby późniejszego tworzenia grafu skanowana jest cała przestrzeń 256 zdefiniowanych reguł automatu.



Rys. 5. Schemat grafu przepływu złożoności zbudowanego dla analizowanego modelu procesu konstruującego kontekst. Każdy wierzchołek składa się z pary system-problem (a_k, e_i) , łukiem połączone są wierzchołki, których problem jest rozwiązywany przez system do którego prowadzi łuk. Dodatkowo, każdy wierzchołek ma określoną złożoność problemu i systemu, a każdy łuk jest ważony zmianą w obu tych złożonościach. Zielona strzałka reprezentuje zmianę problemu przez algorytm, która zachodzi w krótszej skali czasowej.

5.2. Własności otrzymanego demonicznego krajobrazu

Na podstawie przeprowadzonych analiz wyznaczono demoniczny krajobraz, który pozwolił obliczyć złożoność algorytmiczną reguł automatu zgodnie ze wzorem 1 oraz złożoność problemu, która została uproszczona do względnej trudności zdefiniowanej w równaniu 2. Zgodnie ze wspomnianymi równaniami, wybraną miarą względnej złożoności algorytmicznej jest liczba problemów poprawnie rozwiązywanych przez regułę, podzielona przez wszystkie problemy, czyli 65 536. Natomiast względna trudność problemu to para wartości złożona z liczby algorytmów rozwiązujących poprawnie problem podzielona przez liczbę wszystkich algorytmów pierwsza wartość oraz uśredniona złożoność algorytmiczna rozwiązujących problem reguł automatu (druga wartość).

Ukazany na rysunku 6 krajobraz pozwala na wyznaczenie względnych złożoności problemu i algorytmu. Ze wszystkich 256 algorytmów jedynie 15 z nich nie było w stanie rozwiązać żadnego z problemów. Każdy algorytm rozwiązywał średnio $7,35(\pm 7,0218)$ problemu, przy czym duże odchylenie pokazuje na istniejącą dysproporcję w potencjale algorytmów. Cztery reguły wyróżniły się bardzo wysoką skutecznością w rozwiązywaniu problemów względem innych tj. 106 (01101010), 120 (01111000), 169 (10101001) oraz 225 (11100001). Co ciekawe, zgodnie z klasyfikacją i własnościami, reguły 106, 120, 169 oraz 225 są dla siebie ekwiwalentną czwórką [19]. To znaczy, że np. odwrócenie zer i jedynek dla reguły 106 sprawi, że otrzymamy regułę 169, która będzie działała tak samo jak 106, ale na odwróconych kolorach. Odbiciem reguły 106 jest reguła 120. Oznacza jeśli zamienimy prawego i lewego sąsiada dla centralnej komórki w trójce, która jest wykorzystywana do wyliczenia kolejnej komórki w regule

Tabela 1

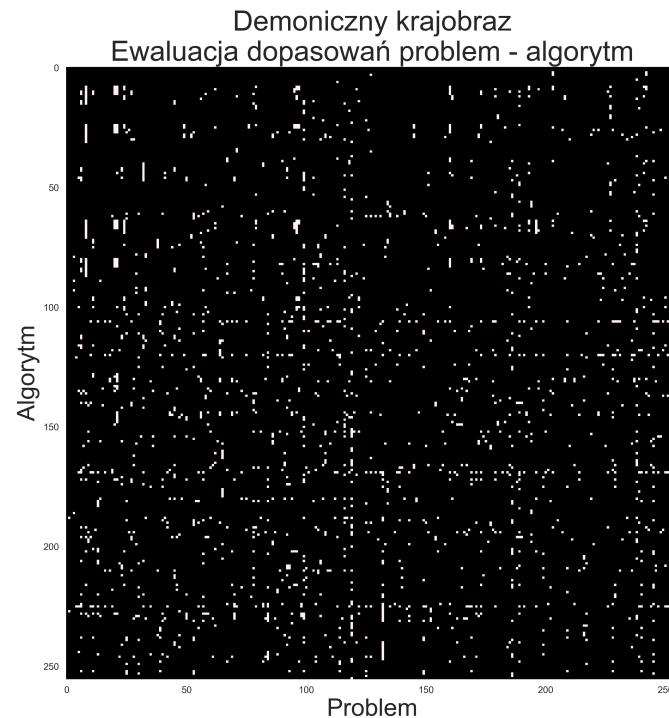
Wygenerowana losowo bijekcja parująca sekwencje wejściowe x na oczekiwane y .

