

Kaja GUTOWSKA<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

## **ALGORYTM PRZESZUKIWANIA TABU DLA PROBLEMU ZNAJDOWANIA PODZBIORÓW KLUCZOWYCH TRANZYCJI W MODELACH SYSTEMÓW BIOLOGICZNYCH OPARTYCH NA SIECIACH PETRIEGO\***

**Streszczenie.** Do modelowania i analizy złożonych systemów biologicznych coraz częściej wykorzystywane są sieci Petriego. Analiza modeli wyrażonych za ich pomocą może być oparta m.in. na t-niezmiennikach, które odpowiadają podprocesom niezmiającym stanu modelowanego systemu. W toku takiej analizy pojawia się konieczność znalezienia pewnych szczególnych podzbiorów tranzycji, które odpowiadają elementarnym procesom o kluczowym znaczeniu. Problemy kombinatoryczne związane ze znajdowaniem takich podzbiorów są często obliczeniowo trudne. W niniejszej pracy zaproponowano algorytm tabu rozwiązujący jeden z tego rodzaju problemów.

## **A TABU SEARCH ALGORITHM FOR THE PROBLEM OF FINDING SUBSETS OF CRUCIAL TRANSITIONS IN PETRI NET BASED MODELS OF BIOLOGICAL SYSTEMS**

**Summary.** For modeling of complex biological systems Petri nets are more and more often used. An analysis of such models can be based on t-invariants. These invariants correspond to subprocesses which do not change a state of the modeled system. During the analysis finding of some particular subsets of transitions, which are counterparts of crucial elementary processes, becomes necessary. Combinatorial problems corresponding to searching for such subsets are often computationally intractable. In this paper a tabu search algorithm for one of these problems is proposed.

### **1. Wstęp**

Wraz z dynamicznym rozwojem nauk biologicznych pojawiła się potrzeba tworzenia precyzyjnych matematycznych modeli analizowanych na ich gruncie zjawisk i procesów. Stopień złożoności układów biologicznych jest tak duży, iż jasne stało się, że bez tego typu modeli trudno będzie w pełni zrozumieć ich naturę. Modele takie mogą

---

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

być tworzone przy wykorzystaniu wielu gałęzi matematyki, jednak najczęściej wykorzystywane są w tym celu równania różniczkowe. Od pewnego czasu konstruowane są też modele mające postać różnego rodzaju grafów lub obiektów matematycznych do nich zbliżonych. Pozwalają one w prosty oraz intuicyjny, a jednocześnie precyzyjny sposób odzwierciedlić strukturę badanego systemu biologicznego, a także umożliwiają nieuwzględnianie wartości parametrów opisujących dany system, które na ogół są nieznanne lub znane tylko w przybliżeniu. Wśród tego rodzaju modeli na szczególną uwagę zasługują modele wyrażone jako sieci Petriego [6]. Sieci takie nie są grafami, ale mają strukturę skierowanego ważonego grafu dwudzielnego. Istotną zaletą modeli wyrażonych jako sieci Petriego w stosunku do modeli grafowych jest fakt, że oprócz struktury systemu opisują one również jego dynamikę, dzięki tokenom, które w takiej sieci przepływają [7]. Istnieje wiele metod analizy własności sieci Petriego, jednak w kontekście biologicznym szczególnie użyteczne są metody oparte na t-niezmiennikach. Niezmienniki takie odpowiadają podprocesom, które nie zmieniają stanu analizowanego systemu biologicznego, są więc dla jego funkcjonowania bardzo ważne [8, 5, 2]. W ramach tego rodzaju analizy często zachodzi potrzeba wyznaczenia pewnych podzbiorów tranzycji, które odpowiadają elementarnym procesom występującym we wspomnianych podprocesach. Są to elementarne procesy, które mogą pełnić kluczowe funkcje w badanym systemie [3]. Część problemów kombinatorycznych związanych ze znajdowaniem tego typu podzbiorów to problemy **NP**-trudne. W niniejszej pracy zaproponowano algorytm przeszukiwania tabu dla jednego z takich problemów.

