

Instytut Informatyki i Matematyki Komputerowej

Uniwersytet Jagielloński

**Recenzja rozprawy doktorskiej Pana mgr. Piotra Brendela pt.  
„Algorytm podzbioru acyklicznego: rozszerzenia i zastosowania”**

**Obszar problemowy rozprawy**

Przedstawiona mi do recenzji praca doktorska Pana mgr Piotra Brendela została napisana pod opieką prof. dr. hab. Mariana Mrozka na Wydziale Matematyki i Informatyki Uniwersytetu Jagiellońskiego. Tematem pracy jest rozszerzenie tzw. algorytmu wyznaczania podzbioru cyklicznego (Mrozek, Pilarczyk, Żelazna, Homology algorithm based on acyclic subspace, Comp. and Math. with App. 55(2008), 2395-2412) działającego na zbiorach kostkowych na przypadek kompleksów symplecjalnych. Rozprawa sytuuje się więc na pograniczu teorii homologii i algorytmiki, tzn. zawiera wyniki będące algorytmicznym podejściem wspomagającym obliczanie grup homologii. Wspomaganie to polega na redukcji kompleksów symplecjalnych z zachowaniem ich homologii, co zmniejsza rozmiar danych wejściowych (opis analizowanej przestrzeni topologicznej) i automatycznie pozwala na zaoszczędzenie pamięci podczas obliczeń.

Teoria homologii zajmuje się analizą i klasyfikowaniem rozmaitości charakteryzując je przez liczbę „różnych” cykli, jakie można wyznaczyć na tej powierzchni, przy czym cykle są „różne”, jeśli nie istnieje ciągła funkcja przekształcająca jeden z nich w drugi. Specyficznym rodzajem analizowanych rozmaitości są tzw. kompleksy symplecjalne, tzn. przestrzenie konstruowane przy pomocy punktów, odcinków, trójkątów oraz ich  $n$ -wymiarowych uogólnień. Rozprawa dotyczy takich właśnie przestrzeni topologicznych. Współczesne podejście do homologii to podejście algebraiczne, gdzie grupy homologii traktowane są jako elementy skończone generowanych grup abelowych. Homologią przestrzeni topologicznej  $X$  nazywa się zbiór niezmienników topologicznych  $X$  opisywanych przez ich grupy homologii  $H_n(X)$ ,  $i = 0, 1, \dots$ , gdzie  $n$ -ta grupa homologii  $H_n$  reprezentuje  $n$ -wymiarowe „dziury” w  $X$ . Formalnie,  $n$ -ta grupa homologii definiowana jest jako grupa ilorazowa  $H_n(X) = \ker(\partial_n) / \text{im}(\partial_{n+1})$ , gdzie  $\partial_n$  jest operatorem brzegu, który  $n$ -wymiarowemu sympleksowi przyporządkowuje kombinację liniową wszystkich jego ścian  $(n-1)$ -wymiaru.

Teoria homologii znajduje praktyczne zastosowanie w tzw. topologicznej analizie danych, która bada duże, wielowymiarowe, niekompletne i/lub zaszumione zbiory danych z wykorzystaniem technik topologicznych, w myśl zasady „data has shape and shape has meaning” (G. Singh). Analizy te mogą

również służyć do budowy modeli predykcyjnych opartych o analizę kształtu danych. Chociażby z tego tylko względu tematykę rozprawy uważam za bardzo aktualną oraz istotną.