x_i	y_i	x_i	y_i	x_i	y_i	x_i	y_i	x_i	y_i	x_i	y_i	x_i	y_i	x_i	y_i
0	234	32	3	64	193	96	8	128	181	160	32	192	28	224	145
1	79	33	109	65	93	97	16	129	118	161	130	193	2	225	166
2	214	34	117	66	166	98	142	130	90	162	59	194	170	226	48
3	171	35	55	67	193	99	99	131	179	163	46	195	27	227	2
4	110	36	202	68	23	100	78	132	189	164	52	196	66	228	14
5	48	37	120	69	112	101	236	133	87	165	78	197	160	229	219
6	192	38	105	70	216	102	233	134	180	166	226	198	21	230	38
7	177	39	249	71	65	103	198	135	230	167	139	199	249	231	18
8	1	40	127	72	30	104	14	136	79	168	185	200	52	232	73
9	146	41	54	73	103	105	98	137	194	169	24	201	60	233	182
10	72	42	143	74	87	106	218	138	118	170	132	202	188	234	167
11	96	43	160	75	38	107	20	139	237	171	62	203	64	235	197
12	147	44	100	76	128	108	85	140	156	172	90	204	240	236	173
13	111	45	180	77	171	109	19	141	238	173	33	205	254	237	28
14	83	46	114	78	156	110	234	142	150	174	101	206	166	238	85
15	172	47	105	79	8	111	125	143	237	175	53	207	22	239	143
16	18	48	100	80	187	112	142	144	83	176	101	208	10	240	31
17	33	49	168	81	58	113	144	145	73	177	253	209	207	241	128
18	131	50	54	82	16	114	207	146	98	178	196	210	53	242	16
19	124	51	91	83	23	115	94	147	163	179	66	211	73	243	140
20	128	52	162	84	253	116	116	148	2	180	104	212	5	244	101
21	64	53	191	85	111	117	79	149	254	181	147	213	165	245	253
22	175	54	188	86	170	118	6	150	215	182	217	214	118	246	98
23	204	55	248	87	46	119	255	151	22	183	17	215	82	247	36
24	128	56	130	88	159	120	161	152	171	184	171	216	39	248	194
25	223	57	237	89	161	121	52	153	13	185	140	217	119	249	76
26	88	58	50	90	14	122	134	154	82	186	239	218	187	250	129
27	76	59	136	91	85	123	158	155	232	187	230	219	85	251	66
28	197	60	142	92	102	124	27	156	225	188	114	220	107	252	40
29	182	61	46	93	159	125	199	157	161	189	33	221	212	253	170
30	247	62	136	94	193	126	176	158	93	190	39	222	14	254	202
31	53	63	90	95	32	127	31	159	157	191	9	223	35	255	9

106, otrzymany regułę 120 (lustrzane odbicie wzorców tworzonych przez regułę 106). Jednoczesna aplikacja operacji odwrócenia barw oraz odbicia dla wzorców reguły 106 wytworzy wzorce zgodne z wynikami działania reguły 225. Wszystkie cztery reguły zostały zaprezentowane na rysunku 7.

Z 256 problemów dla 27 nie znaleziono żadnego algorytmu, który potrafiłby je rozwiązać. 9 problemów było rozwiązanych przez 30 lub więcej algorytmów, ale tylko 3 przez 40 lub więcej. Dla przykładu największa liczba dopasowań algorytmów do jednego problemu wynosi 51, jest to problem transformacji sekwencji 119 (01110111) do sekwencji 255 (11111111). Każdy problem był rozwiązywany średnio przez $7,35(\pm 7,8676)$. Szczegółowe informacje na temat liczby algorytmów rozwiązujących każdy z analizowanych problemów oraz liczby problemów rozwiązywanych przez każdy algorytm zostały zaprezentowane na rysunku 8.

Na rysunku 9 przedstawiono dwa wykresy obrazujące rozkład wyliczonych wartości dla względnych złożoności dla algorytmów (po lewej) oraz dla problemów (po prawej). Histogram złożoności algorytmicznej reguł automatu wyraźnie wskazuje na to, że większość reguł ma relatywnie niską złożoność. Złożoność algorytmiczna spada wraz z liczebnością, pozostawiając pusty obszar po którym znajdują się wcześniej wspomniane cztery ekwiwalentne reguły (106,120,169,225). W przypadku złożoności problemów najwięcej z nich leży w obszarze wykresu, który wcześniej określono jako miejsce prostych problemów rozwiązywanych przez małą liczbę algorytmów (patrz



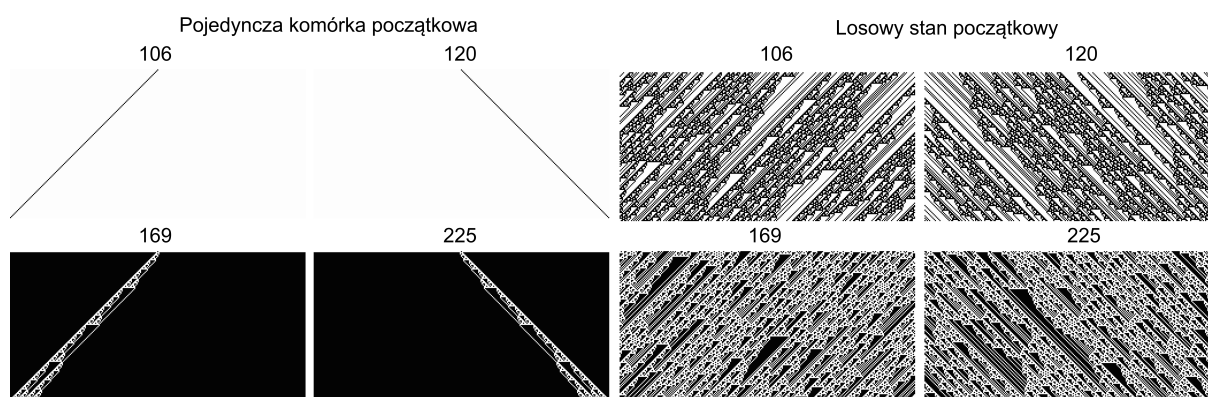
Rys. 6. Uproszczony demoniczny krajobraz przedstawiający dopasowania problem algorytm. Każdy biały kwadracik oznacza, że algorytm reprezentowany w wierszu rozwiązuje problem określony w kolumnie.

Rys 2.). W sektorach poświęconych prostym problemom rozwiązywanym przez wiele algorytmów oraz złożonych problemów rozwiązywanym przez małą liczbą algorytmów znajduje się kilka nielicznych problemów. Nie zaobserwowano natomiast żadnych złożonych problemów rozwiązywanym przez wiele algorytmów, czego można było się spodziewać analizując rozkład złożoności algorytmów.