Układ pracy jest następujący. W rozdziale drugim przedstawione jest sformułowanie rozważanego problemu, natomiast w rozdziale trzecim opisany jest zaproponowany algorytm. W rozdziale czwartym krótko opisano zastosowanie algorytmu do analizy rzeczywistych danych biologicznych. Artykuł kończy się podsumowaniem przedstawionym w rozdziale piątym.

## 2. Sformułowanie problemu

Formalnie sieć Petriego może być zdefiniowana jako piątka  $Q = (P, T, F, W, M_0)$ , gdzie:  $P = \{p_1, p_2, \dots, p_n\}$  jest skończonym zbiorem miejsc,  $T = \{t_1, t_2, \dots, t_m\}$  jest skończonym zbiorem tranzycji,  $F \subseteq (P \times T) \cup (T \times P)$  jest zbiorem łuków,  $W : F \rightarrow \mathbb{Z}^+$  jest funkcją wagi,  $M_0 : P \rightarrow \mathbb{N}$  jest oznakowaniem początkowym,  $P \cap T = \emptyset \wedge P \cup T \neq \emptyset$  [7].

Jeżeli sieć Petriego jest modelem pewnego systemu, miejsca na ogół odpowiadają jego biernym, a tranzycje aktywnym składnikom. Pomiedzy miejscami, poprzez tranzycje, mogą przepływać tokeny, co odpowiada przepływowi przez modelowany system substancji, informacji itp. Przepływem tokenów rządzi reguła uruchomienia tranzycji, zgodnie z którą tranzycja jest aktywna, jeżeli w każdym z miejsc bezpośrednio ją poprzedzających znajduje się liczba tokenów równa co najmniej wadze łuku łączącego dane miejsce z tą tranzycją. Aktywna tranzycja może zostać uruchomiona, co oznacza, że następuje przepływ tokenów z miejsc bezpośrednio ją poprzedzających do miejsc bezpośrednio po niej następujących, przy czym liczba przepływających tokenów jest równa wadze odpowiedniego łuku. Sieć Petriego posiada bardzo intuicyjną reprezentację graficzną, może ona jednak być również opisana za pomocą macierzy incydencji. Element  $a_{ij}$  macierzy incydencji  $A = (a_{ij})_{n \times m}$  jest równy różnicy liczb tokenów znajdujących

się w miejscu  $p_i$  przed i po uruchomieniu tranzycji  $t_j$  [7].

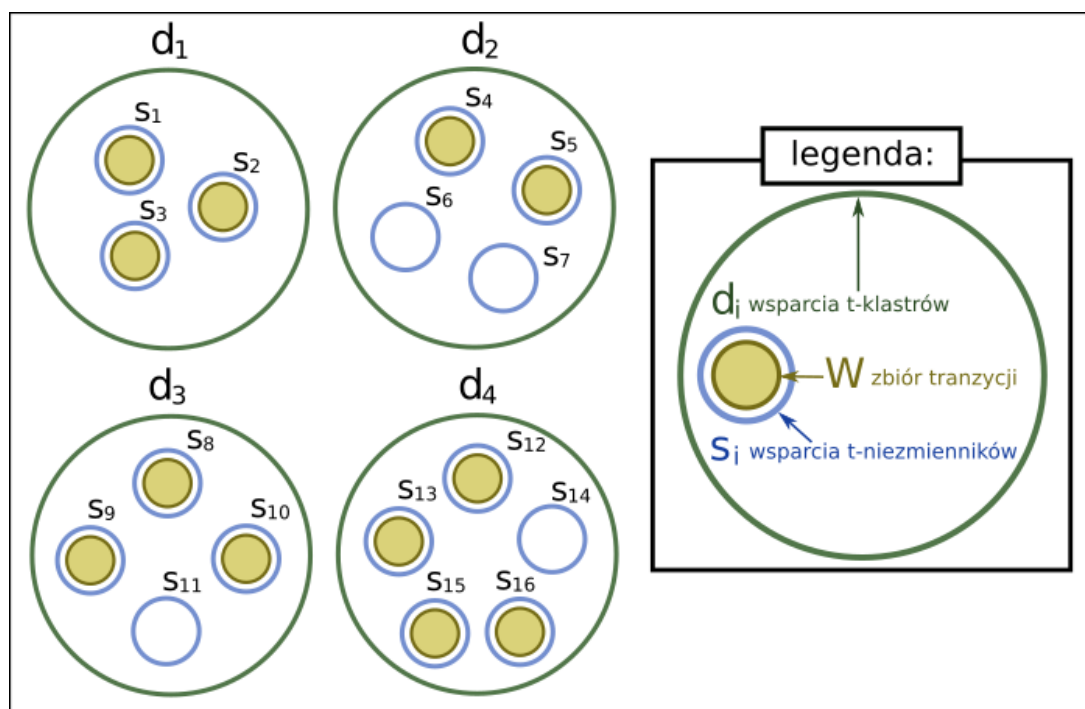
Jak zostało wspomniane wcześniej, analizę modelu systemu biologicznego wyrażonego jako sieć Petriego można oprzeć na t-niezmiennikach. Niezmiennik taki jest wektorem  $x$  spełniającym równanie  $A \cdot x = 0$ . Jest z nim związany zbiór tranzycji  $s(x) = \{t_j : x_j > 0\}$  nazywany wsparciem t-niezmiennika  $x$ . Celem analizy jest m.in. znajdowanie t-niezmienników podobnych do siebie, gdyż mogą one odpowiadać podprocesom oddziałującym na siebie nawzajem. W przypadku dużej liczby t-niezmienników grupuje się je w klastry, nazywane t-klastrami, by znaleźć wspomniane podobieństwa. Wsparcia podobnych do siebie t-niezmienników zawierają części wspólne, których elementami są tranzycje o potencjalnie istotnym znaczeniu, gdyż to poprzez nie podprocesy oddziałują na siebie. Jednym z celów analizy jest znajdowanie tego typu podzbiorów tranzycji [2, 3].

