



ssdnm
środowiskowe
studia doktoranckie
z nauk matematycznych

Wojciech Czarnecki

Uniwersytet Jagielloński

Jakościowe wnioskowanie czasowo przestrzenne z użyciem
algorytmów genetycznych

Praca semestralna nr 2
(semestr zimowy 2012/13)

Opiekun pracy: Krzysztof Jassem

Jakościowe wnioskowanie czasowo-przestrzenne z użyciem algorytmów genetycznych

Wojciech Czarnecki

Pracownia Systemów Informacyjnych,
Wydział Matematyki i Informatyki,
Uniwersytet im. Adama Mickiewicza w Poznaniu, Polska
w.czarnecki@amu.edu.pl

Abstrakt W artykule prezentujemy nowe, oparte o algorytmy genetyczne, podejście do dobrze znanego problemu sprawdzania niesprzeczności grafów jakościowych ograniczeń czasowo-przestrzennych. Po empirycznej analizie najpopularniejszych rozwiązań heurystycznych pokazujemy w jaki sposób można wyrazić ten problem w formie optymalizacji i przedstawiamy ogólny schemat dla zastosowania do niej metod ewolucyjnych. Następnie skupiamy się na rachunku Allena, który jest wciąż najpopularniejszą formą reprezentowania wiedzy czasowej i przedstawiamy szczegółowy opis stworzenia odpowiedniej reprezentacji w naszym schemacie.

Słowa kluczowe jakościowe rozumowanie czasowe, algorytm genetyczny, heurystyki, algebra Allena

1 Wprowadzenie

Problemem spełnialności więzów (CSP) nazywamy problem znalezienia rozwiązania dla pewnych ograniczeń (warunków wymaganych od zmiennych). Jest to stare zagadnienie sztucznej inteligencji, który sięga połowy lat siedemdziesiątych [10], które znalazło zastosowanie w wielu dziedzinach, od szeregowania zadań [3], przez bioinformatykę [9] po optymalizację zapytań do baz danych [12]. W poniższej pracy zajmujemy się egzystencjalną alternatywą CSP: dla zadanego grafu należy zdecydować czy jest on niesprzeczny (czy istnieje pewien zbiór wartości zmiennych, który spełnia zadane ograniczenia).

Skupiamy się na grafach jakościowych ograniczeń czasowo-przestrzennych, które są używane jako modele do reprezentacji i wnioskowania o czasie i przestrzeni. Każdy taki graf posiada n wierzchołków reprezentujących obiekty przestrzenne (bądź wydarzenia) oraz n^2 krawędzi pomiędzy nimi, każda etykietowana zbiorem ograniczeń z pewnego ustalonego formalizmu, reprezentujące proste relacje binarne pomiędzy wierzchołkami (pochodzące ze skończonego zbioru możliwych relacji, niezależnych od innych wierzchołków grafu). Dwa wierzchołki v_1 i v_2 połączone krawędzią o etykiecie $\{c_1, \dots, c_k\}$ reprezentują naszą wiedzę o tym, że $v_1 \text{ } c_1 \text{ } v_2 \vee \dots \vee v_1 \text{ } c_k \text{ } v_2$. Jeśli nie posiadamy wiedzy na temat relacji pomiędzy

dwoma wierzchołkami - ograniczeniem pomiędzy nimi jest relacja pełna (zbiór zawierający wszystkie relacje podstawowe danego formalizmu).

Jako, że problem jest NP-trudny dla większości formalizmów, pojawiło się wiele zróżnicowanych podejść, które można podzielić na trzy kategorie:

- Algorytmy o wykładniczej złożoności, które są kompletne, np. backtracking [13] - metoda przeszukiwania wszystkich możliwych podgrafów z ograniczeniami z pewnego subformalizmu, dla którego problem sprawdzania niesprzeczności jest rozstrzygalny w czasie wielomianowym.
- Algorytmy o wielomianowej złożoności, które nie są kompletne, np. algorytm spójności ścieżkowej [1], będący ogólniejszą formą spójności krawędziowej, sprawdzający czy dowolny trójwierzchołkowy podgraf jest niesprzeczny.
- Algorytmy o wielomianowej złożoności, kompletne tylko w określonych podklasach problemu (np. w określonych klasach grafów).

Pojawiły się również pojedyncze próby aplikacji algorytmów genetycznych. Alessandro Fin pokazał [7] w jaki sposób można wykorzystać tę technikę aby zrównoleglić wykonanie algorytmu backtrackingu dla pewnej subalgebry Allena [14]. Niestety nie jest to bezpośrednio zastosowanie algorytmu genetycznego a raczej metoda kontrolowania kolejności przeszukiwania przestrzeni stanów wykorzystując dobrze znane heurystyki [5] jako operatory mutacji. W naszych badaniach skupiliśmy się na bardziej bezpośrednim podejściu i proponujemy metodę o wielomianowej złożoności, która w ogólnym przypadku nie jest metodą kompletną.