5.3. Struktura i cechy grafu przepływu złożoności

Na podstawie przeprowadzonych symulacji i analizy wszystkich kombinacji par problem-algorytm dla procesu konstrukcji kontekstu opisanego w sekcji 3.2, stworzono graf przepływu złożoności (rys. 10). Pierwotny graf zawierał 65536 wierzchołków z czego 6679 stanowiły wierzchołki izolowane. Z grafu usunięto wierzchołki, których stopień wejściowy wynosił 0 (w tym izolowane), ponieważ stanowiły one warunki startowe dla symulacji, które często ulegały konwergencji do wspólnego stanu pochodnego przez konieczność dopasowania algorytmu do problemu. Finalnie otrzymany graf (ΔG) był spójny i zawierał 1759 wierzchołków oraz 13333 łuki.

Rysunek 11 przedstawia rozkład trzech wartości: stopni wejściowych, wyjściowych oraz statystyki PageRank. Rozkład stopni wejściowych i wyjściowych odpowiada rozkładowi złożoności algorytmów i problemów (w kolejności), co wynika z definicji grafu, jednak zależność nie jest idealna, ponieważ skasowano część wierzchołków. Wyliczone dla wierzchołków wartości PageRank ($\alpha = 0,95$) pokazują, że istnieje mała grupa wierzchołków, które stanowią małe huby, tj. mają większe prawdopodobieństwo odwie-



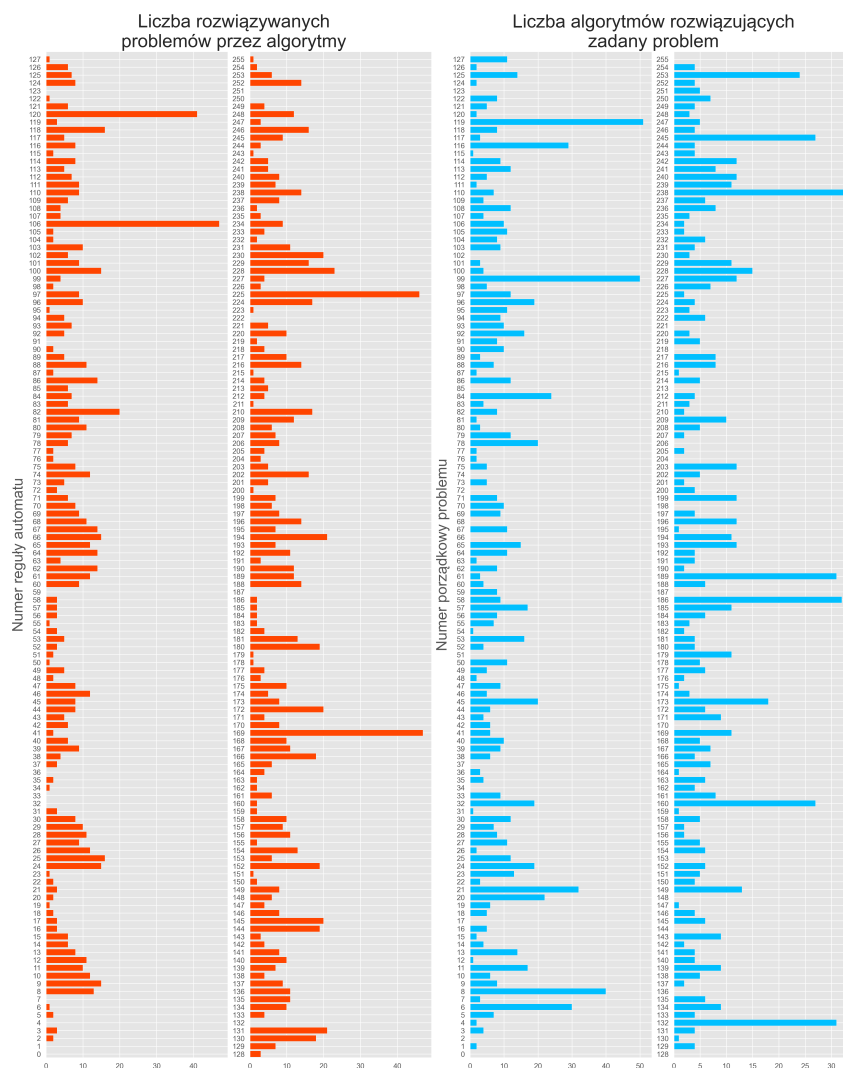
Rys. 7. Przebiegi reguł 106, 120, 169 oraz 225 na wejściu stanowiącej pojedynczą komórkę oraz losowej sekwencji binarnej. Widoczne są relacje opisane w tekście, świadczące o ekwiwalentności reguł. Dodatkowo widoczne są wzory wygenerowane w oparciu o losowe sekwencje na wejściu, które pokazują wrażliwość reguł na warunki początkowe oraz złożoność generowanych wzorców. Wszystkie obrazy wchodzące w skład tej grafiki uzyskano w oparciu o model elementarnych jednowymiarowych automatów komórkowych udostępniony w portalu NetLogo Web [18].

dzenia od innych wierzchołków podczas losowego błędzenia po grafie. Nie mniej jednak, graf jest dość mocno usieciowany, a rozmiar wspomnianych hubów jest względnie niewielki.

Aby zbadać zależności między częścią grafu powiązaną z zestawem najbardziej złożonych względnie reguł (106, 120, 169, 225) podzielono graf przepływu złożoności na dwa rozdzielne podgrafy. Pierwszy z podgrafów jest indukowany zbiorem wierzchołków, które posiadają jedną z wymienionych czterech reguł jako algorytm (ΔG_{106}), drugi z nich jest indukowany zbiorem pozostałych wierzchołków ($\Delta G_{\overline{106}}$). Podgraf składający się tylko z wierzchołków z regułą 106 i jej ekwiwalentami liczy 833 wierzchołki oraz 1538 łuków, stanowiąc odpowiednio 47,35% wierzchołków oraz 11,54% łuków oryginalnego grafu.