W niniejszej pracy zaproponowany jest algorytm przeszukiwania tabu [4] dla jednego z problemów związanych ze znajdowaniem wspomnianych podzbiorów. Problem ten zdefiniowany jest następująco [3].

INSTANCJA: zbiór  $T = \{t_1, t_2, \dots, t_m\}$  tranzycji, zbiór  $S = \{s_1, s_2, \dots, s_p\}$  wsparć t-niezmienników, gdzie  $\forall_{i=1,2,\dots,p} s_i \subseteq T$ , zbiór  $C = \{c_1, c_2, \dots, c_q\}$  klastrów, gdzie  $\forall_{c_i \in C} c_i = \{s_{i_1}, s_{i_2}, \dots, s_{i_{|c_i|}}\}$ ,  $\forall_{i=1,2,\dots,q; j=1,2,\dots,q; i \neq j} c_i \cap c_j = \emptyset$ , liczby  $K \in \mathbb{Z}^+$  i  $R \in \mathbb{Z}^+$ .

ODPOWIEDŹ: zbiór  $W = \{w_1, w_2, \dots, w_r\} \subseteq T$  taki, że  $\exists_{c_{i_1}, c_{i_2}, \dots, c_{i_k} \in C} W \subseteq d_{i_1}, W \subseteq d_{i_2}, \dots, W \subseteq d_{i_k}$ , gdzie  $\forall_{c_i \in C} d_i = \bigcup_{j=1}^{|c_i|} s_{i_j}$  oraz  $k \geq K$  i  $r \geq R$ .

Rozwiązaniem powyższego problemu jest podzbiór tranzycji, które występują we wsparciach t-niezmienników, które z kolei znalazły się w odpowiednio dużej (tj. nie mniejszej niż  $K$ ) liczbie t-klastrów, przy czym podzbiór ten musi składać się z co najmniej  $R$  tranzycji (zbiory  $d_i$  nazywane są wsparciami t-klastrów). Taki zbiór odpowiada elementarnym procesom, które pojawiają się w wielu grupach podprocesów, stąd mogą mieć istotne znaczenie dla funkcjonowania analizowanego systemu biologicznego (por. [3]). Idea ta zilustrowana została na Rysunku 1.



Rys. 1. Ilustracja rozważanego problemu kombinatorycznego

### 3. Algorytm przeszukiwania tabu

Zdefiniowany w poprzednim rozdziale problem poszukiwania podzbioru tranzycji we wsparciach  $t$ -niezmienników należy do klasy problemów **NP**-trudnych, stąd do jego rozwiązania zaproponowano algorytm heurystyczny przeszukiwania tabu. Schemat blokowy wspomnianego algorytmu przedstawiony został na Rysunku 2, natomiast poniżej znajduje się opis jego działania.