Struktura naszej pracy jest następująca- najpierw przedstawiamy rachunek Allena, najbardziej znany formalizm jakościowego rozumowania czasowego, który jest wykorzystany jako główny przykład naszego podejścia. Następnie pokazujemy pewne empiryczne wyniki dotyczące precyzji algorytmu spójności ścieżkowej dla tego formalizmu, uzasadniając nimi potrzebę tworzenia alternatywnych metod dla rozwiązywania problemu niesprzeczności grafów ograniczeń. W kolejnych sekcjach prezentujemy alternatywne sformułowanie problemu, które umożliwia zastosowanie metod optymalizacyjnych oraz podajemy ogólny schemat zastosowania algorytmów genetycznych dla dowolnych formalizmów. Pozostała część opisuje przykładową implementację dla formalizmu Allena wraz z analizą kilku prostych właściwości, które były przydatne dla projektowania tego rozwiązania.

2 Algebra Allena

Rachunek Allena (znany również jako *algebra Allena*) jest formalizmem rozumowania czasowego stworzonym przez Jamesa F. Allena we wczesnych latach osiemdziesiątych [2], ale wciąż pozostaje jednym z najczęściej stosowanych podejść w tej dziedzinie.

Jeśli rozważymy wszystkie możliwe relacje pomiędzy dwoma przedziałami czasowymi opartymi jedynie o relatywny porządek ich punktów końcowych, otrzymamy 13 możliwych relacji (zebranych w tabeli 1), nazywanych podstawowymi relacjami Allena.

Nawet dla tak prostego formalizmu - sprawdzanie niesprzeczności grafu Allena (czyli w tym przypadku udzielenie odpowiedzi czy dla danego grafu istnieje taki zbiór przedziałów, który spełnia zadane ograniczenia) jest NP-kompletne [16].

nazwa	relacja	przedziały	nierówności punktów końcowych
before	b		$X_e < Y_s$
after	bi		$Y_e < X_s$
equals	e		$X_s = Y_s, X_e = Y_e$
starts	s		$X_s = Y_s, X_e < Y_e$
started by	si		$X_s = Y_s, Y_e < X_e$
finishes	f		$X_e = Y_e, X_s > Y_s$
finished by	fi		$X_e = Y_e, Y_s > X_s$
overlaps	o		$X_s < Y_s, X_e > Y_s, X_e < Y_e$
overlapped by	oi		$Y_s < X_s, Y_e > X_s, Y_e < X_e$
during	d		$X_s > Y_s, X_e < Y_e$
contains	di		$Y_s > X_s, Y_e < X_e$
meets	m		$X_e = Y_s$
met by	mi		$Y_e = X_s$

Tabela 1. Podstawowe relacje rachunku Allena. X_s oznacza punkt początkowy pierwszego przedziału, X_e jego punkt końcowy, Y_s punkt początkowy drugiego a Y_e końcowy.

3 Spójność ścieżkowa a niesprzeczność

Algorytm spójności ścieżkowej jest prostą metodą sprawdzania niesprzeczności wszystkich podgrafów o rozmiarze 3 (czasami nazywamy *algorytmem 3-spójności*[6])). Ze względu na jego wydajność jest często wykorzystywany do sprawdzania niesprzeczności grafów ograniczeń dla zróżnicowanych formalizmów (np. Marcin Walas wykorzystuje go dla systemu Q/A wzbogaconego o wiedzę przestrzenną [17]). Niestety to podejście może prowadzić do wielu fałszywych klasyfikacji grafów sprzecznych jako niesprzeczne. Zbadaliśmy empirycznie precyzję algorytmu spójności ścieżkowej analizując jak wiele z grafów które są ścieżkowo spójne są w rzeczywistości niesprzeczne. Wykorzystujemy prostą metodę tworzenia losowych grafów $A(n, p)$ (patrz algorytm 1).