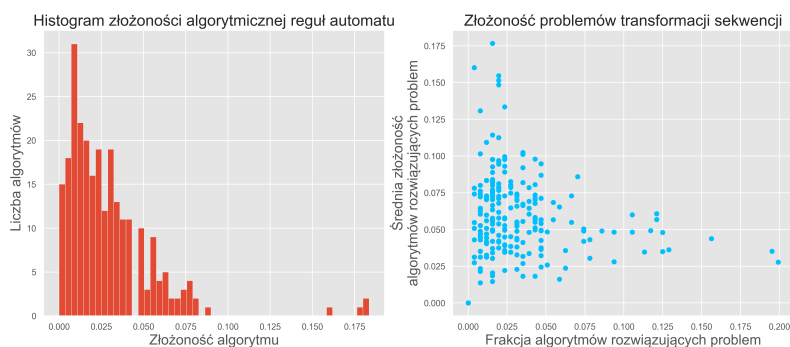
Na rysunku 12 przedstawiono wyniki porównania rozkładów wag łuków oraz dodatkowo zmiany wartości PageRank dla obu podgrafów. Wykresy 12A oraz 12G przedstawiają rozkłady zmian złożoności algorytmów dla odpowiednio podgrafu składającego się ze złożonych reguł i podgrafu złożonego z pozostałych algorytmów. Porównując te rozkłady można zauważyć dwumodalność, przy czym dla ΔG_{106} wartości są przesunięte w prawo (średnia $0,1227 \pm 0,0448$), sugerując wzrost złożoności przy zmianie stanu układu. Rozkład zmian złożoności algorytmów dla $\Delta G_{\overline{106}}$ jest skupiony bliżej 0 (średnia wynosi $-0,014 \pm 0,0516$) przy czym część wartości go obniża, sprawiając, że złożoność algorytmiczna w tej części grafu nie zmienia się lub nawet trochę spada. Test Kolmogorowa-Smirnowa jasno wskazuje, że rozkłady zmian złożoności algorytmicznej są różne w podgrafach $\Delta G_{\overline{106}}$ oraz ΔG_{106} ($D=0,9063$; $p < 10^{-4}$).

Na wykresach 12B, 12C, 12H, 12I przedstawiono rozkłady dla zmian złożoności problemu. Wykresy 12B i 12H dotyczą zmian frakcji algorytmów rozwiązujących problem, odpowiednio w podgrafach ΔG_{106} oraz $\Delta G_{\overline{106}}$, natomiast wykresy 12C i 12I dotyczą zmian średniej złożoności algorytmów rozwiązujących problem (znów odpowiednio

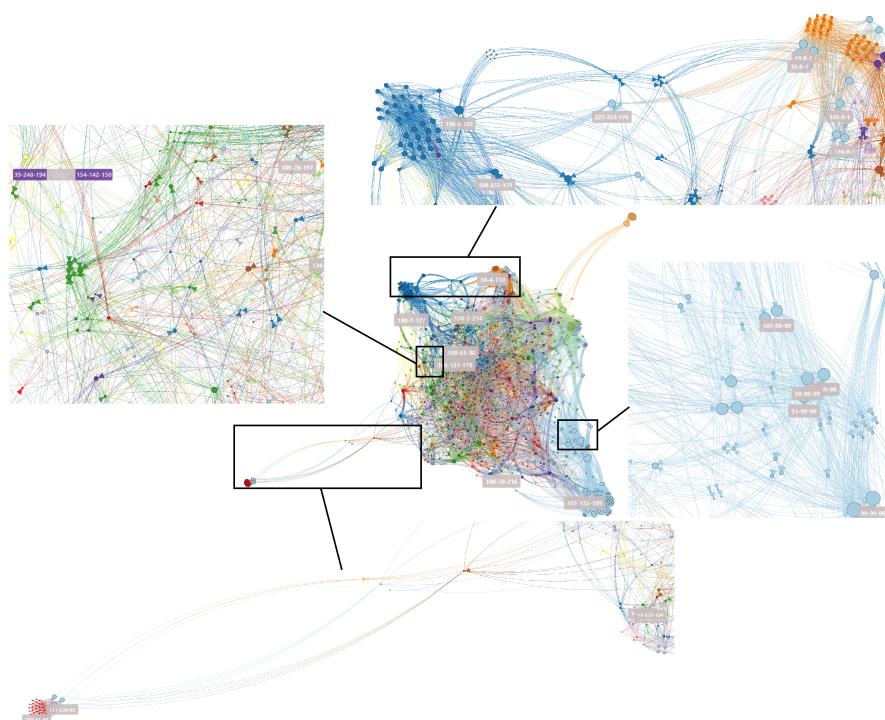


Rys. 8. Wykresy przedstawiające liczbę dopasowań dla każdego z algorytmów (po lewej w barwie czerwonej) i każdego z problemów (po prawej, wykres w kolorze niebieskim). Każdy słupek na dla algorytmów oznacza liczbę rozwiązywanych przez regułę problemów. Analogicznie, po prawej stronie wysokość słupków oznacza liczbę algorytmów, które rozwiązują problem o danym numerze porządkowym.