#### 1. Generowanie rozwiązania początkowego:

Instancja problemu zawiera zbiór wsparć  $t$ -niezmienników, które są zbiorami tranzycji. Pierwszym krokiem podczas generowania rozwiązania początkowego jest znalezienie najbardziej licznych tranzycji, przez co rozumie się tranzycje z maksymalną liczbą wystąpień we wsparciach  $t$ -niezmienników. Na tej podstawie zostaje wyznaczona wartość  $r$ , czyli liczba tranzycji znajdujących się w rozwiązaniu początkowym. Następnie sprawdzana jest zgodność  $t$ -niezmienników dla wyselekcjonowanych tranzycji, na tej podstawie określana jest wartość  $k$ , oznaczająca liczbę wsparć  $t$ -niezmienników, w której występują wszystkie tranzycje ze zbioru  $r$ . Zgodność  $t$ -niezmienników jest to część wspólna  $t$ -niezmienników wyznaczona dla wszystkich tranzycji ze znalezionej zbioru  $r$ .

#### 2. Przegląd sąsiedztwa (możliwe do wykonania ruchy):

Celem polepszenia rozwiązania początkowego przeprowadzane są następujące ruchy: dodanie nowej tranzycji do bieżącego rozwiązania lub wymiana tranzycji z bieżącego rozwiązania na inną tranzycję.

- Dodanie tranzycji: zgodnie z zasadami algorytmu przeszukiwania tabu, dodawana jest najlepsza tranzycja. Aby znaleźć najlepszą tranzycję do dodania

sprawdzone są wszystkie możliwe tranzycje (takie, które nie występują na liście tabu). Po dodaniu każdej tranzycji sprawdzana jest wartość  $r$  odpowiadająca wielkości znalezionej zbioru. Następnie sprawdzana jest zgodność  $t$ -niezmienników dla tranzycji z bieżącego rozwiązania i sprawdzana jest wartość  $k$  odpowiadająca liczbie wsparć  $t$ -niezmienników, w których występuje znaleziony zbiór tranzycji o wielkości  $r$ . Najlepsza tranzycja to taka, która najbardziej poprawia rozwiązanie, gdy zostanie ona znaleziona następuje aktualizacja bieżącego rozwiązania i aktualizacja wartości  $r$  i  $k$ .

- Wymiana tranzycji: początkowo usuwana jest tranzycja z bieżącego rozwiązania i dodawana jest nowa. W związku z tym szukana jest najlepsza tranzycja do usunięcia, nazwana jest ona najlepszą tranzycją ponieważ jej usunięcie najbardziej poprawia rozwiązanie pod względem wartości  $k$  (wsparć  $t$ -niezmienników). Po usunięciu takiej tranzycji dodawana jest najlepsza tranzycja zgodnie z zasadami opisanymi w powyższym podpunkcie. Na końcu, po usunięciu tranzycji i dodaniu nowej rozwiązanie jest aktualizowane, włącznie z wartościami  $r$  i  $k$ .