Algorytm 1, $A(n,p)$ - generuje losowy graf ograniczeń Allena z n wierzchołkami i prawdopodobieństwem połączenia p

test

```

1:  $G \leftarrow \{\}$ 
2: for  $(i, j) \in \{1, \dots, n\}^2, i < j$  do
3:   if  $\text{random}() < p$  then
4:      $\text{constraints} \leftarrow \{\}$ 
5:     for  $c \in$  podstawowe relacje Allena do
6:       if  $\text{random}() < 0.5$  then
7:          $\text{constraints} \leftarrow \text{constraints} \cup \{c\}$ 
8:       end if
9:     end for
10:     $G \leftarrow G \cup \{(i, j, \text{constraints})\}$ 
11:   else
12:     $G \leftarrow G \cup \{(i, j, \text{relacja pełna})\}$ 
13:   end if
14: end for
15: return  $G$ 

```

Twierdzenie 1 Algorytm $A(n,1.0)$ generuje grafy ograniczeń Allena z jednostajnym rozkładem prawdopodobieństwa, przy założeniu, że istotna jest kolejność wierzchołków.

Dowód. Każdy z grafów ograniczeń Allena z n wierzchołkami może zostać zapisany jako liczba binarna o długości $13 \frac{n(n-1)}{2}$ (ponieważ mamy 13 podstawowych relacji, które mogą niezależnie pojawić się na każdej z $\frac{n(n-1)}{2}$ krawędzi) i taka transformacja jest bijekcją dopóki istotna jest kolejność wierzchołków (w przeciwnym razie obrazem wielu kodów binarnych byłby ten sam graf z odpowiednią permutacją wierzchołków). Na algorytm $A(n,1.0)$ można patrzeć jak na proces generowania takich właśnie ciągów zer i jedynek - dla każdej krawędzi ustawia on i'ty bit ograniczeń wtedy i tylko wtedy gdy i'ty warunek z linii 7 był prawdziwy. Stąd też dla każdej sekwencji x , $P(x) = \frac{1}{2^{13 \frac{n(n-1)}{2}}}$. \square

Dzięki twierdzeniu 1 racjonalnym jest założenie, że empiryczne wyniki uzyskane ze sprawdzania pewnych własności grafów wygenerowanych przy pomocy algorytmu $A(n,p)$ ukazują faktyczne zależności pomiędzy wszystkimi grafami Allena danego rozmiaru. W tabeli 2 zebrane zostały wyniki eksperymentów, w których dla każdej kombinacji n i d stworzyliśmy co najmniej 1000 grafów spójnych ścieżkowo (generując je używając $A(n, \frac{d}{n})$ i dodając do zbioru wtedy i tylko wtedy gdy był on ścieżkowo spójny) i zliczyliśmy liczbę grafów niesprzecznych spośród nich (wykorzystując do tego mocno zoptymalizowany algorytm backtrackingu).

Uzyskane rezultaty jasno ukazują, że algorytm spójności ścieżkowej jest metodą o bardzo niskiej precyzji w zdecydowanej większości przypadków. W zasadzie można na nim polegać jedynie przy stosowaniu go dla bardzo małych lub

	n=5	n=10	n=15	n=20	n=25	n=30	n=35	n=45	n=55	n=65	n=75
d=5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
d=10	-	0.625	0.59	0.56	0.52	0.48	0.49	0.47	0.53	0.52	0.5
d=13	-	-	0.04	0.03	0.01	0.01	<0.01	<0.01	<0.01	<0.01	<0.01
d=15	-	-	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01	<0.01

Tabela 2. Precyzja algorytmu spójności ścieżkowej dla problemu sprawdzania niesprzeczności grafu ograniczeń Allena testowana na losowych grafach generowanych za pomocą $A(n, \frac{d}{n})$

bardzo rzadkich grafów. Podobny eksperyment można przeprowadzić dla spójności krawędziowej, ale jako, że jest to słabsza forma spójności, jego precyzja będzie jeszcze niższa.

4 Zastosowanie algorytmu genetycznego dla sprawdzania niesprzeczności

Aby móc wykorzystać algorytm genetyczny do sprawdzania niesprzeczności grafu niezbędne jest przeformułowanie wyjściowego problemu z języka klasyfikacji binarnej do języka optymalizacji. Przedstawiamy tutaj ogólną metodę, która może zostać wykorzystana dla wielu formalizmów, dopóki spełnione będą pewne warunki. W kolejnych sekcjach używamy następującej notacji:

- r_{full} jest relacją pełną (ograniczeniem pełnym) danego formalizmu (np. w rachunku Allena $r_{full} = \{b, bi, e, s, si, f, fi, o, oi, m, mi, d, di\}$)
- A_{ij} jest zbiorem ograniczeń pomiędzy wierzchołkami i oraz j w grafie A
- $A \subset B$ (w kontekście grafów ograniczeń A oraz B) oznacza, że oba grafy mają te same wierzchołki i że $\forall i, j : A_{ij} = r_{full} \vee A_{ij} \subset B_{ij}$.

