

Bartłomiej SZAWULAK<sup>1</sup>, Piotr FORMANOWICZ<sup>1,2</sup>

<sup>1</sup>Instytut Informatyki, Politechnika Poznańska

<sup>2</sup>Instytut Chemii Bioorganicznej, Polska Akademia Nauk

## DEKOMPOZYCJA SIECI JAKO PODSTAWA DO PORÓWNYWANIA SIECI PETRIEGO\*

**Streszczenie.** W niniejszej pracy zaprezentowany został algorytm porównywania sieci Petriego. Metoda opiera się na dekompozycji grafu na podgrafy odpowiadające minimalnym podprocesom o samodzielnym znaczeniu i porównaniu ich struktur z podsieciami uzyskanymi z dekompozycji drugiej sieci. Algorytm bazuje wyłącznie na informacjach zakodowanych w strukturze grafu bez uwzględniania dodatkowych informacji o systemach, których porównywane sieci są modelami.

### NET DECOMPOSITION AS A BASE FOR PETRI NET COMPARISON

**Summary.** In this article a new algorithm for Petri net comparison was presented. Main idea is to decompose graph to subgraphs that represent minimal meaningful subprocesses and comparing with subprocesses from other net. Algorithm bases only on information coded in graph structure without regard to additional information about systems whose compared networks are models.

### 1. Wstęp

Współcześnie modele grafowe są coraz częściej używane w biologii oraz chemii do opisywania złożonych procesów i reakcji. Jednak wraz ze wzrostem skomplikowania grafów gwałtownie rośnie czas potrzebny do jego analizy. Jest to zwłaszcza widoczne podczas określania poziomu podobieństwa pomiędzy dwoma grafami.

Wśród metod służących do modelowania i analizy złożonych systemów biologicznych na szczególną uwagę zasługują te oparte na sieciach Petriego. Nie są one grafami, jednak ich struktura stanowi skierowany ważony graf dwudzielny. Niestety, bezpośrednie zastosowanie znanych metod porównywania grafów do porównywania sieci Petriego nie daje zadowalających wyników.

Do niedawna znana była tylko jedna metoda porównywania sieci Petriego [1]. Metoda ta bazuje na konwersji sieci metabolicznych do sieci Petriego, a następnie porównywaniu niezmienników należących do każdej z sieci.

W niniejszym artykule zaprezentowany został nowy algorytm porównywania sie-

---

\*Badania częściowo przeprowadzone w ramach realizacji projektu finansowanego ze środków Narodowego Centrum Nauki przyznanych na podstawie decyzji numer DEC-2012/07/B/ST6/01537.

ci Petriego. W odróżnieniu od wspomnianej wcześniej metody, jest on oparty na podziale porównywanych sieci na rozłączne fragmenty oraz na porównaniu ich struktur.

## 2. Sieci Petriego

Wprowadzenie do algorytmu zaczniemy od przedstawienia matematycznej definicji sieci Petriego.

### Definicja 1

Sieć Petriego jest 5-elementową krotką  $N = \{P, T, F, W, m_0\}$ :

- $P = \{p_1, p_2, \dots, p_n\}$  i  $T = \{t_1, t_2, \dots, t_m\}$  są skończonymi, rozłącznymi zbiorami miejsc i tranzycji,
- $F \subseteq (P \times T) \cup (T \times P)$  jest skończonym zbiorem łuków,
- $W : F \rightarrow \{1, 2, 3, \dots\}$  jest funkcją wag dla każdego łuku,
- $m_0 : P \rightarrow \{0, 1, 2, 3, \dots\}$  jest funkcją przypisującą początkową liczbę tokenów dla każdego miejsca [5].

Sieci Petriego mają strukturę dwudzielnego, skierowanego grafu ważonego. Dwudzielny graf to graf którego wierzchołki mogą zostać podzielone na dwa rozłączne zbiory. Elementy z jednego zbioru mogą być połączone łukiem tylko z elementami z innego zbioru. Sieci Petriego posiadają dwa rodzaje wierzchołków: tranzycje  $T$  i miejsca  $P$ , które mogą być połączone za pomocą łuków. Miejsca należące do zbioru  $P$ , graficznie przedstawiane jako okrąg, przechowują tokeny. Tranzycje ze zbioru  $T$ , reprezentowane przez kwadrat, są odpowiedzialne za przepływ tokenów wewnątrz sieci. Dla każdej tranzycji możemy wyznaczyć zbiór miejsc wejściowych i zbiór miejsc wyjściowych.