### 3. Aktualizacja najlepszego rozwiązania:

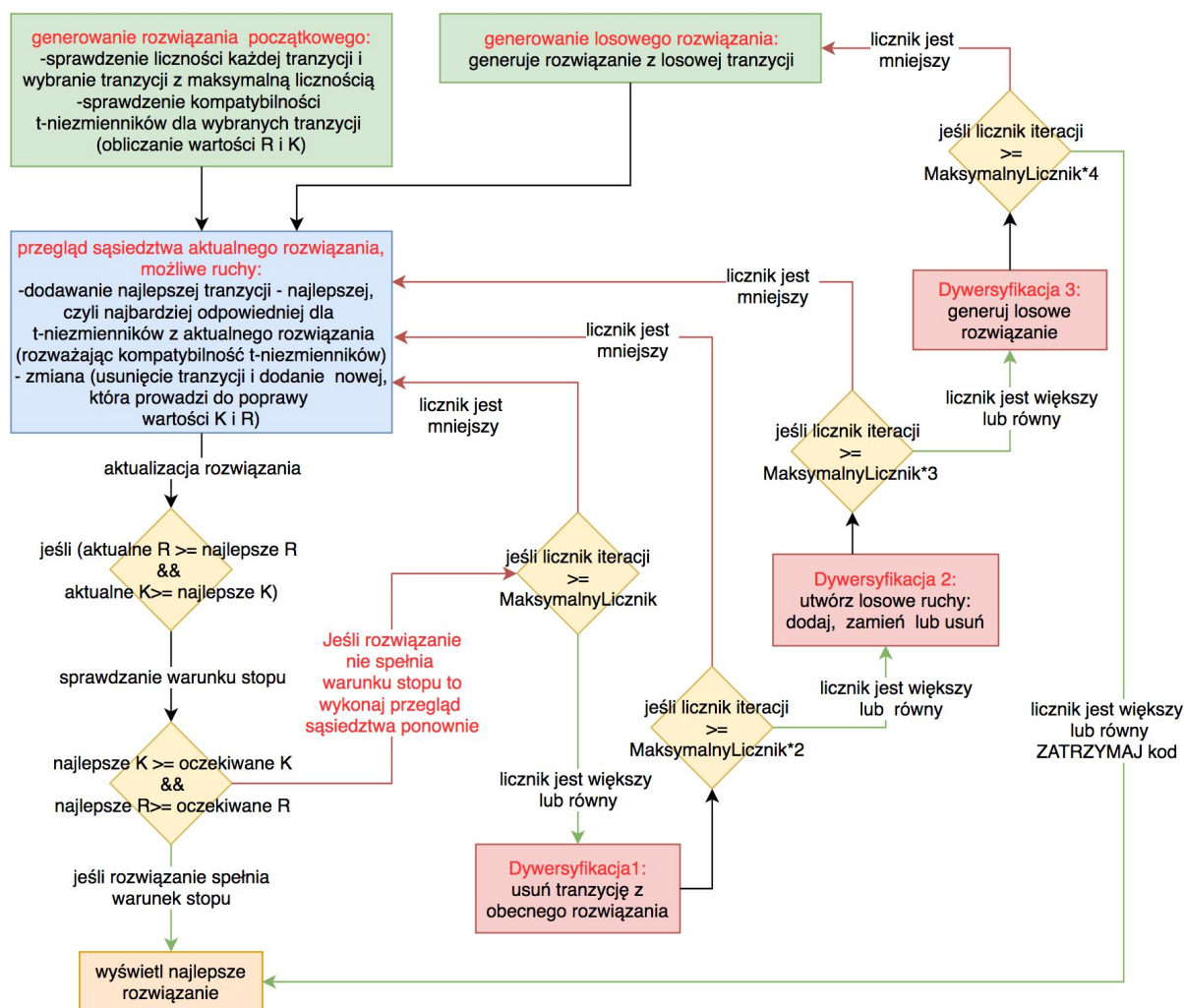
Na podstawie uzyskanych wartości  $r$  i  $k$  można oszacować czy bieżące rozwiązanie jest lepsze od aktualnie najlepszego rozwiązania. Sprawdzany jest następujący warunek: jeśli  $r$  bieżącego rozwiązania jest większe lub równe  $r$  najlepszego rozwiązania oraz  $k$  bieżącego rozwiązania jest większe lub równe  $k$  najlepszego rozwiązania, to bieżące rozwiązanie jest teraz najlepszym rozwiązaniem.

### 4. Warunek stopu:

Zaproponowany algorytm nie jest algorytmem maksymalizacyjnym. Poszukiwane jest rozwiązanie spełniające pewien określony warunek stopu: jeśli  $r$  najlepszego rozwiązania jest większe lub równe oczekiwanej wartości  $R$  oraz wartość  $k$  najlepszego rozwiązania jest większa lub równa wartości oczekiwanej  $K$ , to algorytm kończy swoje działanie zwracając najlepsze rozwiązanie. W przypadku, gdy algorytm nie spełnia warunków stopu przez określoną liczbę iteracji, algorytm kończy działanie zwracając najlepsze rozwiązanie znalezione do tej pory, jednak nie jest to rozwiązanie oczekiwane.