### **Zawartość rozprawy**

Praca liczy 110 stron i podzielona jest na 4 rozdziały poprzedzone wstępem przedstawiającym motywację oraz streszczenie głównych wyników rozprawy, oraz dwa dodatki. Rozdział 1 wprowadza czytelnika w podstawowe pojęcia z zakresu teorii homologii wykorzystywane w późniejszej części rozprawy. Rozdział 2 jest główną częścią pracy, zawierającą oryginalne wyniki doktoranta związane z algorytmem konstrukcji podzbioru cyklicznego. W rozdziale 3 opisano algorytm obliczania tzw. grupy podstawowej (czyli, nieformalnie mówiąc, grupy cykli na rozmaitości z operacją ich „sklejania”) dla pewnej klasy CW-kompleksów. Rozdział 4 to podsumowanie pracy. Dodatek A opisuje notację wykorzystaną do zapisu algorytmów, a dodatek B stanowi techniczną instrukcję kompilacji i użycia dołączonej do pracy implementacji algorytmu. Rozprawa zawiera również indeks pojęć, który bardzo ułatwia czytanie, zwłaszcza gdy chce się wrócić do jakiejś definicji.

### **Oryginalne osiągnięcia**

Problem obliczenia homologii zadanej przestrzeni może być wymagający obliczeniowo, ale można go uprościć poprzez zredukowanie rozmiaru danych wejściowych korzystając z prostego faktu, iż

$$H_n(X) \cong H_n(X, A), n > 0,$$

gdzie  $A$  jest *podzbiorem acyklicznym* zbioru  $X$ , tzn. jest homologicznie równoważny punktowi, a  $H_n(X, A)$  jest homologią relatywną. Równoważność ta oznacza, że homologię  $X$  można liczyć z pominięciem zbioru  $A$ , zatem jeśli  $A$  jest duży w stosunku do  $X$ , obliczenie homologii będzie szybsze niż w przypadku analizowania całego zbioru  $X$ .

Powyższa obserwacja stanowi podstawę do konstrukcji algorytmu. Oryginalne wyniki doktoranta zawarte w rozprawie to:

- projekt, implementacja i analiza złożoności obliczeniowej algorytmu konstrukcji podzbioru acyklicznego będącego rozszerzeniem algorytmu dla zbiorów kostkowych na przypadek kompleksów symplijalnych;
- modyfikacje w/w algorytmu pozwalające na zwiększenie jego efektywności poprzez zwiększenie rozmiaru konstruowanych podkompleksów;
- zaproponowanie sposobu zrównoleglenia algorytmu podzbioru acyklicznego;
- projekt i implementacja algorytmu obliczającego grupę podstawową określonej klasy CW-kompleksów.

Część wyników doktoranta została opublikowanych w dwóch publikacjach:

- Brendel, Dłotko, Ellis, Juda, Mrozek – Computing fundamental groups from point clouds, Appl. Alg. in Eng., Comm. & Comp. 26(2015), 27-48,
- Brendel, Dłotko, Mrozek, Żelazna – Homology Computations via Acyclic Subspace, LNCS 7309(2012), 117-127.

Oba czasopisma są dobre: pierwsze posiada Impact Factor, a w serii LNCS, również utrzymującej wysoki poziom, publikowane są głównie recenzowane artykuły pokonferencyjne.

Za rzecz bardzo wartościową w przedstawionej rozprawie uważam rzeczywistą implementację algorytmu podzbioru cyklicznego. W wielu pracach doktorskich z informatyki (nawet tych w naukach technicznych, o naukach matematycznych nie wspominając) algorytmy występują jedynie na papierze, a mgr Brendel napisał w pełni działający program implementujący zaprojektowany wcześniej algorytm.

Doktorant jest również autorem jednego z twierdzeń oraz części lematów opisanych w pracy (lematy 2.5, 2.11-2.14, 2.22-2.24, twierdzenie 2.15). Wyniki teoretyczne nie budzą moich wątpliwości, doktorant wykazuje się dobrym poziomem kultury matematycznej oraz znajomością teoretycznych podstaw algebry czy topologii algebraicznej i sprawnie porusza się w tym obszarze. W kolejnej części recenzji więcej uwagi chcę poświęcić samym algorytmom oraz wynikom dotyczącym ich analizy. W swojej zasadniczej idei algorytmy te sprowadzają się do szczególnych postaci algorytmów przeszukiwania wszerz, co może wydawać się dosyć proste i nieoryginalne. Jednak gdy zagłębić się w szczegóły, konstrukcja tych algorytmów wymaga bardzo dobrej znajomości topologii algebraicznej. W podejściu algorytmicznym do homologii występuje bowiem wiele subtelności i detali, które są niezwykle istotne dla poprawności działania algorytmów. Sposoby radzenia sobie z większością z nich są formalnie opisane w lematach i twierdzeniach zawartych w rozdziałach 1 i 2. Osobnym problemem – z punktu widzenia projektowania algorytmu – była kwestia doboru odpowiednich struktur danych dla reprezentacji rozmaitości (sympleksów, topleksów itp.) zapewniających nie tylko poprawność reprezentacji, ale i efektywność działania algorytmu.