- $\bullet t$  - zbiór miejsc wejściowych dla tranzycji  $t$ ,
- $t^\bullet$  - zbiór miejsc wyjściowych dla tranzycji  $t$ .

Z perspektywy miejsc można wyznaczyć zbiory tranzycji wejściowych i tranzycji wyjściowych:

- $\bullet p$  - zbiór tranzycji wejściowych dla miejsca  $p$ ,
- $p^\bullet$  - zbiór tranzycji wyjściowych dla miejsca  $p$ .

O tranzycji  $t$  mówimy że jest aktywna, jeżeli dla każdego miejsca  $p$  ze zbioru  $\bullet t$ , liczba tokenów znajdujących się w nim jest równa bądź większa niż  $w(p, t)$ , gdzie  $w(p, t)$  jest wagą łuku  $(p, t)$ .

Aktywna tranzycja może ale nie musi zostać uruchomiona. Uruchomiona tranzycja pobiera z każdego ze swoich miejsc wejściowych liczbę tokenów równą wadze łuku, który łączy ją z tym miejscem. Jednocześnie generuje tokeny dla każdego z miejsc wyjściowych w liczbie określonej przez wagę łuku łączącego te wierzchołki. Proces uruchamiania tranzycji jest automatyczny i niepodzielny [5, 6, 4, 3].

Sieć Petriego można reprezentować za pomocą macierzy incydencji. Macierz ta dla grafu skierowanego zawiera informacje o liczbie i zwrocie łuków występujących pomiędzy dwoma wierzchołkami. Dla sieci Petriego  $N$ , gdzie liczba tranzycji jest równa  $m$  a miejsc  $n$ , macierz incydencji  $C = (c_{ij})$ , jest macierzą  $n \times m$  liczb całkowitych:

$$c_{ij} = c_{ij}^+ - c_{ij}^- \quad (1)$$

gdzie  $c_{ij}^+ = w(p_i, t_j)$  i  $c_{ij}^- = w(t_j, p_i)$  [5, 6, 3].

W zaproponowanej w niniejszej pracy metodzie porównywania sieci Petriego istotną rolę odgrywają t-inwarianty i ich wsparcia [6, 3] oraz zbiory ADT [4].

### Definicja 2

t-inwariant jest wektorem  $x$  dla którego spełniony jest warunek:

$$C \cdot x = 0 \quad (2)$$

Zbiór minimalnych inwariantów  $X$  to zbiór inwariantów spełniających poniższy warunek:

$$\forall x \in X \text{ supp}(z) \subset \text{supp}(x), \quad (3)$$

gdzie  $z$  jest innym dowolnym inwariantem ze zbioru wszystkich inwariantów, natomiast  $\text{supp}(x) = \{t_j : x_j > 0\}$  jest wsparciem t-inwariantu  $x$ .

### Definicja 3

Zbiór tranzycji ADT (Abstract Dependent Transition sets)  $A_i$  jest zbiorem spełniającym warunki:

$$A_i \subseteq T \quad (4)$$

$$\cup A_i = T \quad (5)$$

$$\forall i, j : j \neq i \Rightarrow A_i \cap A_j = \emptyset \quad (6)$$

Maksymalne zbiory ADT określane są przez:

$$\forall A_i, \forall x \in X \quad (7)$$

gdzie  $X$  jest zbiorem minimalnych t-inwariantów [4].

$$A_i \subseteq \text{supp}(x) \vee A_i \cap \text{supp}(x) = \emptyset \quad (8)$$

Maksymalne zbiory ADT są z definicji rozłączne i indukują podsieci, które mogą się pokrywać tylko przez miejsca graniczne. Zbiór miejsc granicznych  $P_{IF}$  jest definiowany przez:

$$P_{IF} = (\bullet A_i \cup A_i^\bullet) \cap (\bullet A_j \cup A_j^\bullet) \subseteq P$$

Zbiory ADT można uznać za najmniejsze części sieci posiadające znaczenie biologiczne [4].