w $\blacktriangle G_{106}$ i $\blacktriangle G_{106}^-$. Dla obu rozkładów dla zmian złożoności problemu nie widać dużych różnic między podgrafami $\blacktriangle G_{106}$ i $\blacktriangle G_{106}^-$, co jest uzasadnione tym, że transformacja problemu dokonywana przez algorytm wskazywała nową parę sekwencji wejściowej i oczekiwanej, które były sparowane losowo. Dla podgrafu $\blacktriangle G_{106}$ średnia zmiana frakcji problemów jaka rozwiązywała problem wynosiła $-0,0098 \pm 0,0367$, a dla $\blacktriangle G_{106}^-$ było to $-0,0369 \pm 0,0615$. Natomiast średnia zmiana złożoności algorytmów rozwiązujących dany problem wynosi dla podgrafu $\blacktriangle G_{106}$ $-0,0311 \pm 0,0520$, a dla $\blacktriangle G_{106}^-$ jest to $0,0056 \pm 0,0429$. Porównanie rozkładów dla podgrafów $\blacktriangle G_{106}$ oraz $\blacktriangle G_{106}^-$ wykonano także za pomocą testu Kolmogorowa-Smirnowa. Dla rozkładów zmian frakcji rozwiązujących problem algorytmów ($D=0,2676$; $p < 10^{-4}$) test wykazał istotne różnice, podobnie było w przypadku zmian złożoności algorytmów rozwiązujących problem w wierzchołku



Rys. 9. Rozkłady złożoności algorytmów (po lewej) i problemów (po prawej).

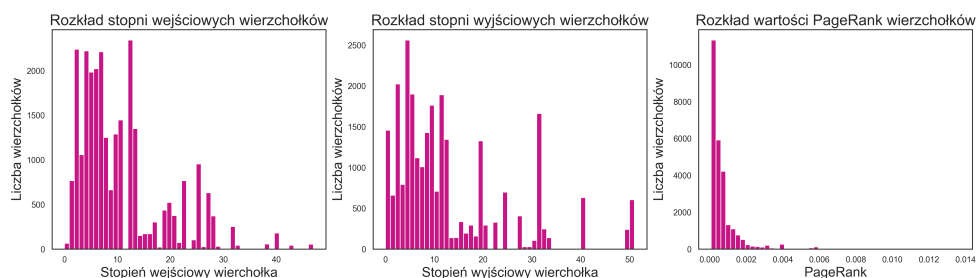


Rys. 10. Graf przepływu złożoności, niektóre fragmenty zostały przybliżone.

($D=0,3175$; $p<10^{-4}$).

Rozkład dla zmian wartości PageRank jest trochę przesunięty w prawo dla ΔG_{106} (12D) z średnią wynoszącą $0,0007 \pm 0,0016$, w porównaniu z rozkładem dla ΔG_{106} (12J), gdzie średnia wyniosła $0,0001 \pm 0,0012$. Oznacza to częstsze podążanie od mniej do bardziej ważnego wierzchołka w podgrafie ΔG_{106} . Test Kolmogorowa-Smirnowa wskazuje na istotne różnice między rozkładami, co są dużo mniejsze niż w przypadku uprzednio omawianych par rozkładów ($D=0,1989$; $p<10^{-4}$).

Szczególnie wyraźne różnice między ΔG_{106} oraz ΔG_{106} są widoczne dla rozkładów liczb transformacji potrzebnych do rozwiązania problemu (wykresy 12E i 12K) oraz zmian średniej liczby transformacji (wykresy 12F i 12L). Reguły 106, 120, 169 i 225 są w stanie wykonywać transformacje o wiele dłużej nie wpadając w powtarzający się wzorec, co czyni je unikalnymi z pośród badanych reguł jednowymiarowego automatu. W konsekwencji rozkład dla liczby kroków w podgrafie ΔG_{106} zwiera w sobie



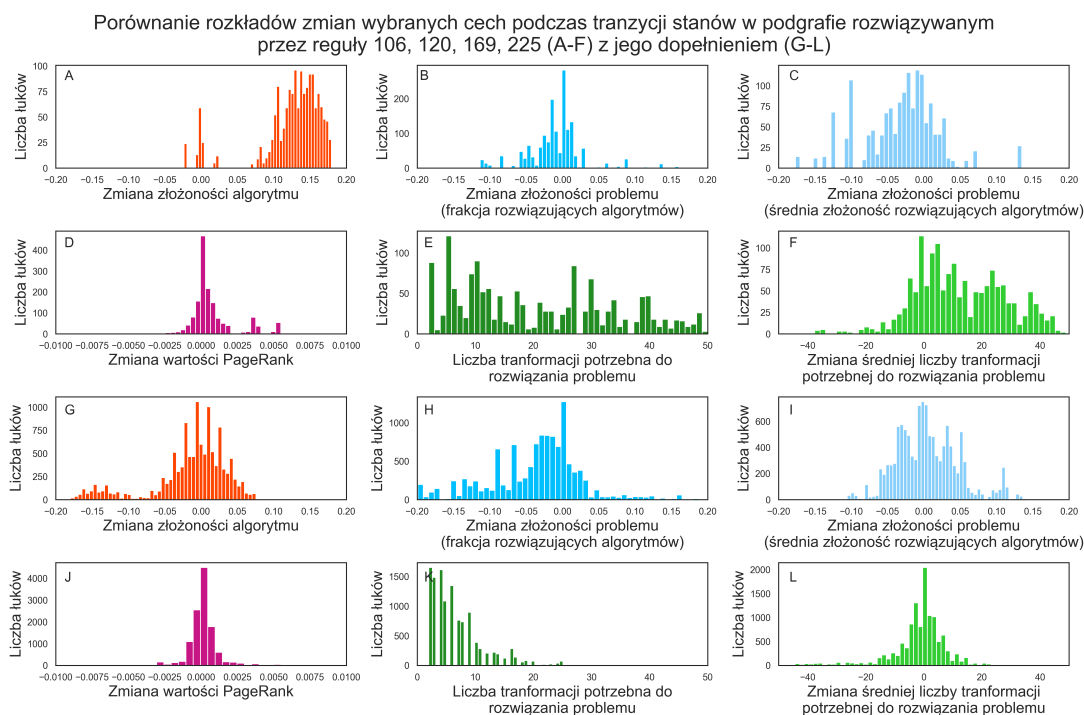
Rys. 11. Rozkłady stopni wejściowych oraz wyjściowych w analizowanym grafie przepływu złożoności oraz histogram dla wartości PageRank wyliczonej dla wierzchołków ($\alpha = 0,95$).