### 5. Dywersyfikacja:

W algorytmie zaproponowano trzy poziomy dywersyfikacji. Jeśli po określonej liczbie iteracji  $n$  możliwych do wykonania ruchów (dodanie lub wymiana tranzycji), rozwiązanie najlepsze nie spełnia warunków oczekiwanych dla parametrów  $k$  i  $r$ , to przeprowadzana jest pierwsza dywersyfikacja, która polega na usunięciu tranzycji. Takie podejście zawsze prowadzi do pogorszenia wartości  $r$ , ponieważ zmniejsza się liczba tranzycji w rozwiązaniu. Jednakże, pomniejszenie  $r$  może niekiedy prowadzić do polepszenia wartości  $k$ . Jeśli po  $2n$  iteracjach możliwych do wykonania ruchów, najlepsze rozwiązanie dalej nie spełnia warunków oczekiwanych, to przeprowadzana jest dywersyfikacja druga. Ta dywersyfikacja polega na wykonaniu losowych ruchów: dodawania, wymiany lub usunięcia tranzycji (zgodnie z listą tabu). Natomiast trzecia dywersyfikacja, uruchamiana po  $3n$  iteracjach bez odnalezienia rozwiązania spełniającego założone oczekiwania polega na utworzeniu całkowicie losowego rozwiązania, składającego się z losowych tranzycji.



Rys. 2. Schemat blokowy działania algorytmu przybliżonego.

Zaprezentowany algorytm jest w fazie testowania oraz dalszego rozwoju. W związku z tym przedstawione w tej pracy wyniki to rezultat wstępnych testów dla wybranych parametrów na podstawie rzeczywistych danych opisanych w punkcie "Zastosowanie algorytmu dla złożonego procesu biologicznego". Testowano wpływ wielkości listy tabu na uzyskane wyniki, co przedstawiono w tabeli 1.

Tabela 1  
Wybrane wyniki działania algorytmu w zależności od wielkości listy tabu.

	Wybrane wyniki		
Liczba iteracji	49	<b>49</b>	49
Rozmiar listy tabu	49	<b>24</b>	10
Uzyskane $r$	15	<b>5</b>	6
Uzyskane $k$	19	<b>72</b>	40

Optymalna wielkość listy tabu wyznaczana jest na podstawie liczby iteracji podzielonej przez dwa ( $liczba\ iteracji/2$ ), podczas gdy liczba iteracji jest to liczba tranzycji zawartych w modelu podzielona przez dwa ( $liczba\ tranzycji/2$ ). Przedstawione w tabeli 1 wyniki pokazują, że gdy wielkość listy tabu jest zbyt duża, to uzyskane wyniki

nie są interesujące z biologicznego punktu widzenia; uzyskany zbiór to 15 tranzycji, które znajdują się w 19 t-niezmiennikach (co stanowi 26% wszystkich t-niezmiennikach). W przypadku, gdy wielkość listy tabu jest zbyt mała, to znalezione rozwiązanie nie jest rozwiązaniem najlepszym; uzyskany zbiór to 6 tranzycji występujących w 40 t-niezmiennikach (co stanowi 55% wszystkich t-inwariantów). W tabeli 1 oznaczono pogrubioną czcionką wartości parametrów, które doprowadziły do znalezienia najlepszego rozwiązania.

W następnej kolejności testowano oczekiwane wartości  $K$  i  $R$ , które powinny być zdeterminowane w kontekście biologicznym, wyniki przedstawiono w tabeli 2.

Tabela 2

Wybrane wyniki działania algorytmu w zależności od wartości dla parametrów oczekiwanych  $K$  i  $R$ .

	Wybrane wyniki		
Liczba iteracji	49	49	49
Rozmiar listy tabu	24	24	24
Oczekiwane $R$	5	15	30
Oczekiwane $K$	5	15	30
Uzyskane $r$	<b>5</b>	15	<b>5</b>
Uzyskane $k$	<b>72</b>	26	<b>72</b>

Jeśli wartości parametrów oczekiwanych są niskie to algorytm ma duże szanse, aby znaleźć rozwiązania spełniające warunki stopu. W przypadku testowanych danych, niskie wartości  $K$  i  $R$  pozwalają na znalezienie najlepszego rozwiązania. Jeśli wartości oczekiwanych parametrów mają średnio duże wielkości to znalezione rozwiązanie nie jest interesujące z biologicznego punktu widzenia. Natomiast jeśli wartości parametrów oczekiwanych są zbyt wysokie to znalezione jest najlepsze rozwiązanie, ale niekoniecznie spełnia ono warunek stopu. Rozwiązanie jest wyświetlone przez przerwanie algorytmu po odpowiedniej liczbie iteracji.