Definicja 1 (Krawędziowo maksymalny podgraf niesprzeczny). S nazywamy krawędziowo maksymalnym podgrafem niesprzeczny grafu G wtw. gdy $S \subset G$, S jest niesprzeczny oraz $\forall S' \subset G : S'$ jest sprzeczny lub

$$\# \{(i, j) : S'_{ij} \neq r_{full}\} < \# \{(i, j) : S_{ij} \neq r_{full}\}$$

Teraz możemy zdefiniować problem optymalizacyjny znajdowania krawędziowo maksymalnego podgrafu niesprzecznego (Edge Maximal Constraint Subgraph - EMCS) dla danego grafu ograniczeń. Łatwo zauważyć, że graf z m niepełnymi ograniczeniami jest niesprzeczny wtw. gdy EMCS ma rozmiar m (w rozumieniu ilości krawędzi o niepełnej relacji).

Spostrzeżenie 1. *Znajdowanie rozmiaru EMCS jest NP-trudne dla każdego typu grafy ograniczeń, dla którego problem niesprzeczności jest NP-trudny.*

Z jednej strony można spojrzeć na ten problem jako na metodę podejścia do sprawdzania niesprzeczności grafu ograniczeń, ale w rzeczywistości jest to znacznie szersze zagadnienie. W zastosowaniach praktycznych, gdy mamy wiele ograniczeń, które mogą prowadzić do sprzeczności (ze względu na błędy na etapie ekstrakcji informacji, dynamikę systemu, niepewność danych etc.) przydatną informacją jest które z ograniczeń należy usunąć by odzyskać niesprzeczność całego grafu wiedzy. Znalezienie EMCS danego grafu daje dokładną odpowiedź na tak postawione pytanie (przy założeniu, że chcemy zminimalizować liczbę usuniętych ograniczeń), a znajdowanie pewnej aproksymacji - daje pewne przybliżenie rozwiązania (daje zbiór krawędzi, których usunięcie przywróci niesprzeczność, acz może być on większy niż optymalny). Dla przykładu, znaną zagadkę *Who killed the duke of Densmore* autorstwa Claude Berge można rozwiązać szukając EMCS w grafie ograniczeń Allena opisującego tę historię.

W naszych rozważaniach rozróżniamy genotyp i fenotyp [8] osobnika. Kiedy piszemy *genotyp* mamy na myśli faktyczną reprezentację danego osobnika w naszej implementacji algorytmu genetycznego, podczas gdy *fenotyp* odnosi się do jego interpretacji w domenie jakościowego rozumowania czasowo-przestrzennego.

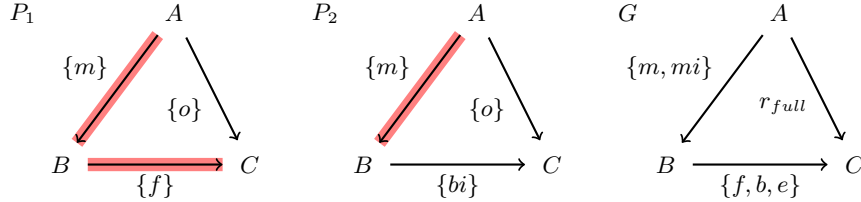
Ogólny schemat zastosowania algorytmu genetycznego dla znajdowania EMCS danego grafu $G = (V, E)$ w pewnej klasie *class* jest następujący:

1. Dla danej klasy grafów *class* potrzebujemy pewnej podklasy, która zawiera relacje $r_1, \dots, r_o \in subclass$ takie że $\forall r \in class : \exists k, a_1, \dots, a_k \in \mathbb{N} \bigcup_{i=1}^k r_{a_i} = r$ (np. klasa wszystkich relacji podstawowych). Dzięki temu każdy graf z *class* może zostać rozłożony na grafy z *subclass*, więc przestrzeń rozwiązań jest podzbiorem przestrzeni rozpiętej przez osobniki. Dodatkowo, *subclass* musi spełniać przynajmniej jeden z dwóch warunków:
 - (a) istnieje prosta metoda generowania niesprzecznych grafów tej subclassy
 - (b) niesprzeczność grafów tej subclassy może być łatwo (wydajnie) zweryfikowana
2. *fenotyp osobnika* - graf ograniczeń z rozważanego subclassy
3. *mutacja/krzyżowanie* - metoda generowania potomstwa o niesprzecznych fenotypach (jeśli warunek 1 jest spełniony) lub przynajmniej potomstwa o niesprzecznym fenotypie z dużym prawdopodobieństwem (wpp.). W przypadku operatora krzyżowania należy również zadbać o to, aby fenotyp potomstwa był możliwie podobny do fenotypów rodziców.
4. *funkcja oceny* - funkcja, która zwraca 0 jeśli fenotyp P jest sprzeczny (co łatwo sprawdzić zgodnie z przyjętymi założeniami) oraz liczbę krawędzi których zbiory ograniczeń są podzbiorem ograniczeń z odpowiednich krawędzi w grafie G (przy założeniu, że nie jest to krawędź o relacji pełnej) w przeciwnym przypadku.