### Definicja 4

*Graf  $G$  jest izomorficzny z grafem  $H$  jeśli istnieje takie, różnowartościowe i na, przyporządkowanie (bijekcja) wierzchołków grafu  $H$  wierzchołkom grafu  $G$ , wtedy i tylko wtedy jeśli jakieś dwa wierzchołki są połączone krawędzią w jednym z grafów, to odpowiadające im wierzchołki w drugim grafie również łączy krawędź.*

$$F : V(G) \rightarrow V(H) \quad (9)$$

Nie zostało udowodnione że problem jest rozwiązywalny w czasie wielomianowym, bądź że jest NP-zupełny, pozostaje więc problemem otwartym. Na przestrzeni pięćdziesięciu lat badań nad problemem porównywania grafów opracowano wiele efektywnych algorytmów zajmujących się porównywaniem konkretnych klas grafów jak np. drzewa, grafy planarne, grafy przedziałowe.

*Graf  $G$  nazywamy automorficznym jeśli istnieje relacja izomorfizmu z nim samym.*

$$F : V(G) \rightarrow V(G) \quad (10)$$

### 3. Dekompozycja do podsieci

Ideą algorytmu jest określanie podobieństwa pomiędzy fragmentami sieci odpowiedzialnymi za konkretne podprocesy w modelu biologicznym. Uzyskiwane jest to za pomocą podziału do maksymalnych zbiorów ADT. Do skonstruowania podgrafu w sieci Petriego potrzebne są informacje o drugim rodzaju wierzchołków. Dlatego konieczne jest wyznaczenie wsparcia dla zbiorów maxADT zawierającego miejsca, które są połączone z tranzycjami należącymi do danego zbioru.

$$s_n = \begin{cases} \text{supp}(A_n) \\ A_n \end{cases} \quad (11)$$

gdzie  $s$  jest podsiecią powstałą na podstawie zbioru tranzycji i jego wsparcia.

### 4. Porównywanie

Proces porównywania jest przeprowadzany na grafach zdekomponowanych do podsieci. Podstawową operacją będzie porównanie pary podsieci i określenia odległości pomiędzy nimi. Odległością nazywamy minimalną liczbę zmian jakie należy wykonać aby pomiędzy sieciami zachodził izomorfizm. Przez pojedynczą zmianę rozumiemy dodanie, usunięcie wierzchołka lub łuku. Podejście to bazuje na popularnej metodzie GED (Graph Edit Distance) [2].

Jako że określając podobieństwo opieramy się wyłącznie na strukturze sieci, należy dokonać porównania podsieci należących do tej samej sieci. Efektem jest sprawdzenie czy w grafie lub pomiędzy jego podgrafami występuje automorfizm. Wiedza ta pozwala dokładniej przeanalizować wyniki porównań podgrafów należących do różnych sieci.

Istotnym jest fakt, że wynik porównania podsieci  $s_i$  należącej do sieci  $S^1$  z siecią  $s_j$  należąca do sieci  $S^2$  może być różny od porównania podsieci  $s_j$  do podsieci  $s_i$ . Jest to spowodowane asymetrycznością porównywania. Najprostszym przykładem jest sytuacja, gdy  $s_i$  stanowi podsieć większej  $s_j$ . Wynik porównania  $s_i$  do  $s_j$  będzie bliski 1 (nieobecne łuki nie pozwolą na uzyskanie 100% podobieństwa), podczas gdy  $s_j$  do  $s_i$  będzie proporcjonalnie mniejszy do rozmiaru  $s_j$ . Odpowiedzią na ten problem jest porównanie obu możliwości -  $s_j$  do  $s_i$  i  $s_i$  do  $s_j$ . Liczba porównań podsieci jakie należy więc wykonać mając na uwadze powyższe spostrzeżenia wynosi:

$$(|S^1| + |S^2|) \cdot (|S^1| + |S^2|). \quad (12)$$

#### 4.1. Ograniczenie górne liczby porównań

Zbiór ADT zawierający tylko jedną tranzycję nazywamy trywialnym. Istnieją więc sieci posiadające liczbę zbiorów ADT równą liczbie tranzycji. Oznacza to, że liczba podsieci, które będą uczestniczyć w procesie porównywania może być co najwyżej równa liczbie tranzycji porównywanej sieci:

$$|A| \leq |T|. \quad (13)$$

Tu ujawnia się przewaga w złożoności obliczeniowej nad algorytmem stworzonym przez zespół z Padowy [1], który jest ściśle powiązany z liczbą inwariantów występujących w sieci. Liczba zbiorów ADT w sieci zawsze będzie co najwyżej równa liczbie inwariantów w sieci.

#### 4.2. Graf sąsiedztwa sąsiadów

Na podstawie miejsc  $p$  należących do zbioru miejsc granicznych danego zbioru ADT można łatwo określić ich sąsiedztwo. Dwie podsieci  $s_1$  i  $s_2$  nazywamy sąsiednimi, jeśli istnieje takie miejsce  $p$  znajdujące się we wsparciach tych podsieci:

$$\exists p \in \text{supp}(A_{s_1}) \wedge p \in \text{supp}(A_{s_2}) \quad (14)$$

Miejsce  $p$  nazywamy miejscem granicznym(interface place). Na podstawie tych informacji możemy skonstruować skierowany graf sąsiedztwa dla każdej z sieci. Wierzchołki tego grafu odpowiadają podsiecią  $s$  natomiast łuki i ich wagi odpowiadają połączeniom przechodzącym przez miejsca graniczne.

#### 4.3. Wyniki

Wyniki porównania podprocesów znajdują się w odpowiadających krotkach tabeli wynikowej

$$R = [r_{ij}] \quad (15)$$

gdzie  $i$  i  $j$  odpowiadają indeksom porównywanych podsieci. W komórce  $[r_{ij}]$  znajdują się trzy liczby odpowiadające tranzycjom  $T_s$ , miejscom  $P_s$  i wierzchołkom  $N_s$ .

$$[r_{ij}] = \{T_s, P_s, N_s\} \quad (16)$$

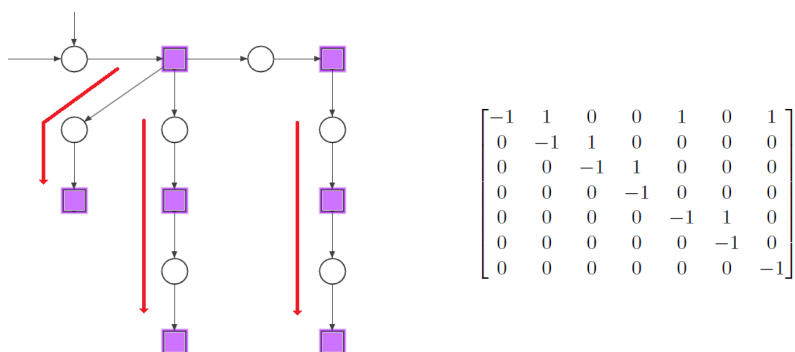
Liczby są z przedziału od 0 do 1, gdzie 1 oznacza pełne podobieństwo pomiędzy danymi typami wierzchołków lub dla  $N_s$  izomorfizm dwóch podsieci. Wypełniona tabela  $R$  zawiera informacje na temat podobieństwa pomiędzy każdą parą podsieci uczestniczących w porównywaniu – z pominięciem porównania do samej siebie, co zawsze będzie skutkowało izomorfizmem.

Korzystając z wygenerowanych wcześniej grafów sąsiedztwa możemy przeanalizować zawartość tabeli wynikowej  $R$  pod względem połączeń pomiędzy podsieciami  $s$ . Dane zawarte w macierzy  $R$  i grafie sąsiedztwa wspólnie kodują informacje o podobieństwie struktury dwóch badanych sieci.