#### 4. Zastosowanie algorytmu do złożonego procesu biologicznego

Przedstawiony w niniejszej pracy algorytm wykorzystano w celu przeprowadzenia bardziej dokładnej analizy t-niezmienników modelu procesu biologicznego dotyczącego rozwoju chorób sercowo-naczyniowych, w których kluczową rolę odgrywa prozapalna cytokina jaką jest interleukina 18 [1].

Zaproponowany model zawiera 83 miejsca, 98 tranzycji, 78 t-niezmienników, 15 t-klastrów. Wstępnym wynikiem działania algorytmu dla tego modelu jest podzbiór 5 tranzycji:  $t_{14}$ ,  $t_{23}$ ,  $t_{45}$ ,  $t_{89}$ ,  $t_{90}$ , które występują we wsparciach 72 t-niezmienników, co stanowi 92% wszystkich t-niezmienników. Znalezione tranzycje odpowiadają elementarnemu procesowi, który związany jest z fosforylacją kompleksu I $\kappa$ B. Interesujący jest fakt, że wykonana do tej pory analiza modelu ([1]), nie pozwoliła na znalezienie tego kluczowego elementarnego procesu. Zatem uzyskane wyniki prowadzą do ukierunkowania dalszej analizy.

## 5. Podsumowanie

Zaproponowany algorytm jest dalej rozwijany. Testowane są najważniejsze parametry, które mogą mieć wpływ na uzyskiwane wyniki. Wśród tych parametrów wyróżnia się: długość listy tabu, liczbę iteracji, podczas których nie zostanie znalezione rozwiązanie spełniające warunki początkowe, oczekiwane wartości stałych  $K$  i  $R$ . Testowana jest również procedura aktualizacji najlepszego rozwiązania. Przedstawiony problem nie jest problemem maksymalizacyjnym – otrzymany w wyniku zbiór tranzycji spełnia oczekiwane założenia początkowe co do wartości stałych  $K$  i  $R$ , a więc zawiera co najmniej  $R$  tranzycji, które występują we wsparciach co najmniej  $K$  niezmienników. W przypadku, gdy wartości oczekiwane są zbyt wysokie, po odpowiedniej liczbie iteracji algorytm zwraca znalezione do tej pory najlepsze rozwiązanie, które nie spełnia warunku stopu. Przedstawiony algorytm pozwala znaleźć interesujące wyniki dla rzeczywistych danych, a tym samym staje się dopełnieniem dla istniejących już metod analizy modeli biologicznych w oparciu o t-niezmienniki.

## LITERATURA

1. Formanowicz D., Gutowska K., Formanowicz P.: Influence of different types of macrophages on IL-18 synthesis – a systems approach, nieopublikowany manuskrypt.
2. 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.
3. Formanowicz P.: Kombinatoryczne aspekty analizy t-niezmienników w modelach systemów biologicznych opartych na sieciach Petriego [w:] Świerniak A., Krystek J., (Red.): *Automatyzacja procesów dyskretnych, Teoria i zastosowania*", Wydawnictwo Pracowni Komputerowej Jacka Skalmierskiego, Vol II, Gliwice, 2016, p. 65–76.
4. Glover F.W., Laguna M.: *Tabu Search*, Springer, New York, 1997.
5. Grafahrend-Belau E., Schreiber F., Heiner M., Sackmann A., Junker B.H., Grunwald S. Speer A., Winder K., Koch I.: Modularization of biochemical networks based on classification of Petri net t-invariants, *BMC Bioinformatics*, 9, 2008, 90.
6. Koch I., Reisig W., Schreiber F. (Red.): *Modeling in systems biology: the Petri net approach*, Springer, London, 2011.
7. Murata T.: Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77, 1989, p. 541–580.
8. Sackmann A., Heiner M., Koch I.: Application of Petri net based analysis techniques to signal transduction pathways, *BMC Bioinformatics*, 7, 2006, 482.