### **Uwagi krytyczne i polemiczne**

Nie mam najmniejszych wątpliwości, że algorytmy zawarte w pracy co do swej idei są poprawne – ich koncepcja jest dla mnie jasna i zrozumiała. Moje wątpliwości budzą natomiast pewne fragmenty dotyczące analizy złożoności oraz sposób samego zapisu algorytmów. W przypadku Algorytmu 2.1 nie przeprowadzono analizy złożoności tworzenia słownika (linia 2), co może mieć przecież wpływ na złożoność całego algorytmu. Nie jestem również pewien, czy w przypadku istnienia np. wierzchołka izolowanego algorytm go przeanalizuje (wydaje mi się, że taki wierzchołek będzie pominięty w liniach 4. i 12.). Algorytm ma zwrócić trójkę  $(V, E, C)$  gdzie  $C$  jest zbiorem topleksów, ale ze względu na linie 5. i 9. do zbiorów  $V$  i  $C$  zawsze będą dodawane te same elementy, w rezultacie czego na samym końcu dostaniemy  $C = V$  (?): po wejściu do if-a w linii 4. w linii 6. element dodany w linii 5. do  $C$

bezwzględnie dodawany jest do kolejki  $Q$ , a więc w pętli 7. w końcu w którymś momencie zostanie dodany do  $V$ . W pętli 10. co prawda do kolejki  $Q$  mogą być dodawane inne elementy, ale po powrocie, z chwilą wejścia do ciała if-a w linii 4. elementy te również w końcu trafią do obu zbiorów  $C$  i  $V$ . W linii 13. Występuje notacja  $(\tau, \eta)$  co sugeruje krawędź skierowaną, a mamy chyba do czynienia z krawędziami nieskierowanymi.

Wydaje mi się, że algorytm 2.1. można w ogóle zapisać prościej, wykorzystując wyłącznie postać słownika  $D$  i informacje, jakie on zawiera, tzn. zbiór topleksów zawierających zadany wierzchołek – poniżej pseudokod uproszczonej wersji algorytmu dla konstrukcji  $V$  i  $E$ :

```
D := TwórzSłownik(S); Q :=  $\bigcup_v D[v]$ 
while Q !=  $\emptyset$ 
   $\tau$  := PobierzPierwszy(Q)
  forall  $\eta \in Q$ 
    if  $\tau \cap \eta \neq \emptyset$  E := E  $\cup$   $(\{\tau, \eta\})$ ; V := V  $\cup$   $\{\tau, \eta\}$ 
```

Natomiast zbiór  $C$  można chyba konstruować wykonując BFS na topleksach i po prostu dodawać tylko jeden topleks dla składowej:

```
Q :=  $\bigcup_v D[v]$ 
while Q !=  $\emptyset$ 
  C := PobierzPierwszy(Q); T := wierzchołki C
  forall  $\tau \in Q$ 
    if  $\tau \cap T \neq \emptyset$  Q := Q  $\setminus$   $\tau$ ; T := T  $\cup$  wierzchołki  $\tau$ 
```