$$fitness(P, G) = \begin{cases} 0, & \text{jeśli } P \text{ jest niespreczny} \\ \#\{(i, j) : P_{ij} \subset G_{ij} \wedge G_{ij} \neq r_{full}\}, & \text{wpp.} \end{cases}$$

Funkcja oceny opisuje więc rozmiar najlepszego kandydata na EMCS znalezionego dotychczas (przykład na rysunku 1). Jeśli $fitness(P, G) > 0$,

faktyczny kandydat może zostać odzyskany z fenotypu poprzez podmianę ograniczeń w P na krawędziach pomiędzy parami wierzchołków ze zbioru $\{(i, j) : \neg(P_{ij} \subset G_{ij}) \vee G_{ij} = r_{full}\}$ na r_{full} .



Rysunek 1. Przykład funkcji oceny dla grafów ograniczeń Allena. $fitness(P_1, G) = 2$, $fitness(P_2, G) = 1$

5 Zastosowanie do rachunku Allena

Jako subclassę grafów ograniczeń Allena wybraliśmy zbiór wszystkich podstawowych grafów (tzn. grafów, które na każdej krawędzi mają relację podstawową), która spełnia oba warunki podane w poprzedniej sekcji. Jest to klasa, w której niesprzeczność można sprawdzić np. algorytmem spójności ścieżkowej oraz istnieje prosta bijekcja pomiędzy tymi grafami a ciągami przedziałów [11] (które spełniają ich ograniczenia) co daje prostą metodę generowania reprezentantów tej klasy.

5.1 Genotyp

Genom każdego osobnika składa się z n przedziałów o naturalnych współrzędnych reprezentujących hipotetyczną pozycją przedziałów czasowych opisanych przez G . Każdy przedział jest pojedynczym *genem*. Generowanie takich przedziałów jest proste, ale istnieje problem z zakresem możliwych współrzędnych. Jeśli wybierzemy zbyt mały zakres to niektóre z grafów ograniczeń Allena nie będą mogły zostać wyrażone w ten sposób a wybranie zbyt dużego prowadzi do niepotrzebnego skomplikowania obliczeń oraz rozbudowy przestrzeni genotypów. Następujące twierdzenie daje dobry punkt wyjścia rozwiązania tego problemu:

Twierdzenie 2 *Dla każdego niesprzecznego grafu ograniczeń Allena o n wierzchołkach istnieje ciąg przedziałów $I = ([s_1, e_1], \dots, [s_n, e_n])$ gdzie $\forall_{1 \leq i \leq n} (s_i, e_i) \in \mathbb{N} \times \mathbb{N}$ spełniających jego ograniczenia, taki że $\bigcup I \subset [0, 2n - 1]$ i jest to najmniejsze możliwe ograniczenie, przy założeniu, że współrzędne są liczbami naturalnymi i nie rozpatrujemy odcinków pustych.*

Dowód. Graf jest niesprzeczny, więc istnieje pewien ciąg przedziałów

$$I' = ([s'_1, e'_1], \dots, [s'_n, e'_n])$$

spełniający jego ograniczenia. Zdefiniujemy

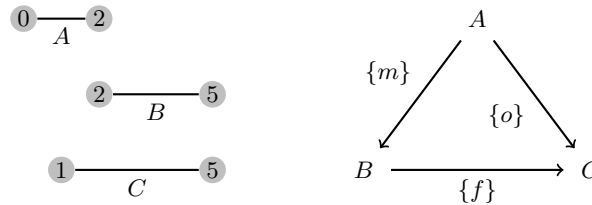
$$I = ([p_{s_1}, p_{e_1}], \dots, [p_{s_n}, p_{e_n}])$$

gdzie p_x jest pozycją (zaczynając od 0) wartości x w uporządkowanej liście elementów zbioru $S = \{x : \exists_{i \in \{1, 2, \dots, n\}} x = s'_i \vee x = e'_i\}$. I wciąż spełnia ograniczenia grafu jako, że zmiana wartości na indeksy porządkowe nie zmienia porządku punktów. Co więcej $\|S\| \leq 2n$ so $\bigcup I \subset [0, 2n - 1]$. Aby udowodnić, że jest to najlepsze możliwe ograniczenie wystarczy rozważyć graf ograniczeń Allena z wierzchołkami a_1, a_2, \dots, a_n takimi, że $\forall_{i < j} a_i \leq a_j$, który jest niesprzeczny i wymaga dokładnie $2n$ różnych współrzędnych ($a_1 = [0, 1], a_2 = [2, 3], \dots, a_n = [2n - 2, 2n - 1]$). \square