przebiegi nawet 50 transformacji prowadzących do poprawnego dopasowania sekwencji (średnia to: $20,5091 \pm 13,599$). Tego samego nie można powiedzieć o pozostałych regułach automatu, gdzie duża część wyników jest prawdopodobnie oportunistyczna, a większość przebiegów jest krótsza od 10 transformacji (średnia: $6,7043 \pm 4,6069$). Porównanie rozkładów w oparciu o test Kolmogorowa-Smirnowa jasno potwierdza wymienione obserwacje ($D=0,5374$, $p < 10^{-4}$). Zmiany średniej liczby transformacji potrzebnej do rozwiązania problemu są miarą bardziej skoncentrowaną na problemach. Jednoznacznie widać z rozkładów (wykresy 12F i 12L), że tylko w przypadku ΔG_{106} średnia liczba transformacji ma tendencję wzrostową, co oznacza, że trudne problemy są wyłącznie rozwiązywane przez regułę 106 oraz reguły ekwiwalentne do niej. Proste problemy są rozwiązywane przez wszystkie reguły zarówno te w podgrafie ΔG_{106} jak i ΔG_{106} . Warto tutaj dodać, że w przypadku ΔG_{106} rozkład dla zmian średniej liczby transformacji jest skoncentrowany poniżej zera (średnia: $-1,6008 \pm 9,5476$), w przeciwieństwie do rozkładu ΔG_{106} , który zwiera rozciągający się duży ogon (średnia: $12,2798 \pm 15,6629$). Porównanie tych rozkładów jednoznacznie potwierdza wspomniane różnice ($D=0,9622$; $p < 10^{-4}$).

6. Zastosowania

6.1. Analiza dynamiki procesów ekologiczno-ewolucyjnych

Analiza dynamiki procesów ekologiczno-ewolucyjnych polega na badaniu, w jaki sposób organizmy i ich populacje zmieniają się na skutek interakcji ekologicznych i powiązanych z nimi presji ewolucyjnych. Możliwość analizy potencjału populacji organizmów w środowisku oraz wpływania na ich ewolucję (np. poprzez introdukcję na nowy obszar lub inżynieria presji selekcyjnych), może przyczynić się do ochrony wymierających gatunków, opracowania nowych technik uprawy roślin lub też projektowania całych ekosystemów. W tej pracy został wykorzystany przykład takiego układu tj. populacja bakterii *E. coli*, której celem była adaptacja do zróżnicowanego i zmieniającego się środowiska. Na tym przykładzie pokazano jak zmienia się złożoność algorytmiczna populacji oraz jak zastosowania złożonego algorytmu wpływa na rozwój i szerokość niszy (rozumianą przez zakres rozwiązywanych problemów). Dodatkowo, automaty komórkowe są często wykorzystywane do symulacji procesów ekologicznych i ewolucyjnych [2]. Grafy przepływu złożoności mogą pomóc w analizie dopasowania różnych populacji do różnych warunków środowiskowych, pozwalając na określenie sytuacji, w których



Rys. 12. Zestawienie rozkładów różnych statystyk dotyczących dwóch części badanego grafu złożoności: podgraf zawierający w wierzchołkach jedną z czterech ekwiwalentnych reguł: 106, 120, 169, 225 (A-F) oraz pozostały podgraf niezawierający tych algorytmów w swoich węzłach (G-L). Można wyróżnić łącznie 6 rodzajów rozkładów: Rozkład zmian złożoności algorytmu (A - dla podgrafu ze złożonymi regułami, G - dla pozostałej części grafu); Rozkład zmian złożoności problemu wyrażonej jako część algorytmów rozwiązujących problem (odpowiednio B i H); Rozkład zmian złożoności problemu wyrażonej średnią złożonością algorytmów go rozwiązujących (C i I); Rozkład zmian wartości PageRank (D i J); Rozkład dla liczby transformacji potrzebnej do rozwiązania problemu przez algorytm (E i K); Rozkład zmian średniej liczby transformacji wymaganej to przekształcenia wejściowej sekwencji w oczekiwaną (F i L).

dobór naturalny może sprzyjać zmianom złożoności.

6.2. Sterowanie ewolucją domen białkowych

Ewolowalność, czyli zdolność systemu biologicznego do generowania dziedzicznych zmian fenotypowych, jest kluczowym czynnikiem w potencjale adaptacyjnym białek. Zrozumienie ewolowalności miejsc aktywnych białek jest niezbędne dla uzyskania wglądu w funkcję enzymów, ich adaptację i inżynierię. Grafy przepływu złożoności mogą stanowić nową metodę kwantyfikacji dynamiki strukturalnej i funkcjonalnej, które przyczyniają się do zmian miejsc aktywnych białek. Grafy przepływu złożoności umożliwiają śledzenie parametrów związanych ze złożonością miejsc aktywnych białka w odniesieniu do trudności związanej z katalizowaniem określonej reakcji chemicznej. Innymi słowy, grafy przepływu złożoności mogą reprezentować gradienty przyrostu informacji, które można zaobserwować podczas ewolucji biologicznej lub stochastycznej optymalizacji. Ponieważ aktywne miejsca białek można z powodzeniem

modelować za pomocą automatów komórkowych [21], mogą one zostać wykorzystane do przetestowania potencjału zastosowania grafów przepływu złożoności. Grafy przepływu złożoności mogą mieć znaczące implikacje dla dziedzin takich jak inżynieria enzymów, czy biologia syntetyczna. Natomiast lepsze zrozumienie ewoluowalności miejsc aktywnych może prowadzić do opracowania nowych narzędzi obliczeniowych ułatwiających proces projektowania nowych enzymów, dzięki inżynierii trajektorii ewolucyjnych.