Lemat 2.9 opisuje złożoność algorytmu 2.1 jako  $O(\text{card } V \cdot (\dim S + 1) \cdot \deg D)$ . Po pierwsze owo „+1” nie ma wpływu na złożoność w notacji  $O$  i spokojnie można je pominąć, ale to akurat jest szczegół. Poważniejszą usterką jest tu fakt, iż złożoność wyrażona jest w terminach elementów, które nie są częścią wejścia. Wejściem jest jedynie  $S$  i złożoność może zależeć tylko od jego rozmiaru. Rozmiar słownika niewątpliwie da się wyrazić w terminach wielkości  $S$  jako  $O(S)$ . Natomiast zmienna  $V$  jest w ogóle modyfikowana w trakcie działania algorytmu i jej rozmiar się zmienia w czasie, więc w ogóle nie można mówić o jej rozmiarze przy definiowaniu złożoności. Należy obliczyć górne ograniczenie na rozmiar  $V$  (które wynosi  $O(S)$ ). Ostatecznie, złożoność algorytmu wynosi  $O(|S|^3)$ .

W algorytmie 2.2 do zbioru  $A$  w linii 3. dodane zostaną wszystkie  $\sigma \in G$  niezależnie od warunku z linii 8, bo linia 3. wykonuje się bezwzględnie na początku każdego przebiegu pętli forall, a chyba nie o to chodziło? Przyjmując nawet, że algorytm działa tak, jak powinien, postać zbioru  $A$  w oczywisty sposób zależeć będzie od wyboru początkowego  $\sigma$  – na szczęście doktorant zauważył to i proponuje w rozdziale 2.3. usprawnienie pozwalające na dodanie do podkompleksu acyklicznego większej liczby elementów, niż by to wynikało z „naiwnej” wersji algorytmu 2.2.

W algorytmie 2.3. dobrze by było rozważyć złożoność dwóch operacji: sprawdzenia predykatu w linii 8. oraz operacji dodania elementu do zbioru w linii 12, gdyż mogą one wpływać na złożoność całego algorytmu.

Dla algorytmu 2.5. nie podano złożoności obliczeniowej. W przypadku algorytmu 2.6. z kolei brakuje mi rozważań na temat tego, czy wybór elementu  $W$  w linii 22. nie będzie aby miał wpływu na efektywność algorytmu (analogicznie do wyboru początkowego  $\sigma$  w algorytmie 2.2.).

W dodatku A.2 opisano składnię algorytmów – blok instrukcji oraz instrukcje warunkowe i pętle; w pracy doktorskiej z informatyki można sobie to darować.

Tablica 2.1 zawiera porównanie algorytmu podzbioru acyklicznego z algorytmem opartym na koredukcjach dla testowych kompleksów symplecjalnych pochodzących z pewnej bazy danych. Eksperymenty przeprowadzono jednak tylko dla trzech kompleksów, a baza ta zawiera ich aż 21 – dlaczego inne nie zostały poddane testom?

Tablica 2.2. zawiera wyniki doświadczeń dla kompleksów o wymiarach 3 i 4. Co z kompleksami większymi? Czy algorytm daje sobie z nimi radę?

Drobniejsze uwagi merytoryczne:

- na str. 6 doktorant prowadzi rozważania na temat złożoności algorytmu obliczania grup homologii i pisze, że w ogólnym przypadku (tzn. gdy współczynniki macierzy operatora brzegu niekoniecznie należą do ciała) jego złożoność jest *ponad* sześcienna – ale nie precyzuje jaką dokładnie, w szczególności interesujące jest to, czy jest ona nadal wielomianowa?
- str. 12: jest  $X \cup_f U$ , powinno być  $X \cup_f Y$
- w definicji CW-struktury na str. 13 podzbiory  $K$  muszą być niepuste
- w tej samej definicji jest  $K^i$  a powinno być  $K_i$
- str. 14d1: jest  $B^{|\varphi|}$ , powinno być  $B^{|\sigma|}$
- str. 213d: jeśli  $|S| < \infty$ , to czy warunek, iż  $c(\sigma) = 0$  dla *prawie wszystkich* sympleksów  $\sigma \in S$  nie jest pusto spełnialny?
- str. 24g16 – brak definicji  $H_*$
- str. 26g4: jest  $v_0 - v_1 \in B_n(S)$ , powinno być  $v_0 - v_1 \in B_0(S)$
- str. 28d3: jest  $n$ , powinno być  $d$
- str. 30g14: jest  $\alpha, \beta \in \mathbb{Z}$ , powinno być  $\alpha_i, \beta_i \in \mathbb{Z}$
- str. 34d4: co z warunkiem  $0 < k < l < n$  w przypadku, gdy  $n = 2$ ?
- czy lematy 2.2, 2.4, 2.7, 2.8 są oryginalnym wynikiem doktoranta, czy też doktorant przytacza ich dowody z jakiegoś źródła?
- str. 39g19: suma  $U$  powinna być indeksowana, bo w takiej postaci  $UK$  oznacza sumę elementów ze zbiorów  $S_k$ , a nie zbiór  $K_n$
- str. 39d8: niezdefiniowane  $H$
- str. 45d5: jest „ścieżka”, powinno być „ścieżka  $W$ ”
- str. 45d14: skoro  $W$  jest ciągiem, to  $W = (\sigma_1, \dots, \sigma_n)$ , a nie  $\{\sigma_1, \dots, \sigma_n\}$
- str. 58, algorytm, l. 17: jest „SympleksyZawierajace”, powinno być „TopleksyZawierajace”
- str. 64d16: struktura drzewa nie będzie zniszczona zawsze, ale tylko wtedy, gdy usuwana krawędź nie zawiera wierzchołka będącego liściem
- czy twierdzenie 3.16 jest wynikiem własnym doktoranta, czy też jego dowód istniał wcześniej, a doktorant jedynie go zweryfikował przy użyciu obliczeń?

- notacja opisana w dodatku A.16 powinna zostać wprowadzona wcześniej, ponieważ była używana w algorytmach w rozdziale 2

Drobne błędy językowe/literówki znalezione w rozprawie (nie wpływające w żadnym stopniu na ocenę merytoryczną pracy):

6g2: przecinek po „matematyczne”; 7d16 potrafią → mogą; 8d6 CW-kompleksów, → CW-kompleksów; 9g3 teoretycznie → teoretyczne; 14g3, 41g19 Ilość → Liczbę; 14d13 odwzorowuje → odwzorowuje; 20g19 symplecjajnym → symplecjajny; 22d5 kowymiariu → kowymiaru; 29d7, 42d18, 43d12 prze → przez; 30d12 Operacje → Operację; 33g18 różnymi → różnymi; 38d13 w postaci → postaci; 50g4 wejściowa → wejściową; 62d7 Obie z nich → Obie; 64g19 algorytm → algorytmu; 69d12 względów → względu; 72g12 równie → również; 73d12 indeks to → indeks; 74d8 orientacje → orientację; 76g1, 82d4 ilość → liczba; 77g14 ztablicowanej → stablicowanej; 77g16 ilość → liczbę; 82g17 wszystkie pary redukcyjne → wszystkich par redukcyjnych; 82d11, 82d12 ilość → liczby; 82d6 nawet znalezienie → znalezienie nawet; 83d6 uruchomiomy → uruchomiony; 84g11 odpowiadającym → odpowiadających; 88d8 3.11 → twierdzenia 3.11; 92g8 Dowiedzono → Dowiedziono;

### Konkluzja

Podsumowując, uważam, że przedstawiona mi do oceny rozprawa doktorska mgr. Piotra Brendela zawiera oryginalne wyniki i pomimo kilku uwag krytycznych moja ogólna jej ocena jest pozytywna. Rozprawa ta w mojej ocenie spełnia wymagania zwyczajowe i ustawowe stawiane rozprawom doktorskim. W związku z tym wnoszę o dopuszczenie doktoranta do dalszych czynności przewodu doktorskiego.

Adam Roman