Wiedząc, że dla każdego G wystarczającym jest używanie przedziałów z zakresu $[0, 2n - 1]$ zdecydowaliśmy się na używanie zakresu $[0, 4n - 1]$ aby dać dodatkowe miejsce dla lepszej elastyczności. Ten dodatkowy narzut jest uzasadniony tym, że w ten sposób upewniamy się, że istnieje wiele genotypów fenotypu EMCS (podczas gdy używając $[0, 2n - 1]$ mielibyśmy dla niektórych grafów tylko jeden taki genotyp). Podczas testów wykorzystywanie większych narzutów nie przyniosło istotnej poprawy w działaniu algorytmu.

5.2 Fenotyp

W naszych badaniach fenotypem genotypu I jest podstawowy graf ograniczeń Allena, którego ograniczenia I spełnia (patrz rysunek 2). Takie grafy mogą być wygenerowane w czasie $O(n^2)$, jako, że dla każdej pary przedziałów możemy wywnioskować relację pomiędzy nimi w czasie stałym. Faktyczny fenotyp tworzony jest jedynie podczas procesu *selekcji elity*, ale jest to teoretyczna interpretacja naszych osobników.



Rysunek 2. Reprezentacja przedziałowa (po lewej) osobnika o genotypie $((0, 2), (2, 5), (1, 5))$ i jego fenotyp (po prawej)

5.3 Krzyżowanie

Zgodnie z zaproponowanym ogólnym schematem zastosowania algorytmu genetycznego, operator krzyżowania powinien tworzyć osobnika o niesprzecznym fenotypie, który jest możliwie podobny do fenotypów rodziców. W tym przypadku, podobieństwo pomiędzy dwoma genotypami może być zdefiniowane jako ilość wspólnych ograniczeń pomiędzy ich fenotypami. Aby osiągnąć taki rezultat używamy klasycznego n -punktowego krzyżowania, gdzie n jest liczbą wierzchołków grafu.

Twierdzenie 3 *n -punktowe krzyżowanie tworzy potomstwo o fenotypie, który jest niespreczny oraz dzieli conajmniej f^2 i $(n - f)^2$ ograniczeń z fenotypami rodziców (gdzie f jest liczbą genów odziedziczonych po pierwszym z rodziców).*

Dowód. Załóżmy, że podczas operacji krzyżowania f genów zostało skopiowanych od pierwszego z rodziców. Łatwo zauważyć, że relacje pomiędzy tymi f przedziałami nie uległy zmianie i jako, że jest dokładnie f^2 takich relacji, dziecko dzieli conajmniej f^2 ograniczeń z fenotypem tego rodzica i conajmniej $(n - f)^2$ z fenotypem drugiego rodzica. Fenotyp jest niespreczny jako, że jest spełniony przez swój genotyp.

Zgodnie z twierdzeniem 3 jeśli chcemy zmaksymalizować podobieństwo pomiędzy potomstwem a obojgiem rodziców powinniśmy upewnić się, że $f = \frac{n}{2}$ (jako rozwiązanie $\operatorname{argmin}_{0 \leq f \leq n} |f^2 - (n - f)^2|$). Niestety brakuje tu jednej rzeczy - rozważamy jedynie w pełni połączone grafy, podczas gdy w rzeczywistych aplikacjach większość relacji jest relacjami pełnymi, które są nieistotne z punktu widzenia dzielenia informacji. Dla lepszych wyników powinniśmy znaleźć sposób podziału wierzchołków grafów na dwie grupy w taki sposób, że różnica w ilości krawędzi (nie licząc tych o pełnej relacji) wewnątrz każdej z grup jest zminimalizowana, jednocześnie minimalizując ilość utraconej informacji. Bardziej formalnie, definiujemy stratę informacji jako f_{is} i funkcję równowagi f_e w sposób następujący:

$$f_{is}(A, G) = \#\{(i, j) : i \in A \wedge j \notin A \wedge G_{ij} \neq r_{full}\}$$

$$f_e(A, G) = |\#\{(i, j) : i, j \in A \wedge G_{ij} \neq r_{full}\} - \#\{(i, j) : i, j \notin A \wedge G_{ij} \neq r_{full}\}|$$

i dla pewnej stałej $\alpha \in [0, 1]$, szukamy rozwiązania dla

$$\operatorname{argmin}_{A \subset V, G=(V,E)} (\alpha f_{is}(A, G) + (1 - \alpha) f_e(A, V))$$

dla $\alpha = 1$ może być postrzegane jako instancja dobrze znanego problemu *podziału grafu (Graph Partitioning)*. Niestety ten problem również jest NP-trudny, acz istnieją pewne heurystyczne metody, które mogą zostać wykorzystane (np. algorytm Kernighan-Lin o złożoności $O(n^2 \lg(n))$).