6.3. Analiza przepływu informacji w sieciach neuronowych

Zastosowanie grafów przepływu złożoności do analizy sieci neuronowych mogłoby umożliwić dokładne określenie, które z fragmentów sieci są aktywne podczas przetwarzania różnych informacji otrzymanych na wejściu. Analiza pozwala na określenie, jak dane wejściowe wpływają na działanie poszczególnych neuronów i warstw w sieci. Dzięki mapom przepływu informacji możliwa jest identyfikacja konkretnych obszarów sieci, które są najbardziej podatne na błędy. Obecnie istnieje wiele podejść adresujących ten problem [10], a samo zagadnienie cieszy się zainteresowaniem badaczy. Dzięki czemu, możliwe jest bardziej precyzyjne wzmacnianie wrażliwych obszarów i minimalizowanie pomyłek. Poprzez wizualizację przepływu informacji w sieci, grafy przepływu złożoności mogłyby zwiększyć przejrzystość działania modelu. Ułatwia to interpretację decyzji podejmowanych przez sztuczną inteligencję, co jest kluczowe w aplikacjach wymagających wysokiego poziomu zaufania i zrozumiałości.

Złożoność warstwy sieci neuronowej można ocenić np. za pomocą analizy entropii rozkładu wag w poszczególnych neuronach, natomiast trudność problemu można wyrazić przez liczbę różnych zestawów wag na warstwie sieci neuronowej, które w satysfakcjonujący sposób przekształcają dane wyjściowe w wyniki na wyjściu. Zarówno miara złożoności jak i miara trudności muszą być zawsze szacowane przez próby statystyczne. Wynikiem analizy dostarczanym przez grafy przepływu złożoności jest informacja o fragmentach sieci neuronowej, które są aktywnie wykorzystywane do przetwarzania informacji oraz takich, które nie są często lub wcale używane, dzięki czemu można zmapować wybrane fragmenty sieci neuronowej i wprowadzić zabezpieczenia i modyfikacje usprawniające model. Możliwe są także badania różnych zbiorów danych, które potrafiłyby w określony sposób przekształcić rozkład wag w głębokiej sieci neuronowej podczas procesu dotrenowywania, zmieniając jego względną złożoność.

7. Podsumowanie

Omawiany przykład procesu konstrukcji kontekstu, który modeluje ewolucję populacji *E. coli* w różnych warunkach środowiskowa, pozwala na analizę podstawowych cech plastyczności ewolucyjnej. W pracy wykazano, że nawet w tak prostym jak zaproponowany układ pojawiają się trzy rodzaje algorytmów. Pierwszy z nich to proste algorytmy zdolne do rozwiązania małej liczby problemów. To najliczniejsza grupa, która ma ograniczony zakres przetwarzania informacji, a większość sukcesów zawdzięcza przypadkowi. Tego typu populacje bakterii zostałyby szybko unicestwione przez zmianę warunków środowiskowych lub konkurencję. Drugi rodzaj algorytmów to takie, które są naprawdę złożone, ale są w stanie rozwiązać jedynie wybrane problemy. Tego typu populacje bakterii nazwalibyśmy specjalistami. Co prawda, przeniesienie bakterii do innego środowiska sprawi, że prawdopodobnie wyginą, jednak są w stanie wygrać z

konkurencją w środowisku do jakiego się dostosowały. Ostatnim rodzajem są algorytmy zdolne do rozwiązywania wielu problemów. Odpowiadające im populacje można określić jako generalistów, co oznacza że radzą w wielu różnych środowiskach, ale w żadnym nie robią tego wyjątkowo dobrze. Algorytmy powiązane z tą grupą to reguła 106 oraz jej odpowiedniki, która mimo iż była w stanie rozwiązać wiele problemów to odbywało się to kosztem potrzebnej do tego dużej liczby transformacji, jakie musiały zostać wykonane na sekwencji otrzymanej na wejściu.

Dzięki losowemu sparowaniu wejściowych i oczekiwanych sekwencji możliwy był podział otrzymanego grafu przepływu złożoności na dwa podgrafy (ΔG_{106} oraz $\Delta G_{\overline{106}}$). Porównanie podgrafów pozwoliło na podkreślenie dynamiki zmian układu zależnych od złożoności algorytmicznej reguł. W szczególności widoczne jest to, że prostsze algorytmy zachowują średnią złożoność problemów i algorytmów na mniej więcej stałym poziomie. Natomiast złożone algorytmy, które są w stanie rozwiązywać zarówno proste jak i trudne problemy, cechowały się średnim przyrostem złożoności. Analiza zmienności złożoności dla zamodelowanego procesu konstrukcji kontekstu pozwala na zbadanie trendów w zmianach złożoności badanych układów.

Sterowanie złożonością układów dynamicznych jest bardzo trudne do wykonania. Jednym z głównych problemów, które wymagają rozwiązania w tej kwestii jest opracowanie sposobu mierzenia i monitorowania złożoności systemów, w taki sposób by objaśniał on rodzaj i kierunek zmian podczas ewolucji. W tym kontekście kluczowa okazuje się informacja, a dokładnie sposób szacowania prawdziwej zawartości informacyjnej układów, a także złożoności powiązanych z nimi problemów. Ta praca wprowadza wiele nowych konceptów, tj. demoniczny krajobraz, graf przepływu złożoności, czy proces konstrukcji kontekstu. Wszystkie te pojęcia zostały opracowane, aby badać dynamikę złożoności rzeczywistych układów takich jak organizmy żywe, czy procesy fizyczne. Zgłębienie mechanizmów rządzących zmianami złożoności może w przyszłości pozwolić na opracowanie nowych metod projektowania i sterowania złożonymi systemami. Może tak być na przykład poprzez rozbijanie dużych i złożonych problemów na mniejsze lub wymuszanie samoorganizacji i specjalizacji u niezróżnicowanych grup robotów, co powodowałoby, że ich zadania zmieniają się w zależności od otoczenia i zadań innych robotów. Monitorowanie złożoności może też okazać się cenne przy analizie i inżynierii sieci neuronowych, co mogłoby pozwolić na detekcje (i instalacje) dodatkowych i potencjalnie niepożądanych funkcjonalności.