### 5. Porównywanie podsieci

W macierzy  $R$  w krotce  $r_{ij}$  znajdują się wartości określające stopień podobieństwa struktur dwóch podsieci. Uzyskiwane są przez porównanie macierzy incydencji





Rys. 2. Przeszukiwanie włąb i szeregowanie wierzchołków

której porównujemy podsieć  $B$  jest mniejsza, dodanych krawędzi i wierzchołków nie bierzemy pod uwagę przy określaniu poziomu podobieństwa.

Dzięki posiadanej wiedzy o długości gałęzi dodawanie pustych wierszy/kolumn jest trywialne.

### Przykład

Dokonajmy porównania dwóch sieci  $A$  i  $B$ , z których każdą można zdekomponować do dwóch podsieci oznaczanych jako  $a_1, a_2$  i  $b_1, b_2$ . Porównajmy podsieci  $a_1$  i  $b_1$ , gdzie  $a_1$  jest podsiecią z rysunków 1 i 2, a  $b_1$  ma strukturę łańcucha otwartego. W tym celu dokonujemy transformacji do macierzy incydencji i jej uszeregowania według podanego wcześniej schematu. Następnie przechodzimy do oceny podobieństwa macierzy:

$$\begin{bmatrix} -1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \rightarrow \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0,5 \\ 1 \\ 1 \\ 0,5 \\ 0,5 \\ 0,5 \\ 1 \end{bmatrix} = \frac{5}{7} \approx 0,714 = T_s$$

Dla określenia podobieństwa miejsc porównanie dokonujemy przez porównanie kolumn.

W macierzy wynikowej w krotce przeznaczony dla porównania podsieci  $a_1$  i  $b_1$  umieszczamy wynik w postaci 3-elementowej:

$$\begin{bmatrix} \{1, 1, 1\} & \{-, -, -\} & \{-, -, -\} & \{-, -, -\} \\ \{-, -, -\} & \{1, 1, 1\} & \{-, -, -\} & \{-, -, -\} \\ \{0, 714, 0, 857, 0, 785\} & \{-, -, -\} & \{1, 1, 1\} & \{-, -, -\} \\ \{-, -, -\} & \{-, -, -\} & \{-, -, -\} & \{1, 1, 1\} \end{bmatrix}$$

W macierzy  $R$  krotki, dla których  $i = j$ , są z definicji równe  $\{1, 1, 1\}$ , gdyż podsieć jest izomorficzna z samą sobą.

## 6. Podsumowanie

W niniejszej pracy zaproponowany został algorytm porównywania sieci Petriego. Porównywanie takie często jest potrzebne m.in. w sytuacji, gdy analizowane są modele systemów biologicznych pełniących podobne, ale nie identyczne, funkcje, wyrażonych za pomocą tego typu sieci. Istotne jest wtedy zlokalizowanie tych fragmentów sieci, które są odpowiedzialne za różnice pełnionych funkcji. Zaproponowana metoda została skonstruowana tak, aby zwracać informacje zarówno na temat podobieństwa, jak i lokalizacji różnic i jest ona podstawą do dalszych badań w dziedzinie porównywania modeli systemów biologicznych wyrażonych jako sieci Petriego.

## LITERATURA

1. Baldan P., Cocco N., Giummolè F., Simeoni M.: Comparing Metabolic Pathways through Reactions and Potential Fluxes. Proc. BioPPN, 2012.
2. Emmert-Streib F., Dehmer M., Shi Y.: Fifty years of graph matching, network alignment and network comparison. Information Sciences, Volumes 346—347, 2016, Pages 180–197.
3. Formanowicz D., Kozak A., Głowacki T., Radom M., Formanowicz P.: Hemojuvelin-hepcidin axis modeled and analyzed using Petri nets. Journal of Biomedical Informatics 46, 2013, p. 1030–1043.
4. Heiner M.: Understanding Network Behavior by Structured Representations of Transition Invariants. Algorithmic Bioprocesses, 2009.
5. Murata T.: Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE, Vol. 77, No. 4, 1989, 541–580.
6. Sackmann A., Heiner M., Koch I.: Application of Petri net based analysis techniques to signal transduction pathways, BMC Bioinformatics, 7, 2006, 482.