5.4 Mutacja

Zdecydowaliśmy się zaimplementować operator prostej mutacji, który dla każdego genu zmienia go z zadanyam prawdopodobieństwem na losowy przedział spełniający ograniczenia z sekcji 4.

5.5 Funkcja oceny

Zgodnie z definicją użytych genotypów - każdy osobnik ma niesprzeczny fenotyp a dodatkowo liczba 'poprawnych' ograniczeń sformułowanych w definicji funkcji oceny może zostać policzona bez generowania faktycznego fenotypu. Dla każdej niepełnej relacji w G sprawdzamy czy nasze przedziały ją spełniają. Metoda ta prowadzi do złożoności czasowej $O(c)$ gdzie c to liczba niepełnych relacji w G (która jest w praktyce istotnie mniejsze niż n^2).

6 Ewaluacja

Używamy algorytmu genetycznego z jednym operatorem krzyżowania (zdefiniowanym w sekcji 5.3) z prawdopodobieństwem krzyżowania 0.7 i jednym operatorem mutacji (zdefiniowanym w sekcji 5.4) z prawdopodobieństwem 0.05. Wykorzystujemy również lekko zmodyfikowaną technikę selekcji elity aby wymusić większe zróżnicowanie naszej populacji. Zamiast po prostu wybierać K najlepszych osobników, które niezależnie od wyniku krzyżowania i mutacji przechodzą do kolejnej generacji, wybieramy K najlepszych osobników o unikalnych genotypach, więc dla przykładu jeśli najlepszy osobnik ma genotyp $([0, 1], [2, 3])$ to osobnik $([2, 3], [6, 10])$ nie jest rozważany przy wyborze elity, jako, że ma identyczny fenotyp. Cały algorytm jest wielomianowy, o złożoności $O(g p(\log(p) + c + n))$, gdzie g jest liczbą pokoleń a p rozmiarem populacji.

	n=5	n=10	n=15	n=20	n=25	n=30	n=35
d=5	1.0	1.0	1.0	0.99	0.99	0.99	0.98
d=10	-	0.97	0.97	0.97	0.96	0.95	0.95
d=13	-	-	0.97	0.96	0.95	0.95	0.94
d=15	-	-	0.96	0.94	0.94	0.93	0.93

Tabela 3. Ewaluacja algorytmu genetycznego użytego do szukania EMCS w losowych grafach ograniczeń Allena wygenerowanych przy użyciu $A(n, \frac{d}{n})$.

Jako, że póki co nie ma konkretnej bazy danych grafów ograniczeń Allena, które mogłyby zostać wykorzystane do ewaluacji naszego algorytmu, przetestowaliśmy go na ponad 10 000 losowych grafów, stworzonych używając:

- modelu $A(p, d)$, analogiczny jak ten opisany w sekcji 3, który umożliwia testy na grafach o jednostajnym rozkładzie (na mocy twierdzenia 1)
- modelu $S(n, p)$ [4], który generuje duże niesprzeczne grafy, ale niestety - nie z jednostajnym rozkładem prawdopodobieństwa (ale każdy z niesprzecznych grafów może zostać wygenerowany w ten sposób, być może z bardzo małym prawdopodobieństwem)

	n=10	n=15	n=20	n=25	n=30	n=35	n=50	n=100	n=150	n=200
p=0.125	1	1	1	1	1	1	0.99	0.93	0.88	0.86
p=0.25	1	1	1	0.99	0.99	0.99	0.97	0.9	0.86	0.84
p=0.5	1	0.99	0.99	0.98	0.98	0.98	0.95	0.88	0.84	0.81
p=1	0.99	0.98	0.98	0.98	0.98	0.96	0.94	0.87	0.83	0.8

Tabela 4. Ewaluacja algorytmu genetycznego użytego do szukania EMCS w losowych grafach ograniczeń Allena wygenerowanych przy użyciu $S(n,p)$.

Tabele 3 i 4 pokazują wyniki tych eksperymentów. użytą miarą poprawności jest średni stosunek rozmiaru znalezionej rozwiązania do właściwego rozmiaru EMCS (pomijając pełne relacje).