LITERATURA

1. Borenstein Y., Poli R.: Information landscapes and problem hardness. In Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation (GECCO '05). Association for Computing Machinery, New York, NY, USA, p. 1425–1431.
2. Breckling B., Pe'er G., Matsinos Y.G.: Cellular Automata in Ecological Modelling. w: Jopp F., Reuter H., Breckling B. (red.) *Modelling Complex Ecological Dynamics*, Springer, Berlin, Heidelberg, 2011, p. 105-117.
3. Buhmann J.M.: Context Sensitive Information: Model Validation by Information Theory. w: Martínez-Trinidad J.F., Carrasco-Ochoa J.A., Ben-Youssef Brants C., Hancock E.R. (red.) *Pattern Recognition*, Mexican Conference on Pattern Recognition 2011, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg 6718, 2011, p. 12-21.

4. Carscadden K.A., Emery N.C., Arnillas C.A., Cadotte M.W., Afkhami M.E., Gravel D., Livingstone S.W., Wiens J.J.: Niche Breadth: Causes and Consequences for Ecology, Evolution, and Conservation. *The Quarterly Review of Biology*, 2020, vol 95(3), p. 179-214.
5. Constant A., Ramstead M.J.D., Veissière S.P.L., Campbell J.O., Friston K.J.: A variational approach to niche construction. *Journal of the Royal Society Interface*, 2018, vol. 15(141), p. 1-14.
6. Cook, M.: Universality in Elementary Cellular Automata. *Complex Systems*, 2004, vol. 15(1), p. 1-40.
7. Devine, S.: Algorithmic Information Theory for Physicists and Natural Scientists, Rozdział 3: AIT and algorithmic complexity, Institute of Physics Publishing, Wielka Brytania, 2020, p. 3.1-3.28.
8. Durand-Lose, J.: Cellular Automata, Universality of. w: Meyers, R. (red.) *Encyclopedia of Complexity and Systems Science*. Springer, New York, 2009, p. 901-913.
9. Fragata I., Blanckaert A., Dias-Louro M.A., Liberles D.A., Bank C.: Evolution in the light of fitness landscape theory, *Trends in Ecology & Evolution*, 2019, vol 34(1), p. 69-82.
10. Ivanovs M., Kadikis R., Ozols K.: Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters*, 2001, vol. 150, p. 228-234.
11. Krakauer D.C.: Darwinian demons, evolutionary complexity, and information maximization. *Chaos*, 2011, vol. 21(3), p. 1-12.
12. Liu J., Abbass H.A., Green D.G., Zhong W.: Motif Difficulty (MD): A Predictive Measure of Problem Difficulty for Evolutionary Algorithms Using Network Motifs. *Evolutionary Computation*, 2012, vol. 20(3), p. 321-347.
13. Matthews B., De Meester L., Jones C. G., Ibelings B.W., Bouma T.J., Nuutinen V., van de Koppel J., Odling-Smee J.: Under niche construction: An operational bridge between ecology, evolution, and ecosystem science. *Ecological Monographs*, 2014, vol. 84(2), p. 245-263.
14. Mustonen V., Lässig M.: Fitness flux and ubiquity of adaptive evolution. *Proceedings of the National Academy of Sciences*, 2010, vol. 107(9), p. 4248-4253.
15. Naoaki C., Reiji S., Takaya A.: Evolution of Complex Niche-Constructing Behaviors and Ecological Inheritance of Adaptive Structures in a Physically Grounded Environment. *Frontiers in Robotics and AI*, 2020, vol. 7(45), p. 1-13.
16. Odling Smee, J., Laland, K., Feldman, M.: *Niche Construction: The Neglected Process in Evolution*. Princeton: Princeton University Press, 2003.
17. Pickover, C.A.: *The Math Book: From Pythagoras to the 57th Dimension, 250 Milestones in the History of Mathematics*. Sterling Publishing Company, Inc., 2009, p. 406.
18. Wilensky, U. (1998). NetLogo CA 1D Elementary model. www.netlogoweb.org, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. (data do-
stepu: 20.07.2024).
19. Wolfram, S.: *Theory and Applications of Cellular Automata*. World Scientific Publishing Co. Ltd., 1986, p. 485-557.
20. Wright, S.: The roles of mutation, inbreeding, crossbreeding, and selection in evolution. *Proceedings of the Sixth International Congress on Genetics*. 1932, vol. 1(8), p. 355-366.
21. Xiao X., Xue G.F., Stamatovic B., Qiu W.R.: Using Cellular Automata to Simulate Domain Evolution in Proteins, *Frontiers in Genetics*, 2020, vol. 11.
22. Zenil, H.: A Review of Methods for Estimating Algorithmic Complexity: Options, Challenges, and New Directions, *Entropy*, 2020, vol. 22(6), p. 1-28.
23. Zenil, H., Adams, A.: Algorithmic Information Dynamics of Cellular Automata. w: Das S., Roy S., Bhattacharjee K. (red.) *The Mathematical Artist, seria: Emergence, Complexity and Computation*, 45, Springer, Cham., 2022, p. 171-190.