Tabela 4 pokazuje, że nawet dla dość dużych grafów algorytm zwraca dobrą aproksymacją (nie mniej niż 0.8 wielkości EMCS dla grafu o 200 wierzchołkach). Średnio dla wszystkich grafów, które wygenerowaliśmy używając $S(n,p)$ nasz algorytm uzyskał średnią poprawność 0.95 z odchyleniem standardowym 0.06, co jest obiecującym rezultatem w kontekście przyszłych systemów wymagających znalezienia EMCS.

7 Wnioski

W tym artykule zaprezentowaliśmy alternatywne podejście do problemu sprawdzania niesprzeczności grafu ograniczeń poprzez wyrażenie go jako problemu optymalizacyjnego opartego o szukanie największego niesprzecznego podgrafu. Zaproponowany schemat zastosowania algorytmu genetycznego jest bardzo ogólny i może zostać zastosowany do szerokiej gamy formalizmów rozumowania czasowo-przestrzennego (np. RCC-8, CDC, XCDC [15]). Pokazaliśmy również w jaki sposób może zostać on zaimplementowany dla rachunku Allena i przetestowaliśmy go na dużym zbiorze losowych grafów. Podejście to może zostać rozszerzone na wiele sposobów, m.in. poprzez implementowanie lepszych heurystyk dla operatorów krzyżowania/mutacji, ale również poprzez zastosowanie go do bardziej złożonych dziedzin jak grafy z rozmytymi ograniczami czy grafy o dynamicznie zmieniających się ograniczeniach, które będą przedmiotem przyszłych badań.

Literatura

1. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The Design and Analysis of Computer Algorithms. Addison-Wesley (1974)
2. Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM 26(11), 832–843 (1983)
3. Angelsmark, O., Jonsson, P.: Some observations on durations, scheduling and allen's algebra. In: Dechter, R. (ed.) Principles and Practice of Constraint Programming, Lecture Notes in Computer Science, vol. 1894, pp. 484–489. Springer Berlin Heidelberg (2000)

4. Beek, P.V., Manchak, D.W.: The design and experimental analysis of algorithms for temporal reasoning. *Journal of Artificial Intelligence Research* 4, 1–18 (1996)
5. Bitner, J.R., Reingold, E.M.: Backtrack programming techniques. *Communication of the ACM* 18(11), 651–656 (Nov 1975)
6. Cooper, M.C.: An optimal k-consistency algorithm. *Artificial Intelligence* 41(1), 89–95 (1989)
7. Fin, A.: A genetic approach to qualitative temporal reasoning with constraints. *International Conference on Computational Intelligence and Multimedia Applications* p. 91 (1999)
8. Jeff Clune, Benjamin E. Beckmann, C.O., Pennock, R.T.: Evolving coordinated quadruped gaits with the hyperneat generative encoding. In: *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2009*. pp. 2764–2771 (2009)
9. Kelsey, T., Linton, S.: Qualitative models of cell population dynamics as constraint satisfaction problems. In: *WCB 12 Workshop on Constraint-Based Methods for Bioinformatics*. p. 16 (2012)
10. Lauriere, J.L.: A language and a program for stating and solving combinatorial problems. *Artificial Intelligence* 10(1), 29–127 (1978)
11. Ligozat, G.: *Qualitative Spatial and Temporal Reasoning*. Iste Series, Wiley (2011)
12. Mouhoub, M., Feng, C.: Csp techniques for solving combinatorial queries within relational databases. In: Nguyen, N., Szczerbicki, E. (eds.) *Intelligent Systems for Knowledge Management, Studies in Computational Intelligence*, vol. 252, pp. 131–151. Springer Berlin Heidelberg (2009)
13. Nadel, B.A.: Constraint satisfaction algorithms. *Computational Intelligence* 5, 188–299 (1989)
14. Nebel, B.: Solving hard qualitative temporal reasoning problems: Evaluating the efficiency of using the ord-horn class. *Constraints* 1 (1997)
15. Osinski, J.: Extended cdc vs other formalisms – the comparative analysis of the models for spatio-temporal reasoning. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L., Zurada, J. (eds.) *Artificial Intelligence and Soft Computing, LNCS*, vol. 7268, pp. 679–687. Springer Berlin Heidelberg (2012)
16. Vilain, M., Kautz, H., van Beek, P.: Constraint Propagation Algorithms for Temporal Reasoning: A Revised Report. In: Weld, D.S., de Kleer, J. (eds.) *Readings in Qualitative Reasoning about Physical Systems*, pp. 373–381. Morgan Kaufmann (1989)
17. Walas, M.: How to answer yes/no spatial questions using qualitative reasoning? In: Gelbukh, A. (ed.) *Computational Linguistics and Intelligent Text Processing, LNCS*, vol. 7182, pp. 330–341. Springer Berlin Heidelberg (2012)